PMBANet: Progressive Multi-Branch Aggregation Network for Scene Depth Super-Resolution

Xinchen Ye, Baoli Sun, Zhihui Wang, Jingyu Yang, Rui Xu, Haojie Li, Baopu Li

***Abstract*—Depth map super-resolution is an ill-posed inverse problem with many challenges. First, depth boundaries are generally hard to reconstruct particularly at large magnification factors. Second, depth regions on fine structures and tiny objects in the scene are destroyed seriously by downsampling degradation. To tackle these difficulties, we propose a progressive multi-branch aggregation network (PMBANet), which consists of stacked MBA blocks to fully address the above problems and progressively recover the degraded depth map. Specifically, each MBA block has multiple parallel branches: 1) The reconstruction branch is proposed based on the designed attention-based error feed-forward/-back modules, which iteratively exploits and compensates the downsampling errors to refine the depth map by imposing the attention mechanism on the module to gradually highlight the informative features at depth boundaries. 2) We formulate a separate guidance branch as prior knowledge to help to recover the depth details, in which the multi-scale branch is to learn a multi-scale representation that pays close attention at objects of different scales, while the color branch regularizes the depth map by using auxiliary color information. Then, a fusion block is introduced to adaptively fuse and select the discriminative features from all the branches. The design methodology of our whole network is well-founded, and extensive experiments on benchmark datasets demonstrate that our method achieves superior performance in comparison with the state-of-the-art methods. Our code and models are available at https://github.com/Sunbaoli/PMBANet_DSR/.**

***Index Terms*—Depth map, super-resolution, aggregation, progressive, multi-branch**

I. Introduction

Scene depth map is essential and widely used as a basic element in many computer vision fields [7], [8], [11]. However, due to the imaging limitation of depth sensors in real conditions, high quality and high resolution (HR) depth maps are often difficult or even impossible to be acquired directly, thus effective pro-precessing depth super-resolution (SR) techniques are needed to yield HR output from the degraded low resolution (LR) counterpart. Usually, a color image and its associated depth map are the photometric and geometrical representations of the same scene, and have strong structural similarity [9]. Therefore, most existing depth SR methods use color information as guidance to recover the degraded depth maps. Traditional model-based methods [44] or filter-based methods [39], [41] constructed the hand-designed objective functions or filters based on naive assumptions, which can not approach the real depth map priors and lead to unsatisfactory results. Recently, CNN-based methods [23], [30], [35], [45] have been proposed to recover depth maps by automatically learning well-designed networks from data.

A. Motivation

Although the above CNN-based methods present impressive performance, the results are unsatisfactory when dealing with the recovery of depth details. First, depth boundaries are generally hard to reconstruct from LR depth maps and easy to lose sharpness particularly at large magnification factors due to the loss of spatial information. In addition, depth regions on fine structures and tiny objects in the scene are destroyed seriously by the downsampling degradation, which further impedes the accurate depth recovery.

As we observe, the backbone networks of existing depth SR methods can be classified into two categories, i.e., 1) cone- or hourglass-shaped architectures [30], [59] that use low-to-high resolution subnetworks to progressively extract features and raise the spatial resolution, and 2) barrel-shaped architectures [36], [53] that extract features and recover the depth map without changing the feature resolution. Both architectures belong to the networks designed based on purely feed-forward connections, which cannot fully exploit effective high-resolution features in representing the LR to HR relation, especially for large scaling factors.

Besides, the decrease of resolution brings different degrees of damage to the objects in the scene, especially to the fine structures and tiny objects. e.g., the sticks in *Art* of Fig. 4, which requires to leverage multi-scale information to enhance the ability of feature representations and then accurately recover each depth region. Besides, a mainstream multi-scale technique is to use multi-scale inputs [17], or integrate the pyramid dilated convolution modules [5] [19] into the backbone network from a simple cascaded way and use a limited number of parallel dilated convolutions to capture features of different receptive fields, which cannot fully exploit the multi-scale information.

Moreover, rich color features can be used as guidance to further resolve the downsamping degradation in depth SR. However, color discontinuities do not always coincide with those of depth map (structure inconsistency), which results in

This work was supported by National Natural Science Foundation of China (NSFC) under Grant 61702078, 61772108, 61976038, 61772106.

X. Ye, B. Sun, Z. Wang, R. Xu, and H. Li are with DUT-RU International School of Information Science & Engineering, Dalian University of Technology, Liaoning, and Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, China (Corresponding author: Z. Wang. E-mail: zhwang@dlut.edu.cn).

J. Yang is with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China.

B. Li is with Baidu Research (USA).

noticeable artifacts such as texture copying and depth bleeding. Therefore, how to leverage color information to help recover the depth map and whether the color image is required for all upsampling rates, especially for the easily recovered $2\times$ and $4\times$ upsampling cases, still need to be developed and verified.

According to the above analysis, we pursue better architectural design aiming at further improvements. In cognition theory [20] and recent progress [25], [26], [37] in deep learning, feedback connections that inversely transmit response signals from the higher-order areas to the lower-order areas, play an important role in human expression and regulation, which can be used to extract more effective image features for the computer vision tasks. Besides, self-attention [54] is widely used to model internal representations and capture feature interdependencies by attending importance to a group of feature maps to select more informative features. Usually, depth map mainly contains smooth areas separated by a few depth boundaries. What really affects the depth quality is the sharpness of depth boundaries, but not the smooth areas. This motivates us to use feedback connections to effectively capture the high-frequency features around depth boundaries. Moreover, we resort to spatial attention mechanism to automatically highlight and strengthen the extracted high-frequency features. By combining the above two techniques, we develop a novel backbone network focusing on the capability of recovering depth details, and thus facilitating the depth map recovery. Besides, to better extract the multi-scale and color features to assist the depth recovery, we construct two separate branches to form a parallel learning architecture in which each branch is accurate and specialized on capturing either multi-scale or color information. We give a detailed analysis to verify the effectiveness of our design methodology in Sec. IV and present more in-depth discussion about our network through visualizing the intermediate features in Sec. VI.

B. Scope and Contributions

This paper presents a progressive multi-branch aggregation network (PMBANet) for depth SR, which consists of stacked multi-branch aggregation (MBA) blocks to progressively recover the degraded depth map, as shown in Fig. 1. Each MBA block has multiple parallel branches: 1) reconstruction branch (RB), which is designed via the attention-based error feed-forward/back modules. It iteratively exploits and compensates the downsampling errors to refine the depth map by imposing the attention mechanism on the feed-forward/-back process to gradually highlight the informative features at depth boundaries. 2) guidance branch (GB), including a multi-scale branch and a color branch, which is formulated as a separate subnetwork to help RB to recover the depth details. The multi-scale branch is to learn a multi-scale representation that pays close attention at objects of different scales, while the color branch regularizes the depth map by using auxiliary color information based on the internal structural correlation between depth-color pairs. Then, a fusion block is introduced to adaptively fuse and select the discriminative features from all the branches. Extensive experiments on benchmark datasets demonstrate the superiority of our method. Our main contributions are summarized as follows:

1) The proposed parallel architecture inherits the advantage of ensemble learning, which can learn effective and diverse features from each branch. Fusion blocks are progressively used to aggregate information by adaptively attending importance to the features from multiple branches.

2) A novel backbone network (RB) is designed based on the feed-forward/-back connections and attention mechanism to boost the high-resolution representations.

3) A separate multi-scale branch is constructed with a recombination of dense connections and dilated convolutions to better capture the multi-scale information.

4) From experiments (Sec. V-C3 and Sec. VI), we also verify that color information is only suitable to be introduced in earlier stages to help depth reconstruction. Besides, it can offer significant assistance and improve the performance obviously for the $8\times/16\times$ cases, but is not helpful, or even harmful for the easily recovered $2\times/4\times$ cases, which provides some new insights for the future work.

II. Related Work

We first present an overview of CNN-based depth SR methods in Sec. II-A. Then, all the related techniques, i.e., feedback mechanism, self-attention and multi-scale methods, are briefly reviewed in Sec. II-B - II-D, respectively.

A. Depth map Super-Resolution

Most existing depth SR methods use color information as guidance to recover the degraded depth maps. Li *et al.* [36] employed a two-path CNN to obtain the HR depth map which is designed based on the concept of joint filters. A simple fusion branch is added to jointly filter the informative feature maps from both depth and color branches. Hui *et al.* [30] also proposed a gradual up-sampling method with a hierarchical color guidance module, which further exploits the dependency between color and depth structure to resolve ambiguity in depth SR. Ye *et al.* [59] proposed a depth SR network to learn a binary map of depth edges from LR depth map and the corresponding HR color image, and then recovered the HR depth map based on a edge-guided bilateral filter. Wen *et al.* [53] used the color information as guidance to infer a initial HR depth map, then proposed a coarse-to-fine networks to progressively optimize the depth map, which can alleviate texture-copying artifacts and preserve edge details effectively. Pan *et al.* [43] proposed to model the structural information of both the guidance and input image by estimating the spatially variant linear representation coefficients. Lutio *et al.* [15] proposed to find a transformation from the guide image to the target HR depth map, which can be regarded as a pixel-wise translation. Kim *et al.* [32] learned explicitly sparse and spatially-variant kernels by a deformable kernel networks for guided depth map upsampling.

To conclude, color information brings significant improvement for depth SR, but may introduce texture-copying artifacts due to the depth-color inconsistency. We also stand on the color-guided category, and use the attention mechanism to fuse the color information under our multi-branch architecture. We additionally verify the appropriate position to integrate

color information into the network and the suitability for a given downsampling case to use color information as guidance, which are ignored by existing methods.

B. Feedback Mechanism

Feedback mechanism, also called back-projection in traditional algorithms, has been applied to various computer vision tasks. At first, back-projection [31] was proposed for the image registration problem in which it iteratively increases the image resolution with sub-pixel accuracy based on the feedback mechanism. Carreira *et al.* [3] proposed an iterative error feedback mechanism by iteratively estimating and applying a correction estimation to the current one. In a few recent studies, feedback mechanism has showed excellent ability in the task of image SR. Zhang *et al.* [64] proposed a model-based optimization method to solve the image SR problem, in which data term and smooth term are optimized by alternately updating a back-projection step and and a well-trained CNN denoising step, respectively. Han *et al.* [25] applied a delayed feedback mechanism which could transmits the information between two recurrent states in a dual-state RNN. Haris *et al.* [26] exploited iterative back-projection units, providing a error feedback connection to progressively upsample the image. Based on [26], Liu *et al.* [40] proposed a novel dual residual connection network which exploits the potential of paired operations, e.g., up-/down-samplings or large/small convolutional kernels, to facilitate several tasks of image restoration. Zamir *et al.* [62] and Li *et al.* [37] both work in a top-down manner, carrying high-level features back to previous layers in the next iteration and refining the low-level encoded features through the hidden states in an RNN.

Note that, the methods [37], [40], [62] and ours are all proposed based on feedback mechanism, but are implemented in different ways. [62] and [37] use hidden states in an RNN to achieve the feedback manner, while [40] uses the proposed 'dual residual connection' to realize the feature reuse between paired operations from different stages. In contrast, we design our feedback modules through the alternating process of attention feed-forward and feedback in an iterative 'High-to-Low' and 'Low-to-High' fashion, in which the high-frequency features are extracted by the squeeze-and-expand strategy and further strengthened by the self-attention operation. Our design is suitable for the task of depth SR, which can better deal with the recovery of high-frequency depth boundaries.

C. Self-Attention Mechanism

Self-attention mechanism steams from human perception and visual system, and has recently been widely used in computer vision to model internal representations by selectively focusing on useful high-level information to guide the network learning. Wang *et al.* [51] proposed a residual attention network which generates attention-aware features from stacking modules to learn more discriminative feature representation. SENet [28] focused instead on the channel-wise feature responses by explicitly modelling interdependencies between channels. CBAM [54] proposed a convolutional block attention module to effectively infer the attention maps along channel and spatial. Besides, attention are widely used in image SR. Hu *et al.* [29] constructed a set of channel-wise and spatial attention residual blocks to dynamically modulate multi-level features in global and local manners. Zhang *et al.* [65] used the channel attention mechanism to adaptively recalibrate the importance of each channel. Dai *et al.* [14] used attention modules to efficiently exploit the feature correlations in spatial and channel dimensions for stronger feature expression. They further proposed a second-order attention network for more powerful feature expression and feature correlation learning [13].

