Depth Super-Resolution via Deep Controllable Slicing Network

Xinchen Ye[1,2,*], Baoli Sun[1], Zhihui Wang[1,2], Jingyu Yang[3], Rui Xu[1,2], Haojie Li[1,2], Baopu Li[4]
[1]International School of Information Science & Engineering, Dalian University of Technology, China
[2]Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, China
[3]School of Electrical and Information Engineering, Tianjin University, China
[4] Baidu Research, USA

ABSTRACT

Due to the imaging limitation of depth sensors, high-resolution (HR) depth maps are often difficult to be acquired directly, thus effective depth super-resolution (DSR) algorithms are needed to generate HR output from its low-resolution (LR) counterpart. Previous methods treat all depth regions equally without considering different extents of degradation at region-level, and regard DSR under different scales as independent tasks without considering the modeling of different scales, which impede further performance improvement and practical use of DSR. To alleviate these problems, we propose a deep controllable slicing network from a novel perspective. Specifically, our model is to learn a set of slicing branches in a divide-and-conquer manner, parameterized by a distance-aware weighting scheme to adaptively aggregate different depths in an ensemble. Each branch that specifies a depth slice (e.g., the region in some depth range) tends to yield accurate depth recovery. Meanwhile, a scale-controllable module that extracts depth features under different scales is proposed and inserted into the front of slicing network, and enables finely-grained control of the depth restoration results of slicing network with a scale hyper-parameter. Extensive experiments on synthetic and real-world benchmark datasets demonstrate that our method achieves superior performance.

CCS CONCEPTS

• **Computing methodologies** → **3D imaging**; **Reconstruction**.

KEYWORDS

Scene Depth, Super-Resolution, Distance-Aware, Slicing, Controllable

ACM Reference Format:
Xinchen Ye[1,2,*], Baoli Sun[1], Zhihui Wang[1,2], Jingyu Yang[3], Rui Xu[1,2], Haojie Li[1,2], Baopu Li[4] . 2020. Depth Super-Resolution via Deep Controllable Slicing Network. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394171.3413874

*Corresponding author: yexch@dlut.edu.cn. This work was supported by National Natural Science Foundation of China (NSFC) under Grant 61702078, 61772108, 61976038, 61772106. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM '20, October 12–16, 2020, Seattle, WA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413874

1 INTRODUCTION

3D content of a given scene consists of two components, i.e., two-dimensional texture (color) image, and depth information of a third dimension. Scene depth map is essential and widely used as a basic element in many computer vision fields, such as 3D reconstruction [18] [43], autonomous navigation [30] [11], monitoring [8] [10], and so on. However, the acquisition of depth maps is still a challenging task in real conditions, which needs costive specialized equipment, e.g., Time-of-Flight (ToF) depth sensors. Due to the imaging limitation of these depth sensors, high-resolution (HR) depth maps are often difficult or even impossible to be acquired directly. Hence, effective depth super-resolution (DSR) algorithms are needed to yield HR output from the degraded low resolution (LR) counterpart. Recently, CNN-based methods [13, 24, 40, 41] have been proposed to recover depth maps by learning a set of kernels or filters from data instead of hand-designed ones. Although these CNN-based methods present impressive performance, the task of DSR still needs to be improved because of its unsatisfactory performance in terms of *accuracy* and *practicality*.

1.1 Motivation

Scene depth recovery depends on scene characteristics, i.e., depth regions on close-view and tiny objects are inclined to be destroyed by downsampling degradation more seriously than the far-view object and background. Meanwhile, most existing DSR methods [13, 40] have unbalanced estimation in one depth map with a same model. Hence, it may cause inaccurate depth recovery for different depth regions. Besides, depth map captured by a depth sensor are usually polluted with distance-dependent Gaussian noise, i.e, the intensity of the noise depends on the scene depth [5]. However, previous studies mainly focus on a single model to process all regions of a depth map without considering the above complex degradation at region-level, which is suboptimal due to the statistical, computational, and representational limitations. Therefore, we expect to develop a new method to discriminately process each region within a depth map by its depth range in the scenario of DSR.

In addition, most existing algorithms treat DSR task of different scale factors (×2, ×4, ×8, ×16) as independent problems, and require many scale-specific networks that need

to be trained independently to deal with various scales. However, in real world applications, the truly-wanted upscaling factors of the given scenes are fractional (not integer) or even unknown. Alternatively testing on the current LR input to find a suitable model among all the well-trained scale-specific ones is time-consuming and impractical, or even cannot obtain the desired results. Moreover, perceptual quality of the restored depth map is relatively subjective, and it is necessary for the model to finely-grained control the depth restoration according to image characteristics, which cannot be done using existing deterministic networks. Therefore, a controllable network with high generalization ability to different upscaling factors is needed.

1.2 Scope and Contributions

Based on the above analyses, this paper breaks away the shackles of general paradigms and introduces a distance-aware deep controllable slicing network from a novel perspective, as shown in Fig. 1. Specifically, we propose a slicing network architecture to learn a set of slicing branches to specify some depth range, and this novel network tends to yield accurate depth recovery in a divide-and-conquer manner. To make each branch more accurate at representing different depth, we propose a distance-aware weighting scheme to generate a set of weighting masks from the depth map features, which are applied on the estimated results from the slicing branches to make these branches focus on their specific depth regions, and adaptively aggregate all the slicing branches in the ensemble. Meanwhile, we design a scale-controllable module that extracts different depth features before the slicing network, which realizes to super-resolve LR images with different downscaling factors. The module consists of three branches, i.e., generalized branch (GB), specialized branch (SB), and fusion branch (FB). GB aims to extract the common features from the input, while SB takes the given scale parameter and its corresponding LR depth map as input, then the generated specialized features that contain different scale information are fused with the features of GB through FB in a multi-scale fashion, enabling a richer depth representation. The proposed module can finely-grained control the depth restoration according to different depth map degradation through the pre-defined scale hyper-parameter. Our main contributions are summarized as follows:

1) An end-to-end deep controllable slicing network to realize region-level depth recovery and high generalization ability for the task of DSR.

2) A scale-controllable module (SCM), which realizes the fine-grained control of depth restoration with arbitrary magnification in one united model.

3) A depth slicing module (DSM), which discriminately uses depth map features with different depth ranges to super-resolve the depth map in a divide-and-conquer manner.

Note that our method stands on single depth map SR without the aid of color information, but performs better than other color-guided DSR methods on both synthetic and real-world datasets. Besides, our model that is trained on all the scales together is also superior to other state-of-the-art scale-specific models that are trained independently on each scale.

2 RELATED WORK

2.1 CNN-based Depth Super-Resolution

Depending on the input data, DSR methods can be mainly divided into single depth map SR with only LR depth map as input and color-guided DSR with LR depth map and its corresponding HR color image as input.

