Hierarchical Features Driven Residual Learning for Depth Map Super-Resolution

Chunle Guo, Chongyi Li, Jichang Guo, Runmin Cong, Huazhu Fu, *Senior Member, IEEE,* and Ping Han

***Abstract*—Rapid development of affordable and portable consumer depth cameras facilitates the use of depth information in many computer vision tasks such as intelligent vehicles and 3D reconstruction. However, depth map captured by low-cost depth sensors (*e.g.*, Kinect) usually suffers from low spatial resolution, which limits its potential applications. In this paper, we propose a novel deep network for depth map super-resolution (SR), called DepthSR-Net. The proposed DepthSR-Net automatically infers a high resolution (HR) depth map from its low resolution (LR) version by hierarchical features driven residual learning. Specifically, DepthSR-Net is built on a residual U-Net deep network architecture. Given LR depth map, we first obtain the desired HR by bicubic interpolation upsampling, and then construct an input pyramid to achieve multiple level receptive fields. Next, we extract hierarchical features from the input pyramid, intensity image, and encoder-decoder structure of U-Net. Finally, we learn the residual between the interpolated depth map and the corresponding HR one using the rich hierarchical features. The final HR depth map is achieved by adding the learned residual to the interpolated depth map. We conduct an ablation study to demonstrate the effectiveness of each component in the proposed network. Extensive experiments demonstrate that the proposed method outperforms the state-of-the-art methods. Additionally, the potential usage of the proposed network in other low-level vision problems is discussed.**

***Index Terms*—convolutional neural network (CNN), depth map super-resolution (SR), residual learning, image reconstruction.**

I. INTRODUCTION

HIGH quality and HR depth map is significant in many computer vision applications (*e.g.*, driving assistance and 3D reconstruction). In recent years, with rapid development of affordable and portable consumer depth cameras such as AUUS Xtion Pro, Microsoft Kinect and Time-of-Flight (TOF), depth information has become increasingly popular in our daily life. However, the depth map taken by low-cost depth sensors usually suffers from low spatial resolution (*e.g.*, the resolution of depth maps acquired by Kinect 2.0 is only 512×424), which limits its potential applications.

To facilitate the use of depth information, numerous methods have been proposed recently, which recovers HR depth map from the corresponding LR version. In the early stage of depth map SR, the HR depth map was achieved by brute-force interpolation upsampling, which produces the blurred and unsharp HR depth map. Several local filter-based methods were proposed. For example, Yang *et al*. [1] proposed a post-processing method to enhance the spatial resolution and precision of depth map by iterative bilateral filtering. Though local filter-based methods have low computational complexity, they tend to introduce obvious artifacts in the resulting HR depth map. In addition, the task of depth map SR was usually regarded as a global energy minimization problem, which designs complex regularization or explores prior to constrain the reconstruction of HR depth map. For example, Jiang *et al*. [2] proposed a depth map SR method based on both frequency and spatial domain regularization. However, the regularization from statistic or prior is not always available for some cases. Moreover, solving the global energy minimization problem is time-consuming. In recent few years, CNNs have presented impressive performance on color image SR [3]. It is interesting that this success of color image SR has not been matched to depth map, though a few depth map SR networks have shown potential performance. For example, Hui *et al*. [4] proposed a deep network, MSG-Net, for depth map SR, which complements LR depth features with HR intensity features.

In fact, exploring the capability of CNNs for depth map SR largely remains open, although CNNs have been introduced into the task of depth map SR. There are two main problems which need to be further explored. 1) Existing network architectures do not make full use of the features extracted from input LR depth map and the corresponding guidance image to reconstruct/recover HR depth map. 2) Directly inferring a HR depth map from its LR or interpolated version usually increases training time and blurs the details of resulting HR depth map. To solve the above-mentioned problems, we propose a hierarchical features driven residual learning for depth map SR, called DepthSR-Net. Different from existing deep depth map SR networks, the proposed DepthSR-Net makes full use of the rich hierarchical features extracted from input pyramid and HR intensity image in a residual U-Net, which outperforms the state-of-the-art depth map SR methods.

Specifically, our network is built on a residual U-Net architecture, which includes an input pyramid branch, an encoder branch (path), a hierarchical intensity image guidance branch, a skip connections, and a decoder branch (path). Given

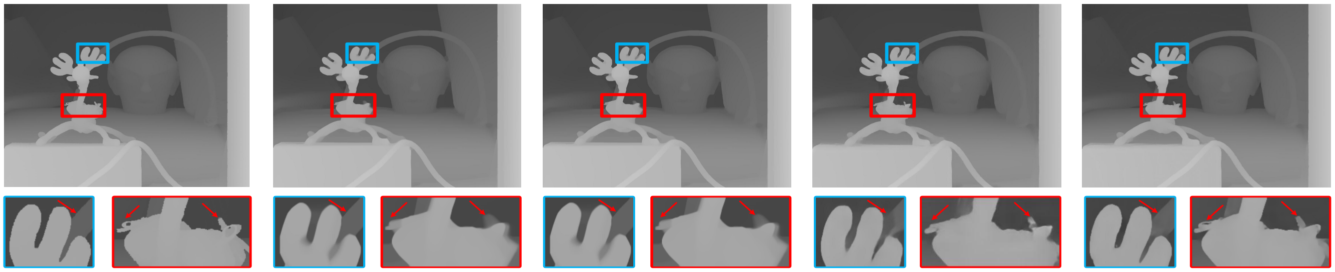
This work was supported in part by the National Natural Science Foundation of China under Grant 61771334 and Grant 61571442, and in part by the National Key Research and Development Program of China under Grant 2016YFB0502405. (*Chunle Guo and Chongyi Li contributed equally to this work.*) (*Corresponding author: Chongyi Li*)

Chunle Guo and Jichang Guo are with the School of Electrical and Information Engineering, Tianjin University, China (e-mail: guochunle@tju.edu.cn; jcguo@tju.edu.cn).

Chongyi Li and Runmin Cong are with the School of Electrical and Information Engineering, Tianjin University, China, and also with the Department of Computer Science, City University of Hong Kong, Hong Kong, China (e-mail: lichongyi25@gmail.com; runmincong@gmail.com).

Huazhu Fu is with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates (e-mail: huazhufu@gmail.com).

Ping Han is with College of Electronic Information and Automation, Civil Aviation University of China, Tianjin, China (e-mail: hanpingcauc@163.com).



(a) GT (RMSE)

(b) Bicubic (5.86)

(c) TSDR [2] (3.12)

(d) MSG [4] (2.92)

(e) DepthSR-Net (2.47)

Fig. 1. Visual comparison results at 16× upsampling for image "Reindeer". In the first row, from (a) to (e) are the HR depth map (*i.e.*, GT) and the results of Bicubic, TSDR [2], MSG [4] and our proposed DepthSR-Net. In the second row, the corresponding amplified details are presented. In the third row, the quantitative values in terms of RMSE are given. **Best viewed with zoom in on a digital display.**

LR depth map, we first upscale it to the desired HR by bicubic interpolation. Then, we construct an input pyramid to achieve multiple level receptive fields, and extract hierarchical intensity features. Next, we concatenate the rich hierarchical features extracted from input pyramid, intensity image and encoder-decoder structure, and then learn the residual map between the interpolated depth map and the corresponding HR one. Finally, we reconstruct the HR depth map by adding the learned residual map to the interpolated depth map. Observing the amplified details and the values of RMSE[1] in Figure 1, our result has sharper boundaries, more complete details, and better quantitative value against the bicubic method, traditional method [2], and deep learning-based method [4].

The main contributions are made as follows:

- Compared to recent deep depth map SR methods, the proposed DepthSR-Net makes full use of rich hierarchical features to recover accurate HR depth map in fully data-driven and end-to-end manner.
- Instead of performing an early spectral decomposition that complements depth features with the associated intensity features in high-frequency domain, we learn the residual between the interpolated depth map and the corresponding HR depth map, which avoids spectral decomposition pre-processing, accelerates training, and is more flexible and suitable for practical applications.
- Benefiting from the novel deep architecture, the proposed DepthSR-Net qualitatively and quantitatively achieves the state-of-the-art performance. In addition, our DepthSR-Net generalizes well to different types of depth data and also can be applied to other low-level vision tasks.

The remainder of this paper is organized as follows. Section II presents a brief overview of the related work. Section III presents the problem formulation and introduces the proposed DepthSR-Net in detail. Section IV conducts extensive experiments and reports the results. In addition, an ablation study is performed. At last, we present an application of the proposed DepthSR-Net on other low-level vision problem. Section V discusses and concludes this paper.

II. Related Work

Depending on the input data, depth map SR methods can be categorized into two categories: depth map SR method with only LR depth map and depth map SR method with LR depth map and the corresponding HR color image/intensity image. We call these two categories as *Non color-guided depth map SR method* and *Color-guided depth map SR method*. Besides, the success of *Deep learning-based color image SR* also inspires our work.

A. Non color-guided depth map SR method

Numerous filtering-based methods have been proposed for depth map SR. Hornacek *et al.* [5] proposed a single depth map SR method by exploring patchwise scene self-similarity. Lei *et al.* [6] proposed a view-synthesis quality based filtering for depth map upsampling, which considers depth smoothness, texture similarity and view syntheses quality jointly.

Some methods treat depth map SR task as a global optimization problem. For an input LR depth patch, Aodha *et al.* [7] searched the HR candidate depth patches from a collected database, then posed the selection of right candidate as a Markov random field labeling problem. Li *et al.* [8] proposed a patchwork assembly method, which disassembles LR depth map into parts and matches the counterparts from a set of HR training images. Xie *et al.* [9] proposed an edge-guided depth map SR method, which constructs the HR edge map from LR depth map by a Markov random field optimization.

Sparse representation and dictionary learning strategies also have been employed in single depth map SR. Liu *et al.* [10] demonstrated depth map can be encoded much more sparsely than natural image and reconstructed depth map from the view of sparse representation. Ferstl *et al.* [11] learned a dictionary of edge priors from an external database, then used the learned edge priors to guide upsampling of LR depth map in a variational sparse coding framework. Xie *et al.* [12] used a robust coupled dictionary learning method with locality coordinate constraints to reconstruct HR depth map. Besides, Xie *et al.* also incorporated an adaptively regularized shock filter to reduce noise and sharpen the edges of the reconstructed depth map. Mandal *et al.* [13] designed a unified framework to restore depth map based on sparse representation, which uses an edge preserving constraint to preserve the discontinuity

[1] In this paper, the root mean squared error (RMSE) is calculated by the code provided by [4].

in the depth map and a pyramidal reconstruction strategy to deal with higher scaling factors. Riegler *et al.* [14] combined a deep convolutional network with a variational method to recover accurate HR depth map, called ATGV-Net.