Usually, depth map mainly contains smooth areas separated by a few depth boundaries. For the task of depth SR, high-frequency depth boundaries are generally hard to reconstruct compared with the smooth areas in a depth map. Note that, the spatial attention can automatically highlight and strengthen the high-frequency features around the regions of depth boundaries. Therefore, we effectively combine the feed-forward/-back connections and spatial attention mechanism, and propose a novel attention-based error feed-forward/-back module to excavate informative features at depth boundaries for depth SR. Besides, our proposed fusion block aggregates multi-branch information by adaptively attending importance to all the features based on channel-wise attention mechanism.

D. Multi-Scale Methods

To achieve precise detection, recognition, or even pixel-level regression, it is necessary to develop multi-scale techniques to enhance the ability of feature representations for objects at different scales. Some methods aim at exploiting multi-scale information use multi-scale inputs [17], the encoder-decoder structures with long connections (U-net) [57], or the recurrent models [6]. For example, Eigen *et al.* [17] proposed a three-scale pyramid CNN as a coarse-to-fine manner for regressing dense depth maps. Feature pyramid network [38] exploited the inherent pyramidal hierarchy of deep convolutional networks to construct feature pyramids for object detection. Another mainstream multi-scale technique is to integrate the pyramid dilated convolution modules [5] [19] into the backbone network from a simple cascaded way and extract features from a limited number of fields-of-views at multiple sampling rates for capturing image context.

In general, all of the above methods combine the multi-scale techniques into the backbone network to help improve the network representation. Different from them, we construct an independent branch from the backbone network to model the multi-scale problem separately, which extracts the multi-scale feature representation more efficiently. Besides, through the effective recombination of dense connections and dilated convolutions, we can obtain more detailed and delicate multi-scale information than previous pyramid pooling modules.

III. Progressive Multi-Branch Aggregation Network

Fig. 1 illustrates the overview of our proposed Progressive Multi-Branch Aggregation Network (PMBANet). Let D_{LR} be the input LR depth map with the size of $w \times h \times 1$ and I_{HR} be

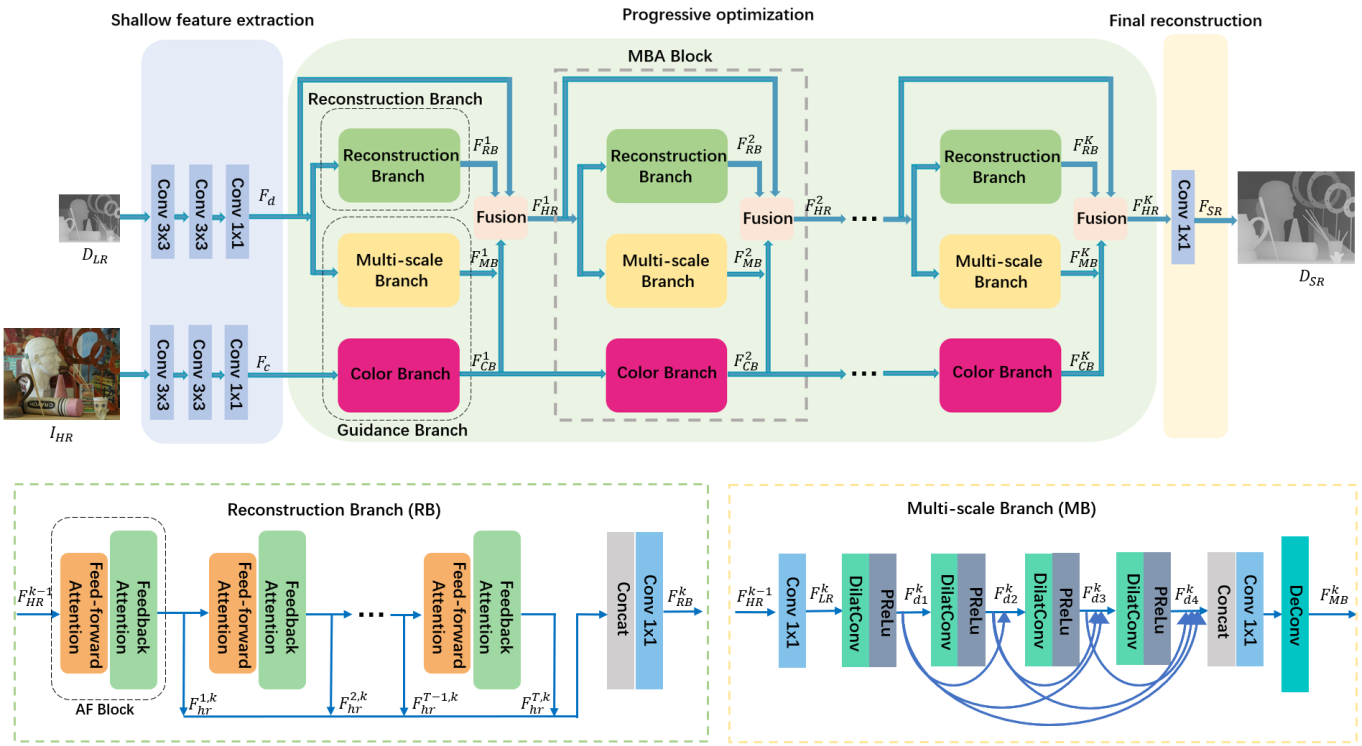


Fig. 1. Network architecture of the proposed PMBANet. To better present the whole framework and implementation details, different colored rectangles are used to represent different stages and different operations in each stage.

the input HR color image with the size of $W \times H \times 3$. The goal is to output the corresponding HR version D_{SR} with the size of $W \times H \times 1$. Note that $W = r*w$ and $H = r*h$, where r is the up-scaling factor. The whole PMBANet is mainly divided into three stages: shallow feature extraction stage, progressive optimization stage and final reconstruction stage.

Shallow feature extraction. Before entering the core progressive optimization stage, we firstly extract initial depth features F_d and color features F_c by shallow convolution layers from D_{LR} and I_{HR}, respectively. Note that, F_d is sent into both reconstruction branch and guidance branch in the next stage as input, while F_c is only sent into guidance branch.

Progressive optimization. Progressive optimization stage consists of K stacked MBA blocks in which each one has two parallel branches, i.e, a reconstruction branch (RB) and a guidance branch (GB). Then, a fusion block is used to adaptively fuse and select the discriminative features from both branches. Through stacked MBA blocks, the missing depth details in HR feature space are progressively recovered.

Final reconstruction. The target super-resolved depth map D_{SR} is reconstructed by using an 1×1 convolution on the output feature map of the last MBA block.

A. Reconstruction Branch

Motivated by the feedback mechanism [3], [26] and self-attention [54], we propose a novel attention-based error feed-forward and feedback module, called attention feed-forward/-back (AF) block to construct our RB, as shown in Fig. 1. In our scenario, feed-forward and feedback can be regarded as the flows of 'High-to-Low' and 'Low-to-High' respectively. Specifically, the attention feed-forward module enhances the feature representations at depth boundaries through projecting HR representations to LR spatial domain and highlights the high-frequency features in the LR space. In contrast, the attention feedback module maps the LR features back into the HR spatial domain and further strengthens the high-frequency features in the HR space. Through the alternating process of attention feed-forward and feedback, the branch gradually makes the reconstruction errors smaller, so as to better recover the depth details.

As shown in Fig. 2, our AF block consists of two stages: attention feed-forward that highlights the informative features in LR domain, and attention feedback that further strengthens the effective features in HR domain. The attention feed-forward in the t-th ($t < T$) AF block is defined as follows:

$$F^t_{lr1} = Down^t(F^{t-1}_{hr}). \tag{1}$$
$$F^t_{pooling} = Avgpool^t(F^t_{lr1}, j). \tag{2}$$
$$F^t_{lr2} = Up^t(F^t_{pooling}). \tag{3}$$
$$W^t_{Res1} = PRelu^t(F^t_{lr2} - F^t_{lr1}). \tag{4}$$
$$F^t_{lr3} = F^t_{lr1} + \gamma(F^t_{lr1} * W^t_{Res1}). \tag{5}$$

where the $Down(\cdot)$ and $Up(\cdot)$ are convolution and deconvolution operations, respectively. $Avgpool(\cdot, k)$ is the average pooling operation with the pooling factor j, while $PRelu(\cdot)$ represents the parametric rectified linear unit.

At the t-th AF block, it takes previous output feature map F^{t-1}_{hr} from $(t-1)$-th AF blocks with the size of $H \times W \times C$ as input. First, we downsample F^{t-1}_{hr} into LR feature map F^t_{lr1} with the size of $h \times w \times C$ and apply average pooling to F^t_{lr1}. The pooling result has the size of $h/j \times w/j$. Then,

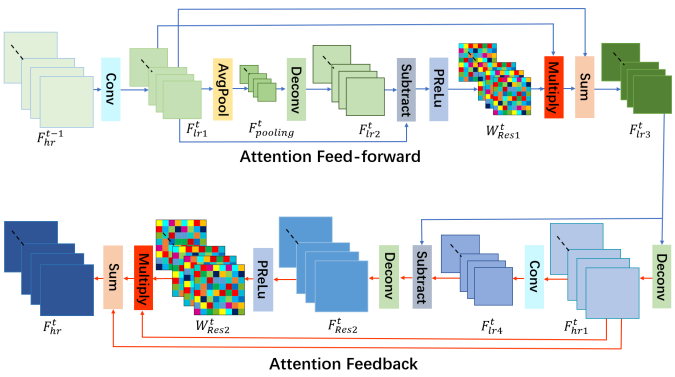


Fig. 2. Network architecture of the proposed AF block

we upsample $F^t_{pooling}$ to the size of $h \times w$. Next, the attention map W^t_{Res1} is calculated by computing the residual between the F^t_{lr2} and F^t_{lr1} activated by $PRelu$. Finally, to highlight the high-frequency regions in F^t_{lr1}, the attention operation is defined as Eq. (5), where $*$ is element-wise multiplication and γ is a hyper-parameter that affects the importance of attention weights.

After the attention feed-forward stage, attention feedback is concatenated to reconstruct the depth details in HR domain, which is defined as follows:

$$F^t_{hr1} = Up^t(F^t_{lr3}). \tag{6}$$
$$F^t_{lr4} = Down^t(F^t_{hr1}). \tag{7}$$
$$F^t_{Res2} = Up^t(F^t_{lr4} - F^t_{lr3}). \tag{8}$$
$$W^t_{Res2} = PRelu^t(F^t_{Res2}). \tag{9}$$
$$F^t_{hr} = F^t_{hr1} + \gamma(F^t_{hr1} * W^t_{Res2}). \tag{10}$$

At the t-th iteration, we take the output F^t_{lr3} of attention feed-forward stage as input, and map it to HR features F^t_{hr1}. Then we map F^t_{hr1} back to LR feature map F^t_{lr4}. The residual F^t_{Res2} is computed between F^t_{lr3} and F^t_{lr4} and then mapped to HR feature space. The attention map W^t_{Res2} is calculated by applying PReLu on F^t_{Res2} and then employed to highlight the informative features (Eq. (10)) to obtain the final output F^t_{hr}.

Finally, we concatenate T HR feature maps from all the AF blocks, and use an 1×1 convolution to output the final feature map F^k_{RB} for the k-th MBA block with the size of $H \times W \times C$:

$$F^k_{RB} = Conv^k_{1\times1}([F^{1,k}_{hr}, F^{2,k}_{hr}, ..., F^{T,k}_{hr}]). \tag{11}$$

where $[\cdot]$ denotes concatenation operation.

Note that, the goal of using a combination of average pooling and de-convolution is to effectively extract the high-frequency features in LR spatial domain. We first use the 'Avgpool' operation to squeeze the features, and expand them by the 'Deconv' operation, which can be regarded as an operation of image blurring. Thus, the high-frequency features are then extracted by the subtraction between the original features and the blurred ones, and are finally highlighted by the attention operation. Similar techniques are also used in our attention feedback block (HR space). Both squeeze-and-expand and attention operations contribute to the extraction of high-frequency features.

B. Guidance Branch

We formulate our GB as a separate parallel subnetwork, including a multi-scale branch (MB) and a color branch (CB) that simultaneously extract multi-scale representation and color information efficiently, to help RB to recover the depth details.