Riegler *et al.* [32] integrated the piecewise affine structures into an CNN which combines the advantage of data driven methods and energy minimization models to recover the accurate HR depth map. Hui *et al.* [17] proposed a low-to-high resolution network to progressively extract features and raise the spatial resolution. Some existing DSR methods use color information as guidance to recover the degraded depth maps. Li *et al.* [25] employed a two-path CNN to obtain the HR depth map which is designed based on the concept of joint filters. Based on the model of single DSR, Hui *et al.* [17] also applied 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 DSR. Wen *et al.* [40] used the color information as guidance to infer an initial HR depth map, then proposed a coarse-to-fine networks to progressively optimize the depth map. Wang *et al.* [39] put forward a DSR network to learn a binary map of depth edges and then recovered the HR depth map based on edge-guided filter or cascaded network modules. However, due to treating all depth regions equally without considering depth range variation, there is still room for the above methods to make further improvement.

2.2 Ensemble Strategy

For image restoration tasks, the ensemble strategy has been explored to boost a network's performance. As studied in [4], a single model or a model with only one branch is usually subject to computational and representational limitations. To alleviate this problem, Zhang *et al.* [42] proposed a light-weight ensemble method to improve the generalization of negative correlation learning for regression problems. Qin *et al.* [31] suggested a difficulty-aware image SR method that use a dual-way network to separately recover easy image regions and hard ones. Li *et al.* [23] applied a learning-based adapting method to ensemble the outputs from multiple models, which can exploit the information among successive video frames. Inspired by the above methods that capture different features, structures or sub-components in an image with separate models or branches, we make use of a slicing network to discriminately process each region within a depth map by its depth range in the scenario of DSR.

2.3 Generalization Ability

Many techniques have been explored to improve the generalization ability of network for image restoration. Kim *et al.* [21] and Gao *et al.* [9] both proposed a joint-training strategy

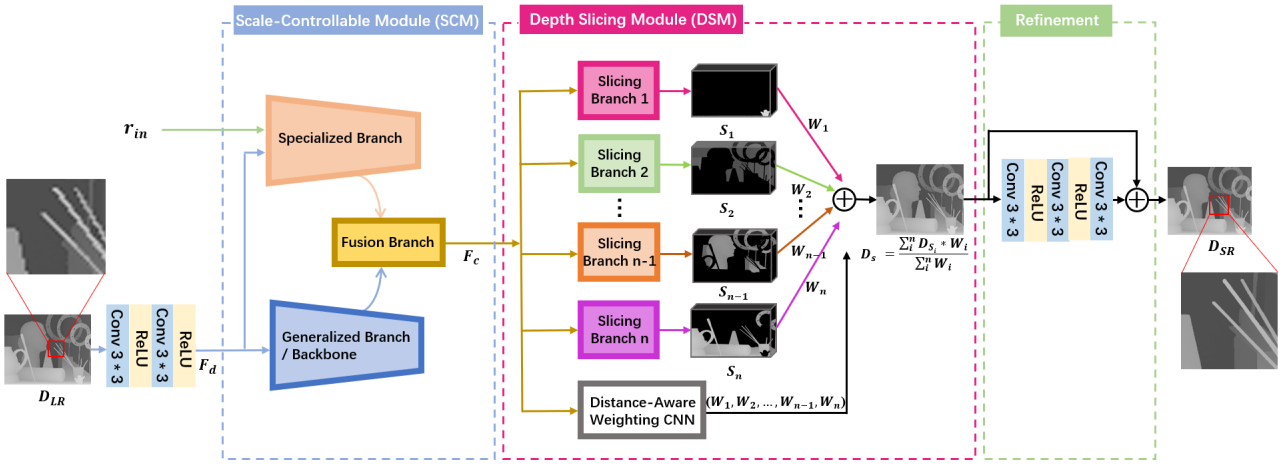

Figure 1: Overview of the network architecture. SCM is to realize DSR with different downscaling factors in an unified model, which allows to finely-grained control the depth restoration results by using a scale parameter, while DSM aims to learn a set of slicing branches in a divide-and-conquer manner, parameterized by a distance-aware weighting scheme to adaptively aggregate all the branches in the ensemble.

to learn a single image SR network with different downsampling inputs together. Based on [21], Lim *et al.* [27] used a single main branch as backbone and further applied three separate scale-specific processing modules ($\times 2$, $\times 3$, $\times 4$) after the backbone to improve the network generalization. However, these methods are still subject to insufficient network representation, leading to a relatively low restoration accuracy. To realize the regulation and control for image restoration, Wang *et al.* [38] performed image interpolation in the parameter space for continuous imagery effect transition. Hu *et al.* [16] and Jo *et al.* [19] advocated to generate dynamic upsampling filters/kernels according to different upscaling factors for image and video SR respectively. However, these methods still need to be further improved with regard to the fine-grained control of image restoration.

3 PROPOSED METHOD

Fig. 1 outlines the whole architecture and detailed configuration of our proposed deep controllable slicing network. Let D_{LR} and γ_{in} be the LR depth map (interpolated to the desired output size) and the upscaling factor as input, respectively. The goal is to predict the corresponding super-resolved depth map D_{SR}. Note that color information can bring improvement for DSR, but may introduce texture-copying and depth bleeding artifacts because of the depth-color inconsistency. Therefore, our method focuses on singe depth map SR, and achieves superior performance to other color-guided methods.

The proposed model can be divided into three components: scale-controllable module (SCM), depth slicing module (DSM) and the final refinement. SCM consists of three branches, i.e., generalized branch (GB, our backbone), specialized branch (SB), and fusion branch (FB). The shallow features F_d extracted from D_{LR} are sent into both GB and SB, and

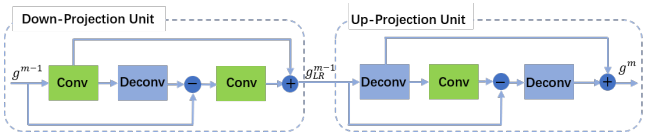

Figure 3: Down- and up-projection units. 'Deconv' and 'Conv' represent de-convolution and convolution layers, which can be regarded as upsampling and downsampling operators to transform features between HR and LR spatial domains, respectively.

γ_{in} is additionally fed into SB to represent the desired scale. FB aggregates features from SB and GB in a multi-scale fashion, and then outputs the intermediate feature map F_c. In DSM, F_c is spatially split into n regions according to different depth ranges, and is sent into the corresponding slicing branches. Then, the intermediate depth map D_s is obtained by weighted-averaging the outputs from slicing branches. Finally, D_s is refined by a simple residual structure to output the final super-resolved depth map D_{SR}.