B. Color-guided depth map SR method

Compared to the HR depth map, HR color image can be easy acquired by color cameras. Thus, various methods have been proposed to enhance the quality of depth map by the guidance of the HR color image or intensity image, also including filtering-based methods [1], [15]–[17], global optimization-based methods [2], [18]–[26], sparse representation-based methods [27], [28] and deep learning-based methods [4], [29].

Yang *et al*. [1] iteratively applied bilateral filtering to refine LR depth map using HR color image as a reference. Inspired by the joint bilateral upsampling method [30], Liu *et al*. [16] utilized geodesic distances to upsample a LR depth map by a registered HR color image. Lu and Forsyth [17] used the relationship between image segmentation boundaries and depth boundaries to produce detailed HR depth structures.

Yang *et al*. [19] formulated the depth recovery problem into a minimization of auto-regressive prediction errors. Ferstle *et al*. [20] formulated a convex optimization problem for depth image upsampling, which guides the depth upsampling by an anisotropic diffusion tensor calculated from a HR intensity image. Park *et al*. [21] extended the nonlocal structure regularization by combining the additional HR color input when upsampling a LR depth map together with a weighting scheme that favors structure details. Dong *et al*. [26] exploited both local and nonlocal structural constraints for depth map recovery. Jiang *et al*. [2] proposed a depth map SR method by frequency and spatial domain regularization. In the frequency domain, nonlocal correlations are exploited by an auto-regressive model. In the spatial domain, a multi-directional total variation prior is used to characterize the geometrical structures.

Kiechle *et al*. [27] introduced a bimodal co-sparse analysis to capture the interdependency of registered intensity and depth information, which has the capability of depth map SR and inpainting of missing depth values. Based on the assumption that local patches in depth maps and RGB images can be represented by a sparse linear combination of basis functions, Kwon *et al*. [28] proposed a data-driven method for depth map refinement through multi-scale dictionary learning.

Recently, deep learning-based depth map SR methods have attracted more and more attention. Hui *et al*. [4] proposed a MSG-Net for depth map SR, which infers HR depth map from a LR depth map and the corresponding HR intensity image of the same scene by CNN. Specifically, MSG-Net uses multi-scale fusion strategy to complement LR depth features with HR intensity features in high-frequency domain. Zhou *et al*. [29] explored how much color image can help for depth map SR problem and drew a conclusion that color images are more helpful when noise is present and/or the scaling factor is large. Additionally, the authors of [29] proposed a fully convolutional network to reconstruct HR depth map.

C. Deep learning-based color image SR method

Compared to traditional methods [31]–[33], CNNs have achieved remarkable success in computer vision task and image processing [34]–[36]. While earlier networks focused on high-level tasks, tremendous attention has been devoted to low-level vision problems in recent years, particularly in color image SR. Dong *et al*. [3] established a relationship between deep learning-based SR method and the traditional sparse coding-based SR method and then proposed an end-to-end network to learn nonlinear mapping between LR and HR color images. This method is called SRCNN which inspires many deep learning-based SR networks such as deeper network architecture [37] and deep recursive network [38]. Lai *et al*. [39] proposed a Laplacian pyramid structure to progressively reconstruct the sub-band residuals of HR color image. To recover the fine texture details in the reconstructed HR color image, a generative adversarial network (*i.e*., SR-GAN) was proposed in [40]. By introducing perceptual loss function which consists of an adversarial loss and a content loss, SRGAN can infer photo-realistic HR color image. Tai *et al*. [41] proposed a very deep network for color image SR task, which uses recursive learning to control the model parameters while increasing the network depth. In [42], the authors recovered the realistic texture in color image SR by a novel spatial feature transformation layer.

Among the previous works, MSG-Net [4] is the most related one to the proposed DepthSR-Net. However, DepthSR-Net is different from the MSG-Net in the following aspects: 1) Instead of performing an early spectral decomposition, we learn the residual map to avoid the spectral decomposition pre-processing, which is more flexible and suitable for practical applications; 2) Different from direct applying LR depth map as input, we first upscale it to the desired solution by bicubic interpolation, which relaxes the constraint on the size of output. In other word, the proposed DepthSR-Net can process any scaling factors while the MSG-Net only generalizes to 2^N scaling factors due to the constraint of automatic upsampling operation utilized in the MSG-Net; 3) Compared with the MSG-Net, we make full use of the multi-level features extracted from input pyramid to recover HR depth map; 4) Although both MSG-Net and DepthSR-Net employ the intensity image as guidance, they extract intensity features by different network architectures. We acknowledge that the features extracted from the intensity image can boost the performance of depth map SR and further demonstrate this conclusion in our ablation study.

III. Proposed Method

In this part, we first briefly formulate the problem that this paper focuses on, and then illustrate the details of the proposed DepthSR-Net architecture. At last, we present the loss function, and training and implementation details.

A. Problem Formulation

The purpose of the proposed DepthSR-Net is to learn an end-to-end nonlinear mapping between LR depth map and HR depth map with the help of HR intensity image. Specifically, given a LR depth map $D_l \in \mathbb{R}^{pH \times pW}$, we first upscale it to the desired solution by bicubic interpolation, denoted as $D_l^{up} \in \mathbb{R}^{H \times W}$, where $p < 1$ is the downscaling factor (*e.g.*, 1/2, 1/3, 1/4, 1/8, and 1/16). Different from direct applying D_l as network input, such upscaling pre-processing can reduce the computational burden for network training and relax the constraint on the size of output [29]. Then, the corresponding HR depth map $D_h \in \mathbb{R}^{H \times W}$ can be formulated as:

$$D_h = \mathcal{F}(D_l^{up}, D_Y; \mathbf{\Theta}) + D_l^{up}. \tag{1}$$

where $\mathcal{F}$ denotes the residual mapping function between the HR depth map D_h and the interpolated depth map D_l^{up}. $\mathbf{\Theta}$ represents the learned network parameters. $D_Y \in \mathbb{R}^{H \times W}$ is the HR intensity image captured under same scene.

Following the conclusion proposed in [44], when the original mapping is more like an identity mapping, the residual mapping will be much easier to be optimized. Observing the interpolated depth map, we found that it is much more like the latent HR depth map than the residual image. Accordingly, we learn the residual between the interpolated depth map and the corresponding HR depth map that is the missed high-frequency component in the process of bicubic interpolation upsampling.

B. Proposed DepthSR-Net Architecture

The overview of the proposed network architecture and parameter settings is shown in Figure 2. In the following, we will give some important details in our network architecture. The proposed DepthSR-Net architecture is based on U-Net network [43] with residual learning and mainly consists of

- **input pyramid branch** that achieves multiple level receptive fields and produces hierarchical representation;
- **encoder branch** that concatenates the hierarchical features from input pyramid and produces a set of hierarchical encoder features;
- **hierarchical Y guidance branch** that extracts hierarchical intensity features to transfer useful structure to the final HR depth map;
- **skip connections** that transmits the encoder features to decoder path;
- **decoder branch** that produces the residual map by fusing rich hierarchical concatenated features.

*1) **Input Pyramid Branch**:* We employ the max pooling layer with 2 × 2 filters and stride 2 to progressively downsample depth map for input pyramid generation. For input pyramid, we extract hierarchical representation by convolution layers. At last, the multi-level input features are transmitted to the encoder path. The operations in the input pyramid branch can be expressed as follows:

$$D_{py}^1 = maxpool(D_l^{up}), \tag{2}$$

$$F_{py}^1 = \sigma(\mathbf{W}_{py}^1 * D_{py}^1 + \mathbf{b}_{py}^1), \tag{3}$$

$$D_{py}^{i+1} = maxpool(D_{py}^i), \tag{4}$$

$$F_{py}^{i+1} = \sigma(\mathbf{W}_{py}^{i+1} * D_{py}^{i+1} + \mathbf{b}_{py}^{i+1}). \tag{5}$$

where $i \in \{1, 2, 3\}$, D_{py}^1 is the 2 × downsampling input depth map D_l^{up}, F_{py}^1 are the features extracted from D_{py}^1, $*$ represents convolution operation, $maxpool$ is max pooling operation. $\mathbf{W}_{py}^1$ and $\mathbf{b}_{py}^1$ stand for the weight and bias in the 1^{th} convolution operation in the input pyramid branch, σ is the element-wise rectified linear unit (ReLU) activation function, D_{py}^{i+1} denotes the 2 × downsampling D_{py}^i, F_{py}^{i+1} are the features extracted from D_{py}^{i+1}, $\mathbf{W}_{py}^{i+1}$ and $\mathbf{b}_{py}^{i+1}$ stand for the weight and bias in the $(i+1)^{th}$ convolution operation.

Input pyramid branch has following advantages: (1) providing hierarchical feature representation extracted from input depth map; (2) achieving multiple level receptive fields; (3) reducing the risk of over-fitting by providing an abstract form of the representation.

*2) **Encoder Branch**:* The encoder path is similar to the left side of U-Net [43], which produces a set of hierarchical encoder features. Different from U-Net, the encoder path in our DepthSR-Net concatenates the hierarchical features extracted from input pyramid branch, which fuses multi-level feature representation. The encoder branch is expressed as:

$$F_{ecb}^1 = \sigma(\mathbf{W}_{ecb}^1 * D_l^{up} + \mathbf{b}_{ecb}^1), \tag{6}$$

$$D_{ecb}^{\frac{j+1}{2}} = maxpool(F_{ecb}^j), \tag{7}$$

$$F_{ecb}^{j+1} = \sigma(\mathbf{W}_{ecb}^{j+1} * D_{ecb}^{\frac{j+1}{2}} + \mathbf{b}_{ecb}^{j+1}), \tag{8}$$

$$F_{ecb}^{j+2} = \sigma(\mathbf{W}_{ecb}^{j+2} * (F_{ecb}^{j+1}, F_{py}^{\frac{j+1}{2}}) + \mathbf{b}_{ecb}^{j+2}). \tag{9}$$

where $j \in \{1, 3, 5, 7\}$, F_{ecb}^1 are the features extracted from D_l^{up}, $D_{ecb}^{\frac{j+1}{2}}$ is the 2 × downsampling F_{ecb}^j by max pooling, F_{ecb}^{j+1} are the features extracted from $D_{ecb}^{\frac{j+1}{2}}$, $(F_{ecb}^{j+1}, F_{py}^{\frac{j+1}{2}})$ represents the concatenation of features F_{ecb}^{j+1} and $F_{py}^{\frac{j+1}{2}}$.