(1) Multi-Scale Branch. The sufficient multi-scale information is crucial to achieve high accuracy reconstruction for depth SR when addressing the different impact to multi-scale objects caused by downsampling degradation. As shown in Fig. 1, MB consists of a stack of four dilated convolution layers (DilatConv) followed by PReLu, an 1×1 convolution layer and a deconvolution layer (DeConv). all the DilatConvs have 3×3 kernels with the dilation factors set to 1, 2, 3 and 4, respectively. Dense connections are used to alleviate the vanishing-gradient problem and make use of all the features from different stages, which can obtain more detailed and delicate multi-scale information. Therefore, the input of each dilated convolution layer is the concatenation of the output from all previous dilated convolution layers:

$$F^k_{d_i} = DilatConv([F^k_{d_1}, ..., F^k_{d_{i-1}}], i), i = 2, 3, 4. \tag{12}$$

where $F^k_{d_i}$ is the feature map from the output of i-th dilation convolution. i is dilated factor. Finally, we map the features with different receptive fields to the HR features F^k_{MB} by the last deconvolution operation:

$$F^k_{MB} = DeConv^k(Conv^k_{1\times1}[F^k_{d_1}, F^k_{d_2}, F^k_{d_3}, F^k_{d_4}]) \tag{13}$$

(2) Color Branch. Since the HR color image can be easily obtained by consumer camera sensors in most cases, the available color image can be used as prior information to upsample the LR depth map, under the assumption of structural similarity between color-depth pairs. In the color branch, we extract the rich color features as additional prior knowledge for CB by a shallow CNN. We use three convolution layers with 3×3, 3×3 and 1×1 kernels continuously to obtain the color feature F^k_{CB} for the k-th MBA block. Note that the color features cannot be used for all the MBA blocks, which can lead to the texture copying or depth bleeding artifacts due to the depth-color inconsistency. Through the following experiments, we will verify that color information is only suitable to be introduced in earlier MBA blocks to avoid the above artifacts. Besides, we also demonstrate that it can offer significant assistance and improve the performance obviously for the higher upsampling cases, but is not helpful, or even harmful for the easily recovered lower upsampling cases.

C. Fusion

The goal of our fusion block is to effectively mine the relationship between feature channels in different branches, and then select useful feature channels to facilitate the depth SR. Channel attention [61] aims to learn a weight distribution of image features along the channel dimension, and apply the learned weights to the original feature channels to make the task focus on some key feature channels and ignore the unimportant ones, which is suitable for our fusion block that needs to automatically and adaptively aggregate different

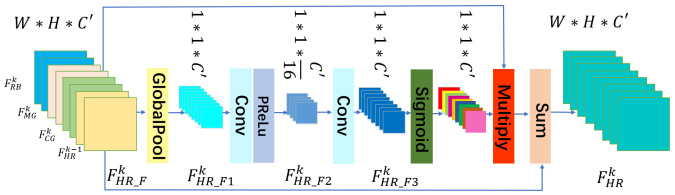


Fig. 3. The proposed fusion block.

feature channels from all the branches. We formulate our fusion block to fuse the features F^k_{RB}, F^k_{MB} and F^k_{CB} from all the branches in k-th MBA block and the output F^{k-1}_{HR} from $(k-1)$-th MBA block, as shown in Fig. 3. The fusion strategy is defined as follows:

$$F^k_{HR\_F} = [F^k_{RB}, F^k_{MB}, F^k_{CB}, F^{k-1}_{HR}]. \tag{14}$$

$$F^k_{HR\_F1} = Globalpool^k(F^k_{HR\_F}). \tag{15}$$

$$F^k_{HR\_F2} = Conv^k_1(F^k_{HR\_F1}). \tag{16}$$

$$F^k_{HR\_F3} = Conv^k_2(F^k_{HR\_F2}). \tag{17}$$

$$F^k_{HR} = F^k_{HR\_F} + F^k_{HR\_F} * \sigma(F^k_{HR\_F3}). \tag{18}$$

where $Globalpool(\cdot)$ is a global pooling operation to generate a feature vector $F^k_{HR\_F1}$. We use two convolution operations to further obtain statistic correlation of each channel in $F^k_{HR\_F}$. σ is sigmoid gateway. Finally, the channels of F_{HR_F} get reweighted as shown in Eq. (18), and we obtain the final feature map F^k_{HR}.

Note that all branches work complementarily, and we conduct repeated information aggregation by fusing the parallel multi-branch networks over and over through the whole process to boost the high-resolution representations.

IV. Design Methodology

Traditionally, color-guided depth SR is formulated as an optimization problem, which includes a fidelity term and a prior term to make the ill-posed problem well constrained. It can be roughly summarized into the following optimization function:

$$x^* = \arg\min_x \frac{1}{2}\|y - Kx\|^2 + \lambda \sum_l \omega_l * \rho_l(f_l \otimes x) \tag{19}$$

where K is a downsampling degradation matrix, λ is a trade-off parameter. f_l and $\rho_l(\cdot)$ are a set of filtering kernels and penalty functions, respectively. ω is the weighting matrix computed from the corresponding color image. $\otimes$ is the convolution operator.

A key point for these methods is to design any component in the prior term, e.g., total variation (TV) [60], TGV [18] or RBF [22] for the kernel f, L1 norm [60] or Welsch's function [33] for the penalty function $\rho(\cdot)$, and nonlocal filter [58] or anisotropic diffusion [18] for computing the weight ω. However, these hand-crafted priors cannot approach the real image prior. There have been several attempts to incorporate plug-and-play denoisers into model-based optimization methods and deduce particular iterative schemes [10] based on a given algorithm, e.g., gradient descent [22], HQS [64] or ADMM [4], to calculate the solution. For example, we may utilize a coordinate descent flow to minimize the above energy, i.e.,

$$x^{t+1} = x^t + \mathcal{R}_f(x) + \lambda \mathcal{R}_p(x) \tag{20}$$

where $\mathcal{R}_f(x)$ and $\mathcal{R}_p(x)$ are gradients or residuals with respect to the energies of fidelity term and prior term, which can be regarded as a progressive residual learning that iteratively applys the fidelity and prior residuals on x^t to find a new x^{t+1}.

Usually, $\mathcal{R}_p(x)$ is replaced by a off-the-shelf denoiser, e.g., BM3D [34] or a more robust CNN denoiser [64]. Meanwhile, $\mathcal{R}_f(x)$ is computed through least squares or using iterative back-projection to refine the result x^t without explicitly computing the inversion of K [12] [64]. However, the unknown of K, e.g., including some mixed degradations (noise, depth missing and downsampling), leads to difficulties in modelling the fidelity term with limited expressivity. Second, existing CNN denoisers are not specifically designed for depth SR, in which the use of color and multi-scale information are often ignored.

Motivated by the above analysis, we pursue better architectural design aiming at further improvements for depth SR. First, we formulate two parallel branches, i.e., RB and GB, to separately model the fidelity and prior residuals, respectively. Instead of a simple summation with a parameter λ, we introduce a fusion block to aggregate both branches. Through iterative optimization and repeated aggregation, we progressively recover the degraded depth map. Next, we design RB based on back-projection technique and self-attention mechanism to better exploit the HR features. Instead of hard-coding the fidelity term, let the network to freely learn in what point the forward operator K should be evaluated. For GB, to better characterize the filter bank f_l, we design a multi-scale branch based on different dilated kernels with dense connections to extract various features with different field of views. PReLu is used after each dilated convolution as a nonlinear penalty. Besides, color branch is also used as prior knowledge to further exploit color features to help recover the degraded depth map. GB is dynamically adjusted to account for the updating together with RB. In general, the design methodology of our whole network originates from the model-based optimization methods, but simultaneously models the fidelity term and the prior term with deep networks.

A similar work related to us is DG-CNN (dynamic guidance with CNN nonlinearity parametrization) [21], which is also designed based on the optimization models. It unfolds the optimization process, then uses a simple CNN that contains two separate encoders and a shared decoder to parameterize the stage-wise operation. In each stage, LR depth map and color image are feed into their corresponding encoders and the intermediate depth map is obtained from the shared decoder. In contrast, we also use the stage-wise learning framework, but design a more sophisticated and professional parallel network architecture in each stage, which can learn effective and diverse features (including high-frequency, multi-scale, and color features) from every branch, and adaptively aggregate all the branch in the ensemble.

TABLE I
QUANTITATIVE DEPTH UPSAMPLING RESULTS ON MIDDLEBURY 2005 DATASET. (LOWER MAD AND PE VALUES, BETTER PERFORMANCE)

	Art			*Books*			*Dolls*			*Laundry*			*Moebius*			*Reindeer*		
	×4	×8	×16	×4	×8	×16	×4	×8	×16	×4	×8	×16	×4	×8	×16	×4	×8	×16
CLMF [41]	0.76/8.12	1.44/17.28	2.87/33.25	0.28/3.27	0.51/7.25	1.02/16.09	0.34/4.40	0.60/8.76	1.01/18.32	0.50/5.50	0.80/12.67	1.67/25.40	0.29/4.13	0.51/8.42	0.97/17.27	0.51/4.65	0.84/9.96	1.55/18.34
JGF [39]	0.47/3.25	0.78/7.39	1.54/14.31	0.24/2.14	0.43/5.41	0.81/12.05	0.33/3.23	0.59/7.29	1.06/15.87	0.36/2.60	0.64/4.54	1.20/8.69	0.25/3.36	0.46/6.45	0.80/12.33	0.38/2.27	0.64/5.17	1.09/11.84
EDGE [44]	0.65/6.82	1.03/13.49	2.11/25.90	0.30/3.35	0.56/8.50	1.03/19.32	0.31/2.90	0.56/6.84	1.05/17.97	0.32/2.82	0.54/5.46	1.14/13.57	0.29/3.72	0.51/7.36	1.10/14.05	0.37/2.67	0.63/6.22	1.28/16.80
TGV [18]	0.65/5.14	1.17/10.51	2.30/21.37	0.27/2.48	0.42/4.65	0.82/11.20	0.33/4.45	0.70/11.12	2.20/45.54	0.55/6.99	1.22/16.32	3.37/53.61	0.29/3.68	0.49/6.84	0.90/14.09	0.49/4.67	1.03/11.22	3.05/43.48
KSVD [63]	0.64/3.46	0.81/5.18	1.47/8.39	0.23/2.13	0.52/3.97	0.76/8.76	0.34/4.53	0.56/6.18	0.82/12.98	0.35/2.19	0.52/3.89	1.08/8.79	0.28/2.08	0.48/4.86	0.81/8.97	0.47/2.19	0.57/5.76	0.99/12.67
CDLLC [55]	0.53/2.86	0.76/4.59	1.41/7.53	0.19/1.34	0.46/3.67	0.75/8.12	0.31/4.61	0.53/5.94	0.79/12.64	0.30/2.08	0.48/3.77	0.96/8.25	0.27/1.98	0.46/4.59	0.79/7.89	0.43/2.09	0.55/5.39	0.98/11.49
PB [42]	0.79/3.12	0.93/6.18	1.98/12.34	0.16/1.39	0.43/3.34	0.79/8.12	0.53/3.99	0.83/6.22	0.99/12.86	1.13/2.68	1.89/5.62	2.87/11.76	0.17/1.95	0.47/4.12	0.82/8.32	0.56/6.04	0.97/12.17	1.89/21.35
EG [56]	0.48/2.48	0.71/**3.31**	1.35/**5.88**	0.15/1.23	0.36/3.09	0.70/7.58	0.27/2.72	0.49/5.59	0.74/12.06	0.28/1.62	0.45/2.86	0.92/7.87	0.23/1.88	0.42/4.29	0.75/7.63	0.36/1.97	0.51/4.31	0.95/9.27
SRCNN [16]	0.63/7.61	1.21/14.54	2.34/23.65	0.25/2.88	0.52/7.98	0.97/15.24	0.29/3.93	0.58/8.34	1.03/16.13	0.40/6.25	0.87/13.63	1.74/24.84	0.25/3.63	0.43/7.28	0.87/14.53	0.35/3.84	0.75/7.98	1.47/14.78
DSP [52]	0.73/7.83	1.56/15.21	3.03/31.32	0.28/3.19	0.61/8.52	1.31/16.73	0.32/4.74	0.65/9.53	1.45/19.37	0.45/6.19	0.98/12.86	2.01/22.96	0.31/3.89	0.59/8.23	1.26/16.58	0.42/3.59	0.84/7.23	1.73/14.12
ATGVNet [45]	0.65/3.78	0.81/3.78	1.42/9.68	0.43/5.48	0.51/7.16	0.79/10.32	0.41/4.55	0.52/6.27	**0.56**/12.64	0.37/2.07	0.89/3.78	0.94/8.69	0.38/3.47	0.45/4.81	0.80/8.56	0.41/3.82	0.58/5.68	1.01/12.63
MSG [30]	0.46/2.31	0.76/4.31	1.53/8.78	0.15/1.21	0.41/3.24	0.76/7.85	0.25/2.39	0.51/4.86	0.87/9.94	0.30/1.68	0.46/2.78	1.12/7.62	0.21/1.79	0.43/4.05	0.76/7.48	0.31/1.73	0.52/2.93	0.99/7.63
DGDIE [22]	0.48/2.34	1.20/13.18	2.44/26.32	0.30/3.21	0.58/7.33	1.02/14.25	0.34/4.79	0.63/9.44	0.93/11.66	0.35/2.03	0.86/3.69	1.56/16.72	0.28/1.98	0.58/8.11	0.98/16.22	0.35/1.76	0.73/7.82	1.29/15.83
DEIN [59]	0.40/2.17	0.64/3.62	1.34/6.69	0.22/1.68	0.37/3.20	0.78/8.05	0.22/1.73	0.38/3.38	0.73/9.95	0.23/1.70	0.36/3.27	0.81/7.71	0.20/1.89	0.35/3.02	0.73/7.42	0.26/1.40	0.40/2.76	0.80/5.88
CCFN [53]	0.43/2.23	0.72/3.59	1.50/7.28	0.17/1.19	0.36/3.07	0.69/7.32	0.25/1.98	0.46/4.49	0.75/9.84	0.24/1.39	0.41/2.49	**0.71**/7.35	0.23/2.18	0.39/3.91	0.73/7.41	0.29/1.51	0.46/2.79	0.95/6.58
GSRPT [15]	0.48/2.53	0.74/4.18	1.48/7.83	0.21/1.77	0.38/4.23	0.76/7.67	0.28/2.84	0.48/4.61	0.79/10.12	0.33/1.79	0.56/4.55	1.24/8.98	0.24/2.02	0.49/4.70	0.80/8.38	0.31/1.58	0.61/5.90	1.07/10.35
Ours	**0.26**/**1.95**	**0.51**/3.45	**1.22**/6.28	**0.15**/**1.13**	**0.26**/**2.87**	**0.59**/**6.79**	**0.19**/**1.35**	**0.32**/**3.22**	0.59/**8.92**	**0.17**/**1.27**	**0.34**/**2.41**	**0.71**/**6.88**	**0.16**/**1.21**	**0.26**/**2.87**	**0.67**/**6.73**	**0.17**/**1.28**	**0.34**/**2.40**	**0.74**/**5.66**