3.1 Scale-Controllable Module

To realize the fine-grained control of depth restoration with arbitrary upscaling factors, we propose a scale-controllable module (SCM) to learn a set of controllable features. As shown in Fig. 2, GB can be regarded as our backbone and contains M stacked generalized blocks, while SB includes M stacked specialized blocks together with M successive fully connected blocks. FB contains M fusion blocks to aggregate the output features from generalized blocks, specialized blocks and fully connected blocks at different stages in a multi-scale fashion.

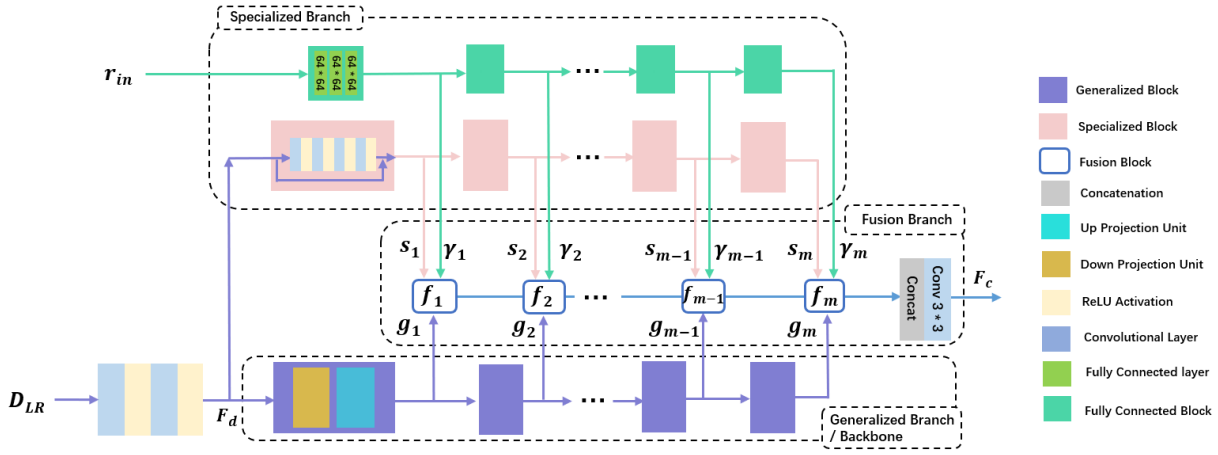

Figure 2: Network architecture of our SCM. GB aims to extract the common features from the input, while SB takes the given scale parameter and its corresponding LR depth map as input, then generates specialized features to adaptively tune the features of GB through FB in a linear fusion fashion, e.g., $f^m = g^m + \gamma^m \cdot s^m$.

Each generalized block is built upon a down-projection unit and an up-projection unit [14], as shown in Fig. 3. It can effectively improve the feature representations at depth boundaries through iterative projecting HR representations to LR spatial domain and then mapping the reconstruction errors back into the HR domain. The specialized block is a simple residual architecture consisting of four convolution layers, each followed by a ReLU activation.

The initial depth feature F_d is extracted from D_{LR} through a series of shallow feature extraction operations such as convolution. Then, F_d is sent into GB and SB, and we get:

$$g^m = \mathcal{G}^m_{gb}(g^{m-1}), g^0 = F_d, \tag{1}$$

$$s^m = \mathcal{G}^m_{sb}(s^{m-1}), s^0 = F_d, \tag{2}$$

where $\mathcal{G}^m_{gb}(\cdot)$ and $\mathcal{G}^m_{sb}(\cdot)$ denote the m-th generalized block and specialized block respectively. g^m and s^m denote the output features of m-th generalized block and specialized block respectively.

We introduce γ_{in} to explicity control the scale of the depth feature, and it may further lead to adaptive fusion of the outputs of generalized blocks and specialized blocks. Specifically, we introduce fully connected block to map an input scale hyper-parameter γ_{in} into different vectors. Each vector has different fusion coefficients for the multi-channel features of generalized blocks and specialized blocks. We use three fully connected layers to implement this mapping operation:

$$\gamma^m = \mathcal{G}^m_{fc}(\gamma^{m-1}), \gamma^0 = \gamma_{in}, \tag{3}$$

where $\mathcal{G}^m_{fc}(\cdot)$ denotes the m-th fully connected block and γ^m is the fusion vector of m-th stage. Then, we introduce the linear fusion operation to selectively treat the output features at every feature extraction stage, which is defined as follows:

$$f^m = g^m + \gamma^m \cdot s^m, \tag{4}$$

where each value in γ^m is multiplied on the corresponding channel of s^m. Finally, all the outputs from fusion blocks are concatenated and filtered by an 3×3 convolution to output the feature map F_c. Through varying the value of γ_{in}, the fine-grained control of depth restoration with arbitrary magnification can be realized.

To obtain the generalized features and specialized features discriminatively from GB and SB, the whole network is learned with a two-stage training strategy based on different optimization objectives. More training details are described in Sec. 3.3.

3.2 Depth Slicing Module

We formulate our depth slicing module (DSM) to discriminately mitigate complex degradation at region-level. DSM consists of a set of slicing branches and a distance-aware weighting CNN, in which each has a same network architecture with the specialized block, i.e., a simple residual block. Each slicing branch S_n that specifies some depth range tends to yield accurate depth estimation D_{S_n} from the shared depth feature F_c on the corresponding depth region:

$$D_{S_n} = \mathcal{G}^n_{slice}(F_c), \tag{5}$$

where $\mathcal{G}^n_{slice}(\cdot)$ denotes the n-th slicing block. To achieve accurate regression, we also propose a distance-aware weighting scheme to learn the fusion weights for adaptively aggregating all the slicing branches in an ensemble. The output of the weighting CNN is a set of weighting masks $\{W_n, n \in [1, N]\}$, which are defined as follows:

$$[W_1, W_2, ..., W_{n-1}, W_n] = \mathcal{G}_w(F_c), \tag{6}$$

where $\mathcal{G}_w(\cdot)$ denotes the weighting CNN. The weighting mask W_n consists of 0 and 1, where 1 represents that this pixel belongs to corresponding depth ranges. Our proposed distance-aware weighting scheme is automatically aware of

multiple depth regions according to different depth ranges (distance within D_{S_n}), and assists each slicing branch to recover the depth values focusing on its corresponding depth region. Thus, the outputs of all the slicing branches are aggregated with the estimated weighting masks, and generate a accurate recovered depth map D_S on all the depth regions:

$$D_S = \frac{\sum_i^n D_{S_i} * W_i}{\sum_i^n W_i}, \tag{7}$$

where $*$ denotes the element-wise multiplication.

