Learned Dynamic Guidance for Depth Image Reconstruction

Shuhang Gu, Shi Guo, Wangmeng Zuo, Yunjin Chen, Radu Timofte, Luc Van Gool, and Lei Zhang, *Fellow, IEEE*

Abstract—The depth images acquired by consumer depth sensors (e.g., Kinect and ToF) usually are of low resolution and insufficient quality. One natural solution is to incorporate a high resolution RGB camera and exploit the statistical correlation of its data and depth. In recent years, both optimization-based and learning-based approaches have been proposed to deal with the guided depth reconstruction problems. In this paper, we introduce a weighted analysis sparse representation (WASR) model for guided depth image enhancement, which can be considered a generalized formulation of a wide range of previous optimization-based models. We unfold the optimization by the WASR model and conduct guided depth reconstruction with dynamically changed stage-wise operations. Such a guidance strategy enables us to dynamically adjust the stage-wise operations that update the depth image, thus improving the reconstruction quality and speed. To learn the stage-wise operations in a task-driven manner, we propose two parameterizations and their corresponding methods: dynamic guidance with Gaussian RBF nonlinearity parameterization (DG-RBF) and dynamic guidance with CNN nonlinearity parameterization (DG-CNN). The network structures of the proposed DG-RBF and DG-CNN methods are designed with the the objective function of our WASR model in mind and the optimal network parameters are learned from paired training data. Such optimization-inspired network architectures enable our models to leverage the previous expertise as well as take benefit from training data. The effectiveness is validated for guided depth image super-resolution and for realistic depth image reconstruction tasks using standard benchmarks. Our DG-RBF and DG-CNN methods achieve the best quantitative results (RMSE) and better visual quality than the state-of-the-art approaches at the time of writing. The code is available at https://github.com/ShuhangGu/GuidedDepthSR

1 INTRODUCTION

HIGH quality, dense depth images play an important role in many real world applications such as human pose estimation [1], hand pose estimation [2], [3] and scene understanding [4]. Traditional depth sensing is mainly based on stereo or lidar, coming with a high computational burden and/or price. The recent proliferation of consumer depth sensing products, e.g., RGB-D cameras and Time of Flight (ToF) range sensors, offers a cheaper alternative to dense depth measurements. However, the depth images generated by such consumer depth sensors are of lower quality and resolution. It therefore is of great interest whether depth image enhancement can make up for those flaws [5], [6], [7], [8], [9], [10], [11]. To improve the quality of depth images, one category of methods [5], [6] utilize multiple images from the same scene to provide complementary information. These methods, however, heavily rely on accurate calibration and are not applicable in dynamic environments. Another category of approaches [7], [8], [9], [11], [12] introduce structure information from a guidance image (for example, an RGB image) to improve the quality of the depth image. As in most cases the high quality RGB image can be acquired simultaneously with the depth image, such guided depth reconstruction has a wide range of applications [13].

A key issue of guided depth enhancement is to appropriately exploit the structural scene information in the guidance image. By incorporating the guidance image in the weight calculating step, joint filtering methods [12], [14], [15], [16] directly transfer structural information from the intensity image to the depth image [17], [18]. Yet, due to the complex relationship between the local structures of intensity and depth, such simple joint filtering methods are highly sensitive to the parameters, and often copy unrelated textures from the guidance image into the depth estimation. To better model the relationship between the intensity image and the depth image, optimization-based methods [7], [8], [9] adopt objective functions to characterize their dependency. Although the limited number of parameters in these heuristic models has restricted their capacity, these elaborately designed models still capture certain aspects of the joint prior, and have delivered highly competitive enhancement results. Recently, discriminative learning solutions [10], [19], [20], [21] have also been proposed to capture the complex relationships between

- S. Gu and R. Timofte are with the Computer Vision Lab, ETH Zurich 8092 Zürich, Switzerland. E-mail: shuhanggu@gmail.com, timofter@vision.ee.ethz.ch.
- S. Guo and W. Zuo are with the Harbin Institute of Technology, Harbin 150001, China. E-mail: guoshi28@outlook.com, cswmzuo@gmail.com.
- Y. Chen is with the ULSee Inc., San Francisco, CA 94105. E-mail: cheny@icg.tugraz.at.
- L. Van Gool is with the Computer Vision Lab, ETH Zurich 8092 Zürich, Switzerland, and also with the KU Leuven 3000, Leuven, Belgium. E-mail: vangool@vision.ee.ethz.ch.
- L. Zhang is with the Hong Kong Polytechnic University, Kowloon, Hong Kong. E-mail: cslzhang@comp.polyu.edu.hk.

Manuscript received 11 Oct. 2018; revised 19 Sept. 2019; accepted 8 Dec. 2019. Date of publication 0 . 0000; date of current version 0 . 0000. (Corresponding author: Shuhang Gu.)
Recommended for acceptance by M. Bennamoun, Y. Guo, F. Tombari, K. Youcef-Toumi, and K. Nishino.
Digital Object Identifier no. 10.1109/TPAMI.2019.2961672

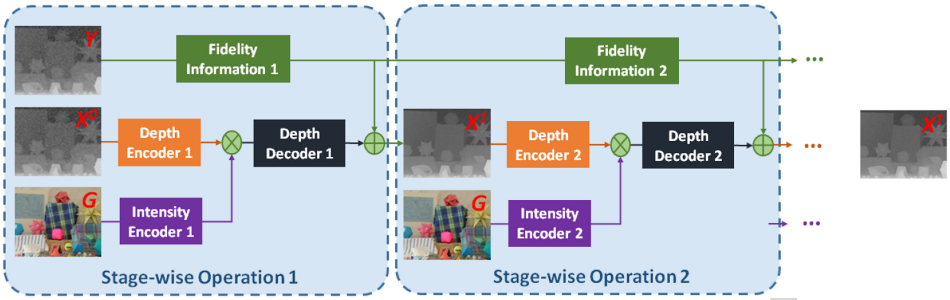


Fig. 1. Illustration of the unfolded optimization process of a WASR model. The WASR model takes low quality depth estimation Y and guidance intensity image G as input, aims to achieve a high quality depth image X. Each step of the optimization process can be termed as a stage-wise operation. By dynamically changing the stage-wise operation, we construct the DG-RBF and DG-CNN model for fast and accurate guided depth reconstruction.

intensity and depth. Due to the unparalleled non-linear modeling capacity of deep neural networks as well as the paired training data, deep learning based methods [20], [21] have achieved better enhancement performance than conventional optimization-based approaches.

To deal with the guided depth reconstruction task, recent solutions [19], [20], [21] utilize deep neural networks (DNN) to build the mapping function from the low quality inputs and the guidance images to the high quality reconstruction results. As for other dense estimation tasks [22], [23], [24], an appropriate network structure plays a crucial role in the success of the DNN-based guided depth reconstruction system. Recently, a large number of works [24], [25], [26], [27] have shown that some successful optimization-based models could provide useful guidelines for designing network architectures. By unrolling the optimization process of variational or graphical models, network structures have been designed to solve image denoising [25], [26], compressive sensing [28] and semantic segmentation [24]. These networks employ domain knowledge as well as paired training data and have achieved state-of-the-art performance for different tasks. In this paper, we analyze and generalize previous optimization-based approaches, and propose better network structures to deal with the guided depth reconstruction task.

Work related to this paper is that of Riegler *et al.* [29], which unrolls the optimization steps of a non-local variational model [30] and proposes a primal-dual network (PDN) to deal with the guided depth super-resolution task. Yet, PDN follows the unrolled formula of the non-local regularization model [30] strictly, and only adopts the predefined operator (Huber norm) to penalize point-wise differences between depth pixels. As a result, the PDN method [29] has limited flexibility to take full advantage of paired training data. In this paper, we propose a more flexible solution to exploit paired training data as well as prior knowledge from previous optimization-based models. We analyze previous dependency modeling methods and generalize them as a weighted analysis sparse representation regularization (WASR) term. By unfolding the optimization process of the WASR model, we get the formula of a stage-wise operation for guided depth enhancement, and use it as departure point for our network structure design. In Fig. 1, we provide a flowchart of the general formula of the unfolded optimization process of the WASR model. Each iteration of the optimization algorithm can be regarded as a stage-wise operation to enhance the depth map.

WASR is a generalized model which shares many of the characteristics common to previous optimization-based approaches [7], [31]. Unfolding its optimization process provides us with a framework to leverage the previous expertise while leaving our model enough freedom to take full advantage of training data. With the general formula of the stage-wise operation established, we adopt two approaches to parameterize the operations. The first approach parameterizes the unfolded WASR model in a direct way. Based on the unfolded optimization process, the stage-wise operations consist of simple convolutions and nonlinear functions. We learn the filters and nonlinear functions (parameterized as the summation of Gaussian RBF kernels [25], [26]) for each stage-wise operation, in a task-driven manner. Although such model shares its formula for the optimization with a simple WASR model, its operations are changed dynamically to account for the depth enhancement. As a result, it can generate better enhancements in just a few stages. In the remainder of this paper, we denote this model as dynamic guidance with RBF nonlinearity parameterization (DG-RBF). An illustration of one stage of the DG-RBF operation can be found in Fig. 2.