V. EXPERIMENTAL RESULTS

In this section, the implementation details are given in (Sec.V-A). Our proposed method is first evaluated on the performance of depth SR under different datasets (Sec.V-B). Then, ablation study is presented to analyze the design choices of the proposed scheme (Sec.V-C).

A. Implementation Details

During training, we use 36 RGB-D images (6, 21, 9 images from 2001 [2], 2006 [27] and 2014 [46] datasets, respectively) from Middlebury dataset[1]. To evaluate the performance of our PMBANet, we test on 6 standard depth maps (*Art, Books, Moebius, Dolls, Laundry, Reindeer*) from Middlebury 2005 [47], 4 standard depth maps (*Tsukuba, Venus, Teddy, Cones*) from Middlebury 2003 [48]. We evaluate the generalization ability on 5 depth maps (*Alley_1-48, Ambush_2-15, Ambush_4-12, Ambush_5-41, Temple_3-23*) from MPI dataset [2] and 3 real depth maps captured by ToF sensor from ToFMark dataset [18]. Another training and testing dataset is NYU v2 RGB-D dataset [49] captured from Kinect. Following the common splitting method, we use the first 1000 images of the NYU dataset as training data, and evaluate on the last 449 images. To produce LR depth maps, we downsample the HR depth maps to the target size using Bicubic interpolation. We augment the training dataset by 180-rotation and randomly extracted 10000+ depth patches of a fixed size of 16×16 from LR depth maps. The corresponding HR depth patches are the squared size of 32, 64, 128, and 256 according to 2, 4, 8, and 16 upscaling factors respectively. Similar to other works, the metric of Mean Absolute Difference (MAD), Root Mean Square Error (RMSE), and percentage of error pixels (PE) [53] is used to measure the difference between the predicted depth map and the corresponding ground truth.

During training, we set the number of MBA blocks as $K = 3$ and the number of AF blocks in each MBA block as $T = 4$. The ablation study presented below will demonstrate the effectiveness of our configurations. In each AF block, we set the pooling factor as $j = 4$ and the hyper-paramter γ to 0.1. At the both attention feed-forward and attention feedback stages, we used the kernel size of 6×6, 8×8, 12×12 and 20×20 with a stride size of 2, 4, 8 and 16 for upsampling (deconvolution) and downsampling (convolution) operations in 2×, 4×, 8× and 16× upsampling cases, respectively.

Our models are trained end-to-end using L1 loss between the predicted HR depth map and ground truth. For optimization, we used Adam optimizer with momentum = 0.9, β_1 = 0.9, β_2 = 0.99 and $\epsilon = 10^{-8}$. The initial learning rate is set to 0.0001 and decreased by multipling by 0.1 for every 100 epochs. We implemented our models with PyTorch framework and trained them on a NVIDIA 1080Ti GPU.

B. Performance Comparison

1) ***Middlebury dataset (Noiseless Case):*** To demonstrate the capacity of our proposed PMBANet, we compare with other state-of-the-art depth SR methods under different upscaling factors (4×, 8× and 16×) in Table I [3], Note that, from all the compared methods, deep learning based methods are SRCNN [16], DSP [52], ATGVNet [45], MSG [30], DGDIE [22], DEIN [59], CCFN [53], GSRPT [15] and ours, which are all trained an tested on the same datasets for fairly comparison.

As shown in Table I (objective results on Middlebury 2005), traditional filtering or optimization based methods obtain relatively higher MAD and PE values compared to the CNN-based methods. Compared among these CNN-based methods, our PMBANet almost obtains the best objective results, especially for the 8× and 16× cases which is more difficult to recover. Similar conclusion can be obtained in Table II, which shows the evaluation on Middlebury 2003. Fig. 4 further demonstrates the visual performance of our method under the 8× case. Obviously, we obtain the most similar reconstruction results compared to ground truth depth patches in terms of structure and details. Notice that for the large scene objects in 2rd and 4th rows, all the methods present similar

[1]Middlebury datasets, http://vision.middlebury.edu/.
[2]MPI Sintel datasets, http://sintel.is.tue.mpg.de/.
[3]The cases of 2× are omitted to save space.

TABLE II
QUANTITATIVE DEPTH UPSAMPLING RESULTS ON MIDDLEBURY 2003 DATASET. (LOWER MAD AND PE VALUES, BETTER PERFORMANCE.)

	Tsukuba			*Venus*			*Teddy*			*Cones*		
	×2	×4	×8	×2	×4	×8	×2	×4	×8	×2	×4	×8
EDGE [44]	0.61/2.35	0.77/4.44	1.32/6.95	0.23/0.44	0.29/0.90	0.56/2.65	0.78/3.12	1.08/6.27	2.13/13.73	1.03/3.26	1.52/7.18	2.98/14.38
TGV [18]	0.53/1.79	0.71/3.08	1.18/5.31	0.17/0.41	0.24/0.60	0.43/1.76	0.75/2.31	0.83/3.72	1.62/7.51	0.83/2.54	1.13/4.34	2.23/8.17
KSVD [63]	0.51/2.48	0.66/4.30	1.09/6.78	0.23/0.59	0.30/1.22	0.59/3.15	0.70/2.97	0.92/5.17	2.07/8.93	0.91/3.97	1.15/6.45	2.28/12.51
CDLLC [55]	0.48/2.41	0.61/4.15	0.98/6.59	0.21/0.71	0.27/1.18	0.53/3.08	0.67/2.99	0.85/4.72	1.59/9.13	0.85/3.68	1.07/5.79	2.12/11.23
PB [42]	0.62/1.57	0.86/2.52	1.71/3.69	0.30/0.39	0.38/0.66	0.62/1.83	0.89/4.13	1.26/8.03	2.73/17.90	1.18/4.35	1.56/9.73	3.11/17.69
EG [56]	0.45/1.27	0.67/2.36	1.09/3.50	0.19/0.37	0.29/0.54	0.49/1.62	0.63/1.61	0.95/3.11	1.51/6.18	0.76/1.72	1.16/3.09	2.14/6.27
SRCNN [16]	0.64/2.99	0.79/5.52	1.43/8.64	0.28/0.71	0.34/1.30	0.61/3.23	0.88/3.98	1.10/6.92	2.35/14.12	1.12/4.99	1.41/8.64	2.91/16.18
DSP [52]	0.65/3.12	0.68/3.24	0.83/5.68	0.26/0.68	0.34/1.21	0.69/2.87	0.75/3.92	1.24/4.27	3.01/5.67	1.86/4.83	1.35/8.72	4.86/9.35
ATGVNet [45]	0.46/1.52	0.72/2.41	0.88/3.59	0.23/0.40	0.31/0.63	0.52/1.76	0.69/3.35	1.03/5.37	1.60/7.62	0.83/4.63	1.27/5.74	2.42/7.36
MSG [30]	0.41/1.22	0.62/2.21	0.75/3.44	0.14/0.35	0.34/0.51	0.57/1.58	0.65/1.59	0.82/3.07	2.76/3.67	0.73/1.71	1.06/2.92	2.22/3.71
DEIN [59]	0.40/1.19	0.58/1.98	0.63/2.24	0.10/0.29	0.21/0.42	0.36/1.11	0.64/1.72	0.73/2.55	1.25/2.48	0.69/1.44	0.92/2.17	1.87/3.40
CCFN [53]	0.39/1.16	0.61/2.18	0.71/3.42	0.12/0.33	0.25/0.51	0.44/1.56	**0.61/1.58**	0.79/2.98	1.42/3.58	0.71/1.64	1.05/2.89	2.09/3.70
GSRPT [15]	0.41/1.53	0.60/2.14	0.73/3.27	0.14/0.34	0.32/0.60	0.51/1.46	0.61/1.62	0.80/3.02	2.35/3.78	0.71/1.66	1.03/2.78	2.15/3.62
Ours	**0.35/1.14**	**0.50/1.92**	**0.58/2.04**	**0.08/0.26**	**0.19/0.35**	**0.33/1.04**	0.64/1.71	**0.70/2.45**	**1.20/2.24**	**0.67/1.39**	**0.87/2.00**	**1.78/3.32**

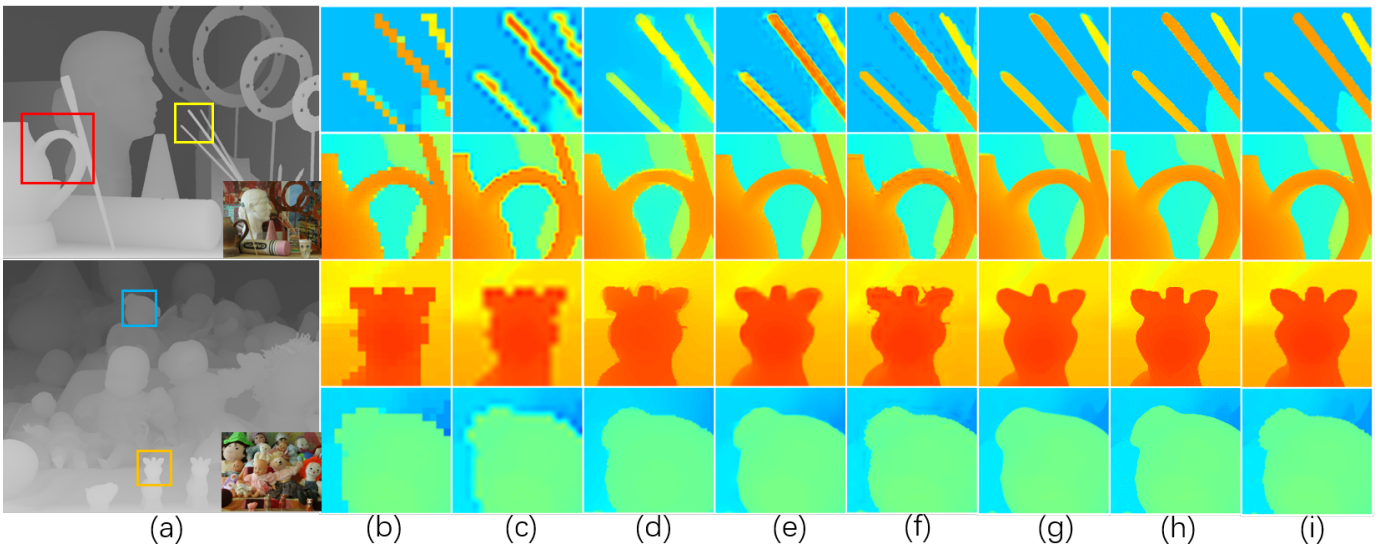


Fig. 4. Visual comparison of 8× upsampling results on two examples (*Art, Dolls*). (a) GT depth maps and color images; (b) LR; (c) Bicubic; (d) JGF [39]; (e) DGDIE [22]; (f) DEIN [59]; (g) GSRPT [15]; (h) PMBANet; (i) GT. Depth patches are enlarged and colored to enhance the contrast for clear visualization.

recovered results. However, for the tiny objects, i.e., the stick in *Art* and the toy's head in *Dolls*, the compared methods present obvious jaggy artifacts and wrong estimation on the stick and some blurring on the head, which demonstrates that the downsampling degradation brings more severe damages on fine structures and thus makes the recovery more difficult on these regions. In contrast, our method accurately and clearly recovers the depth boundaries of these tiny objects, and achieves the best performance.