*3) **Hierarchical Y Guidance Branch**:* Different from the multi-scale guidance utilized in [4], we use the fixed size of convolution kernel (*i.e.*, 3 × 3) and guide the reconstruction of residual map in the decoder branch. Hierarchical intensity features extracted by 3×3 convolution kernel can make full use of the discontinuities in intensity image to locate the associated depth discontinuities in the process of reconstruction, and also reduces the computational burden. The hierarchical Y guidance branch can be expressed as:

$$F_Y^1 = \sigma(\mathbf{W}_Y^1 * Y + \mathbf{b}_Y^1), \tag{10}$$

$$D_Y^r = maxpool(F_Y^r), \tag{11}$$

$$F_Y^{r+1} = \sigma(\mathbf{W}_Y^{r+1} * D_Y^r + \mathbf{b}_Y^{r+1}). \tag{12}$$

where $r \in \{1, 2, 3\}$, F_Y^1 are the features extracted from intensity image Y, D_Y^r is the 2 × downsampling F_Y^r, F_Y^{r+1} are the features extracted from D_Y^r.

Hierarchical Y guidance branch has following advantages: (1) transferring hierarchically useful structure of intensity image to the final HR depth map; (2) increasing the network width of the decoder branch.

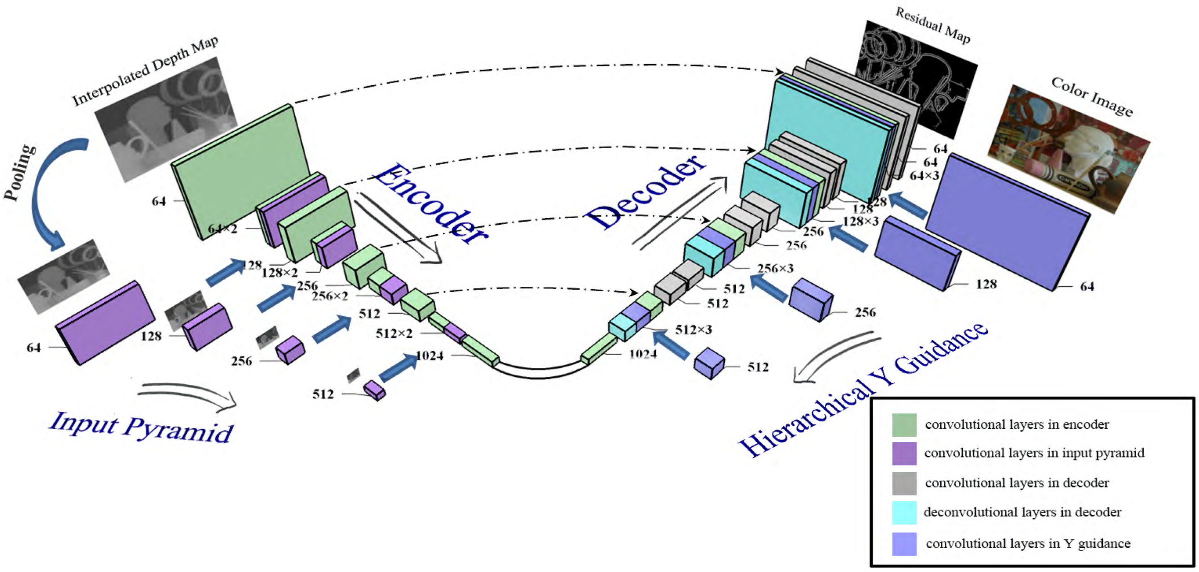


Fig. 2. An overview of the proposed DepthSR-Net. DepthSR-Net is built on a residual U-Net network with hierarchical Y (*i.e.*, intensity image) guidance branch and input pyramid branch. LR depth map is first interpolated to the desired HR depth map denoted as **Interpolated Depth Map**. Then we generate input pyramid by progressive 2× max polling. **Color Image** is first transformed to intensity image. **Residual Map** is the difference between HR depth map and the interpolated depth map. Different color blocks represent different convolution/deconvolution layers in the different branches. The convolution/deconvolution layers include ReLU activation function, except for the last one in the decoder path. Blocks with same color and size presented in different branches stand for concatenation operation. "× 2" and "× 3" represent the concatenation of 2 and 3 sets of features from different branches, respectively. Numbers represent the numbers of feature maps. The direction of arrows denotes the direction of data flow.

*4) **Skip Connections**:* Skip connections operation is the extra connection between nodes in different layers of neural network that skip one or more layers of nonlinear processing. The purpose is to transfer the corresponding features from encoder branch to decoder branch. The advantages of the skip connections are as follows: (1) alleviating the vanishing gradient problem; (2) ensuring maximum information flow between layers; (3) encouraging feature reuse.

*5) **Decoder Branch**:* In the decoder branch, we progressively fuse the decoder features with the hierarchical features from other branches to predict the residual map. The decoder branch can be expressed as:

$$F_{dec}^{1} = \sigma(\mathbf{W}_{dec}^{1} * F_{ecb}^{9} + \mathbf{b}_{dec}^{1}), \tag{13}$$

$$F_{dec}^{k} = \sigma(\mathbf{W}_{dec}^{k} \star F_{dec}^{k-1} + \mathbf{b}_{dec}^{k}), \tag{14}$$

$$F_{dec}^{k+1} = \sigma(\mathbf{W}_{dec}^{k+1} * (F_{dec}^{k}, F_{Y}^{\frac{14-k}{3}}, F_{ecb}^{\frac{25-2k}{3}}) + \mathbf{b}_{dec}^{k+1}), \tag{15}$$

$$F_{dec}^{k+2} = \sigma(\mathbf{W}_{dec}^{k+2} * F_{dec}^{k+1} + \mathbf{b}_{dec}^{k+2}). \tag{16}$$

where $k \in \{2, 5, 8, 11\}$, F_{dec}^{1} are the features extracted from encoder features F_{ecb}^{9} , $\star$ represents deconvolution operation. F_{dec}^{k} are the features extracted from F_{dec}^{k-1}, F_{dec}^{k+1} are the features extracted from the concatenation of features F_{dec}^{k}, $F_{Y}^{\frac{14-k}{3}}$ and $F_{ecb}^{\frac{25-2k}{3}}$, F_{dec}^{k+2} are the features extracted from F_{dec}^{k+1}. For the case of $k = 11$, F_{dec}^{13} is the final residual map which is from the convolution layer without ReLU active function. At last, we reconstruct the HR depth map by adding the learned residual map to the interpolated depth map.

Additionally, following popular network design [37], [44], we use 3×3 filter size in the entire DepthSR-Net architecture and pad zeros for convolution/deconvolution layers. The number of features in different layers is set as shown in Figure 2. The effects of each component will be further discussed.

C. Loss Function

Following previous SR methods, we minimize the ℓ_2 loss function to learn the residual mapping:

$$L(\mathbf{\Theta}) = \frac{1}{N} \sum_{ii=1}^{N} \| \mathcal{F}(D_{l_{ii}}^{up}, Y_{ii}; \mathbf{\Theta}) - (D_{h_{ii}} - D_{l_{ii}}^{up}) \|^2. \tag{17}$$

where $\mathcal{F}$ represents the residual mapping function, $\mathbf{\Theta}$ represents the unknown network hyperparameters that need to be learned, N is the total number of training samples, $D_{l_{ii}}^{up}$ is the ii^{th} interpolated depth map, Y_{ii} is the corresponding intensity image, $D_{h_{ii}}$ is the corresponding HR depth map, $D_{h_{ii}} - D_{l_{ii}}^{up}$ represents the residual between the HR depth map $D_{h_{ii}}$ and the interpolated depth map $D_{l_{ii}}^{up}$. The $D_{h_{ii}}$, $D_{l_{ii}}^{up}$ and Y_{ii} are all normalized to the range $[0, 1]$.

D. Network Training and Implementation

*1) **Training Details**:* We utilized the same training datasets with [4], namely 58 RGB-D images from MPI Sintel depth dataset [45] and 34 RGB-D images (6, 10, 18 images are from 2001, 2006 and 2014 datasets respectively) from Middlebury dataset [46]–[48]. We totally used 82 images for training and 10 images for validation. To get more training data, we augmented data with flipping and rotation. We obtained 7 additional augmented versions of the original training data.

In the training phase, the HR depth maps D_h were cropped to 96 × 96 image patches by overlapping sampling with a stride of 48 for scaling factors 2, 3, 4, and 8, which reduces the training time. We used the size of 128×128 image patches by overlapping sampling with a stride of 64 for scaling factor 16. At last, the augmented training data provided roughly 450,000 image patches for scaling factors 2, 3, 4, 8 and roughly 240,000 image patches for scaling factor 16. To synthesize LR depth maps, we downsampled each full-resolution image patch by bicubic downsampling with the given scaling factor.

*2) **Implementation Details**:* During training, a batch-mode learning method with a batch size of 64 was applied. The filter weights of each layer were initialized by "Xavier". We used ADAM with β_1=0.9 and β_2=0.999 for network optimization. We fixed the learning rate in the entire training procedure. The learning rate was set to $1e^{-4}$, $1e^{-4}$, $1e^{-3}$, $1e^{-2}$, and $1e^{-2}$ for the scaling factors 2, 3, 4, 8, and 16. We trained a specific DepthSR-Net for each scaling factor. The training was performed with TensorFlow on a PC with an i7-6700 CPU, 32GB RAM, and a GTX 1080Ti GPU.

IV. Experiments

In this section, we first conduct both quantitative and qualitative experiments, and compare the proposed DepthSR-Net with the baseline bicubic interpolation upsampling and several state-of-the-art methods: local filtering-based method (*i.e.*, GF (TPAMI'13) [15]), global optimization-based methods (*i.e.*, TGV (ICCV'13) [20] and TSDR (TIP'18) [2]), sparse representation-based method (*i.e.*, JID (ICCV'13) [27]), deep learning-based color image SR methods (*i.e.*, SRCNN (TPAMI'16) [3] and VDSR (CVPR'16) [37]), and deep learning-based depth map SR method (*i.e.*, MSG (ECCV'16) [4]).