When fusing slicing branches, the intermediate depth map D_S may demonstrate some errors at the junction of different depth regions. Therefore, we add a refinement module $\mathcal{G}_{refine}(\cdot)$ at the end of the DSM module to further enhance the intermediate depth map. The final super-resolved depth map D_{SR} is reconstructed from D_S by using a sample residual block with three 3×3 convolution layers interlaced by two ReLU activations.

Note that, by dividing the DSR task into multiple subproblems applied on each specific depth range, our method shares the essence of ensemble strategy and yields more robust estimations than a single DSR network.

3.3 Training Algorithm

Training Data Generation. Given that we have a collection of LR-HR paired depth maps, which contain HR depth maps $D_{HR} \in \mathcal{D}_{HR}$ and the corresponding LR depth maps $D_{LR} \in \mathcal{D}_{LR}$ (×2, ×4, ×8, ×16 cases downsampled from HR depth maps). We produce the ground truth depth regions $D_{S_n}^{gt} \in \mathcal{D}_S$ by slicing the corresponding HR depth map D_{HR} into N regions with equal depth range according to the maximum distance. Meanwhile, we generate the ground truth weighting masks $W_n^{gt} \in \mathcal{W}$ by setting the non-zero values of the corresponding sliced depth map $D_{S_n}^{gt}$ as one to form a binary map.

Loss Functions. All the slicing branches are trained in a supervised manner such that each one becomes accurate and specialized on a specific depth range. The loss function for training the slicing branches can be formulated as follows:

$$\mathcal{L}_S = \sum_i^n ||D_{S_i} - D_{S_i}^{gt}||_1, \tag{8}$$

The weighting CNN is also trained in a supervised manner, and its loss function is engaged to enforce the regions of concern with regard to a specific depth range, which is defined as

$$\mathcal{L}_W = \sum_i^n ||W_i - W_i^{gt}||_1, \tag{9}$$

For the final depth map reconstruction, we directly measure the pixel-wise difference between the predicted depth map D_{SR} and its corresponding ground truth D_{HR} as a task loss to encourage an accurate regression:

$$\mathcal{L}_T = ||D_{SR} - D_{HR}||_1, \tag{10}$$

Combining the above three losses, then we have the overall loss for our model:

$$\mathcal{L} = \mathcal{L}_T + \lambda_1 \mathcal{L}_S + \lambda_2 \mathcal{L}_W, \tag{11}$$

where λ_1 and λ_2 are the trade-off parameters.

Training Strategy. We realize the fine-grained controllable feature learning by a two-stage training strategy. For the first stage, we fix all vectors $\{\gamma^m, m \in [1, M]\}$ as all-one vectors and remove the fully connected block from SB. We train the rest modules together on the dataset with all the downsampling inputs. Note that, this stage aims to learn the generalization features from diverse degraded depth maps. For the second stage, we fix the parameters of GB and add the fully connected blocks back to SB. We set the scale factor γ_{in} as a designated value that dynamically matches the type of input depth map, i.e., γ_{in} can be set as 1, 2, 3, 4 corresponding to ×2, ×4, ×8, and ×16 LR depth maps, respectively. The network learns the specialized features by updating the parameters of SB and DSM. At testing phase, the scale factor γ_{in} can be set at any values (from 1 to 4) to tune the restoration performance without knowing the real upscaling factors beforehand, which is very practical for the DSR task tested on real-world inputs.

4 EXPERIMENTAL RESULTS

To generate training data, we use 38 depth maps (6, 2, 21, 9 depth maps from 2001 [2], 2003 [35], 2006 [15] and 2014 [33] datasets, respectively) from Middlebury dataset. To test the performance, we conducted experiments on Middlebury 2005 [34] dataset (6 standard test depth maps, i.e., *Art, Books, Moebius, Dolls, Laundry, Reindeer*), and evaluate the generalization on MPI Sintel dataset (5 depth frames, i.e., *Ambush_2-15, Ambush_4-12, Ambush_5-41, Twmple_3-23*) and TofMark dataset [7] (3 real-world depth maps, i.e., *Books, Shark, Devil*, captured by ToF depth sensors). Another training and testing dataset is NYU v2 RGB-D dataset [36] 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. We randomly extract 15000+ depth patches of a fixed size of 256×256 from HR depth maps and augment the training dataset by 180-degree-rotation. The corresponding LR depth patches are the squared size of 128, 64, 32, and 16 according to 2, 4, 8, and 16 scale factors respectively. The metric of Mean Absolute Difference (MAD) is used for objective evaluation. During training, we set the number of generalized/specialized blocks as $M = 4$ and the number of slicing branches as $N = 5$. We set the trade-off parameters as $\lambda_1 = 0.2$ and $\lambda_2 = 0.2$ after trials. For optimization, we used Adam optimizer with momentum = 0.9, $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-8}$. The initial learning rate is set to 0.0001 and decreased by multipling by 0.1 for every 50 epochs. We implemented our models by PyTorch framework with GPU acceleration.

Table 1: Quantitative depth upsampling results on Middlebury dataset. (lower MADs, 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	0.48	0.97	1.85	3.59	0.13	0.29	0.59	1.15	0.20	0.36	0.66	1.18	0.28	0.54	1.04	1.95	0.13	0.30	0.59	1.13	0.30	0.55	0.99	1.88
FGI [26]	0.70	1.29	2.41	4.51	0.43	0.74	1.16	1.91	0.54	0.93	1.44	2.12	0.51	0.91	1.59	2.68	0.42	0.72	1.13	1.81	0.50	0.87	1.58	2.72
TGV [6]	0.45	0.65	1.17	2.30	0.18	0.27	0.42	0.82	0.21	0.33	0.70	2.20	0.31	0.55	1.22	3.37	0.18	0.29	0.49	0.90	0.32	0.49	1.03	3.05
DJF [25]	0.12	0.40	1.07	2.78	0.05	0.16	0.45	1.00	**0.06**	0.20	0.49	0.99	0.07	0.28	0.71	1.67	0.06	0.18	0.46	1.02	0.07	0.23	0.60	1.36
MSG [17]	-	0.46	0.76	1.53	-	0.15	0.41	0.76	-	0.25	0.51	0.87	-	0.30	0.46	1.12	-	0.21	0.43	0.76	-	0.31	0.52	0.99
DGDIE [12]	0.20	0.48	1.20	2.44	0.14	0.30	0.58	1.02	0.16	0.34	0.63	0.93	0.15	0.35	0.86	1.56	0.14	0.28	0.58	0.98	0.16	0.35	0.73	1.29
DEIN [41]	0.23	0.40	0.64	1.34	0.12	0.22	0.37	0.78	0.12	0.22	0.38	0.73	0.13	0.23	0.36	0.81	0.11	0.20	0.35	0.73	0.15	0.26	0.40	0.80
CCFN [40]	-	0.43	0.72	1.50	-	0.17	0.36	0.69	-	0.25	0.46	0.75	-	0.24	0.41	**0.71**	-	0.23	0.39	0.73	-	0.29	0.46	0.95
GSRPT [3]	0.22	0.48	0.74	1.48	0.11	0.21	0.38	0.76	0.13	0.28	0.48	0.79	0.12	0.33	0.56	1.24	0.12	0.24	0.49	0.80	0.14	0.31	0.61	1.07
DSR_N [39]	0.12	0.25	0.61	1.80	**0.04**	0.11	0.28	0.69	**0.06**	0.14	0.33	0.73	**0.06**	0.15	0.43	1.24	**0.05**	0.13	0.29	0.67	**0.07**	0.15	0.35	0.92
Ours	**0.10**	**0.23**	**0.58**	**1.30**	0.06	**0.09**	**0.26**	**0.63**	0.07	**0.11**	**0.31**	**0.69**	**0.06**	**0.14**	**0.34**	0.77	0.06	**0.12**	**0.27**	**0.64**	**0.07**	**0.14**	**0.33**	**0.79**