Besides the DG-RBF model, we also propose to parameterize the stage-wise operation in a loose way. In particular, we analyze the stage-wise operation's formula and divide the operation into three sub-components: the depth encoder, the intensity encoder and the depth decoder. Instead of using one large filter and one nonlinear function to form the encoder and the decoder in the stage-wise operation, we use several layers of convolutional neural networks (CNN) to improve

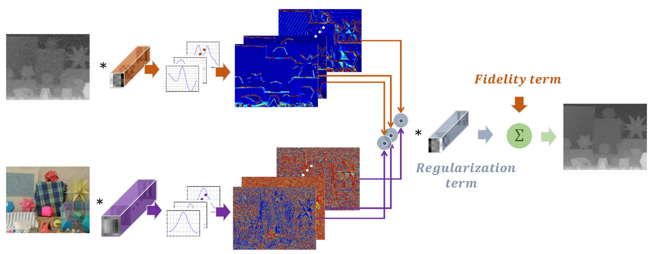


Fig. 2. Illustration of one stage-wise operation in the DG-RBF model. DG-RBF follows the unfolded optimization process of WASR strictly, the current enhancement result x_t and the guidance image g are first convolved with the corresponding L analysis filters, respectively. After a nonlinear transform, the filtering responses of x_t and g are combined via an element-wise product, and further convolved with the L adjoint filters to form the result with a regularization term. Finally, the results of regularization and the fidelity terms are summarized to obtain the updated result x_{t+1}.

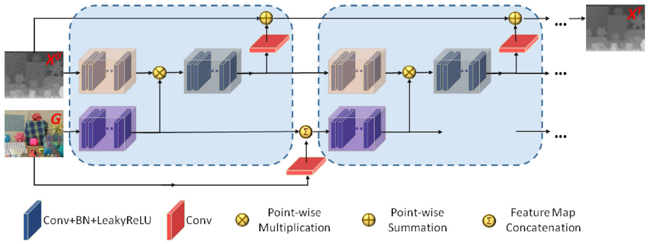


Fig. 3. Illustration of DG-CNN structure (with two stage-wise operations) for guided depth reconstruction. The light orange, purple and gray components in the figure correspond to the depth encoder, the intensity encoder and the joint decoder, respectively.

the capacity of each sub-component. The overall model of this dynamic guidance with CNN nonlinearity parameterization (DG-CNN) is designed based on the unfolded optimization process of the WASR model, while its sub-components are parameterized with powerful CNNs. As DG-CNN builds upon the conventional optimization-based approach and the recent advances in deep learning, it generates better enhancement results than the existing methods. An illustration of a two stage DG-CNN model can be found in Fig. 3, details of the networks will be introduced in Section 5.

The formula of the WASR model and some experimental results of the DG-RBF method have been introduced in our earlier conference paper [32]. In this paper, we provide more information about the WASR model and DG-RBF method, and provide the DG-CNN approach, a new parameterization of the WASR model. Due to its unparalleled nonlinearity modeling capacity, CNN based parameterization often generates better enhancement results than the Gaussian RBF based method, especially in challenging cases with large zooming factors. Furthermore, the well optimized deep learning tool box makes the CNN based method (DG-CNN) more efficient than DG-RBF in both training and testing.

The contributions of this paper are summarized as follows:

- By analyzing previous guided depth enhancement methods, we formulate the dependency modeling of depth and RGB images as a weighted analysis sparse representation (WASR) model. We unfold the optimization process of the WASR objective function, and propose a task-driven training strategy to learn stage-wise dynamic guidance for different tasks. A Gaussian RBF kernel nonlinearity modeling method (DG-RBF) and a special CNN (DG-CNN) are trained to conduct depth enhancement at each stage.
- We conduct detailed ablation experiments to analyze the model hyper-parameters and network architecture. The experimental results clearly demonstrate the effectiveness of the optimization-inspired network architecture design.
- Experimental results on depth image super-resolution and noisy depth image reconstruction validate the effectiveness of the proposed dynamic guidance approach. The proposed algorithm achieves the best quantitative and qualitative depth enhancement results among the state-of-the-art methods that we compared to.

The rest of this paper is organized as follows. Section 2 briefly introduces some related work. Section 3 analyzes previous objective functions of guided depth enhancement approaches, and introduces the task-driven formulation of the guided depth enhancement task. By unrolling the optimization process of the task-driven formulation, Sections 4 and 5 introduce two parameterization approaches, i.e., parameterize the nonlinear operation in each step with Gaussian RBF kernels or parameterize each gradient-descent stage with convolutional neural networks. Section 6 conducts ablation experiments to analyze the model hyper-parameters and to show the advantage of the optimization-inspired network architecture design. Sections 7 and 8 provide experimental results of the different methods for guided depth super-resolution and enhancement. Section 9 discusses the DG-RBF and DG-CNN models. Section 10 concludes the paper.

2 Related Work

In this section, we introduce related work. We start by briefly surveying the analysis representation model literature to then review prior guided depth enhancement methods. Finally, we discuss previous work on optimization-inspired network architecture design.

2.1 Analysis Sparse Representation

Sparse analysis representations have been widely applied in image processing and computer vision tasks [25], [26], [33], [34], [35], [36]. An analysis operator [37] operates on image patches or analysis filters [35], [38] operate on whole images to model the local structure of natural images. Compared with sparse synthesis representations, the analysis model adopts an alternative viewpoint for union-of-subspaces reconstruction by characterizing the complement subspace of signals [39], and usually results in more efficient solutions.

Here we only consider the convolutional analysis representation, with one of its representative forms given by:

$$\hat{X} = \arg\min_{X} \mathcal{L}(X,Y) + \sum\nolimits_l \sum\nolimits_i \rho_l((\boldsymbol{k}_l \otimes X)_i), \tag{1}$$

where X is the latent high quality image and Y is its degraded observation. $\otimes$ denotes the convolution operator, and $(\cdot)_i$ denotes the value at position i. The penalty function $\rho_l(\cdot)$ is introduced to characterize the analysis coefficients of latent estimation, which are generated by the analysis dictionaries $\{\boldsymbol{k}_l\}_{l=1,\ldots,L}$ in a convolutional manner. $\mathcal{L}(X,Y)$ is the data fidelity term determined by the relationship between X and its degraded observation Y. For example, for the task of Gaussian denoising, $\mathcal{L}(X,Y) = \frac{1}{2\sigma^2}\|X-Y\|_F^2$ shows that the difference between X and Y is zero mean white Gaussian noise with standard deviation value σ. In the remainder of this paper, we denote $\rho_l((\boldsymbol{k}_l \otimes X)_i)$ by $\rho_{l,i}(\boldsymbol{k}_l \otimes X)$ for the purpose of simplicity. For Gaussian denoising, one can simply let $\mathcal{L}(X,Y) = \frac{1}{2\sigma^2}\|X-Y\|_F^2$.

Sparse analysis representation has been studied for several decades. Rudin *et al.* proposed a total variation (TV) model [33], where the analysis filters are gradient operators and the penalty function is the ℓ_1-norm. Subsequently, many attempts were made to provide better analysis filters and penalty functions, and an emerging topic is to learn sparse models from training data. Zhu *et al.* [40] proposed a

FRAME model which aims to learn penalty functions for predefined filters. Roth *et al.* [35] proposed a field-of-expert (FoE) model in which analysis filters are learned for predefined penalty functions. Although FRAME and FoE are originally introduced from a MRF perspective, they can also be interpreted as analysis representation models [37]. Recently, Schmidt *et al.* [25] and Chen *et al.* [26] suggested to model the related functions with linear combinations of Gaussian RBF kernels, and can learn both analysis filters and penalty functions from training data. Moreover, by incorporating the specific optimization methods, stage-wise parameters can be learned in a task driven manner.

Despite their achievements in image restoration, most existing methods are used for learning analysis representation of images from a single modality and cannot be applied to guided depth image reconstruction. Kiechle *et al.* went a step forward by introducing a bimodal analysis model to learn a pair of analysis operators [19]. But the issue of explicit and dynamic guidance from intensity images remains unaddressed in analysis representation learning. In this work, we extend the analysis model by introducing a guided weight function for modeling the guidance from intensity image and by adopting a task-driven learning method to learn stage-wise parameters for dynamic guidance.

2.2 Guided Depth Enhancement

The wide availability of consumer depth sensing equipment has made depth enhancement an important application. To estimate high quality depth images, guided depth enhancement can incorporate an intensity image of the same scene, as supplementary information, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2019.2961672. Based on the co-discontinuous assumption between the guidance and target images, general joint filtering methods, such as bilateral filters [16] and guided filters [17], can be directly applied to transfer structural information from intensity to depth images. Yet, due to the complex dependency between depth and intensity, such simple joint filtering methods may transfer irrelevant texture into the depth estimation.