*2) **Middlebury dataset (Noisy Case):*** The noisy datasets are built according to [58], where the authors first add Gaussian noise with a variance of 25 to the original Middlebury datasets, and then downsample the polluted datasets at four scales. We choose some methods that are also applied in noisy cases for comparison. The quantitative results for 2×. 4×, 8× and 16× noisy cases are shown in Table III. We can clearly see that our PMBANet can better deal with noisy removal when upsampling the depth maps, even compared to the CNN-based methods, i.e., MSG, DEIN, DGDIE and GSRPT. Fig. 5 further presents the qualitative results from 8× downsampling and noisy degradation. To conclude, filter based methods, i.e, JGF cannot remove the noise and generate blurring and cotton-like artifacts. TGV belongs to the category of optimization methods that inherits the advantage of exploiting global information, thus achieves better performance when addressing the problem of noise removal. Besides, the learning-based methods (DGDIE, GSRPT and our PMBANet) are qualified for the noise removal compared to previous methods. However, the results of DGDIE present excessive blurring. In contrast, GSRPT and our PMBANet can remove noise and keep the sharpest depth boundaries at the same time.

*3) **NYU datasset:*** Additionally, we evaluate on NYU dataset to demonstrate the effectiveness of our method. All the deep learning based methods (DJF [36], DGDIE [22], GbFT [1], PAC [50], SVLRM [43], DKN [32]) are trained and tested on NYU dataset with the same training-testing splitting method for fairly comparison. As shown in Table IV, our PMBANet obtains the best objective results for all the upsampling cases. Fig. 6 further demonstrates the visual performance of our method under the 8× case. Focusing on the yellow rectangle, our method successfully recovers the right depth information of the vase on the table. Besides, we achieve

TABLE III
QUANTITATIVE DEPTH UPSAMPLING RESULTS ON SYNTHETIC NOISY MILDDLEBURRY DATASET. (LOWER MAD VALUES, BETTER PERFORMANCE.)

	Art				*Books*				*Dolls*				*Laundry*				*Moebius*				*Reindeer*			
	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×
Bicubic	3.52	3.84	4.47	5.72	3.30	3.37	3.51	3.82	3.28	3.34	3.47	3.72	3.35	3.49	3.77	4.35	3.28	3.36	3.50	3.80	3.39	3.52	3.82	4.45
EDGE [44]	1.69	2.40	3.60	5.75	1.12	1.44	1.81	2.59	1.14	1.54	2.07	3.02	1.28	1.63	2.20	3.34	1.13	1.45	1.95	2.91	1.20	1.60	2.40	3.97
CLMF [41]	1.19	1.77	2.95	4.91	0.90	1.48	2.38	3.36	0.96	1.54	2.37	3.25	0.94	1.55	2.50	3.81	0.87	1.44	2.32	3.30	0.96	1.56	2.54	3.85
JGF [39]	2.36	2.74	3.64	5.46	2.12	2.25	2.49	3.25	2.09	2.22	2.49	3.25	2.16	2.37	2.85	3.90	2.09	2.24	2.56	3.28	2.18	2.40	2.89	3.94
TGV [18]	0.82	1.26	2.76	6.87	0.50	0.74	1.49	2.74	0.66	1.63	1.75	3.71	0.61	1.59	1.89	4.16	0.56	0.89	1.72	3.99	0.59	0.84	1.75	4.40
MSG [30]	0.58	0.84	1.57	2.98	0.46	0.62	1.18	1.48	0.59	0.84	1.12	1.78	0.51	0.78	1.03	1.89	0.48	0.66	1.13	1.76	0.45	0.57	1.12	1.87
DEIN [59]	0.91	1.32	2.44	4.24	0.48	0.73	1.44	2.38	0.64	1.54	1.55	2.45	0.61	1.49	1.77	3.20	0.52	0.78	1.64	3.29	0.52	0.77	1.46	3.87
DGDIE [22]	0.61	0.99	1.84	3.34	0.52	0.81	1.29	2.04	0.63	0.95	1.39	2.05	0.58	1.10	1.73	2.67	0.53	0.84	1.37	2.16	0.52	0.79	1.33	2.19
GSRPT [15]	0.46	0.68	1.33	2.47	0.38	0.52	0.87	1.37	0.56	0.78	1.26	2.03	0.53	0.76	1.24	1.86	0.45	0.65	1.03	1.68	0.48	0.55	1.04	1.70
Ours	**0.44**	**0.59**	**0.98**	**1.89**	**0.34**	**0.44**	**0.71**	**1.23**	**0.50**	**0.64**	**1.01**	**1.56**	**0.42**	**0.54**	**0.89**	**1.62**	**0.37**	**0.48**	**0.81**	**1.30**	**0.37**	**0.47**	**0.78**	**1.52**

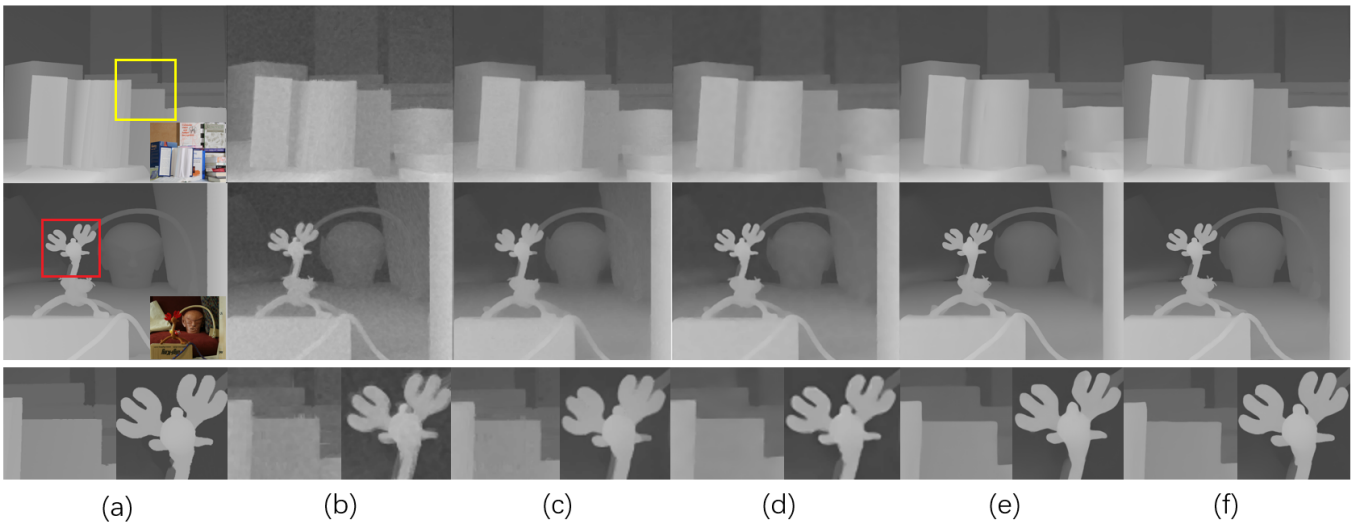


Fig. 5. Visual comparison for recovered depth maps from 8× downsampling and noisy degradation on two examples (*Books, Reindeer*): (a) GT depth maps and color images; (b) JGF [39]; (c) TGV [18]; (d) DGDIE [22]; (e) GSRPT [15]; (f) PMBANet.

TABLE IV
QUANTITATIVE DEPTH UPSAMPLING RESULTS (IN RMSE) ON REAL NYU DATASET. (LOWER RMSE VALUES, BETTER PERFORMANCE.)

Method	Bicubic	TGV [18]	EDGE [44]	DJF [36]	DGDIE [22]	GbFT [1]	PAC [50]	SVLRM [43]	DKN [32]	Ours
×4	8.16	6.98	5.21	3.54	1.56	3.35	2.39	1.74	1.62	**1.06**
×8	14.22	11.23	9.56	6.20	2.99	5.73	4.59	5.59	3.26	**2.28**
×16	22.32	28.13	18.10	10.21	5.24	9.01	8.09	7.23	6.51	**4.98**

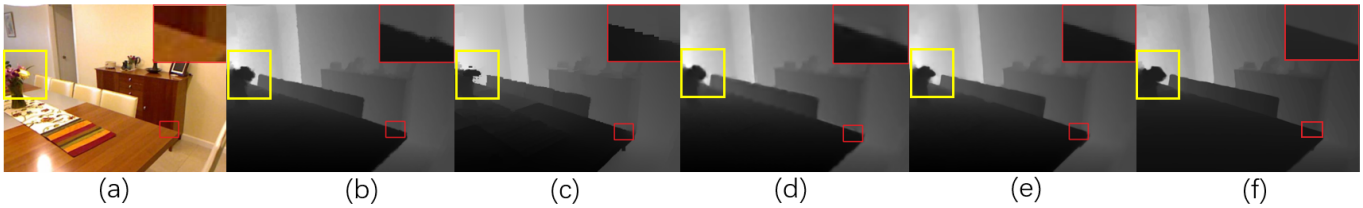


Fig. 6. Visual comparison for recovered depth maps from ×8 downsampling on NYU v2 dataset. (a) color image; (b) GT; (c) SDF [24]; (d) DJF [36] ; (e) SVLRM [43]; (f) PMBANet.

the sharpest and clearest results (red rectangle).

*4) **Evaluation on Generalization:*** we choose MPI dataset to evaluate the generalization and compare with DJF [36], MSG [30], DEIN [59], DGDIE [22] and GSRPT [15], from which their source codes are available. Table V presents the quantitative performance on the chosen five depth maps from MPI. DJF achieves comparable results with ours on the first two cases, but is totally inferior to us on the last three ones. Fig. 7 shows the visual comparison between GSRPT and ours. We obtain the right and clear depth boundaries in the recovered results (red cycles).

TABLE V
QUANTITATIVE DEPTH UPSAMPLING RESULTS ON MPI DATASET. (LOWER MAD VALUES, BETTER PEFORMANCE.)

	Alley_1-48				*Ambush_2-15*				*Ambush_4-12*				*Ambush_5-41*				*Temple_3-23*			
	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×
DJF [36]	**0.07**	**0.17**	0.46	0.90	**0.06**	**0.20**	0.48	**0.96**	0.21	0.54	1.14	2.49	0.28	**0.72**	1.42	2.67	0.15	0.40	0.79	1.76
MSG [30]	0.13	0.20	0.39	0.87	0.09	0.26	0.51	1.12	0.22	0.43	1.10	1.82	0.20	0.77	1.36	2.01	0.15	0.44	0.82	1.78
DEIN [59]	0.13	0.21	0.42	0.88	0.09	0.25	0.47	1.08	0.25	0.55	1.12	1.76	0.24	0.81	1.69	2.32	0.17	0.41	0.89	1.82
DGDIE [22]	0.23	0.18	0.44	**0.79**	0.23	0.21	0.65	1.24	0.23	0.57	1.26	2.23	0.23	0.73	1.79	3.10	0.23	0.40	1.01	1.90
GSRPT [15]	0.14	0.22	0.52	0.93	0.17	0.29	0.62	1.44	0.15	0.62	1.32	2.45	0.15	0.75	1.98	3.46	0.17	0.49	1.19	2.07
Ours	0.11	0.20	**0.38**	0.90	0.08	0.25	**0.45**	1.05	**0.14**	**0.50**	**0.92**	**1.71**	**0.14**	**0.72**	**1.21**	**1.90**	**0.13**	**0.38**	**0.74**	**1.72**

TABLE VI
QUANTITATIVE DEPTH UPSAMPLING RESULTS ON TOFMARK. (LOWER MAD VALUES, BETTER PEFORMANCE.)