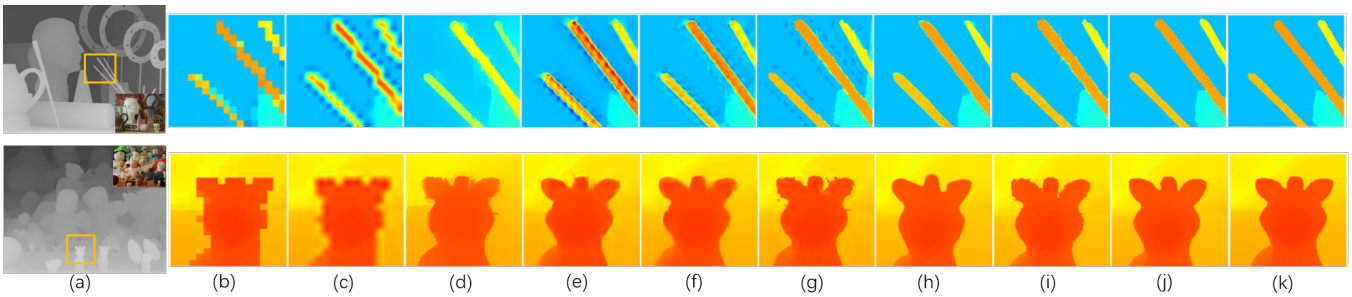

Figure 4: Visual comparison of ×8 upsampling results on *Art, Dolls*: (a) GT; (b) LR; (c) Bicubic, (d) FGI [26], (e) DJF [25], (f) DGDIE [12], (g) DEIN [41], (h) GSRPT [3], (i) DSR_N [39], (j) Ours, (k) GT.

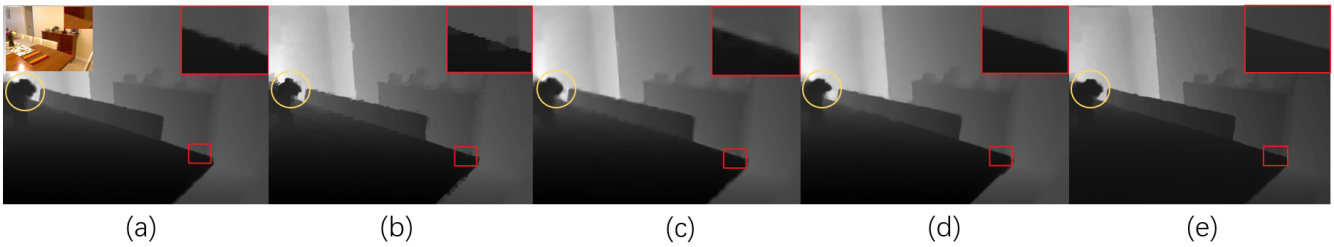

Figure 5: Visual comparison for recovered depth maps from ×8 downsampling on NYU v2 dataset. (a) GT, (b) JBU [22]. (c) DJF [25], (d) SVLRM [28], (e) Ours.

Table 2: Quantitative depth upsampling results on real NYU v2 dataset.

Method	JBU [22]	DJF [25]	DGDIE [12]	GbFT [1]	PAC [37]	SVLRM [28]	DKN [20]	Ours
×4	4.07	3.54	1.56	3.35	2.39	1.74	1.62	**1.33**
×8	8.29	6.20	2.99	5.73	4.59	5.59	3.26	**2.87**
×16	13.35	10.21	5.24	9.01	8.09	7.23	6.51	**5.12**

4.1 Performance Comparison

Depth SR under Noiseless Cases: To validate the superiority of our method, we first evaluate noiseless cases on Middlebury and NYU datasets, respectively.

Table 3: Quantitative depth upsampling results noisy Milddleburry dataset.(lower MADs, better performance)

	Art		*Books*		*Dolls*		*Laundry*		*Moebius*		*Reindeer*	
	×8	×16	×8	×16	×8	×16	×8	×16	×8	×16	×8	×16
TGV [6]	2.76	6.87	1.49	2.74	1.75	3.71	1.89	4.16	1.72	3.99	1.75	4.40
MSG [17]	1.57	2.98	1.18	1.48	1.12	1.78	**1.03**	1.89	1.13	1.76	1.12	1.87
DGDIE [12]	1.84	3.34	1.29	2.04	1.39	2.05	1.73	2.67	1.37	2.16	1.33	2.19
DEIN [41]	2.44	4.24	1.44	2.38	1.55	2.45	1.77	3.20	1.64	3.29	1.46	3.87
GSRPT [3]	1.33	2.47	**0.87**	1.37	1.26	**2.03**	1.24	1.86	**1.03**	1.68	1.04	**1.70**
DSR_N [39]	1.60	3.25	1.21	1.98	1.33	2.16	1.44	2.64	1.24	2.16	1.29	2.35
Ours	**1.20**	**2.13**	0.96	**1.33**	**1.11**	2.06	1.19	**1.82**	**1.03**	**1.61**	**0.91**	1.82

For Middlebury dataset, we compare with DJF [25], MSG [17], DGDIE [12], DEIN [41], CCFN [40], GSRPT [3], DSR_N [39], which are learning-based methods based on color