To better model the dependency, the optimization based methods combine the input image $\boldsymbol{Y}$, the output image $\boldsymbol{X}$ and the guidance image $\boldsymbol{G}$ into an optimization model [7], [8], [9], [31], [41]. In [7], Diebel and Thrun proposed an MRF-based method to characterize the pixel-wise co-difference between the depth and intensity images. Their prior potential function is defined as

$$\sum_i \sum_{j\in\mathcal{N}(i)} \phi_\mu(\boldsymbol{G}_i - \boldsymbol{G}_j)(\boldsymbol{X}_i - \boldsymbol{X}_j)^2, \tag{2}$$

where i and j are the pixel indexes of image, $\mathcal{N}(i)$ is the set of neighboring index of i, and $\phi_\mu(z) = \exp(-\mu z^2)$. Similar weight functions have also been adopted in other models, e.g., non-local mean (NLM) [8], for guided depth enhancement. Besides pixel-wise differences, other cues such as color, segmentation and edges, are also considered to design proper weight functions. Instead of modifying the weight function, Ham *et al.* [31] adopt Welsch's function to regularize the depth differences

$$\sum_i \sum_{j\in\mathcal{N}(i)} \phi_\mu(\boldsymbol{G}_i - \boldsymbol{G}_j)(1 - \phi_\nu(\boldsymbol{X}_i - \boldsymbol{X}_j))/\nu. \tag{3}$$

Moreover, several hand-crafted high order models have also been proposed, to model the weight function and the depth regularizer [9].

Recently, learning-based methods started to exploit training data to enhance the results. To model the statistical dependency between the local structures of corresponding intensity and depth images, analysis [19] and synthesis [10] dictionary learning methods have been suggested in a data-driven manner. Taking the low quality depth image and the guidance intensity image as inputs, [20], [22], [29] directly train a CNN to generate the high quality enhanced output result.

2.3 Optimization-Inspired Network Architecture Design

The idea of unfolding the optimization or inference steps of variational model as neural networks has been investigated from different perspectives. Some early work [27], [42] proposed to only conduct a limited number of steps in the optimization algorithm for the purpose of efficiency. Gregor *et al.* [42] shown that learning the filters and the mutual inhibition matrices of truncated versions of FISTA [43] and CoD [44] leads to a dramatic reduction in the number of iterations to reach a given code prediction error. Domke [27] proposed a truncated fitting approach which only runs a fixed number of iterations of an inference algorithm to combat computational complexity.

In addition to the efficiency issue, recent works found that unfolding the inference steps of optimization algorithm also helps to increase model flexibility and improve the estimation results for different applications. Schmidt *et al.* [25] unfolded the inference process of conditional random field and proposed a shrinkage field approach to solve the image denoising problem. Chen *et al.* [26] proposed to learn time varying linear filters and penalties from a reaction-diffusion model point of view. Recently, Kobler *et al.* [45] explored links between variational energy minimization methods and deep learning approaches, and proposed a variational network for different image reconstruction tasks. Compared with exact minimization, unfolded networks are able to perform different operations in each step [46]. Consequently, these methods [25], [26], [45], [46] achieved great improvements in both run-time and reconstruction performance over conventional models. Besides single image reconstruction, the idea of optimization-inspired network architecture design has also been exploited in other tasks. To incorporate the CRF model in a CNN-based semantic segmentation method, Zheng *et al.* [24] unrolled the mean-field approximate inference algorithm as a recurrent neural network. Their proposed CRF-RNN integrates a CRF model with CNNs, and achieved state-of-the-art performance on the semantic segmentation task. Compressive Sensing (CS) is an effective approach for fast Magnetic Resonance Imaging (MRI). To improve the MRI reconstruction accuracy and speed, Yang *et al.* [28] proposed an ADMM-Net, which is derived from the ADMM algorithm for optimizing a CS-based MRI model.

In the field of guided depth super-resolution (SR), Riegler *et al.* [29] introduced a two-stage primal-dual network (PDN) approach. PDN [29] utilizes a fully convolutional network to estimate a coarse high resolution depth image, and adopts an unrolled variational model to refine the coarse estimation. The PDN method combines the advantages of a CNN and variational methods to achieve top depth SR performance. Nonetheless, PDN still strictly follows the optimization steps of a concrete variational model, and has limited capacity in adapting to the training data. The latest DNN-based methods [20], [21] improved over the depth SR results of PDN. In this paper, we generalize conventional guided depth reconstruction models, and provide a more flexible solution to benefit from domain knowledge and training data.

3 Task-Driven WASR Model for Dependency Modeling

In this section, we first suggest a weighted analysis sparse representation (WASR) model to introduce guidance information from the intensity image. Then, a task-driven parameter training formulation of the proposed model is derived for training parameters in the objective function.

3.1 Weighted Analysis Regularization for Dependency Modeling

For the conventional analysis sparse representation from Eq. (1), the regularization term is only a function of the output image $\boldsymbol{X}$. Actually, the models in Eqs. (2) and (3) can be treated as special handcrafted analysis models, in which a group of inter-pixel difference operators are used as the analysis filters and the weight function on $\boldsymbol{G}$ is introduced for explicit guidance. Motivated by this observation, we propose a generalized weighted analysis model for guided depth reconstruction. Instead of regularizing the first order inter-pixel differences, the proposed weighted analysis model adopts high order filters to capture better the structural dependency between intensity and depth image

$$\sum\nolimits_i \sum\nolimits_l w_{l,i}(\boldsymbol{G})\rho_{l,i}(\boldsymbol{k}_l \otimes \boldsymbol{X}), \tag{4}$$

where the weight for the lth analysis operator at position i is denoted as $w_{l,i}(\boldsymbol{G})$. The weight function extracts information from the guidance image $\boldsymbol{G}$ to adaptively regularize the analysis coefficients.

Eq. (4) is a generalized version of Eqs. (2) and (3). Like the previous methods, WASR aims to capture the co-discontinuous property between depth and intensity images for better depth reconstruction. Specifically, by extracting the local information of the guidance image, the weight function in Eq. (4) adaptively regularizes the penalty on the analysis coefficient of the depth image, and consequently determines the locations of sharp edges in the depth image. Analyzing previously proposed guided depth enhancement methods [7], [8], [9] under our WASR framework, we note that different weighting and penalty functions have been suggested in a handcrafted manner. In the next subsection, we introduce the task-driven formulation of the proposed WASR model, which provides a method to learn better model parameters to fit the guided depth reconstruction task.

3.2 Task-Driven Learning of WASR Parameters

Having the weighted analysis regularization term, the depth enhancement can be achieved by solving

$$\min_{\boldsymbol{X}} \mathcal{L}(\boldsymbol{X},\boldsymbol{Y}) + \sum\nolimits_i \sum\nolimits_l w_{l,i}(\boldsymbol{G})\rho_{l,i}(\boldsymbol{k}_l \otimes \boldsymbol{X}), \tag{5}$$

where the data fidelity term $\mathcal{L}(\boldsymbol{X},\boldsymbol{Y})$ in Eq. (5) is specified by the depth reconstruction task to indicate the relationship between latent high quality estimation $\boldsymbol{X}$ and the observation $\boldsymbol{Y}$. The WASR regularization term provides prior information to reconstruct the depth image and plays a crucial role to the reconstruction quality.

Since the model parameters may vary for different tasks, we provide a task-driven formulation to learn task-specific parameters for Eq. (5) [47], [48].

We denote by $\mathcal{D} = \{\boldsymbol{Y}^s, \boldsymbol{X}^s_{gt}, \boldsymbol{G}^s\}_{s=1}^S$ a training set of S samples, and by $\boldsymbol{Y}^s$, $\boldsymbol{X}^s_{gt}$, and $\boldsymbol{G}^s$ the sth input depth image, ground truth depth image, and ground truth intensity image, respectively. Following [47], [48], the task-driven formulation can be written as a bi-level optimization problem

$$\begin{aligned}
&\{\rho_l^*, w_l^*, \boldsymbol{k}_l^*\}_{l=1}^L = \arg \min_{\{\rho_l, w_l, \boldsymbol{k}_l\}_{l=1}^L} \sum_{s=1}^S \|\boldsymbol{X}^s_{gt} - \boldsymbol{X}^s\|_2^2 \\
&s.t.\ \boldsymbol{X}^s = \arg\min_{\boldsymbol{X}} \mathcal{L}(\boldsymbol{X},\boldsymbol{Y}^s) + \sum_l \sum_i w_{l,i}(\boldsymbol{G}^s)\rho_{l,i}(\boldsymbol{k}_l \otimes \boldsymbol{X}).
\end{aligned} \tag{6}$$

Eq. (6) optimizes the parameters in the objective function (5), makes the solution $\boldsymbol{X}^s$ of (5) as close (in terms of ℓ_2 distance as chosen in (6)) as its corresponding ground truth image $\boldsymbol{X}^s_{gt}$.