Method	Bicubic	CLMF [41]	JGF [39]	TGV [18]	MSG [30]	DEIN [59]	DGDIE [22]	GSRPT [15]	PMBANet
Books	16.23	13.89	17.39	12.36	12.26	12.78	12.31	13.21	**12.08**
Shark	17.78	15.10	18.17	15.29	14.11	15.11	14.06	15.03	**10.15**
Devil	16.66	14.55	19.02	14.68	12.45	14.25	**9.66**	12.27	11.95

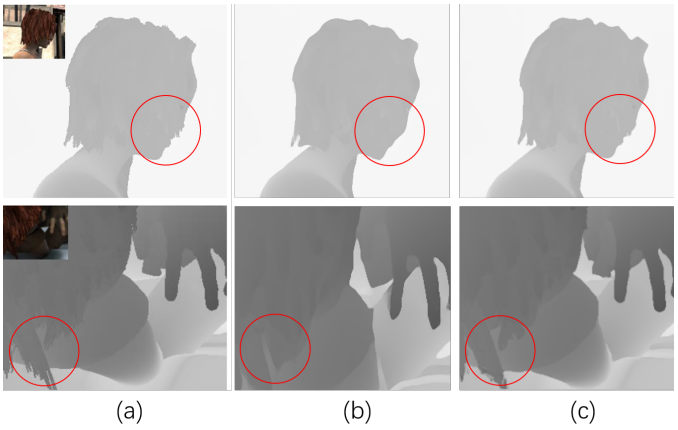


Fig. 7. Generalization on MPI dataset (8× cases on *Alley_1-48, Ambush_4-12*): (a) GT; (b) GSRPT [15] and (c) PMBANet.

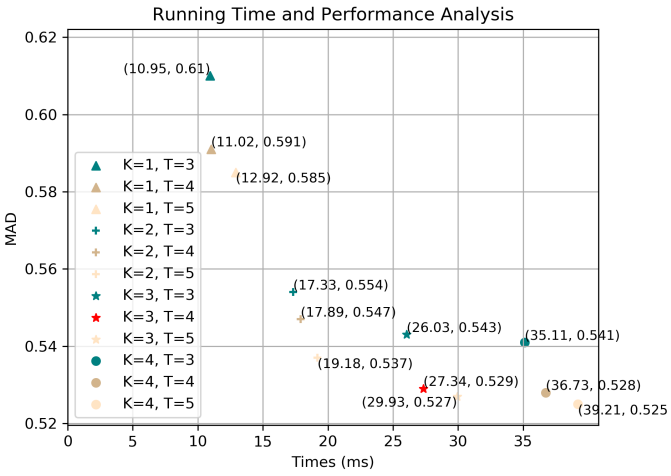


Fig. 8. Trade-offs between runtime and accuracy on 8× case under different combinations of K and T. The input image size is 1376×1104 (*Art*).

*5) **Evaluation on Real data:*** We evaluate the proposed method on ToFMark, which is a real ToF sensor dataset that contains only three test cases. Different from DGDIE that first fills the missing points in captured depth map and then synthesizes training datasets to train a new model by computing the noise distribution between depth pairs, we just fill the missing points and downsample the input by 2× rate, then directly send it into our model ('2× with noise') to acquire the final results. Other methods, i.e., MSG, DEIN, and GSPRT, are tested with the same strategy with ours. We compare the above deep learning based methods and some traditional methods shown in Table VI and Fig. 9. We achieve the best performance on *Books, Shark*, and relatively lower MAD values on *Devil* than DGDIE but higher than other mehods, including MSG, DEIN, GSPRT, and all the non-learning methods. Without reconstructing the training dataset, we also obtain satisfactory results. The most essential reason is that what we really learn is the mapping from the blurring on depth edges to the accurate edge location (high-frequency information), but not directly inferring the depth value of every pixel, especially on the depth smooth regions. Our model focuses on the learning of high-frequency features extraction through iterative attention feed-forward and feedback modules, which can support a great variety of test images that involve substantial appearance and geometric variations.

TABLE VII
ABLATION STUDY OF RECONSTRUCTION BRANCH TO VERIFY THE EFFECTIVENESS OF OUR DESIGNED ELEMENTS (8× CASE).

Method	MAD Values (the lower the better)					
	Art	*Books*	*Dolls*	*Laundry*	*Moebius*	*Reindeer*
1) w/o RB	1.212	0.841	0.512	0.742	0.499	0.663
2) RB→IBP	1.171	0.389	0.503	0.662	0.452	0.670
3) RB→LtH	0.553	0.290	0.367	0.397	0.289	0.375
4) RB→DBPN	0.540	0.277	0.358	0.395	0.281	0.368
5) S-E→Plain Residual	0.531	0.273	0.345	0.390	0.272	0.366
6) w/o Attention	0.540	0.276	0.354	0.390	0.279	0.368
Ours	**0.529**	**0.270**	**0.344**	**0.387**	**0.268**	**0.364**

C. Ablation Study

In this section, we further verify the key designed modules, i.e., reconstruction branch, multi-scale branch and color branch in our framework by ablation study.

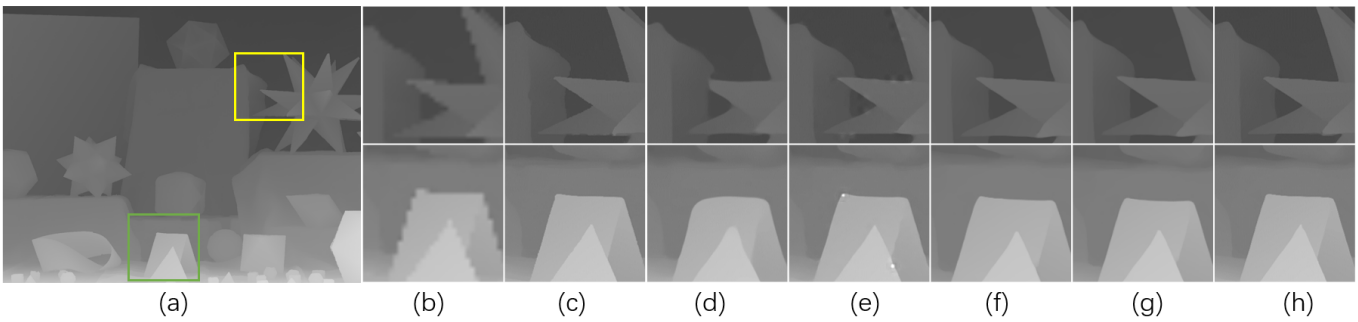


Fig. 10. Visual comparison between different backbone configurations: (a) GT depth map; (b) LR; (c) GT; Results obtained by (d) 'w/o RB (only multi-scale branch)', (e) 'RB $\rightarrow$ IBP', (f) 'RB $\rightarrow$ DBPN', (g) 'RB $\rightarrow$ Low-to-High network' and (h) Ours (RB with multi-scale branch).

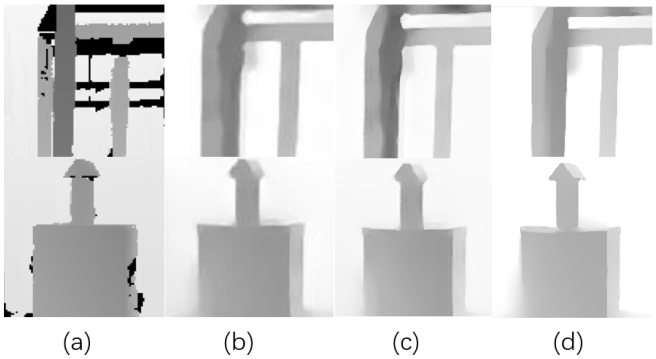


Fig. 9. Visual comparison on ToFMark. (a) GT; (b) DGDIE [22]; (c) GSRPT [15]; (d) ours. Depth pathes are shown for clear visualization.

*1) **Effectiveness of Reconstruction Branch:*** To evaluate the capability of RB, we firstly explore the influence of different configurations in RB, i.e., the number of MBA blocks (denoted as K) and the number of attention feed-forward/-back (AF) blocks (denoted as T) in each MBA block, which are the fundamental components of our RB. For easy comparison, we keep the multi-scale branch along with the RB, but remove the color branch to avoid the negative impact from color information. Fusion blocks are also used to fuse the features from both RB and multi-scale branch. We construct our RB with a maximum number of MBA blocks and AF blocks as $K = 5$ and $T = 5$, respectively. The performance improves as K and T get larger. However, the performance saturates when T approaches to 4 , i.e, the cases of $T = 4$ and $T = 5$ almost achieve the similar results. We also report the runtime-accuracy relationship under the different combinations of K and T in Fig. 8. With the consideration of the network simplicity and effectiveness of training and testing simultaneously, the case of $T = 4$ and $K = 3$ (red star) is our final choice in PMBANet.

Next, to demonstrate the improvements obtained by the proposed elements in RB, i.e., the feedback module and the attention module, we further apply four ablation experiments by integrating different backbones in our framework:

1) w/o RB (only multi-scale branch): The simple single-path stacked multi-scale branches are used as backbone, but without RB.

2) RB $\rightarrow$ IBP [31]: Instead of training our PMBANet with RB, we use a traditional algorithm, i.e., iterative back projection (IBP) [31] to replace our RB, which is defined in the following:

$$HR^i = HR^{i-1} + \alpha f_{Up}(LR^i - f_{Down}(HR^{i-1})). \quad (21)$$

where f_{Up} and f_{Down} are linear interpolation (Bicubic) operators with a $8\times$ scale rate. α is step size and set to 1.75. To obtain a fast convergence, we repeat Eq. (21) five times in each MPA block. Note that IBP can be regarded as a simple iterative strategy to approximately solve the fidelity term, as analyzed in Sec. IV.

3) RB $\rightarrow$ Low-to-High network [30]: We substitute RB with a simple low-to-high resolution network architecture proposed by [30], which is a pure feed-forward architecture.

4) RB $\rightarrow$ DBPN [26]: A similar architecture is the deep back-projection network (DBPN) [26], which also exploits iterative back-projection units to formulate a network. The main difference between DBPN and our RB is that we also employ the self-attention mechanism, and effectively combine the back-projection connections and spatial attention to better excavate informative features at depth boundaries for depth SR.

The quantitative results in MAD for the above cases are shown in Table VII. It is clear that the use of stacked multi-scale branches alone cannot obtain satisfactory results. Replacing the network with a simple IBP algorithm also cannot improve performance very well, which indicates the necessity of simultaneous learning fidelity term and prior term. The next three cases can incrementally verify the effectiveness of feedback mechanism (between 'Low-to-High' and 'DBPN') and attention mechanism (between 'DBPN' and ours), and demonstrates the superior performance of our RB. Fig. 10 further demonstrates the visual performance of our proposed methods. We achieve the best performance with the sharpest and most similar results to the groundtruth.

Besides, we also conduct the ablation experiments at the component-level. As Sec. III-A illustrates, both squeeze-and-expand (S-E) and attention operations contribute to the extraction of useful high-frequency features. Therefore, we further show the ablation experiments on these two key components:

5) S-E $\rightarrow$ Plain Residual: Replacing the squeeze-and-expand operation with a plain residual block (just including two same convolution layers without successive downsampling and upsampling).

6) w/o Attention: Removing the spatial attention operation from 'Ours'.