In this paper, all results are generated by the authors's codes and tuned to generate best results with bicubic downsampling LR depth map, with exception of TSDR [2] and JID [27] (we generate the input depth maps by the downsampling methods provided by the authors), and MSG [4] (we directly use the RSME values reported by the authors when these values are available). Moverover, we directly apply the released models of SRCNN [3] and VDSR [37] to depth map SR. The results of SRCNN and VDSR for depth map SR are only used as the reference since they are designed for color image SR.

Similarly to recent methods [2] and [4], we validate the accuracy of different methods on the hole-filled Middlebury RGB-D datasets (denoted as **Test**) for various scaling factors (*i.e.*, 2, 3, 4, 8, and 16). The testing images in the **Test** are collected from [11], [20], [22]. To assess the robustness of our method, we conduct two experiments. First, following [2], we simulate the ToF-like degradation by adding Gaussian noise with standard deviation of 5 to the LR depth maps in the **Test**, and denote this dataset as **Test-ToF**. Second, we add Gaussian noise with standard deviation of 5 to the guidance intensity images in the **Test**, and denote this dataset as **Test-Ynoise**. We report the qualitative and quantitative (in terms of RMSE and PSNR (dB)) performance. The best result for each evaluation is in red, whereas the second and third best one are in blue and brown respectively. To show the generalization of our method, we present several representative results of depth SR methods for real data taken by Kinect. Note that we ensure none of the testing images are in the training set. Besides, we compare the running time of different methods and carry out an ablation study to demonstrate the effectiveness of each component in the proposed DepthSR-Net. At last, we present an example of the DepthSR-Net applied to other low-level task.

A. Experiment on Middlebury Dataset

We first conduct quantitative comparisons on **Test** for different scaling factors. From Tables I-V, MSG [4], TSDR [2], and our DepthSR-Net have obvious advantages when compared with other methods. It can be observed that the proposed DepthSR-Net achieves best performance on all scaling factors in terms of average RMSE and PSNR values, which benefits from the multiple level receptive fields and rich hierarchical features provided by our architecture. Compared with the second best results (in blue), our results outperform them (a gain of 0.12/0.93 (2×), 0.27/1.82 (3×), 0.26/1.00 (4×), 0.54/1.29 (8×), and 0.37/0.69 (16×)) for average RMSE/PSNR values. Furthermore, our method almost obtains the best results in terms of RMSE and PSNR values for all testing images, especially for large scaling factors which are difficult for traditional methods. Such results also demonstrate that the discontinuities in intensity image are more helpful for guiding the reconstruction of the large scaling upsampling depth map.

To further analyze the performance of our method, we present the visual results for 4× and 8× scaling factors on several images from **Test** in Figures 3 and 4. Some results for 16× scaling factor have been presented in Figure 1. Observing Figure 3, almost all results fill the hole of the camera bracket except for our result (the details presented in the blue box), which demonstrates the proposed DepthSR-Net can produce clear boundaries and does not introduce blurred details. Besides, the details of our result in the red box are closer to the GT. In Figures 4 and 1, 8× and 16× upsampling are difficult, especially for traditional methods, due to the much missed information. However, our DepthSR-Net can well deal with these missed information and produce unbroken details (*e.g.*, the handle in the boxes in Figure 4) and sharp boundaries (*e.g.*, the boundaries in the boxes in Figure 1).

In summary, Bicubic, GF [15], TGV [20], and JID [27] tend to produce over-smooth results. GF [15], TGV [20], and JID even introduce extra texture, noise, and artifacts. SRCNN [3] and VDSR [37] designed for color image SR do not perform well for depth map SR. Recent TSDR [2] and MSG [4] produce competitive results, which benefits from dual domain regularization and data-driven deep learning. The proposed DepthSR-Net outperforms the compared methods from the aspects of quantitative and qualitative comparisons.

B. Experiment on Test-ToF Dataset

We carry out experiments on **Test-ToF** and present the quantitative results in Table VI and an example of visual comparison in Figure 5. We added Gaussian noise with standard

TABLE I
Quantitative comparisons on **Test** for 2× scaling factor in terms of RMSE/PSNR(dB) values.

Method	Art	Books	Laundry	Reindeer	Tsukuba	Teddy	Average
Bicubic	2.64/39.70	1.07/47.58	1.61/43.97	1.94/42.38	5.82/32.84	1.96/42.30	2.51/41.46
GF [15]	3.15/38.16	1.43/44.99	2.01/42.09	2.25/41.11	7.63/30.48	2.47/40.28	3.16/39.52
TGV [20]	3.16/38.14	1.33/45.62	1.87/42.69	2.40/40.53	7.05/31.17	2.38/40.60	3.03/39.79
JID [27]	1.18/46.70	0.45/55.16	0.68/51.43	0.90/49.04	3.69/36.80	1.20/46.58	1.35/47.62
SRCNN [3]	1.92/42.45	0.79/50.14	1.10/47.34	1.34/45.57	3.91/36.28	1.55/44.32	1.77/44.35
VDSR [37]	1.37/45.41	0.40/56.08	0.72/50.93	0.94/48.67	3.08/38.35	1.15/46.92	1.28/47.73
MSG [4]	0.66/51.74	0.37/56.77	0.79/50.18	0.42/55.67	1.85/42.79	0.71/51.11	0.80/51.38
DepthSR-Net	0.53/53.63	0.42/55.66	0.44/55.25	0.51/53.88	1.33/45.68	0.83/49.73	0.68/52.31

TABLE II
Quantitative comparisons on **Test** for 3× scaling factor in terms of RMSE/PSNR(dB) values.

Method	Art	Books	Laundry	Reindeer	Tsukuba	Teddy	Average
Bicubic	3.31/37.73	1.37/45.39	2.06/41.85	2.42/40.46	7.18/31.01	2.44/40.37	3.13/39.47
GF [15]	3.48/37.31	1.60/44.06	2.22/41.19	2.50/40.16	8.23/29.82	2.69/39.52	3.45/38.68
TGV [20]	3.21/38.00	1.41/45.15	1.89/42.60	2.33/40.78	7.78/30.31	2.32/40.82	3.16/39.61
TSDR [2]	1.08/47.45	0.93/48.69	0.74/50.78	0.80/50.06	3.15/38.17	1.16/46.81	1.31/46.99
JID [27]	1.63/43.90	0.59/52.69	0.92/48.83	1.21/46.50	4.91/34.31	1.53/44.45	1.80/45.11
SRCNN [3]	2.36/40.66	1.00/48.14	1.37/45.42	1.60/44.03	4.91/34.31	1.91/42.53	2.19/42.52
VDSR [37]	1.73/43.36	0.56/53.12	0.97/48.39	1.21/46.48	3.87/36.38	1.48/44.75	1.64/45.41
DepthSR-Net	0.89/49.10	0.56/53.14	0.62/52.24	0.77/50.35	2.25 /41.08	1.15/46.94	1.04/48.81

TABLE III
Quantitative comparisons on **Test** for 4× scaling factor in terms of RMSE/PSNR(dB) values.

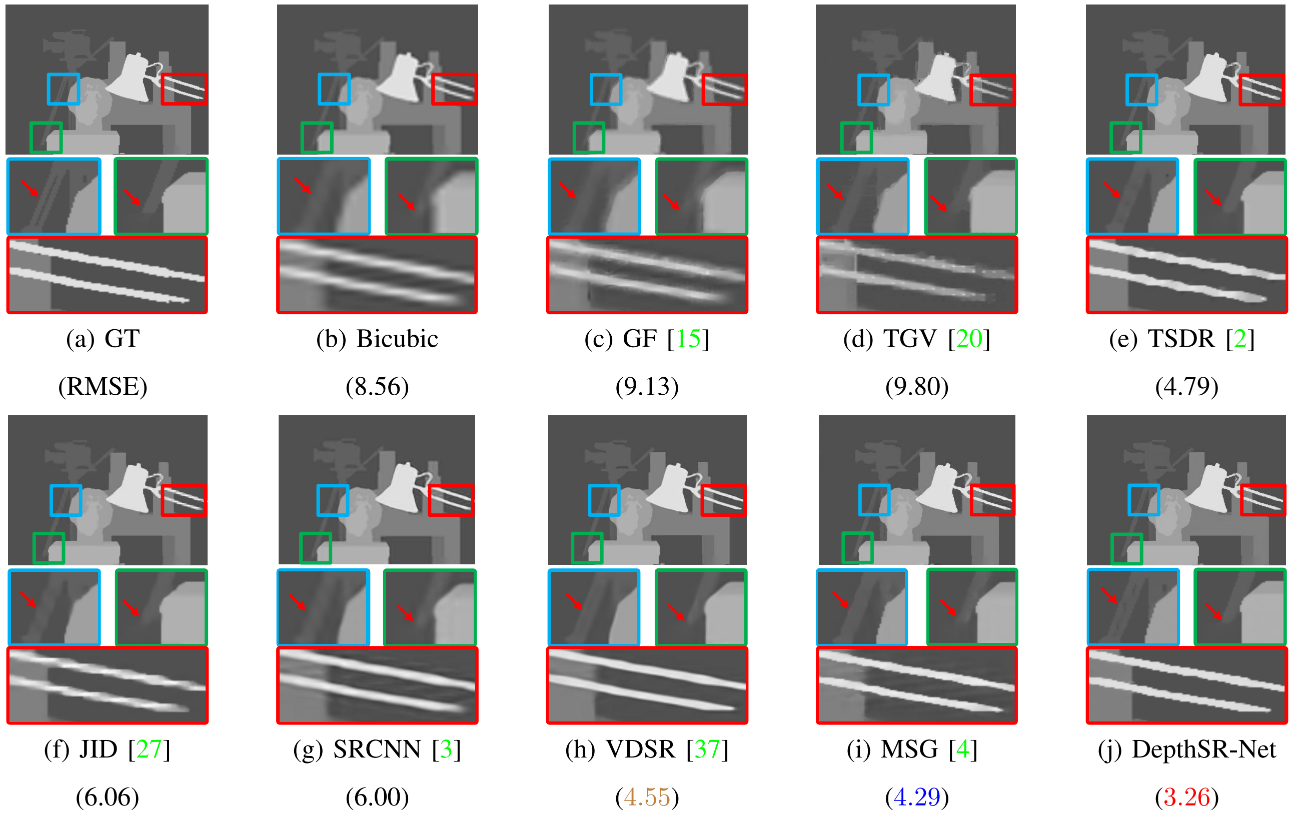
Method	Art	Books	Laundry	Reindeer	Tsukuba	Teddy	Average
Bicubic	3.87/36.37	1.61/43.97	2.41/40.50	2.81/39.16	8.56/29.48	2.86/39.01	3.69/38.08
GF [15]	3.90/36.30	1.76/43.21	2.47/40.29	2.82/39.14	9.13/28.92	3.01/38.56	3.85/37.74
TGV [20]	3.73/36.70	1.67/43.68	2.25/41.09	2.65/39.67	9.80/28.31	2.81/39.16	3.82/38.10
TSDR [2]	1.57/44.19	1.05/47.69	0.98/48.34	1.19/46.64	4.79/34.53	1.46/44.88	1.84/44.38
JID [27]	1.92/42.45	0.71/51.14	1.10/47.32	1.41/45.13	6.06/32.48	1.78/43.14	2.16/43.61
SRCNN [3]	2.65/39.66	1.15/46.92	1.63/43.87	1.89/42.63	6.00/32.60	2.18/41.38	2.58/41.18
VDSR [37]	1.99/42.14	0.72/50.96	1.19/46.60	1.38/45.32	4.55/34.97	1.72/43.41	1.93/43.90
MSG [4]	1.47/44.78	0.67/51.61	0.79/50.18	0.98/48.31	4.29/35.48	1.49/44.67	1.62/45.84
DepthSR-Net	1.20/46.55	0.60/52.49	0.78/50.26	0.96/48.51	3.26/37.85	1.37/45.39	1.36/46.84