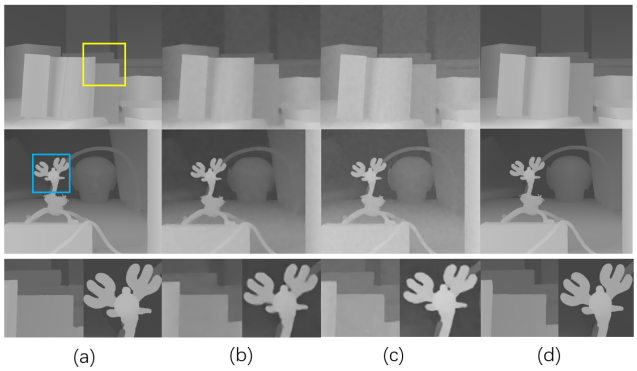

Figure 6: Visual comparison of ×8 upsampling and denoising results on *Books, Reindeer*: (a) Noisy; (b) DGDIE [12], (c) DSR_N [39], (d) Ours.

guidance. Table 1 presents the ×2, ×4, ×8 and ×16 upsampling performance of different methods. Our network almost obtains the best objective scores in all cases, especially for the ×8 and ×16 cases which are more difficult to restore. As shown in Fig. 4, obviously, our method can recover more pleasing structures and depth details, e.g., less jaggy artifacts around the stick in *Art*, more sharper and cleaner results on the toy's head in *Dolls*. Note that our unified model performs better than these color-guided and scale-specific models that are trained independently on each scale.

Additionally, we evaluate on NYU dataset to demonstrate the effectiveness of our method. These state-of-the-art methods (JBU [22], DJF [25], DGDIE [12], GbFT [1], PAC [37], SVLRM [28], DKN [20]) that are also evaluated on NYU dataset are compared. As illustrated in Table 2, our method obtains the best objective results for all the upsampling cases. Fig. 5 further shows the visual performance under the ×8 case. Focusing on the highlighted regions, we achieve the sharpest and clearest results.

Depth SR under Noisy Cases: Following [29], to simulate the acquisition process of a ToF depth sensor, we also add depth-dependent Gaussian noise to the training data, and then downsample the polluted depth maps at ×8 and ×16 scales. As shown in Table 3, our method achieves the best objective performance. We further provide perceptual comparisons in Fig. 6. The results of other methods present excessive cotton-like blurring, while our method can remove the noise and keep the sharpest depth boundaries on each depth range thanks to our distance-aware slicing network.

4.2 Evaluation on Generalization

We validate the generalization ability of our method on MPI Sintel dataset, the degraded depth inputs with unseen scale factors, and real ToFMark dataset.

MPI Sintel Dataset As demonstrated in Table 4, we almost achieve the best results on each case. Fig. 7 shows the visual comparison on *Ambush_4-12*. We obtain more accurate and clear depth details in the recovered results.

Table 4: Generalization on MPI Sintel datasets.

	Ambush_2-15		*Ambush_4-12*		*Ambush_5-41*		*Twmple_3-23*	
	×8	×16	×8	×16	×8	×16	×8	×16
MSG [17]	0.51	1.12	1.10	1.82	1.36	2.01	0.82	1.78
DGDIE [12]	0.65	1.24	1.26	2.23	1.79	3.10	1.01	1.90
DEIN [41]	0.47	1.08	1.12	1.76	1.69	2.32	0.89	1.82
GSRPT [3]	0.62	1.44	1.32	2.45	1.98	3.46	1.19	2.07
DSR_N [39]	**0.30**	0.85	0.88	2.21	0.98	2.07	**0.59**	1.35
Ours	0.42	**0.64**	**0.82**	**1.62**	**0.77**	**1.88**	0.81	**1.30**

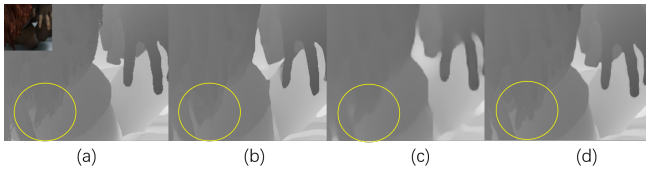

Figure 7: Visualization on *Ambush_4-12* from MPI (×8): (a) GT, (b) GSRPT, (c) DSR_N, (d) Ours.

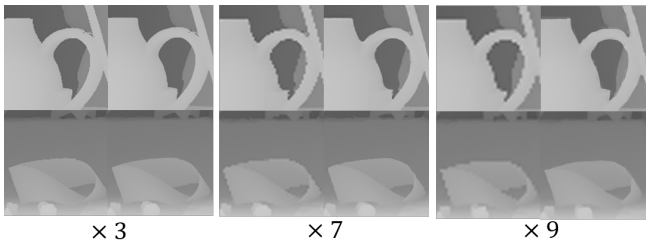

Figure 8: Generalization on unseen upsampling scales in training data.

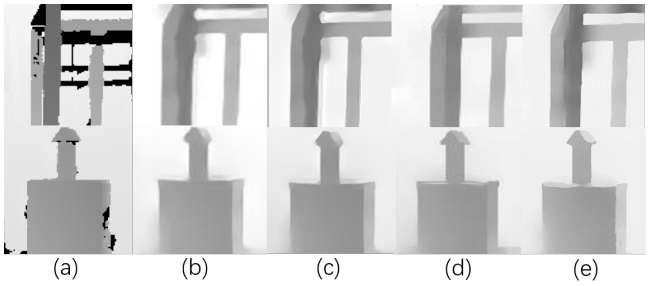

Figure 10: Visualization on ToFMark: (a) GT; (b) DGDIE[12], (c)GSRPT[3], (d) DSR_N[39];(e) Ours.

Inputs with Unseen Scales We also provide the visualization results which are tested on the inputs with unseen scale factors. Fig. 8 lists paired input and output of different scales. Our model also achieves high perceptual quality which validates our high generalization ability due to the proposed scale-controllable module.

ToFMark Dataset Different from other methods that need to synthesize the suitable training datasets for testing on ToFMark or revise the inputs from ToFMark to adapt to

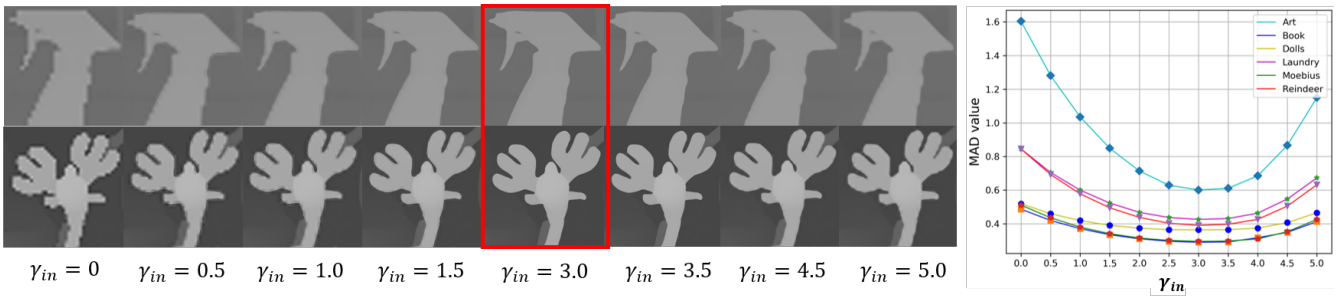

Figure 9: Visual comparison of $\times 8$ ($\gamma_{in} = 3$) upsampling results at different γ_{in} vaules.