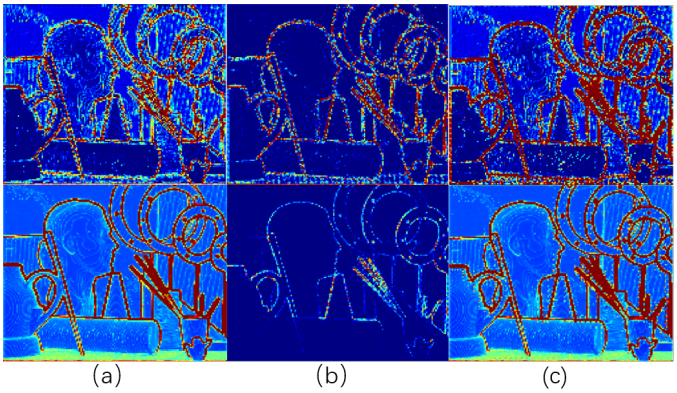


Fig. 11. Visualization of feature maps: (a) high-frequency feature map captured by the squeeze-and-expand operation, (b) attention map, (c) feature map highlighted by the attention operation. The first and second rows represent the features at the stages K=1 and K=3 respectively.

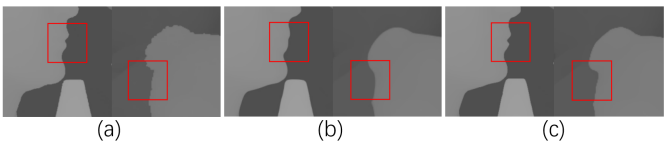


Fig. 12. Visual comparison between the results of cascaded and parallel backbone networks: (a) GT depth map; Results obtained by (b) cascaded mode, (c) parallel mode.

As shown in Table VII, objective results verify that both squeeze-and-expand and attention operations can contribute to our final performance compared to our final configuration (Ours). We also visualize the intermediate feature maps obtained from our RB at the stages K=1 and K=3, respectively. The squeeze-and-expand operation pays more attention on the high-frequency features, and the attention can further highlight the high-frequency locations through the attention map. Compared between K=1 and K=3, the extracted features are more obvious and clear at depth boundaries through progressive refinement.

*2) **Effectiveness of Multi-scale Branch**:* To evaluate the performance of multi-scale branch, we test the cases of our RB with or without the multi-scale branch (MB) under different number of MBA blocks. Color branches are also removed from the whole network to facilitate the comparison. Table VIII shows the qualitative results at 8× upsampling rate. The case with MB presents lower MAD values for all the three configurations of number of MBA blocks. Besides, For the case $K = 3$, we further verifies the effectiveness of dense connections (DC) employed by MB.

Moreover, though PMBANet consists of two parallel branches, an alternative is to combine the RB and GB in a cascaded manner, i.e., concatenating GB to the top of RB and removing the fusion blocks. In the cascade model, the first stage estimates the fidelity residuals while the second stage estimates the prior residuals. However, as shown in Fig. 12 and Table IX, the cascaded architecture does not generate high-quality results compared to the parallel mode. Our parallel architecture inherits the advantage of ensemble learning, which can learn more effective features from each branch.

TABLE VIII
THE EFFECTIVENESS OF MULTI-SCALE BRANCH (8× CASE).

Method	MAD Values (the lower the better)					
	Art	*Books*	*Dolls*	*Laundry*	*Moebius*	*Reindeer*
K=1						
only RB	0.660	0.311	0.378	0.451	0.320	0.426
RB + MB	**0.556**	**0.284**	**0.352**	**0.394**	**0.284**	**0.386**
K=2						
only RB	0.592	0.287	0.365	0.413	0.293	0.391
RB + MB	**0.543**	**0.267**	**0.349**	**0.387**	**0.277**	**0.354**
K=3						
only RB	0.573	0.279	0.360	0.400	0.289	0.378
RB + MB (w/o DC)	0.536	0.275	0.351	0.392	0.274	0.382
RB + MB	**0.529**	**0.270**	**0.344**	**0.387**	**0.268**	**0.364**

TABLE IX
QUANTITATIVE EVALUATION OF DIFFERENT COMBINATION MODES BETWEEN RB AND MULTI-SCALE BRANCH. ($K = 3$)

Method	MAD Values (the lower the better)					
	Art	*Books*	*Dolls*	*Laundry*	*Moebius*	*Reindeer*
Cascade	0.566	0.284	0.353	0.390	0.283	0.379
Parallel	**0.529**	**0.270**	**0.344**	**0.387**	**0.268**	**0.364**

TABLE X
EFFECTIVENESS OF COLOR BRANCH ON EACH UPSAMPLING RATE.

Method	Average MAD Values			
	2×	4×	8×	16×
PMBANet w/o color branch	0.067	0.192	0.360	0.937
PMBANet	**0.064**	**0.183**	**0.338**	**0.682**

TABLE XI
QUANTITATIVE COMPARISONS UNDER DIFFERENT COMBINATION MODES OF COLOR BRANCH (8× CASE).

Method	MAD Values (the lower the better)					
	Art	*Books*	*Dolls*	*Laundry*	*Moebius*	*Reindeer*
PMBANet_v1	0.529	0.270	0.344	0.387	0.268	0.364
PMBANet_v2	0.542	0.271	0.350	0.387	0.274	0.355
PMBANet_v3	0.536	0.271	0.348	0.386	0.273	0.366
PMBANet	**0.508**	**0.263**	**0.318**	**0.340**	**0.264**	**0.335**

*3) **Effectiveness of Color Branch**:* According to previous analysis, color information brings significant improvement for depth SR, but may introduce depth bleeding artifacts due to the depth-color inconsistency. Here, we verify the suitability for a given upsampling case to use color information as guidance. As shown in Table X, we evaluate the role of color branch and compute the average MAD values at each upsampling factor. The results clearly show that color branch can offer significant assistance and improve the performance about 16.4% and 27.2% for the 8× and 16× cases, respectively, but is almost no helpful for the easily recovered 2× and 4× cases (only 4.5% and 2.5% respectively). This is because that lower upsampling cases are not damaged severely by downsampling degradation, which is an easy inverse problem that can be solved by a light-weight CNN to balance the accuracy and complexity. Inappropriately integrating inconsistent color information may lead to the decreased performance easily.

Furthermore, we also investigate the appropriate positions to fuse the color information into the network. We choose the 8× upsampling rate, and test the following four cases, i.e., PMBANet without color branch (PMBANet_v1), PMBANet

Fig. 13. Visual comparison of different positions to fuse the color information into the network (16× case). (a) Color patch; (b) GT; (c) PMBANet_v1; (d) PMBANet_v3; (e) PMBANet.

TABLE XII
EFFECTIVENESS OF FUSION BLOCK.

Method	MAD Values (the lower the better)					
	Art	*Books*	*Dolls*	*Laundry*	*Moebius*	*Reindeer*
w/o fusion	0.535	0.276	0.351	0.392	0.273	0.375
with fusion	**0.529**	**0.270**	**0.344**	**0.387**	**0.268**	**0.364**

that integrating color branch at the last MBA block (PMBANet_v2), at the every MBA block (PMBANet_v3) and at the first MBA block (PMBANet, our final choice). As shown in Table XI, PMBANet_v2 and PMBANet_v3 generate even worse performance than PMBANet_v1, which demonstrates that the color information is not suitable to be introduced for the last several blocks. Fig.13 further demonstrates this. Without color guidance, PMBANet_v1 cannot recover the right structure of the objects. In contrast, there is some improvement in shape retention for PMBANet_v3, but the depth bleeding artifacts around depth boundaries is obvious. Therefore, we choose to integrate color branch only in the first MBA block, and achieve the best results.

*4) **Effectiveness of Fusion Block**:* As shown in Table. XII, we test the effectiveness of our fusion block. For the case of 'w/o fusion', we replace the fusion block by a simple concatenation. With the aid of fusion strategy, the performance has been slightly improved.

VI. VISUALIZATION AND DISCUSSION

Fig. 14 illustrates the PMBANet pipeline at the feature level on 8× upsampling case. We visualize the output of each branch at different stages to fully validate the capability of each designed branch. For clearly presenting the feature extraction process progressively, we increase the MPA blocks up to K = 5. It is interesting to see that RB can effectively enhance the features through iterative attention-based error feed-forward and feedback mechanism, and finally focus on extracting features at depth boundaries.

For multi-scale branch, it extracts the multi-scale feature representation more efficiently, and pays more attention on the fine structures and tiny objects obviously, e.g., the sticks in feature map F^3_{MB}, which increase the probability to recover the depth details from severe damages. Note that slight grid artifacts appear in the feature map F^K_{MB} due to the use of dilated convolutions. Therefore, developing other effective multi-scale methods to replace dilated convolutions would be helpful in the future work.

For color branch, it provides a lot of prior knowledge to reconstruct the depth details. As the data flows forward and aggregates repeatedly, the helpful color features are selected and retained. However, some undesired texture information cannot be removed even in the last feature map, i.e., F^K_{CB}, which may inhibit the improved performance for depth reconstruction. By watching F^K_{CB} carefully, there are still some inconsistent features, and the following fusion block cannot eliminate the negative effects totally. Meanwhile, together with the objective verification in Sec. V-C3, we conclude that color information is only suitable to be introduced in the front stages, in which the harmful color features can be slowly discarded by successive data propagation and aggregation. In future, our analysis can provide new insights for constructing more sophisticated architectures.

VII. CONCLUSION

We propose a progressive multi-branch aggregation network (PMBANet), which consists of stacked MBA blocks to progressively recover the degraded depth map. Specifically, each MBA block has multiple parallel branches, i.e., a reconstruction branch (RB) and a guidance branch (GB) including multiscale color sub-branches. A fusion block is introduced to adaptively fuse and select the discriminative features from all the branches. The design methodology of our whole network is well-founded, and extensive experiments on benchmark datasets demonstrate that our method achieves superior performance in comparison with the state-of-the-art methods.

REFERENCES

[1] Badour Albahar and Jia-Bin Huang. Guided image-to-image translation with bi-directional feature transformation. In *IEEE ICCV*, pages 9015–9024. IEEE, 2019.

[2] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1):1–31, 2011.

[3] João Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *IEEE CVPR*, pages 4733–4742, 2016.

[4] Stanley H. Chan, Xiran Wang, and Omar A. Elgendy. Plug-and-play admm for image restoration: Fixed point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2016.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018.

[6] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *ECCV*, 2018.

[7] Runmin Cong, Jianjun Lei, Huazhu Fu, Junhui Hou, Qingming Huang, and Sam Kwong. Going from RGB to RGBD saliency: A depth-guided transformation model. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99):1–13, 2019.

[8] Runmin Cong, Jianjun Lei, Huazhu Fu, Qingming Huang, Xiaochun Cao, and Chunping Hou. Co-saliency detection for rgbd images based on multi-constraint feature matching and cross label propagation. *IEEE Trans. Image Processing*, 27(2):568–579, 2018.

[9] Runmin Cong, Jianjun Lei, Huazhu Fu, Qingming Huang, Xiaochun Cao, and Nam Ling. HSCS: Hierarchical sparsity based co-saliency detection for RGBD images. *IEEE Transactions on Multimedia*, 21(7):1660–1671, 2019.

[10] Runmin Cong, Jianjun Lei, Huazhu Fu, Weisi Lin, Qingming Huang, Xiaochun Cao, and Chunping Hou. An iterative co-saliency framework for RGBD images. *IEEE Transactions on Cybernetics*, 49(1):233–246, 2019.

[11] Runmin Cong, Jianjun Lei, Changqing Zhang, Qingming Huang, Xiaochun Cao, and Chunping Hou. Saliency detection for stereoscopic images based on depth confidence analysis and multiple cues fusion. *IEEE Signal Processing Letters*, 23(6):819–823, 2016.