TABLE IV
Quantitative comparisons on **Test** for 8× scaling factor in terms of RMSE/PSNR(dB) values.

Method	Art	Books	Laundry	Reindeer	Tsukuba	Teddy	Average
Bicubic	5.46/33.39	2.34/40.76	3.45/37.38	3.99/36.12	12.33/26.31	4.03/36.02	5.27/35.00
GF [15]	5.43/33.44	2.38/40.59	3.44/37.41	3.95/36.20	12.47/26.21	4.06/35.96	5.29/34.97
TGV [20]	7.12/31.08	2.27/41.01	4.05/35.98	4.33/35.40	15.60/24.27	3.89/36.33	6.21/34.01
TSDR [2]	2.30/40.86	1.06/47.62	1.58/44.15	1.75/43.30	9.39/28.67	1.99/42.15	3.01/41.13
JID [27]	2.76/39.31	1.01/48.04	1.83/42.89	2.12/41.62	9.54/28.55	2.72/39.44	3.33/39.98
VDSR [37]	5.46/33.38	2.33/40.77	3.45/37.37	3.99/36.12	12.35/26.30	4.03/36.03	5.27/35.00
MSG [4]	2.46/40.31	1.03/47.87	1.51/44.55	1.76/43.22	8.43/29.61	2.76/39.31	2.99/40.81
DepthSR-Net	2.22/41.22	0.89/49.12	1.31/45.81	1.57/44.21	6.89/31.37	1.85/42.79	2.45/42.42

TABLE V
Quantitative comparisons on **Test** for 16× scaling factor in terms of RMSE/PSNR(dB) values.

Method	Art	Books	Laundry	Reindeer	Tsukuba	Teddy	Average
Bicubic	8.17/29.89	3.34/37.65	5.07/34.04	5.86/32.77	16.57/23.75	6.02/32.53	7.51/31.77
GF [15]	8.15/29.90	3.34/37.65	5.03/34.10	5.83/32.81	16.62/23.72	6.04/32.52	7.50/31.78
TGV [20]	12.08/26.49	4.89/34.34	8.01/30.06	9.05/29.00	23.36/20.76	7.59/30.53	10.83/28.53
TSDR [2]	4.30/35.45	1.59/44.10	2.19/41.26	3.12/38.22	14.13/25.13	3.18/38.18	4.75/37.06
JID [27]	9.93/28.20	8.43/29.62	8.73/29.31	7.19/30.99	16.46/23.80	6.63/31.70	9.56/28.94
VDSR [37]	8.16/29.90	3.33/37.67	5.06/34.05	5.85/32.78	16.57/23.75	6.02/32.54	7.50/31.78
MSG [4]	4.57/34.93	1.63/43.88	2.63/39.73	2.92/38.79	13.83/25.31	3.95/36.20	4.92/36.47
DepthSR-Net	3.90/36.30	1.51/44.54	2.26/41.06	2.47/40.29	13.10/25.78	3.02/38.53	4.38/37.75



(a) GT (RMSE)

(b) Bicubic (8.56)

(c) GF [15] (9.13)

(d) TGV [20] (9.80)

(e) TSDR [2] (4.79)

(f) JID [27] (6.06)

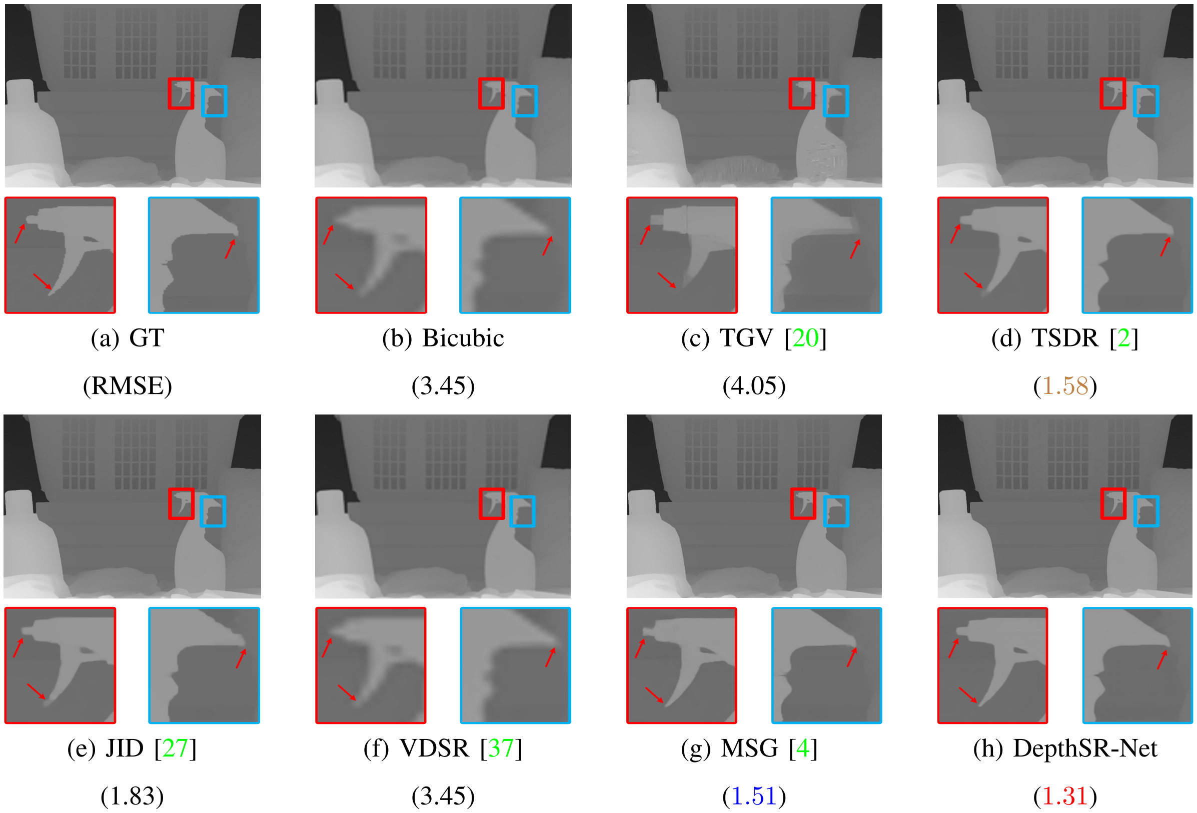
(g) SRCNN [3] (6.00)

(h) VDSR [37] (4.55)

(i) MSG [4] (4.29)

(j) DepthSR-Net (3.26)

Fig. 3. Visual comparison results at 4× upsampling for image “Tsukuba”. From (a) to (j) are the HR depth map and the results of Bicubic, GF [15], TGV [20], TSDR [2], JID [27], SRCNN [3], VDSR [37], MSG [4] and our proposed DepthSR-Net. The details are amplified in the boxes.



(a) GT (RMSE)

(b) Bicubic (3.45)

(c) TGV [20] (4.05)

(d) TSDR [2] (1.58)

(e) JID [27] (1.83)

(f) VDSR [37] (3.45)

(g) MSG [4] (1.51)

(h) DepthSR-Net (1.31)

Fig. 4. Visual comparison results at 8× upsampling for image “Laundry”. From (a) to (h) are the HR depth map and the results of Bicubic, TGV [20], TSDR [2], JID [27], VDSR [37], MSG [4] and our proposed DepthSR-Net. The details are amplified in the boxes.

deviation of 5 to the LR depth maps in our training datasets, and then retrained the DepthSR-Net (denoted as DepthSR-Net-Ddenoise). Besides, inspired by the work [49], we also present the blind denoising results of our method. We added Gaussian noise with random standard deviation varying from 3 to 7 to the LR depth maps in our training datasets, and then retrained the DepthSR-Net (denoted as DepthSR-Net-DdenoiseB). Thus, we show the results of original DepthSR-Net, DepthSR-Net-Ddenoise, and DepthSR-Net-DdenoiseB on **Test-ToF**. For the limited space, we only present part results.

Compared with Table V, the results in Table VI indicate that the noise in the LR depth maps has significant effect on the HR depth maps reconstructed by all methods. In Figure 5, our original DepthSR-Net produces relatively accurate and sharp HR depth map while the compared methods produce over-smooth boundaries (*e.g.*, Bicubic, TGV [20], and VDSR [37]) or tend to magnify noise (*e.g.*, TSDR [2], JID [27], and MSG [4]). Our blind denoising model DepthSR-Net-DdenoiseB can well deal with the effect of noise. As expected, our DepthSR-Net-Ddenoise has superior performance.