3.3 Dynamic Guidance With Unfolded WASR Model

The lower-level problem in Eq. (6) defines an implicit function on $\{\rho_l, w_l, \boldsymbol{k}_l\}_{l=1\cdots L}$, making the training problem very difficult to optimize. The high non-convexity of the lower-level problem further adds difficulty to obtaining the exact solution. Moreover, along with the enhancement procedure, more details of $\boldsymbol{X}^s$ will be recovered. Thus, instead of employing the same model parameters in all the iterations, by dynamically adjusting the model to better fit the reconstruction task both the efficiency and the enhancement result may benefit. To address this issue, we unfold the optimization process of the lower-level problem and train stage-wise operations for guided depth enhancement. Such stage-wise formulation not only reduces the difficulty of training, but also enables us to introduce the guidance information dynamically to cooperate with the newly updated estimation $\boldsymbol{X}^{t+1}$.

To unfold the optimization process of (5), we assume that both the fidelity term $\mathcal{L}(X,Y)$ and the penalty function $\rho_{l,i}(\boldsymbol{k}_l \otimes \boldsymbol{X})$ are differentiable with respect to $\boldsymbol{X}$. Then, solving (5) with gradient descent, the updated result $\boldsymbol{X}^{t+1}$ can be obtained by

$$\begin{aligned}
&\boldsymbol{X}^{t+1} = \\
&\boldsymbol{X}^t - \tau^t\Big(\mathcal{L}'(\boldsymbol{X}^t,\boldsymbol{Y}) + \sum\nolimits_l \overline{\boldsymbol{k}}_l^t \otimes \big(\boldsymbol{W}_l^t(\boldsymbol{G}) \odot \boldsymbol{P}_l^{t\prime}(\boldsymbol{k}_l^t \otimes \boldsymbol{X}^t)\big)\Big),
\end{aligned} \tag{7}$$

where $\mathcal{L}'(\cdot)$ is the derivative of the fidelity term, and τ^t is the step-length in step t. $\boldsymbol{P}_l^{t\prime}(\boldsymbol{k}_l^t \otimes \boldsymbol{X}^t)$ has the same size as $\boldsymbol{k}_l^t \otimes \boldsymbol{X}^t$, and its value in position i is the derivative of the penalty function $\rho_{l,i}^{t\prime}(\boldsymbol{k}_l^t \otimes \boldsymbol{X}^t)$. $\boldsymbol{W}_l^t(\boldsymbol{G})$ is the corresponding weight function, and its value in position i is $w^t_{l,i}(\boldsymbol{G})$. $\overline{\boldsymbol{k}}^t_l$ is obtained by rotating $\boldsymbol{k}^t_l$ 180 degrees.

Eq. (7) enables us to write $\boldsymbol{X}^{t+1}$ as a function of the input variables $\{\boldsymbol{X}^t, \boldsymbol{G}, \boldsymbol{Y}\}$. With $\{\tau^t, \{\rho^t_l, w^t_l, \boldsymbol{k}^t_l\}^L_{l=1}\}$, the function determines one stage of operation which generates $\boldsymbol{X}^{t+1}$ from the current estimation $\boldsymbol{X}^t$. Instead of solving Eq. (6) which requires the operations in each step to be the same, we propose to adopt different operations in each step. Concretely, by allowing $\{\tau^t, \{\rho^t_l, w^t_l, \boldsymbol{k}^t_l\}^L_{l=1}\}$ to be different in each stage t, we adopt a series of stage-wise operations to conduct the guided depth reconstruction. Compared with keeping the model parameters unchanged and solving the optimization problem in Eq. (5), such dynamic guidance approach allows the proposed model to generate high quality depth estimations in several stages.

In order to get the optimal stage-wise operations, we propose to adopt a similar task-driven strategy as we introduced in Eq. (6). In the next two sections, we introduce two parameterization strategies for the stage-wise operation, which enable us to learn optimal operations in a task-driven manner.

4 Learned Dynamic Guidance With RBF Kernel Parameterization

In the previous section, we analyzed the WASR model and analyzed the formula of the stage-wise operation for the guided depth reconstruction. Based on Eq. (7), the $(t+1)$th estimation $\boldsymbol{X}^{t+1}$ is determined by the current estimation $\boldsymbol{X}^t$, guidance image $\boldsymbol{G}$, observation $\boldsymbol{Y}$ and the stage-wise operations. In order to learn stage-wise operations, we adopt a greedy training strategy to train the stage-wise operations sequentially. Concretely, we minimize the difference between $\boldsymbol{X}_{gt}$ and the new estimation $\boldsymbol{X}^{t+1}$ with respect to the operation parameters. In this section, we introduce one parameterization strategy of the stage-wise operation. We follow the formula of Eq. (7) and parameterize the stage-wise operation of the WASR model in a direct way. The derivation of the penalty function is parameterized with a group of RBF kernels, and we call the proposed model dynamic guidance with RBF nonlinearity parameterization (DG-RBF).

4.1 Learning Step Length τ

In Eq. (7), τ^t is the step length for the tth stage-wise operation. τ^t is a scalar and we can directly learn it without any parameterization. However, as τ affects both the two components $\mathcal{L}'(\boldsymbol{X}^t, \boldsymbol{Y})$ and $\sum_l \overline{\boldsymbol{k}}^t_l \otimes \left(\boldsymbol{W}^t_l(\boldsymbol{G}) \odot \boldsymbol{P}^{t\prime}_l(\boldsymbol{k}^t_l \otimes \boldsymbol{X}^t)\right)$, calculating its gradient with respect to the training loss is time consuming. Since we will parameterize the prior term in our DG-RBF model, the stage-variant step length for the prior term can be absorbed into the parameterization of $\sum_l \overline{\boldsymbol{k}}^t_l \otimes \left(\boldsymbol{W}^t_l(\boldsymbol{G}) \odot \boldsymbol{P}^{t\prime}_l(\boldsymbol{k}^t_l \otimes \boldsymbol{X}^t)\right)$. Thus, in the proposed DG-RBF model, we assume τ^t only affects the gradient of fidelity term, i.e., $\boldsymbol{X}^{t+1} = \boldsymbol{X}^t - \tau^t \mathcal{L}'(\boldsymbol{X}^t, \boldsymbol{Y}) - \sum_l \overline{\boldsymbol{k}}^t_l \otimes \left(\boldsymbol{W}^t_l(\boldsymbol{G}) \odot \boldsymbol{P}^{t\prime}_l (\boldsymbol{k}^t_l \otimes \boldsymbol{X}^t)\right)$.

4.2 Parameterizing the Filter k

$\boldsymbol{k}$ in Eq. (7) are the analysis filters used to extract structural information from the depth image. Previous works have found that meaningful analysis filters often are zero-mean, thus, we also parameterize the filters $\{\boldsymbol{k}_l\}^L_{l=1}$ to ensure them to be zero-mean filters. Specifically, we require that each $\boldsymbol{k}_l$ is the summation of a zero-mean Discrete Cosine Transform (DCT) basis

$$\boldsymbol{k}_l = \sum_{i=1}^{I} \alpha_{l,i} \boldsymbol{b}_i, \tag{8}$$

where $\{\boldsymbol{b}_i\}^I_{i=1}$ are the zero-mean DCT basis. The above parameterization helps us to constrain the filters $\{\boldsymbol{k}^t_l\}^L_{l=1}$ to be zero-mean.

4.3 Parameterizing the Penalty Functions ρ

A good penalty function plays a crucial role in the success of analysis sparse representation models. Different functions have been suggested for generating sparse analysis coefficients in conventional optimization models. In this paper, we parameterize $\{\rho_l(\cdot)\}^L_{l=1}$ to allow them to have more flexible shapes. Actually, from Eq. (7) one can see that what we should parameterize is not the penalty function $\rho^t_l(z)$ but the influence function $\rho^{t\prime}_l(z)$. Here we write the influence function $\rho^{t\prime}_l(z)$ as

$$\rho^{t\prime}_l(z) = \sum\nolimits_j^M \beta^t_{l,j} \exp\left(\frac{-(z-\mu_j)^2}{2\sigma_j^2}\right), \tag{9}$$

which is the summation of M Gaussian RBF kernels with centers μ_j and scalar factors σ_j. This formulation can provide a group of highly flexible functions for image restoration [25], [26].

The number M as well as the means $\{\mu_j\}^M_{j=1}$ and scaling factor σ are the hyper-parameters of our model. The means $\{\mu_j\}^M_{j=1}$ determine the location of the kernels and the scaling factors their band width. The two parameters cooperate to determine the flexibility and cover range of the parameterization.

4.4 Parameterizing the Weight Functions w

As we have analyzed in Section 3.1, the weight function extracts local structures from the intensity image to adaptively regularize the penalty of the depth analysis coefficients. In previous hand-crafted models, some simple weight functions have been suggested to capture the co-difference of the depth and intensity images. In this paper, we adopt a similar form which utilizes filters to extract local structures of the intensity image to adaptively regularize the depth discontinuities.

However, although the intensity and the depth images arise from the same scene and are strongly dependent, the values in the two images have different physical meaning. For example, a black box in front of a white wall or a gray box in front of a black wall may correspond to the same depth map but totally different edge gradients for the intensity images. Therefore, the weight function should be able to avoid the interference of such structure-unrelated intensity information, while extracting useful salient structures to help the depth map locate its discontinuities. To this end, the intensity map is locally normalized, to avoid the effect of different intensity magnitude. Specifically, given the vectorization of the guided intensity image $\boldsymbol{g}$, we introduce the operator $\boldsymbol{R}_i$ to extract the local patch at position i by $\boldsymbol{R}_i\boldsymbol{g}$.