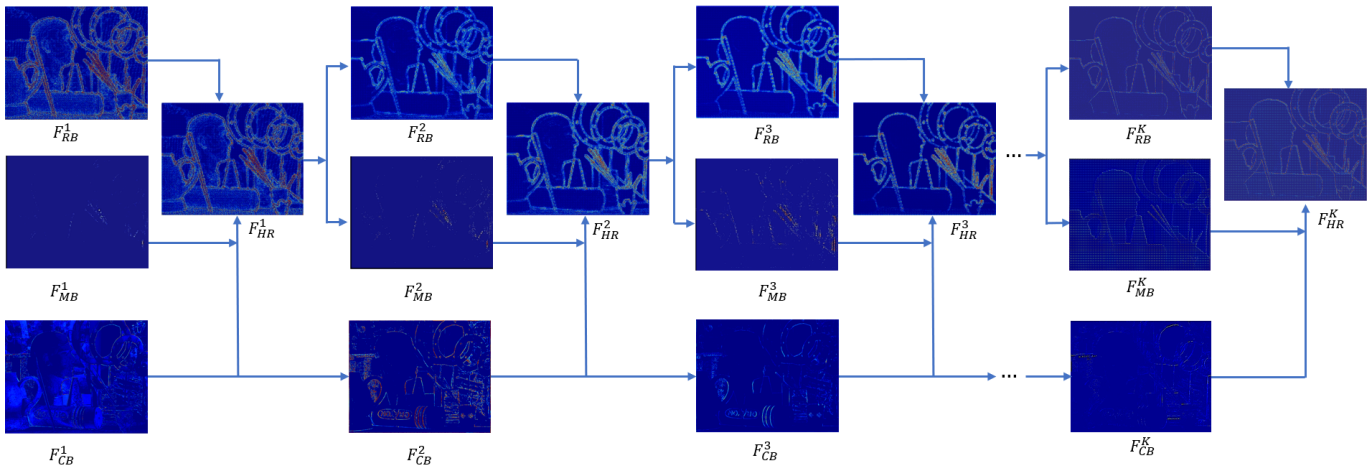


Fig. 14. Visualization of intermediate feature maps from all the branches in PMBANet. For clearly presenting the feature extraction process progressively, we increase the MPA blocks up to $K = 5$.

[12] Shengyang Dai, Mei Han, Ying Wu, and Yihong Gong. Bilateral back-projection for single image super resolution. In *IEEE ICME*, pages 1039–1042, 2007.

[13] T. Dai, J. Cai, Y. Zhang, S. Xia, and L. Zhang. Second-order attention network for single image super-resolution. In *IEEE CVPR*, pages 11057–11066, 2019.

[14] T. Dai, H. Zha, Y. Jiang, and S. Xia. Image super-resolution via residual block attention networks. In *IEEE ICCV Workshop*, pages 3879–3886, 2019.

[15] Riccardo de Lutio, Stefano D'Aronco, Jan Dirk Wegner, and Konrad Schindler. Guided super-resolution as pixel-to-pixel transformation. In *IEEE ICCV*, pages 8828–8836, 2019.

[16] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV*, volume 8692, pages 184–199, 2014.

[17] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE ICCV*, pages 2650–2658, 2015.

[18] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *Proc. ICCV*, pages 993–1000, 2013.

[19] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE CVPR*, pages 2002–2011, 2018.

[20] Stephen Grossberg and Jonathan A. Marshall. Stereo boundary fusion by cortical complex cells: A system of maps, filters, and feedback networks for multiplexing distributed data. *Neural Networks*, 2(1):29–51, 1989.

[21] S. Gu, S. Guo, W. Zuo, Y. Chen, R. Timofte, L. Van Gool, and L. Zhang. Learned dynamic guidance for depth image reconstruction. *IEEE Trans. PAMI*, pages 1–1, 2019.

[22] Shuhang Gu, Wangmeng Zuo, Shi Guo, Yunjin Chen, Chongyu Chen, and Lei Zhang. Learning dynamic guidance for depth image enhancement. In *IEEE CVPR*, pages 712–721, 2017.

[23] Chunle Guo, Chongyi Li, Jichang Guo, Runmin Cong, Huazhu Fu, and Ping Han. Hierarchical features driven residual learning for depth map super-resolution. *IEEE Transactions on Image Processing*, 28(5):2545–2557, 2019.

[24] Bumsub Ham, Minsu Cho, and Jean Ponce. Robust guided image filtering using nonconvex potentials. *IEEE Trans. PAMI*, 40(1):192–207, 2018.

[25] Wei Han, Shiyu Chang, Ding Liu, Mo Yu, Michael Witbrock, and Thomas S. Huang. Image super-resolution via dual-state recurrent networks. In *IEEE CVPR*, pages 1654–1663, 2018.

[26] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *IEEE CVPR*, pages 1664–1673, 2018.

[27] Heiko Hirschmüller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *IEEE CVPR*, 2007.

[28] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE CVPR*, pages 7132–7141, 2018.

[29] Y. Hu, J. Li, Y. Huang, and X. Gao. Channel-wise and spatial feature modulation network for single image super-resolution. *IEEE Trans. CSVT*, 2019.

[30] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *ECCV*, pages 353–369, 2016.

[31] Michal Irani and Shmuel Peleg. Improving resolution by image registration. *CVGIP: Graphical Model and Image Processing*, 53(3):231–239, 1991.

[32] Beomjun Kim, Jean Ponce, and Bumsub Ham. Deformable kernel networks for guided depth map upsampling. *CoRR*, abs/1903.11286, 2019.

[33] Youngjung Kim, Bumsub Ham, Changjae Oh, and Kwanghoon Sohn. Structure selective depth superresolution for rgb-d cameras. *IEEE Trans. Image Processing*, 25(11):5227–5238, 2016.

[34] Dabov Kostadin, Foi Alessandro, Katkovnik Vladimir, and Egiazarian Karen. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Processing*, 16(8):2080–2095, 2007.

[35] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep joint image filtering. In *ECCV*, pages 154–169, 2016.

[36] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Joint image filtering with deep convolutional networks. *CoRR*, abs/1710.04200, 2017.

[37] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. *CoRR*, abs/1903.09814, 2019.

[38] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *IEEE CVPR*, pages 936–944, 2017.

[39] Ming-Yu Liu, Oncel Tuzel, and Yuichi Taguchi. Joint geodesic upsampling of depth images. In *IEEE CVPR*, pages 169–176, 2013.

[40] Xing Liu, Masanori Suganuma, Zhun Sun, and Takayuki Okatani. Dual residual networks leveraging the potential of paired operations for image restoration. In *IEEE CVPR*, 2019.

[41] Jiangbo Lu, Keyang Shi, Dongbo Min, Liang Lin, and Minh N. Do. Cross-based local multipoint filtering. In *IEEE CVPR*, pages 430–437, 2012.

[42] Oisin Mac Aodha, Neill D. F. Campbell, Arun Nair, and Gabriel J. Brostow. Patch based synthesis for single depth image super-resolution. In *ECCV*, pages 71–84, 2012.

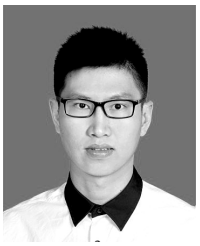
[43] Jinshan Pan, Jiangxin Dong, Jimmy S. J. Ren, Liang Lin, Jinhui Tang, and Ming-Hsuan Yang. Spatially variant linear representation models for joint filtering. In *IEEE CVPR*, pages 1702–1711, 2019.

[44] Jaesik Park, Hyeongwoo Kim, Yu-Wing Tai, Michael S. Brown, and In-So Kweon. High quality depth map upsampling for 3d-tof cameras. In *IEEE ICCV*, pages 1623–1630, 2011.

[45] Gernot Riegler, Matthias Rüther, and Horst Bischof. Atgv-net: Accurate depth super-resolution. In *ECCV*, pages 268–284, 2016.

[46] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nesic, Xi Wang, and Porter Westling. High-resolution stereo

datasets with subpixel-accurate ground truth. In *Pattern Recognition - 36th German Conference*, pages 31–42, 2014.
[47] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *IEEE CVPR*, 2007.
[48] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *IEEE CVPR*, pages 195–202, 2003.
[49] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.
[50] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik G. Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *IEEE CVPR*, pages 11166–11175. IEEE, 2019.
[51] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *IEEE CVPR*, pages 6450–6458, 2017.
[52] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas S. Huang. Deep networks for image super-resolution with sparse prior. In *IEEE ICCV*, pages 370–378, 2015.
[53] Yang Wen, Bin Sheng, Ping Li, Weiyao Lin, and David Dagan Feng. Deep color guided coarse-to-fine convolutional network cascade for depth image super-resolution. *IEEE Trans. Image Processing*, 28(2):994–1006, 2019.
[54] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: convolutional block attention module. In *ECCV*, pages 3–19, 2018.
[55] Jun Xie, Cheng-Chuan Chou, Rogério Schmidt Feris, and Ming-Ting Sun. Single depth image super resolution and denoising via coupled dictionary learning with local constraints and shock filtering. In *IEEE ICME*, pages 1–6, 2014.
[56] Jun Xie, Rogério Schmidt Feris, and Ming-Ting Sun. Edge-guided single depth image super resolution. *IEEE Trans. Image Processing*, 25(1):428–438, 2016.
[57] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *IEEE CVPR*, pages 161–169, 2017.
[58] Jingyu Yang, Xinchen Ye, Kun Li, Chunping Hou, and Yao Wang. Color-guided depth recovery from RGB-D data using an adaptive autoregressive model. *IEEE Trans. Image Processing*, 23(8):3443–3458, 2014.
[59] Xinchen Ye, Xiangyue Duan, and Haojie Li. Depth super-resolution with deep edge-inference network and edge-guided depth filling. In *IEEE ICASSP*, pages 1398–1402, 2018.
[60] Xinchen Ye, Xiaolin Song, Jingyu Yang, Chunping Hou, and Wang Yao. Depth recovery via decomposition of polynomial and piece-wise constant signals. In *IEEE VCIP*, 2017.
[61] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *IEEE CVPR*, pages 1857–1866, 2018.
[62] A. R. Zamir, T. Wu, L. Sun, W. B. Shen, B. E. Shi, J. Malik, and S. Savarese. Feedback networks. In *IEEE CVPR*, pages 1808–1817, 2017.
[63] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces - 7th International Conference*, volume 6920, pages 711–730, 2010.
[64] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep CNN denoiser prior for image restoration. In *IEEE CVPR*, pages 2808–2817, 2017.
[65] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018.



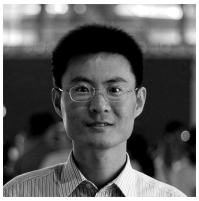
Xinchen Ye (M'17) received the B.E. degree and Ph.D. degree from the Tianjin University, Tianjin, China, in 2012 and 2016, respectively. He was with the Signal Processing Laboratory, EPFL, Lausanne, Switzerland in 2015 under the Grant of the Swiss federal government. He has been a Faculty Member of Dalian University of Technology, Dalian, Liaoning, China, since 2016, where he is currently an Associate Professor with the DUT-RU International School of Information Science and Engineering. His current research interests include image/video processing and 3D imaging. As a co-author, he received the Platinum Best Paper Award in the IEEE ICME 2017. He won the Rising Star Award in 2018 ACM Turing Celebration Conference-China (ACM TURC 2018).

Baoli Sun received the B.S degree in microelectronics science and engineering in 2018 from the Hefei University of Technology , Anhui , China. He is currently a graduate student at the school of Software in Dalian University of Technology in Liaoning, China. His research interests include image processing, computer vision and deep learning.

Zhihui Wang received the B.S. degree in software engineering in 2004 from the North Eastern University, Shenyang, China. She received her M.S. degree in software engineering in 2007 and the Ph.D degree in software and theory of computer in 2010, both from the Dalian University of Technology, Dalian, China. Since November 2011, she has been a visiting scholar of University of Washington. Her current research interests include information hiding and image compression.

Jingyu Yang (M'10-SM'17) received the B.E. degree from Beijing University of Posts and Telecommunications in 2003, and Ph.D. (Hons.) degree from Tsinghua University in 2009. He is currently a Professor with the School of Electrical and Information Engineering. He was with Microsoft Research Asia (MSRA) in 2011, within the MSRA's Young Scholar Supporting Program, and the Signal Processing Laboratory, EPFL, Lausanne, Switzerland, in 2012, and from 2014 to 2015. His research interests include image/video processing, 3D imaging, and computer vision.

Rui Xu received the Ph.D. degree in 2007 from the graduate school of science and engineering, Ritsumeikan University, Japan. He worked in the digital technology research center, Sanyo Electric Co., Ltd., Japan, from 2008 to 2010. He worked as a senior researcher successively in Yamaguchi University and Ritsumeikan University from 2010 to 2015. Since December 2015, he served as an associate professor at Dalian University of Technology. His research fields include intelligent computing in medical images and computer vision.



Haojie Li is a Professor in the School of Software, Dalian University of Technology. His research interests include social media computing and multimedia information retrieval. Dr. Li received the B.E. and the Ph. D. degrees from Nankai University, Tianjin and the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 1996 and 2007 respectively. From 2007 to 2009, he was a Research Fellow in the School of Computing, National University of Singapore.

Baopu Li obtained his PhD degree at the Chinese University of Hong Kong at year 2008, and his current major research interests focus on computer vision, deep learning, robotics and so on.