C. Experiment on Test-Ynoise Dataset

We present the quantitative results of the color-guided depth map SR methods on **Test-Ynoise** in Table VII and an example of visual comparison in Figure 6. We added Gaussian noise with standard deviation of 5 to the guidance intensity images in our training datasets, and retrained the DepthSR-Net (denoted as DepthSR-Net-Ydenoise). We added Gaussian noise with random standard deviation varying from 3 to 7 to the guidance intensity images in our training datasets, and retrained the DepthSR-Net (denoted as DepthSR-Net-YdenoiseB).

Compared the results of DepthSR-Net on **Test-Ynoise** in Table VII with the results of DepthSR-Net on **Test** (presented in Table V) and DepthSR-Net on **Test-ToF** (presented in Table VI), it is obvious that our method is insensitive to the noise in the guidance image. Besides, the retrained DepthSR-Net-Ydenoise and DepthSR-Net-YdenoiseB can slightly improve the performance of original DepthSR-Net. An example in Figure 6 also indicates that our method can tolerate the noise in the guidance image. TSDR [2] and MSG [4] have relative good performance on **Test-Ynoise**. By contrast, the noise in the guidance intensity images has severe effect on the results of TGV [20] and JID [27] since they obviously transfer the noise in the intensity image to the reconstructed depth map.

D. Experiment on Real Data

To validate the generalization of our method, we carry out experiment on the real data captured by Kinect. The resolution of the captured depth maps is 512 × 424 while the resolution of the corresponding RGB images is 1920 × 1080. There are the problems of misalignment and missing pixel in the real data. Before depth SR processing, we first upsample the LR depth map to the same size of the RGB image, and then calibrate them according to the parameter settings of Kinect and fill the missing values by griddata interpolation. For the limited space, we only show several results in Figure 7.

As shown in Figure 7, compared with the results of TGV [20] and the original depth maps, the results of TSDR [2] and the proposed DepthSR-Net have sharper edges. Usually, deep learning-based depth SR methods are sensitive to the misalignment because they generally do not take the issue of misalignment into account during the procedure of networks training. Thus, the performance of deep learning-based methods is easy to be affected by the accuracy of alignment algorithms. In future work, we will take the issues of misalignment and missing data into consideration when we design depth SR networks and collect the training data.

E. Running Time

We compare the running time (s) of different methods for different scaling factors on a PC with an Intel(R) i7-6700 CPU, 32GB RAM, and a NVIDIA GeForce GTX 1080Ti GPU. The average running time for an image with size 1080×1320 is shown in Table VIII. The codes of Bicubic, GF [15], TGV [20], JID [27], and SRCNN [3] are written in Matlab. The code of TSDR is written in Matlab and C++. The running time of VDSR [37] (implemented with MatConvNet), MSG [4] (implemented with Caffe), and our DepthSR-Net (implemented with TensorFlow) is calculated with GPU acceleration. Besides, we also compute the number of floating-point operations (FLOPs) of the deep learning-based methods. For an image with size 128×128, the FLOPs of SRCNN [3], VDSR [37], MSG [4], and the proposed DepthSR-Net are about $2.6e^8$, $2.2e^{10}$, $8.2e^9$, and $3.5e^{10}$.

TABLE VIII
AVERAGE RUNNING TIME FOR DIFFERENT SCALING FACTORS (SECONDS).

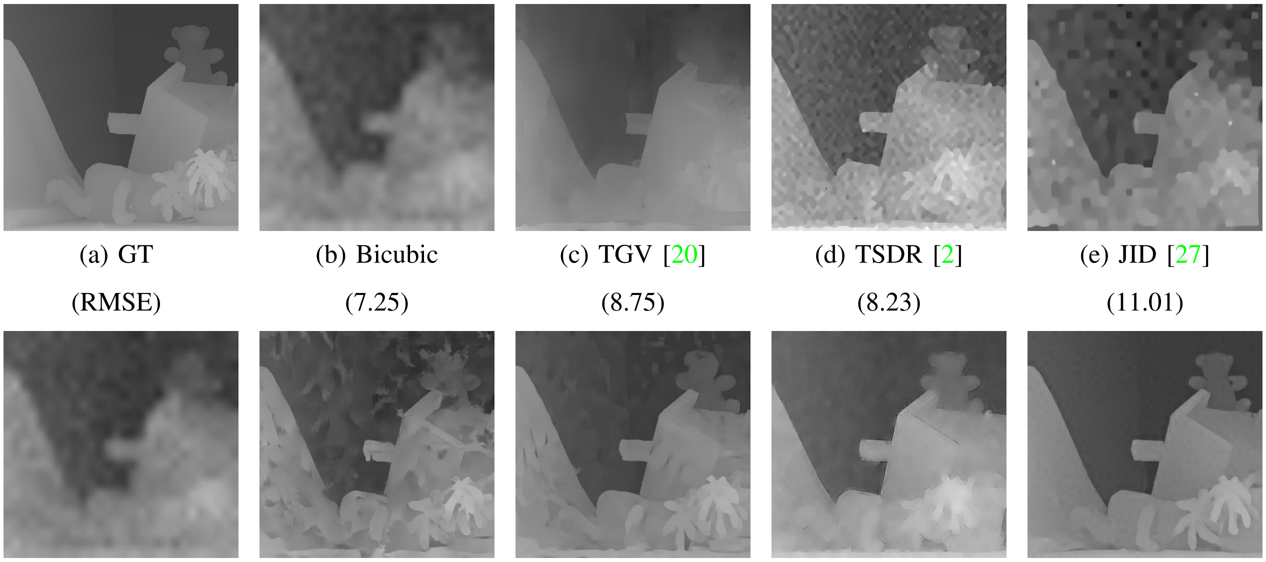
Method	2×	3×	4×	8×	16×
Bicubic	0.01	0.01	0.01	0.01	0.01
GF [15]	0.15	0.15	0.15	0.15	0.15
TGV [20]	893.77	893.69	895.57	894.27	881.87
TSDR [2]	-	838.02	832.93	991.16	1119.16
JID [27]	1169.55	1072.78	984.70	986.01	847.30
SRCNN [3]	46.63	46.55	46.87	-	-
VDSR [37]	0.44	0.44	0.45	0.44	0.47
MSG [4]	0.26	-	0.30	0.38	0.42
DepthSR-Net	1.84	1.85	1.85	1.86	1.85

Observing Table VIII, we found that deep learning-based methods including our DepthSR-Net, except for SRCNN [3] (the released testing model is implemented with Matlab) are fast thanks to the GPU acceleration. The speed of Bicubic and GF [15] is also fast because of the low complexity of codes. In contrast, the speed of TGV [20], TSDR [2], and JID [27] is very slow, which limits their practical applications [50], [51].

F. Ablation Study

To demonstrate the improvements obtained by each component in the proposed DepthSR-Net, we carry out an ablation study involving the following four experiments:

- Original U-Net network (**U-Net**)
- DepthSR-Net without input pyramid (**DepthSR-Net-w/o IP**)



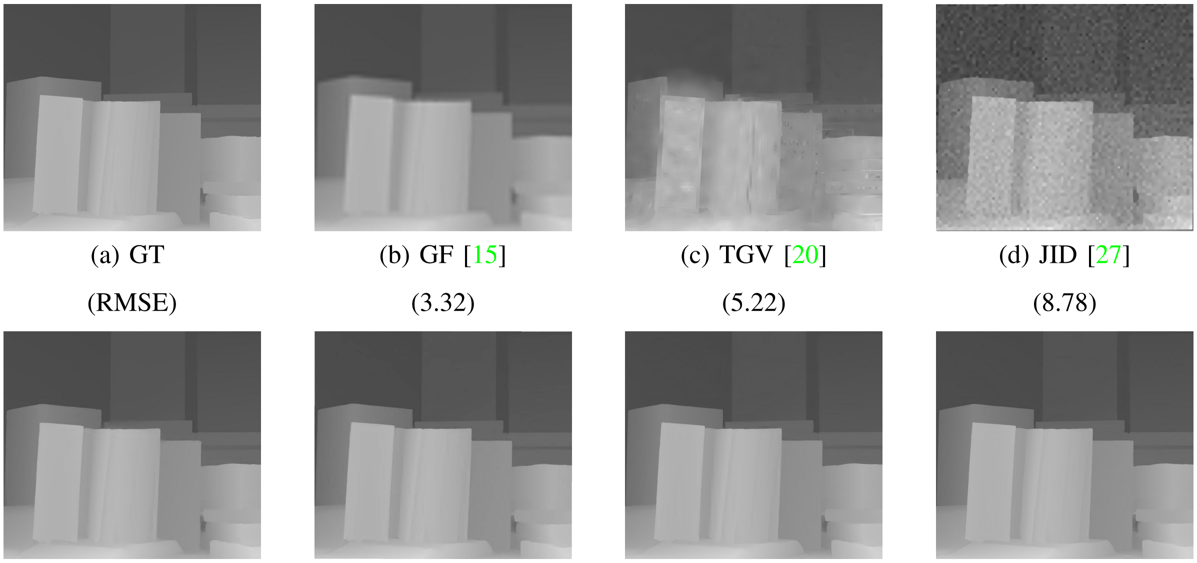
(a) GT (RMSE) (b) Bicubic (7.25) (c) TGV [20] (8.75) (d) TSDR [2] (8.23) (e) JID [27] (11.01)

(f) VDSR [37] (7.38) (g) MSG [4] (8.17) (h) DepthSR-Net (5.77) (i) -DdenoiseB (5.29) (j) -Ddenoise (4.43)

Fig. 5. An example of the results at 16× upsampling on **Test-ToF**. From (a) to (j) are the HR depth map and the results of Bicubic, TGV [20], TSDR [2], JID [27], VDSR [37], MSG [4], our DepthSR-Net, DepthSR-Net-DdenoiseB and DepthSR-Net-Ddenoise.

TABLE VI
QUANTITATIVE COMPARISONS ON **TEST-TOF** FOR 16× SCALING FACTOR IN TERMS OF RMSE/PSNR (DB) VALUES.