The local normalization of $\boldsymbol{R}_i \boldsymbol{g}$ can then be attained by $\boldsymbol{e}_i = \frac{\boldsymbol{R}_i \boldsymbol{g}}{||\boldsymbol{R}_i \boldsymbol{g}||_2}$.

With $\boldsymbol{e}_i$, we define the weight function for the lth analysis operator $\boldsymbol{\beta}_l$ at position i as

$$w_{l,i}(\boldsymbol{G}) = \exp\Big(-(\boldsymbol{\gamma}_l^T \boldsymbol{e}_i)^2\Big). \tag{10}$$

The analysis operator $\boldsymbol{\gamma}_l$ can serve as a special local structure detector. If the local normalized patch $\boldsymbol{e}_i$ contains local structures such as edges, $w_{l,i}(\boldsymbol{G})$ will be very small to encourage that the depth patch exhibits the corresponding local structure.

4.5 Training of DG-RBF Parameters

After parameterization, the stage-wise operations can be determined by the parameters $\boldsymbol{\Theta}^t = \{\tau^t, \{\boldsymbol{\alpha}_l^t, \boldsymbol{\beta}_l^t, \boldsymbol{\gamma}_l^t\}_{l=1}^L\}$. Plugging $\boldsymbol{X}^{s,t+1}(\boldsymbol{X}^t, \boldsymbol{G}, \boldsymbol{Y}; \boldsymbol{\Theta}^t)$ into the task-driven formula of Eq. (6), we are able to learn optimal stage-wise operations by minimizing

$$\boldsymbol{\Theta}^t = \arg\min_{\boldsymbol{\Theta}} \frac{1}{2} \sum\nolimits_{s=1}^S ||\boldsymbol{X}_{gt}^s - \boldsymbol{X}^{s,t+1}(\boldsymbol{X}^t, \boldsymbol{G}^s, \boldsymbol{Y}^s; \boldsymbol{\Theta}^t)||_F^2. \tag{11}$$

The gradient of the loss function with respect to the parameters $\boldsymbol{\Theta}^t = \{\tau^t, \{\boldsymbol{\alpha}_l^t, \boldsymbol{\beta}_l^t, \boldsymbol{\gamma}_l^t\}_{l=1}^L\}$ can be achieved by the chain rule

$$\frac{\partial loss(\boldsymbol{X}_{gt}, \boldsymbol{X}^{t+1})}{\partial \boldsymbol{\Theta}^t} = \frac{\partial loss(\boldsymbol{X}_{gt}, \boldsymbol{X}^{t+1})}{\partial \boldsymbol{X}^{t+1}} \cdot \frac{\partial \boldsymbol{X}^{t+1}}{\partial \boldsymbol{\Theta}^t}. \tag{12}$$

The detailed derivations of $\frac{\partial \boldsymbol{X}^{t+1}}{\partial \boldsymbol{\Theta}^t}$ are introduced in the Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2019.2961672.

Having the gradients, we learn the parameters for each stage with the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm [49], [50]. We learn the stage-wise parameters in a greedy manner. Given initialization $\boldsymbol{X}^0$, we learn one stage operator to generate estimation $\boldsymbol{X}^1$ by minimizing the difference between $\boldsymbol{X}^1$ and target ground truth $\boldsymbol{X}$; then, taking $\boldsymbol{X}^1$ as input, we learn another operation for estimating $\boldsymbol{X}^2$ in the same manner. For both the noise-free and noisy depth SR experiments, we use the results of bicubic interpolation as the initialization of $\boldsymbol{X}^0$. The initialization of $\boldsymbol{X}^0$ for other tasks will be introduced in each experiment. We experimentally found that we can get very good results after only a few stages of processing, i.e., T. After greedy learning, joint training is utilized to learn the parameters of the T stages simultaneously. All the experiments for the DG-RBF model were implemented with Matlab. We used the L-BFGS toolbox provided by [50] to train our model. For all the models, we first conduct 200 iterations of the L-BFGS algorithm for each stage in a greedy manner, and then perform another 50 iterations on all the stages simultaneously. More implementation details are given in the experiments sections.

5 Learned Dynamic Guidance With CNN

In the previous section, we proposed a DG-RBF model which parameterizes the filters as well as the nonlinear functions in the stage-wise operations introduced in Eq. (7). By exploring the dynamic guidance strategy and learning optimal parameters in a task-driven manner, the proposed DG-RBF method greatly improves the flexibility of the original WASR model. But since DG-RBF follows the formula of stage-wise operation strictly - which only conducts one group of convolutions and nonlinear functions on the depth image - we adopted a group of RBF kernels to parameterize the penalty function in order to have a strong capacity towards nonlinearities. Furthermore, we utilize the L-BFGS algorithm [49] to train DG-RBF and it needs to calculate the gradient on the whole training set. The above reasons render the training of the complex DG-RBF model on a large training dataset time and memory consuming. In this section, we provide another parameterization of stage-wise operations for the guided depth enhancement. Specifically, we analyze the formula of Eq. (7) and use convolutional neural networks (CNNs) to approximate the stage-wise operations in a more flexible way.

5.1 Stage-Wise Operation With Intensity/Depth Encoder and Joint Decoder

In Eq. (7), the difference between the current estimation $\boldsymbol{X}^t$ and the new estimation $\boldsymbol{X}^{t+1}$ consists of two components. The first component $\mathcal{L}'(\boldsymbol{X}^t, \boldsymbol{Y})$ comes from the data fidelity term of the objective function. It put the residual between current estimation and input observation back into the next estimation. The second component $\sum_l \overline{\boldsymbol{k}}_l^t \otimes \big(\boldsymbol{W}_l^t(\boldsymbol{G}) \odot \boldsymbol{P}_l^{t\prime}(\boldsymbol{k}_l^t \otimes \boldsymbol{X}^t)\big)$ comes from the regularization term. It extracts high-dimensional features (analysis coefficients in the case of the WASR model) from the local structure in the image, and adjusts the features in the feature space to let the new estimation better fit the prior model.

When the optimization algorithm is adopted to minimize the objective function, the backward part $\mathcal{L}'(\boldsymbol{X}^t, \boldsymbol{Y})$ prevents the estimation $\boldsymbol{X}$ to move too far away from the observation $\boldsymbol{Y}$, and the algorithm converges when the two components get in balance. Since, in this paper, only a fixed number of stage-wise operations are performed to generate the high quality estimation, the backward part can be ignored for the purpose of simplicity. By ignoring the fidelity part, we get the following residual formulation of the stage-wise operation

$$\boldsymbol{X}^{t+1} = \boldsymbol{X}^t + \sum\nolimits_l \overline{\boldsymbol{k}}_l^t \otimes \big(\boldsymbol{W}_l^t(\boldsymbol{G}) \odot \boldsymbol{P}_l^{t\prime}(\boldsymbol{k}_l^t \otimes \boldsymbol{X}^t)\big). \tag{13}$$

In the residual component, an intensity encoder $\boldsymbol{W}_l^t(\boldsymbol{G})$, a depth encoder $\rho_l^{t\prime}(\boldsymbol{X}_t)$ and a joint decoder $\sum_l \overline{\boldsymbol{k}}_l^t \otimes (\cdot)$ cooperate to adjust the local structure in the current estimation. In particular, the intensity encoder and depth encoder extract local features from the intensity and depth images, resp.; then, after generating the joint coefficients with the point-wise product operator, the joint decoder reconstructs the final residual estimation. Denoting the intensity encoder, depth encoder and joint decoder by $F_I(\cdot)$, $F_D(\cdot)$ and $F_R(\cdot)$, we can rewrite Eq. (13) in the form

$$\boldsymbol{X}^{t+1} = \boldsymbol{X}^t + F_R\big(F_D(\boldsymbol{X}^t) \odot F_I(\boldsymbol{G})\big). \tag{14}$$

In our DG-CNN model, we formulate the encoders and decoders in Eq. (14) with several layers of CNN. Compared with the DG-RBF model, the CNN parameterization is able to provide more powerful encoders and decoders with stronger nonlinear modeling capacity. Furthermore, well optimized

CNN toolboxes enable us to train the DG-CNN model easily on large training datasets.