Table 5: Generalization on real ToFMark dataset.

Method	MSG [17]	DEIN[41]	DGDIE [12]	GSRPT [3]	DSR_N [39]	Ours
Books	12.26	12.78	12.31	13.21	11.15	**11.03**
Shark	14.11	15.11	14.06	15.03	13.26	**12.08**
Devil	12.45	14.25	9.66	12.27	9.54	**9.33**

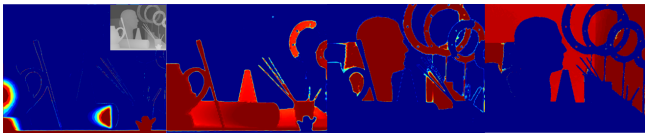

Figure 11: Visualization of the weighting masks.

Table 6: Ablation Study on $\times 8$ downsampling cases.

Method	MAD Values (the lower the better)					
	Art	*Books*	*Dolls*	*Laundry*	*Moebius*	*Reindeer*
w/o γ_{in}	0.591	0.280	0.317	0.365	0.278	0.340
Ours	**0.578**	**0.263**	**0.310**	**0.342**	**0.272**	**0.334**
N=0	0.607	0.279	0.322	0.370	0.288	0.355
N=3	0.588	0.270	0.307	0.356	0.275	0.350
N=5 (Ours)	0.578	0.263	0.310	**0.342**	0.272	0.334
N=5 (w/o Weight)	0.619	0.286	0.327	0.403	0.299	0.363
N=7	**0.563**	**0.247**	**0.302**	0.344	**0.239**	**0.322**

the well-trained models, we directly send the depth maps into our model to acquire the results. As shown in Table 5 and Fig. 10, through controlling the scale parameter, we achieve the best performance, which demonstrates our better generalization ability on real data.

4.3 Ablation Study

Scale-Controllable Module (SCM). Previous experiments have already validated our SCM on the recovery of designated scales and the generalization of unseen datasets or scales. As shown in Table 6, we further demonstrates the effectiveness of SB according to different training stages ('w/o γ_{in}' for the first training stage, while 'Ours' for the complete training) through ablation study. Without the control of γ_{in}, the performance decreases a lot. Besides, we also provide perceptual comparisons in Fig. 9 to illustrate the fine-grained control of SCM. For the same $\times 8$ LR input, we change γ_{in} from 0 to 5 with an interval of 0.5 to obtain different visual results. Our network presents a mild performance variation, and achieves the best performance at $\gamma_{in} = 3$.

Depth Slicing Module (DSM). We investigate our DSM via ablation study on two aspects, i.e., the distance-aware weighting scheme and the number of slicing branches. As listed in Table 6, removing the whole DSM module ('N = 0') or the weighting scheme ('N = 5 (w/o Weight)') will decrease the performance a lot compared to our final configuration ('N =5 (Ours)'). Meanwhile, as the number of slicing branches N increases, the performance improves obviously, but saturates at $N = 5$ (our final choice). In addition, we visualize the probabilistic weighting masks to further validate the capability of our distance-aware weighting scheme as shown in Fig. 11. Given a test input, the proposed scheme can automatically distinguish different depth regions depending on the depth range, and generate the corresponding weighting mask on each specific depth range.

5 CONCLUSION

We propose a novel distance-aware deep controllable slicing network, which learns a set of slicing branches parameterized by a distance-aware weighting scheme to adaptively aggregate all the branches in an ensemble. In addition, a scale-controllable module is designed to realize the fine-grained control of depth restoration objectives. Comprehensive experiments demonstrate the superiority of our model.

REFERENCES

[1] Badour Albahar and Jia-Bin Huang. 2019. Guided Image-to-Image Translation With Bi-Directional Feature Transformation. In *IEEE ICCV*. IEEE, 9015–9024.

[2] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. 2011. A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision* 92, 1 (2011), 1–31.

[3] Riccardo de Lutio, Stefano D'Aronco, Jan Dirk Wegner, and Konrad Schindler. 2019. Guided Super-Resolution As Pixel-to-Pixel Transformation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 8828–8836.

[4] Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems, First International Workshop, MCS, Cagliari, Italy, Proceedings (Lecture Notes in Computer Science)*, Josef Kittler and Fabio Roli (Eds.), Vol. 1857. Springer, 1–15.

[5] T. Edeler, K. Ohliger, S. Hussmann, and A. Mertins. 2010. Time-of-flight depth image denoising using prior noise information.

In *IEEE 10th International Conference on Signal Processing Proceedings*. 119–122.

[6] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. 2013. Image Guided Depth Upsampling using Anisotropic Total Generalized Variation. In *Proc. ICCV*.

[7] David Ferstl, Christian Reinbacher, René Ranftl, Matthias Rüther, and Horst Bischof. 2013. Image Guided Depth Upsampling Using Anisotropic Total Generalized Variation. In *IEEE International Conference on Computer Vision, ICCV, Sydney, Australia*. IEEE Computer Society, 993–1000.

[8] Zhihang Fu, Zhongming Jin, Guo-Jun Qi, Chen Shen, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. 2018. Previewer for Multi-Scale Object Detector. In *2018 ACM Multimedia Conference on Multimedia Conference, MM, Seoul, Republic of Korea, October 22-26*. 265–273.

[9] Shangqi Gao and Xiahai Zhuang. 2019. Multi-Scale Deep Neural Networks for Real Image Super-Resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Long Beach, CA, USA*. Computer Vision Foundation / IEEE, 0.

[10] Xu Gao and Tingting Jiang. [n.d.]. OSMO: Online Specific Models for Occlusion in Multiple Object Tracking under Surveillance Scene. In *ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea*. ACM, 201–210.

[11] Zan Gao, Li-Shuai Gao, Hua Zhang, Zhiyong Cheng, and Richang Hong. [n.d.]. Deep Spatial Pyramid Features Collaborative Reconstruction for Partial Person ReID. In *ACM International Conference on Multimedia, MM 2019, Nice, France*. ACM, 1879–1887.