Method	Art	Books	Laundry	Reindeer	Tsukuba	Teddy	Average
Bicubic	9.12/28.93	5.28/33.69	6.48/31.89	7.12/31.08	17.04/23.50	7.25/30.93	8.72/30.00
GF [15]	9.11/28.94	5.16/33.88	6.45/31.94	7.05/31.17	17.08/23.48	7.29/30.88	8.69/30.05
TGV [20]	12.34/26.31	5.42/33.46	8.47/29.57	9.30/28.76	23.94/20.55	8.75/29.29	11.37/27.99
TSDR [2]	9.43/28.64	7.86/30.22	8.15/29.91	8.44/29.60	16.68/23.69	8.23/29.83	9.80/28.65
JID [27]	16.59/23.73	11.57/26.86	11.71/26.76	10.81/27.45	18.95/22.58	11.01/27.29	13.44/25.78
VDSR [37]	9.16/28.89	5.28/33.68	6.51/31.86	7.07/31.14	17.05/23.50	7.38/30.77	8.74/29.97
MSG [4]	8.45/29.60	8.19/29.87	7.47/30.66	9.22/28.84	17.76/23.14	8.17/29.89	9.88/28.67
DepthSR-Net	6.96/31.28	5.66/33.07	7.54/30.58	5.25/33.73	14.41/24.96	5.77/32.91	7.60/31.09
DepthSR-Net-DdenoiseB	6.24/32.23	3.36/37.62	4.95/34.24	4.65/34.79	15.02/24.60	5.29/33.68	6.59/32.86
DepthSR-Net-Ddenoise	5.76/32.92	2.41/40.50	2.66/39.63	3.47/37.33	12.87/25.94	4.43/35.20	5.27/35.25



(a) GT (RMSE) (b) GF [15] (3.32) (c) TGV [20] (5.22) (d) JID [27] (8.78)

(e) MSG [4] (1.74) (f) DepthSR-Net (1.54) (g) -YdenoiseB (1.51) (h) -Ydenoise (1.42)

Fig. 6. An example of the results at 16× upsampling on **Test-Ynoise**. From (a) to (h) are HR depth map, and the results of GF [15], TGV [20], JID [27], MSG [4], and our DepthSR-Net, DepthSR-Net-YdenoiseB, and DepthSR-Net-Ydenoise.

TABLE VII
QUANTITATIVE COMPARISONS ON **TEST-YNOISE** FOR 16× SCALING FACTOR IN TERMS OF RMSE/PSNR (DB) VALUES.

Method	Art	Books	Laundry	Reindeer	Tsukuba	Teddy	Average
GF [15]	8.14/29.92	3.32/37.72	5.03/34.09	5.82/32.83	16.59/23.73	6.02/32.53	7.49/31.80
TGV [20]	12.32/26.32	5.22/33.77	8.23/29.83	9.23/28.83	23.47/20.72	7.83/30.25	11.05/28.29
TSDR [2]	4.33/35.41	2.00/42.10	2.15/41.48	3.14/38.20	14.12/25.10	3.19/38.05	4.81/36.72
JID [27]	10.53/27.68	8.78/29.27	9.14/28.91	9.18/28.87	18.03/23.01	9.13/28.92	10.80/27.78
MSG [4]	4.82/34.46	1.74/43.31	2.88/38.94	3.55/37.12	14.39/24.97	3.98/36.14	5.23/35.82
DepthSR-Net	3.98/36.15	1.54/44.44	2.31/40.88	2.55/39.96	13.21/25.71	3.74/36.73	4.56/37.31
DepthSR-Net-YdenoiseB	3.88/36.35	1.51/44.57	2.22/41.19	2.57/39.92	13.23/25.70	3.81/36.51	4.54/37.37
DepthSR-Net-Ydenoise	3.88/36.35	1.42/45.10	2.10/41.70	2.49/40.21	13.50/25.52	3.66/36.85	4.51/37.62

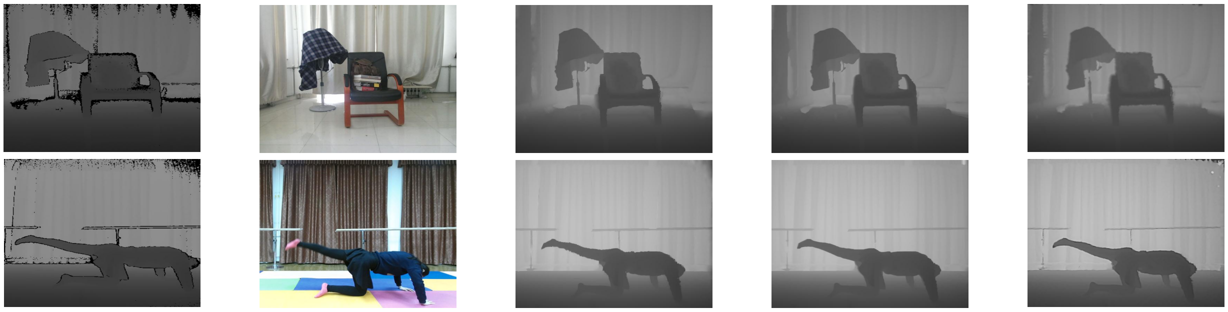

(a) Depth (b) RGB (c) TGV [20] (d) TSDR [2] (e) DepthSR-Net

Fig. 7. An example of the results on real data taken by Kinect. From (a) to (e) are raw depth maps from [2], RGB images, and the results of TGV [20], TSDR [2] and DepthSR-Net. **Best viewed with zoom in on a digital display.**

TABLE IX
QUANTITATIVE COMPARISONS ON **TEST** BY AVERAGE RMSE VALUES.

Method	Average RMSE Values				
	2×	3×	4×	8×	16×
U-Net	0.82	1.75	1.92	2.74	5.23
DepthSR-Net-w/o IP	0.75	1.64	1.83	2.66	4.67
DepthSR-Net-w/o HYG	0.68	1.66	1.86	2.73	5.10
DepthSR-Net-w/o RL	0.74	1.24	1.53	2.60	4.53
DepthSR-Net	0.68	1.04	1.36	2.45	4.38

- DepthSR-Net without hierarchical Y guidance (**DepthSR-Net-w/o HYG**)
- DepthSR-Net without residual learning (**DepthSR-Net-w/o RL**)

In these experiments, we use the same network parameters with the aforementioned settings. The evaluation is performed on the **Test** dataset. As shown in Table IX, we observed that 1) compared with the results of U-Net, DepthSR-Net has better performance for all scaling factors, which demonstrates the effectiveness of the combination of input pyramid, hierarchical Y guidance and residual learning; 2) comparisons between DepthSR-Net-w/o HYG and DepthSR-Net indicate that the hierarchical Y guidance branch is more helpful for large scaling factors; 3) comparisons between DepthSR-Net-w/o IP and DepthSR-Net demonstrate that the input pyramid branch can improve the accuracy of depth map SR; 4) residual learning can slightly improve the performance of depth SR.

V. APPLICATION

Aside from depth map SR problem, the proposed DepthSR-Net can handle other low-level vision tasks, such as transmission guided image dehazing and density-aware image deraining, where the guidance image is required or can boost the final performance. Here, we use single image dehazing as an example based on the limited space.

According to the atmospheric scattering model [52], a common hazy image formation can be described as

$$I(x) = J(x)t(x) + A(x)(1 - t(x)). \tag{18}$$

where x denotes the pixel coordinates, $I(x)$ is the observed image, $J(x)$ is the haze-free image, $A(x)$ is the global atmospheric light, and $t(x)$ is the medium transmission. The purpose of single image dehazing is to restore $J(x)$ from $I(x)$. Here, the purpose of DepthSR-Net is to produce clear image $J(x)$ with the guidance of transmission map $t(x)$. In other word, the input to the DepthSR-Net is a hazy image while the guidance image is a transmission map. Different from the intensity image used in the depth map SR, the transmission map is not directly available in single image dehazing. Thus, we first train a network (here we use original U-Net) to estimate accurate transmission map from an input hazy image. Other networks also can be used as the transmission estimation network. The main purpose of transmission estimation network is to provide the transmission map for haze removal network. We follow [53] to synthesize pairs of training data (hazy image/transmission map/clear image), and then separately optimize the transmission estimation network (*i.e.*, U-Net) and haze removal network (*i.e.*, DepthSR-Net) until convergence.

In Figure 8, it is visible that the compared methods leave haze on the results or make the resulting images darker. In comparison, our network can remove the effect of haze and unveil the characteristics of contrast and colors, which benefits from the rich hierarchical features extracted by the novel

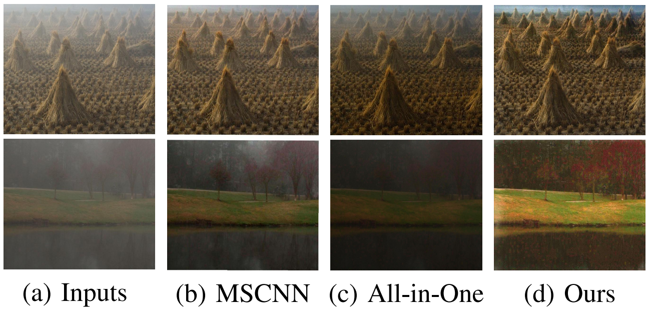


Fig. 8. Visual comparison results for image dehazing. From (a) to (d) are the input hazy images and the results of MSCNN [53], All-in-One [54], and our network.

network architecture. In short, experiment on this application provides additional evidence for the potential usage of the proposed DepthSR-Net in other low-level vision problems.

VI. Discussion and Conclusion

We have proposed a deep learning model for depth image SR, called DepthSR-Net. DepthSR-Net achieves the state-of-the-art performance on depth image SR task, especially for large scaling factors, which benefits from the multiple level receptive fields and rich hierarchical features provided by our novel network architecture. Besides, we observe that deep network architectures are more helpful when they are used for large scaling factors. The proposed DepthSR-Net also generalizes to other low-level vision problem.

Similarly to current deep learning-based depth SR methods, our DepthSR-Net did not take the misalignment and missing data into account. Thus, our method needs extra pre-processing steps when it suffers from the problems of misalignment and missing data. In future work, we will take these two issues into consideration in the design and training of depth SR networks.