5.2 DG-CNN Network Structure

Based on our analysis from the previous Section 5.1, the stage-wise operation for the WASR can be formulated with an intensity encoder, a depth encoder and a joint decoder. To parameterize the encoder and decoder with a CNN, one simple solution is to directly use several convolution and activation layers to form the encoder and the decoder, and to gradually improve the quality of the depth estimations $\{X^t\}_{t=1,\ldots,T}$. Yet, such a strategy reconstructs the joint features back into the image domain where several stages of operation are concatenated together and the reconstructed image acts as a bottleneck in the deep neural network. The bottlenecks may affect the training speed of the neural networks. Furthermore, reconstructing the feature maps back into the image domain impedes the increasing of the network perceptual field. In order to avoid the appearance of bottlenecks in the networks, for the multi-stage DG-CNN model, the tth depth encoder takes the feature maps of the $(t-1)$th joint decoder as input. Furthermore, in order to increase the perceptual field of the intensity encoder, the intensity encoder in each stage takes the output feature maps from previous intensity encoder as well as the guidance intensity image as inputs. An illustration of a two-stage DG-CNN model can be found in Fig. 3. The orange, the purple and the gray blocks represent the depth encoder, the intensity encoder and the joint decoder, respectively. Each encoder consists of 5 convolution, batch normalization [51] and leakyReLU [52] layers, and each decoder consists of 3 convolution, batch normalization [51] and leakyReLU [52] layers. Each convolution layer generate 32 feature maps. Except for the first depth encoder block which takes the observed depth image as input, all the remaining depth encoders take the feature maps of the joint decoder as input. Another convolution layer (red rectangle in Fig. 3) is utilized to reconstruct the feature maps of the decoder back into the image domain.

All the DG-CNN experiments conducted in this paper were implemented with the Pytorch toolbox [53]. We train our model with the Adam [54] solver ($\beta_1 = 0.9$), and set the weight decay parameter to 10^{-4}. We start from a learning rate of 0.001 and divide it by 10 every 10^5 iterations. The total number of training iterations is 3×10^5. An Nvidia Titan XP GPU was utilized to train our model. More details on each dataset can be found in the experiments sections.

6 Model Analysis and Discussion

Before comparing the proposed method with state-of-the-art approaches, we conduct ablation experiments to analyze the effect of hyper-parameters and network architecture design choices. We first introduce the general setting of our ablation experiments, and then present experimental results to analyze the proposed DG-RBF and DG-CNN models, respectively.

6.1 Experimental Setting

We utilize the commonly used Middlebury dataset [55] to conduct our ablation experiments. Following the experimental settings from previous works [9], [31], we use the *Art*, *Books* and *Moebius* images as testing images. To prepare training data, we use 46 depth and intensity image pairs from the Middlebury dataset [55] and augment them with flipping, rotation and scaling operations [56]. Both the training and testing samples are generated by a bicubic resizing of the high quality depth maps. The training and testing datasets are strictly separated, and there is no overlap between the scenes of the training and testing images. To train our DG-RBF model, we crop 3,000 small images of resolution 72×72 from the 46 images as training set. We did not use all the patches from the 46 training images because the L-BFGS method [49] used to train DG-RBF needs to calculate the gradient on the whole training set, and training the model on large datasets is time and memory expensive. In comparison, for our DG-CNN model, all the 46 large images and their augmentations have been adopted as the training dataset. In each training iteration, we randomly crop 32 136×136 patches from the 46 images to train our model. Although the augmentation improves the structural variety of the training samples, the training data is still not diverse enough as the color palette is rather poor. In our experiments, we use only the gray intensity image to guide the reconstruction.

6.2 Analyzing DG-RBF

6.2.1 *Initialization and Model Regularization*

Before investigating the hyper-parameters of our model, we study two key aspects of the proposed method: the initialization and the model regularization. Specifically, DG-RBF has two main groups of parameters for the filters and the non-linear functions, and we investigated the effect of initialization approaches for both parameter groups. Furthermore, we follow [26] and require the filters in DG-RBF to be zero-mean. We also provide experimental results to show the effect of the zero-mean constraint.

To analyze the effect of zero-mean constraint, we compare two parameterization schemes for the filters. The first scheme adopts the zero-mean constraint and requires the filters to be the summation of zero-mean DCT filters. While, the second scheme does not regularize the filters, and directly learns the values in the filters. For both the filters and the penalty functions, we test two kinds of initialization approaches: random initialization and model-inspired initialization. In particular, we initialize the filters with random values or point-wise difference filters, as widely done in previous optimization-based depth enhancement work; and initialize the penalty functions with random values or the commonly used influence function as adopted in [26]. We adopt different initialization settings to train our DG-RBF models to super-resolve the testing images with a factor 8. We train a 5-stage DG-RBF model with 48 7×7 filters on 3,000 training samples. We first initialize the penalty function with the commonly used influence function and evaluate the effect of initialization and parameterization methods on the filters. The experimental results are reported in Table 1. The initialization approach as well as the parameterization method for the filters greatly affect the performance of the unrolled network. Domain knowledge such as zero-mean filters and point-wise difference filters are beneficial in designing as well as initializing network structures.

The effect of the initialization method for the penalty functions is not as significant as that for the filters, changing from the model-inspired initialization to random initialization will

TABLE 1
Experimental Results (Avg. RMSE) on the 3 Test Images [56] With Different Initialization Methods and Constraints for the Filters

	Random Init.	Model Init.
W/ Zero-mean Cons.	3.00	2.25
W/o Zero-mean Cons.	3.22	3.23

only slightly increase the RMSE value on the Middlebury dataset [55] from 2.25 to 2.37.

6.2.2 Filter Size and Number

After investigating the effect of initialization and model regularization, we study the most important hyper-parameters for DG-RBF: the filter size and the number of filters. We train DG-RBF models with different numbers of filters as well as filter sizes with 3,000 training samples. We utilize the same initialization and parameterization scheme for all the models. The SR results as well as the average inference time on the 3 testing images [55] of different models are shown in Table 2. The experiments were conduct in the Matlab environment and we test different models on a PC with Intel i7-4790 CPU. All the models utilize 5 stage-wise operations to super-resolve the testing images with a factor 8. Generally, increasing the filter number and size both help to improve the SR performance. The filter size plays a more import role than the number of filters in the DG-RBF model. In the remainder of this paper, we set the filter size to 9×9 and filter number to 24, seeking a balance between performance and speed.

6.2.3 Number of RBF Kernels

In the DG-RBF model, the parameterization of non-linear penalty functions is the same as in [26]. In [26], 65 kernels with scaling parameter 10 have been utilized to cover the activation range between -310 to 310. This said, we experimentally found that the penalty functions work well even when we only parameterize a smaller activation range. The SR results with different kernel numbers and scaling factors are reported in Table 3. All the models utilize 5 stage-wise operations to super-resolve the testing images with a factor 8. The proposed DG-RBF model achieves good results for a wide range of kernel numbers. It is robust to this hyper-parameter. For similar parameterization ranges, scaling factors 2.5, 5 and 10 can achieve similar SR results and a scaling factor 20 will lead to a performance drop due to insufficient parameterization accuracy. In addition, although DG-RBF cannot achieve good SR performance with very small parameterization range, we do not need to parameterize the penalty function for the complete possible activation range. Outside [-170, 170], a further enlargement of the parameterization range will not improve the SR results. Due to the above reasons, we utilize 33 kernels with scaling factor 10 to parameterize the penalty functions used in DG-RBF method.

TABLE 2
Experimental Results (Avg. RMSE / Runtime [s]) on the 3 Testing Images [56] by DG-RBF Variations With Different Filter Sizes and Numbers

F. num.	12	24	48	72
5×5	2.47 / 3.29s	2.45 / 5.70s	2.42 / 10.56s	2.39 / 15.88s
7×7	2.34 / 4.69s	2.32 / 8.02s	2.25 / 14.77s	2.28 / 21.57s
9×9	2.28 / 6.52s	2.18 / 11.26s	2.14 / 20.40s	2.15 / 29.31s
11×11	2.29 / 9.03s	2.16 / 15.27s	2.13 / 28.32s	2.13 / 41.98s

TABLE 3
Experimental Results (Avg. RMSE) on the 3 Test Images [56] by DG-RBF Variations With Different Penalty Parameterization Approaches

Scaling Factor / Kernel Num.	2.5	5	10	20
17	-	2.33	2.22	2.30
33	2.32	2.20	2.18	2.23
65	2.24	2.17	2.19	-

6.2.4 Stage Number

Another important hyper-parameter in the proposed DG-RBF model is the number of stages. As we utilize the L-BFGS [49] algorithm to train the stage-wise operations in a greedy manner, more stages can always lead to smaller training error. Yet, despite reducing the training error, adopting more stage-wise operations will also introduce more computational burden and increase the risk of over-fitting. In Table 4, we present the average RMSE and run-time on the three testing images in the Middlebury dataset [55]. For simple cases such as zooming factors 2 and 4, DG-RBF is able to achieve good results with a small number of stage-wise operations; whereas for challenging cases the proposed model needs more operations to deliver a good estimation. As the DG-RBF model provides a very easy way to vary computational complexity, we propose to adopt different operation points to process different zooming factors. For SR experiments with zooming factor 2, 4, 8 and 16, we utilize 3, 4, 5 and 6 stage-wise operations, respectively, in the DG-RBF model. Note that we adopt different numbers of stage-wise operations for the purpose of balancing the computational burden and the reconstruction performance. As can be found in Table 4, with a large stage number, DR-RBF is able to achieve high quality depth reconstruction results for different zooming factors.