[12] Shuhang Gu, Wangmeng Zuo, Shi Guo, Yunjin Chen, Chongyu Chen, and Lei Zhang. 2017. Learning Dynamic Guidance for Depth Image Enhancement. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Honolulu, HI, USA*. 712–721.

[13] Chunle Guo, Chongyi Li, Jichang Guo, Runmin Cong, Huazhu Fu, and Ping Han. 2019. Hierarchical Features Driven Residual Learning for Depth Map Super-Resolution. *IEEE Trans. Image Processing* 28, 5 (2019), 2545–2557.

[14] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. 2018. Deep Back-Projection Networks for Super-Resolution. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Salt Lake City, UT, USA*. 1664–1673.

[15] Heiko Hirschmüller and Daniel Scharstein. 2007. Evaluation of Cost Functions for Stereo Matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR, Minneapolis, Minnesota, USA*.

[16] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. 2019. Meta-SR: A Magnification-Arbitrary Network for Super-Resolution. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Long Beach, CA, USA*. Computer Vision Foundation / IEEE, 1575–1584.

[17] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. 2016. Depth Map Super-Resolution by Deep Multi-Scale Guidance. In *Computer Vision - ECCV - 14th European Conference, Amsterdam, The Netherlands, Proceedings, Part III*. 353–369.

[18] Sunghoon Im, Hyowon Ha, Gyeongmin Choe, Hae-Gon Jeon, Kyungdon Joo, and In So Kweon. 2019. Accurate 3D Reconstruction from Small Motion Clip for Rolling Shutter Cameras. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 4 (2019), 775–787.

[19] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. 2018. Deep Video Super-Resolution Network Using Dynamic Upsampling Filters Without Explicit Motion Compensation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Salt Lake City, UT, USA*. 3224–3232.

[20] Beomjun Kim, Jean Ponce, and Bumsub Ham. 2019. Deformable kernel networks for guided depth map upsampling. *CoRR* abs/1903.11286 (2019). arXiv:1903.11286 http://arxiv.org/abs/1903.11286

[21] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. 2016. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Las Vegas, NV, USA*. 1646–1654.

[22] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matthew Uyttendaele. 2007. Joint bilateral upsampling. *ACM Trans. Graph.* 26, 3 (2007), 96.

[23] Chao Li, Dongliang He, Xiao Liu, Yukang Ding, and Shilei Wen. 2019. Adapting Image Super-Resolution State-Of-The-Arts and Learning Multi-Model Ensemble for Video Super-Resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Long Beach, CA, USA*.

[24] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. 2016. Deep Joint Image Filtering. In *Computer Vision - ECCV - 14th European Conference, Amsterdam, The Netherlands, Proceedings, Part IV*. 154–169.

[25] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. 2019. Joint Image Filtering with Deep Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 8 (2019), 1909–1923.

[26] Yu Li, Dongbo Min, Minh N. Do, and Jiangbo Lu. 2016. Fast Guided Global Interpolation for Depth and Motion. In *Computer Vision - ECCV - 14th European Conference, Amsterdam, The Netherlands, Proceedings, Part III*. 717–733.

[27] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. 2017. Enhanced Deep Residual Networks for Single Image Super-Resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Honolulu, HI, USA*. 1132–1140.

[28] Jinshan Pan, Jiangxin Dong, Jimmy S. J. Ren, Liang Lin, Jinhui Tang, and Ming-Hsuan Yang. 2019. Spatially Variant Linear Representation Models for Joint Filtering. In *IEEE CVPR*. 1702–1711.

[29] Jaesik Park, Hyeongwoo Kim, Yu-Wing Tai, Michael S. Brown, and In-So Kweon. 2011. High quality depth map upsampling for 3D-TOF cameras. In *IEEE International Conference on Computer Vision, ICCV, Barcelona, Spain*. 1623–1630.

[30] Guo-Jun Qi. 2016. Hierarchically Gated Deep Networks for Semantic Segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Las Vegas, NV, USA, June 27-30*. 2267–2275.

[31] Jinghui Qin, Ziwei Xie, Yukai Shi, and Wushao Wen. 2019. Difficulty-Aware Image Super Resolution via Deep Adaptive Dual-Network. In *IEEE International Conference on Multimedia and Expo, ICME, Shanghai, China*. IEEE, 586–591.

[32] Gernot Riegler, Matthias Rüther, and Horst Bischof. 2016. ATGV-Net: Accurate Depth Super-Resolution. In *Computer Vision - ECCV - 14th European Conference, Amsterdam, The Netherlands, Proceedings, Part III*. 268–284.

[33] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nesic, Xi Wang, and Porter Westling. 2014. High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth. In *Pattern Recognition - 36th German Conference, GCPR, Münster, Germany, Proceedings*. 31–42.

[34] Daniel Scharstein and Chris Pal. 2007. Learning Conditional Random Fields for Stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR, Minneapolis, Minnesota, USA*.

[35] Daniel Scharstein and Richard Szeliski. 2003. High-Accuracy Stereo Depth Maps Using Structured Light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR, Madison, WI, USA*. 195–202.

[36] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*.

[37] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik G. Learned-Miller, and Jan Kautz. 2019. Pixel-Adaptive Convolutional Neural Networks. In *IEEE CVPR*. IEEE, 11166–11175.

[38] Xintao Wang, Ke Yu, Chao Dong, Xiaoou Tang, and Chen Change Loy. 2019. Deep Network Interpolation for Continuous Imagery Effect Transition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*.

[39] Zhihui Wang, Xinchen Ye, Baoli Sun, Jingyu Yang, Rui Xu, and Haojie Li. 2020. Depth upsampling based on deep edge-aware learning. *Pattern Recognition* 103 (2020), 107274.

[40] Yang Wen, Bin Sheng, Ping Li, Weiyao Lin, and David Dagan Feng. 2019. Deep Color Guided Coarse-to-Fine Convolutional Network Cascade for Depth Image Super-Resolution. *IEEE Trans. Image Processing* 28, 2 (2019), 994–1006.

[41] Xinchen Ye, Xiangyue Duan, and Haojie Li. 2018. Depth Super-Resolution with Deep Edge-Inference Network and Edge-Guided Depth Filling. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, Calgary, AB, Canada*. 1398–1402.

[42] Le Zhang, Zenglin Shi, Ming-Ming Cheng, Yun Liu, and Zeng Zeng. 2019. Nonlinear Regression via Deep Negative Correlation Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP, 99 (2019), 1–1.

[43] Kecheng Zheng, Zheng-Jun Zha, Yang Cao, Xuejin Chen, and Feng Wu. [n.d.]. LA-Net: Layout-Aware Dense Network for Monocular Depth Estimation. In *ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea*. ACM, 1381–1388.