References

[1] Q. Yang, R. Yang, J. Davis, and D. Nister, "Spatial-depth super resolution for range images," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2007, pp. 1-8. 1, 3

[2] Z. Jiang, Y. Hou, H. Yue, J. Yang, and C. Hou, "Depth super-resolution from RGB-D pairs with transform and spatial domain regularization," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2587-2602, 2018. 1, 2, 3, 6, 7, 8, 9, 10, 11

[3] C. Dong, C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38 no. 2, pp. 295-307, 2016. 1, 3, 6, 7, 8, 9

[4] T. Hui, C. Loy, and X. Tang, "Depth map super-resolution by deep multi-scale guidance," in *Proc. of Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 353-369. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

[5] M. Hornacek, C. Rhemann, M. Gelautz, and C. Rother "Depth super resolution by rigid body self-similarity in3D," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2013, pp. 1123-1130. 2

[6] J. Lei, L. Li, H. Yue, F. Wu, N. Ling, and C. Hou, "Depth map super-resolution considering view synthesis quality," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1732-1745, 2017. 2

[7] O. Aodha, N. Campbell, A. Nair, and G. Brostow, "Patch based synthesis for single depth image super-resolution," in *Proc. of Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 71-84. 2

[8] J. Li, Z. Lu, G. Zeng, R. Gan, and H. Zha, "Similarity-aware patchwork assembly for depth image super-resolution," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2014, pp. 3374-3381. 2

[9] J. Xie, R. Feris, and M. Sun, "Edge-guided single single depth image super resolution," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 428-438, 2016. 2

[10] L. Liu, S. Chan, and T. Nguyen, "Depth reconstruction from sparse samples: representation, algorithm, and sampling," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1983-1996, 2015. 2

[11] D. Ferstl, M. Ruther, and H. Bischof, "Variational depth superresolution using example-based edge representations," in *Proc. of IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 513-521. 2, 6

[12] J. Xie, R. Feris, S. Yu, and M. Sun, "Joint super resolution and denosing from a single depth image," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1525-1537, 2015. 2

[13] S. Mandal, A. Bhavsar, and A. Sao, "Depth map restoration from undersampled data," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 119-134, 2017. 2

[14] G. Riegler, M. Ruther, and H. Bischof, "Atgv-net: Accurate depth super-resolution," in *Proc. of Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 268-284. 3

[15] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35 no. 6, pp. 1397-1409, 2013. 3, 6, 7, 8, 9, 10, 11

[16] M. Liu, O. Tuzel, and Y. Taguchi, "Joint geodesic unpsampling of depth images," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2013, pp. 169-176. 3

[17] J. Lu and D. Forsyth, "Sparse depth super resolution," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2015, pp. 2245-2253. 3

[18] J. Park, H. Kim, Y. Tai, M. Brown, and I. Kweon, "High quality depth map upsampling for 3D-TOF cameras," in *Proc. of IEEE Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 1623-1630. 3

[19] J. Yang, X. Ye, K. Li, and C. Hou, "Depth recovery using an adaptive color-guided auto-regressive model," in *Proc. of Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 158-171. 3

[20] D. Ferstle, C. Reinbacher, R. Ranftl, M. Ruther, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *Proc. of IEEE Int. Conf. Comput. Vis. (ICCV)*, 2013, pp. 993-1000. 3, 6, 7, 8, 9, 10, 11

[21] J. Park, H. Kim, Y. Tai, M. Brown, and I. Kweon, "High-quality depth map upsampling and completion for RGB-D cameras," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5559-5572, 2014. 3

[22] J. Yang, X. Ye, K. Li, C. Hou, and Y. Wang, "Color-guided depth recovery from RGB-D data using an adaptive autoregressive model," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3962-2969, 2014. 3, 6

[23] B. Ham, M. Cho, and J. Ponce, "Robustimage filtering using joint static and dynamic guidance," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2015, pp. 4823-4831. 3

[24] B. Ham, D. Min, and K. Sohn, "Depth superresolution by transduction," *IEEE Trans. Image Process.*, vol. 24, no. 5, pp. 1524-1535, 2015. 3

[25] W. Liu, X. Chen, J. Yang, and Q. Wu, "Robust color guided depth map restoration," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 315-327, 2017. 3

[26] W. Dong, G. Shi, X. Li, K. Peng, J. Wu, and Z. Guo, "Color-guided depth recovery via joint local structural and nonlocal low-rank regularization," *IEEE Trans. Multi.*, vol. 19, no. 2, pp. 293-301, 2017. 3

[27] M. Kiechle, S. Hawe, and M. Kleinsteuber, "A joint intensity and depth co-sparse analysisi model for depth map super-resolution," in *Proc. of IEEE Int. Conf. Comput. Vis. (ICCV)*, 2013, pp. 1545-1552. 3, 6, 7, 8, 9, 10, 11

[28] H. Kwon, Y. Tai, and S. Lin, "Data-driven depth map refinement via multi-scale sparse representation," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2015, pp. 159-167. 3

[29] W. Zhou, X. Li, and D. Reynolds, "Guided deep network for depth map super-resolution: How much can color help?," in *Proc. of IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 1457-1461. 3, 4

[30] J. Kopf, M. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Trans Graph.*, vol. 26, no. 3, pp. 169-176, 2007. 3

[31] R. Cong, J. Lei, H. Fu, Q. Huang, X. Cao, and C. Hou, "Co-saliency detection for RGBD images based on multi-constraint feature matching and cross label propagation," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 568-579, 2018. 3

[32] W. Wang, J. Shen, Y. Yu, and K. Ma, "Stereoscopic thumbnail creation via efficient stereo saliency detection," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 8, pp. 2014-2027, 2017. 3

[33] C. Li, J. Guo, R. Cong, Y. Pang, and B. Wang, " Underwater image enhancement by dehazing with minimum information loss and histogram distribution prior," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5664-5677, 2016. 3

[34] W. Wang and J. Shen, "Deep Visual attention prediction," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2368-2378, 2018. 3

[35] C. Li, J. Guo, and C. Guo "Emerging from water: underwater image color correction based on weakly supervised color transfer," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 323-327, 2018. 3

[36] S. Anwar, Z. Hayder, and F. Porikli, "Depth estimation and blur removal from a single out-ot-focus image," in *Proc. of British Conf. Mach. Vis. (BMVC)*, 2017, pp. 1-12. 3

[37] J. Kim, J. Lee, and K. Lee, "Accurate image super-resolution using very deep conlulutional networks," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2016, pp. 1646-1654. 3, 5, 6, 7, 8, 9, 10

[38] J. Kim, J. Lee, and K. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2016, pp. 1637-1645. 3

[39] W. Lai, J. Huang, N. Ahujia, and M. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2017, pp. 624-632. 3

[40] C. Ledig, L. Theis, F. Huszar, etal, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2017, pp. 4681-4690. 3

[41] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive redisuasl network," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2017, pp. 3147-3155. 3

[42] X. Wang, K. Yu, C. Dong, and C. Loy, "Recovering realistic texutre in image super-resolution by deep spatial feature transform," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2018. 3

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. of Med. Image Comput. Comput. Assit. Inter. (MICCAI)*, 2015, PP. 234-241. 4

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2016, pp. 770-778. 4, 5

[45] D. Butler, J. Wulff, G. Stanley, and M. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. of Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 612-625. 5

[46] D. Scharstein and R. Szeliski, "A taxonolmy and evaluation of dense two-fram stereo correspondence algorithms," *Inter. J. Comput. Vis.*, vol. 47, no. 1, pp. 7-42, 2002. 5

[47] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Proc. of IEEE Int. Conf. Comput. Vis. Pattern Rec. (CVPR)*, 2007, pp. 1-8. 5

[48] D. Scharstein, H. Hirschmuller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westlingl "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. of German Conf. Pattern Rec. (GCPR)*, 2014, pp. 31-42. 5

[49] F. Albluwi, V. Krylov, and R. Dahyot, "Image deblurring and super-resolution using deep convolutional neural networks," in *IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP)*, 2018, pp.1-6. 9

[50] R. Cong, J. Lei, H. Fu, W. Lin, Q. Huang, X. Cao, C. Hou "An iterative co-saliency framework for RGBD images," *IEEE Trans. Cybern.*, vol. 49, no. 1, pp. 233-246, 2019. 9

[51] L. Qu, S. He, J. Zhang, J. Tian, Y. Tang, and Q. Yang, "RGBD salient object detection via deep fusion," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2274-2285, 2017. 9

[52] H. Koschmieder, "Theorie der horizontalen sichtweite," in *Beitrage zur Physik der freien Atmosphare*, 1924. 11

[53] W. Ren, S. Liu, H. Zhang, J. Pan, and X. Cao, "Single image dehazing via multi-scale convolutional neural networks," in *Proc. of Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 154-169. 11, 12

[54] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "Aod-net: All-in-one dehazing network," in *Proc. of IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 4780-4788. 12

Chunle Guo received his B.S. degree from School of Electronic Information Engineering in Tianjin University. He is currently pursuing his Ph.D. degree with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China. His current research focuses on image processing and computer vision, particularly in the domains of deep learning-based image restoration and enhancement.

Chongyi Li received his Ph.D. degree from Tianjin University, China, in June 2018. From 2016 to 2017, he took one year study at the Research School of Engineering, Australian National University (ANU) as a visiting Ph.D. student. Now, he is a Postdoc Research Fellow at the Department of Computer Science, City University of Hong Kong (CityU), Hong Kong. His research interests include image processing, computer vision, and deep learning.

Jichang Guo received his M.S. and Ph.D. degrees from the School of Electronic Information Engineering, Tianjin University, Tianjin, China, in 1993 and 2006, respectively. He is a Professor at School of Electrical and Information Engineering, Tianjin University. His research interests include image processing, video coding, and computer vision.

Runmin Cong is currently pursuing his Ph.D. degree in information and communication engineering with Tianjin University, Tianjin, China. He was a visiting student with Nanyang Technological University (NTU), Singapore, from Dec. 2016 to Feb. 2017. Since May 2018, he has been working as a Research Associate at the Department of Computer Science, City University of Hong Kong (CityU), Hong Kong. His research interests include computer vision, image processing, and saliency detection.

Huazhu Fu (SM'18) received the Ph.D. degree in computer science from Tianjin University, China, in 2013. He spent about two years at Nanyang Technological University, Singapore, as a research fellow. In 2015 to 2018, he was a Research Scientist at the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. He is currently a Senior Scientist at Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates. His research interests include computer vision, image processing, and medical image analysis.

Ping Han received her Ph.D. degree from Tianjin University, Tianjin, China in 2004. She is currently a professor with the College of Electronic Information and Automation, Civil Aviation University of China (CAUC), Tianjin, China. Her current research interests include SAR image processing, target detection, and pattern recognition.