6.3 Analyzing DG-CNN

Our DG-CNN also has a large number of hyper-parameters, including the feature map number and filter size, as well as training parameters such as the learning rate. For most of these parameters, we follow some commonly used settings in other CNN based approaches, and did not conduct experiments to analyze the effect of these parameters. In this subsection, we first present the depth reconstruction performance of

TABLE 4
Experimental Results (Avg. RMSE and Run-Time) on the 3 Testing Images [56] by DG-RBF Variations With Different Stage Numbers

Stage	S=1	S=2	S=3	S=4	S=5	S=6	S=7	S=8
$\times 2$	0.84	0.73	0.73	0.74	0.74	0.74	0.74	0.75
$\times 4$	1.74	1.39	1.29	1.27	1.27	1.27	1.27	1.27
$\times 8$	2.88	2.40	2.26	2.22	2.18	2.18	2.19	2.19
$\times 16$	5.73	4.08	3.82	3.76	3.74	3.73	3.72	3.72
Time [s]	3.65	5.50	7.36	9.10	10.93	12.80	14.55	16.32

TABLE 5
Experimental Results (Avg. RMSE) on the 3 Testing Images [56] by DG-CNN Variations With Different Numbers of Stage-Wise Sub-Networks

Stage	S=1	S=2	S=3	S=4	S=5
×2	0.45	0.43	0.43	0.43	0.42
×4	0.88	0.84	0.82	0.82	0.81
×8	1.57	1.42	1.35	1.37	1.35
×16	2.80	2.50	2.40	2.36	2.36

DG-CNN with different stage numbers. Then, we analyze two properties of the proposed DG-CNN, which come from the unrolled optimization steps of the WASR model. Our ablation experiments show the advantages of the optimization-inspired network architecture design.

6.3.1 Stage Number

We evaluate the proposed DG-CNN method with different stage numbers (from one to four) on the Middlebury data set. Table 5 summarizes the SR results for all the different factors with different numbers of stage-wise operations. Similarly to our DG-RBF model, with complex networks (more stage-wise operations), the DG-CNN is able to achieve good results on all the zooming factors. For simple cases with small zooming factors a large number of stage-wise operations is not necessary and the DG-CNN is able to deliver high quality results with a small number of stage-wise operations. The same as for the DG-RBF model, we adopt different numbers of stage-wise operations in the DG-CNN for SR tasks with different zooming factors. For zooming factors 2, 4, 8 and 16, we utilize 1, 2, 3 and 4 stage-wise operations, respectively, in the proposed DG-CNN method.

6.3.2 Stage-Wise Residual Learning

In each stage of the DG-CNN, we utilize encoder networks $\{F_I, F_D\}$ and a decoder network $\{F_R\}$ to approximate the difference between the current estimation and the next estimation $X^{t+1} - X^t$. Each stage-wise operator can be seen as a special residual block, which has been proved to be a highly effective structure in deep neural networks [57]. In this part, we conduct ablation experiments to show the advantage of stage-wise residual learning. In particular, we compare the proposed network architecture with two ablation architectures, which are shown in Fig. 4. The first ablation network (Fig. 4a) adopts a one-stage encoder-decoder network to estimate the residual between the input and the target high quality depth image. The second ablation network (Fig. 4b) adopts stage-wise operations but only contains a global skip connection between the input and output image. For multi-stage networks with/without stage-wise residual learning we utilize the same encoder-decoder sub-networks, whereas for the single stage network

TABLE 6
Experimental Results (Avg. RMSE) on the 3 Testing Images [56] by DG-CNN and Ablation Network Architectures Shown in Fig. 4

Single Stage + Global Res.	Multi-Stage + Global Res.	Multi-Stage + Stage-wise Res.
1.42	1.53	1.35

TABLE 7
Experimental Results (Avg. RMSE) on the 3 Testining Images [56] by DG-CNN Variations With Different Feature Maps Combinations

Feature maps combination	×2	×4	×8	×16
concatenation	0.44	0.86	1.36	2.41
multiplication	0.45	0.84	1.35	2.36

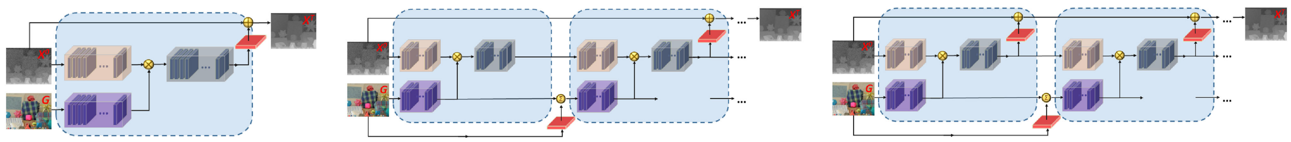
we incorporate two times more convolutional layers in the encoder and decoder sub-networks. All three networks have the same computational complexity. The competing results of different networks can be found in Table 6, showing that the optimization-inspired stage-wise residual learning is beneficial for the guided depth reconstruction task.

6.3.3 Dependency Modeling

WASR summarizes previous optimization-based methods and uses point-wise multiplication to combine the intensity and depth features. We adopt the multiplication strategy also in our DG-CNN network structure. Most of previous CNN-based guided depth reconstruction approaches [20], [21] use the concatenation operation to combine the intensity and depth features. Compared with concatenation, the point-wise multiplication helps to reduce the number of parameters as well as the computational burden of the network. By exchanging multiplication with concatenation, each stage-wise operation gets about 5 percent more parameters and running time. Furthermore, as reported in Table 7, combining feature maps with multiplication instead of concatenation achieves comparable or slightly better SR results on the Middlebury dataset.

7 Guided Depth Super-Resolution Experiments

In this section, we compare the proposed methods with other depth super-resolution methods. Two commonly used data-sets (Middlebury [55] and NYU [4]) are utilized to evaluate the depth upsampling performance of the proposed methods. Besides the baseline bicubic and bilinear upsampling methods, we compare the proposed methods with a variety of guided depth super-resolution methods. The comparison methods include three filtering based methods [17], [58],



(a) Single Stage + Global Res. (b) Multi-Stage + Global Res. (c) Multi-Stage + Stage-wise Res.

Fig. 4. Ablation networks used to validate the effectiveness of the stage-wise residual learning structure. More details can be found in Section 6.3.2.

TABLE 8
Experimental Results (RMSE) on the 3 Noise-Free Test Images

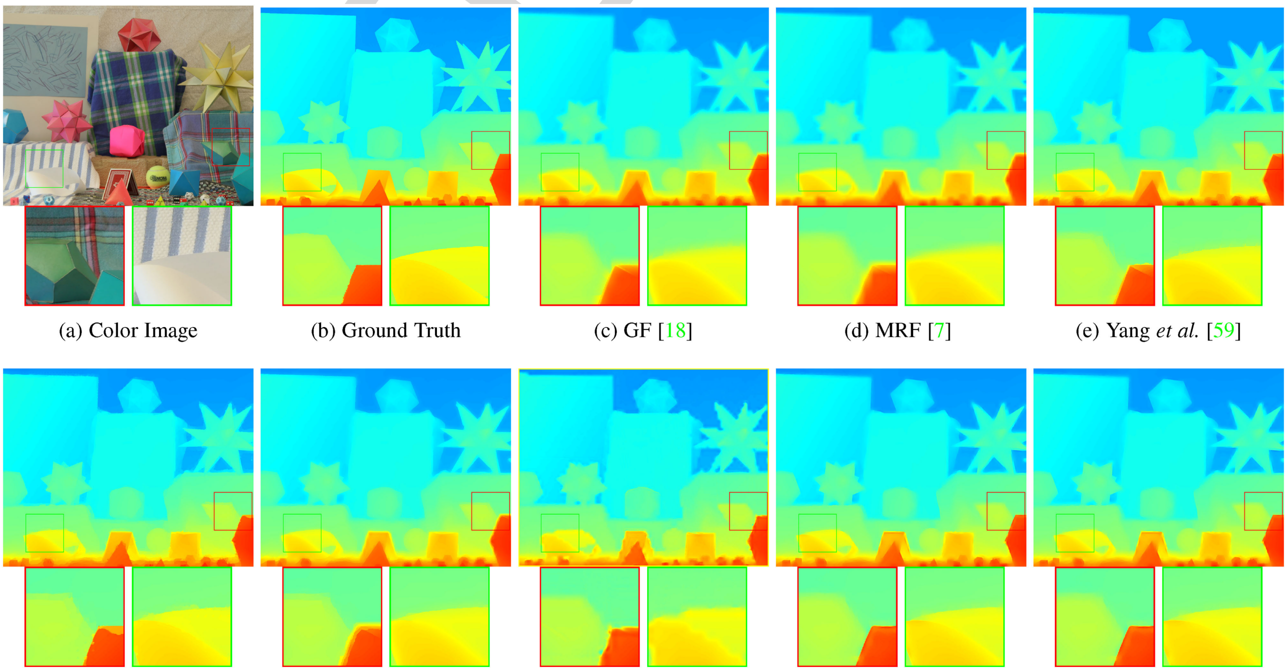
	Art				Books				Moebius				Average			
	×2	×4	×8	×16	×2	×4	×8	×16	×2	×4	×8	×16	×2	×4	×8	×16
Bicubic	2.57	3.85	5.52	8.37	1.01	1.56	2.25	3.35	0.91	1.38	2.04	2.95	1.50	2.26	3.27	4.89
Bilinear	2.83	4.15	6.00	8.93	1.12	1.67	2.39	3.53	1.02	1.50	2.20	3.18	1.66	2.44	3.53	5.21
GF [18]	2.93	3.79	4.97	7.88	1.16	1.58	2.10	3.19	1.10	1.43	1.88	2.85	1.73	2.27	2.98	4.64
MRF [7]	3.12	3.79	5.50	8.66	1.21	1.55	2.21	3.40	1.19	1.44	2.05	3.08	1.84	2.26	3.25	5.05
Yang 2007 [59]	4.07	4.06	4.71	8.27	1.61	1.70	1.95	3.32	1.07	1.39	1.82	2.49	2.25	2.38	2.83	4.69
Park [8]	2.83	3.50	4.17	6.26	1.20	1.50	1.98	2.95	1.06	1.35	1.80	2.38	1.70	2.12	2.65	3.86
TGV [9]	3.03	3.79	4.79	7.10	1.29	1.60	1.99	2.94	1.13	1.46	1.91	2.63	1.82	2.28	2.90	4.22
Yang 2014 [60][1]	3.13	4.76	7.79	13.44	1.30	2.16	5.44	13.00	1.16	1.99	3.30	7.02	1.86	2.97	5.51	11.15
SDF [61]	3.31	3.73	4.60	7.33	1.51	1.67	1.98	2.92	1.56	1.54	1.85	2.57	2.13	2.31	2.81	4.27
DJF [21]	2.77	3.69	4.92	7.72	1.11	1.71	2.16	2.91	1.04	1.50	1.99	2.95	1.64	2.30	3.02	4.53
MSG-Net [22][2]	0.66	1.47	2.46	4.57	0.37	0.68	1.03	1.60	0.36	0.66	1.02	1.63	0.46	0.94	1.50	2.60
DG-RBF (ours)	1.06	1.98	3.40	6.07	0.57	0.92	1.62	5.57	0.55	0.92	1.56	2.55	0.73	1.27	2.19	4.73
DG-CNN (ours)	0.63	1.31	2.17	3.94	0.36	0.61	0.95	1.60	0.33	0.58	0.92	1.47	0.44	0.83	1.35	2.34

[61], an MRF based optimization method [7], a non-local mean regularized depth upsampling method [8], a total generalized variation (TGV) method [9], the joint static and dynamic filtering (SDF) method [60], and the recently proposed CNN-based deep joint filtering method [20] and primal-dual network (PDN) [29]. In [21], Hui *et al.* also evaluated their proposed MSG-Net on the 3 testining images in the Middlebury [55] dataset. However, Hui *et al.* [21] utilized the Gaussian blur + downsampling operation to generate the low resolution input images, which is considered to be easier than the bicubic downsampling setting in the SR literature [62]. Here we also reported the performance by the MSG-Net [21] for reference. Details about the experimental setup will be introduced in the following subsections.

7.1 Super-Resolution Results on the Middlebury Dataset

Following the experimental setting of [9], we conduct super-resolution experiments with both the noise-free and noisy low resolution depth map for four zooming factors, i.e., 2, 4, 8 and 16. The settings of the noise-free experiment have been introduced in Section 6. To compare different methods with noisy low-resolution inputs, we utilize the testing images provided in [8]. To synthesize real noisy depth images, Park *et al.* [8] added conditional Gaussian noise to the low resolution depth maps. The Gaussian noise variance depends on the distance between the camera and the scene, and Park *et al.* did not provide the details for the noise hyper-parameters. To generate training data, we add i.i.d Gaussian white noise with $\sigma = 6$ to the 46 clean images used in our noise-free experiments.

The super-resolution results on the 3 noise-free testing images of the different methods are shown in Table 8. The proposed DG-RBF and DG-CNN methods consistently show their advantage over the competing methods. The proposed DG-RBF method outperforms all the optimization-based approaches as well as a recently proposed CNN-based method DJF [20]. DG-CNN achieves the best results on all the 3 images with different zooming factors. In Fig. 5,



(a) Color Image (b) Ground Truth (c) GF [18] (d) MRF [7] (e) Yang *et al.* [59]
(f) TGV [9] (g) SDF [61] (h) DJF [21] (i) DG-RBF (j) DG-CNN

Fig. 5. Depth restoration results of different methods based on noise-free data (Moebius).

TABLE 9
Experimental Results (RMSE) on the 3 Noisy Test Images

	Art				Books				Moebius				Average			
	×2	×4	×8	×16	×2	×4	×8	×16	×2	×4	×8	×16	×2	×4	×8	×16
Bicubic	5.32	6.07	7.27	9.59	5.00	5.15	5.45	5.97	5.34	5.51	5.68	6.11	5.22	5.58	6.13	7.22
Bilinear	4.58	5.62	7.14	9.72	3.95	4.31	4.71	5.38	4.20	4.57	4.87	5.43	4.24	4.83	5.57	6.84
GF [18]	3.55	4.41	5.72	8.49	2.37	2.74	3.42	4.53	2.48	2.83	3.57	4.58	2.80	3.33	4.24	5.87
MRF [7]	3.49	4.51	6.39	9.39	2.06	3.00	4.05	5.13	2.13	3.11	4.18	5.17	2.56	3.54	4.87	6.56
Yang 2007 [59]	3.01	4.02	4.99	7.86	1.87	2.38	2.88	4.27	1.92	2.42	2.98	4.40	2.27	2.94	3.62	5.51
Park [8]	3.76	4.56	5.93	9.32	1.95	2.61	3.31	4.85	1.96	2.51	3.22	4.48	2.56	3.23	4.15	6.22
TGV [9]	3.19	4.06	5.08	7.61	1.52	2.21	2.47	3.54	1.47	2.03	2.58	3.56	2.06	2.77	3.38	4.90
Chan [12]	3.44	4.46	6.12	8.68	2.09	2.77	3.78	5.45	2.08	2.76	3.87	5.57	2.54	3.33	4.59	6.57
Yang 2014 [60]	5.37	6.06	9.33	15.02	4.98	5.06	7.62	16.13	4.73	5.32	5.73	9.19	5.03	5.48	7.56	13.45
SDF [61]	3.36	3.86	4.93	7.85	1.59	1.92	2.60	4.16	1.64	1.85	2.67	4.21	2.20	2.54	3.40	5.41
PDN [30]	1.87	3.11	4.48	7.35	1.01	1.56	2.24	3.46	1.16	1.68	2.48	3.62	1.35	2.12	3.07	4.81
FBS [62]	2.93	3.79	4.95	7.13	1.39	1.84	2.38	3.29	1.38	1.80	2.38	3.23	1.90	2.48	3.24	4.55
DG-RBF (ours)	1.91	3.06	4.75	8.10	1.21	1.77	2.55	4.12	1.32	1.84	2.86	4.13	1.48	2.22	3.39	5.45
DG-CNN (ours)	1.74	2.53	3.51	5.14	1.09	1.40	1.93	2.80	1.20	1.47	2.01	2.91	1.34	1.80	2.48	3.62

we give visual examples of the super-resolution results for the Moebius image with zooming factor 16. In the figure we can see that the guided filter method [17] and the MRF method [7] cannot generate very sharp edges. The results of [58], [8] and [9] have some artifacts around the edges. Our methods are able to generate high quality depth maps with sharper edges and fewer artifacts.

We further evaluate the proposed methods for noisy depth super-resolution. For both the DG-RBF and DG-CNN models, we utilize the same hyper-parameters as we adopted in the noise-free experiment. The results by different methods are shown in Table 9. We do not provide the results of DJF [20] because the authors have not provided their network and have not reported results for such setting. The results by [12] are also included, a method designed to handle noise in depth super-resolution tasks. The proposed methods again achieve the best results.

7.2 Super-Resolution Results on the NYU Dataset

In [20], Li *et al.* utilize the first 1,000 images of the NYU dataset [4] as training data, and evaluate their DJF method on the last 449 images of the NYU dataset. In this section, we follow their experimental setting and compare different methods on the 449 images. The results of the other methods are provided by the authors of [20]. For the DG-RBF model, we crop 3000 72×72 subimages as the training set. For the DG-CNN model, we use all the 1,000 images as training dataset. The hyper-parameters for both the DG-RBF and DG-CNN models are the same as our settings on the Middleburry [55] dataset. The experimental results are shown in Table 10. Compared with other methods, the proposed DG-RBF and DG-CNN achieve the best results in terms of RMSE. Some visual examples of the SR results of different algorithms have been provided in Fig. 6.

TABLE 10
Experimental Results (RMSE) on the 449 NYU Test Images

	NYU		
	×4	×8	×16
MRF [7]	4.29	7.54	12.32
GF [18]	4.04	7.34	12.23
JBU [16]	2.31	4.12	6.98
TGV [9]	3.83	6.46	13.49
Park [8]	3.00	5.05	9.73
Ham [32]	3.04	5.67	9.97
DJF [21]	1.97	3.39	5.63
DG-RBF (ours)	1.35	2.69	5.11
DG-CNN (ours)	0.87	1.78	3.53

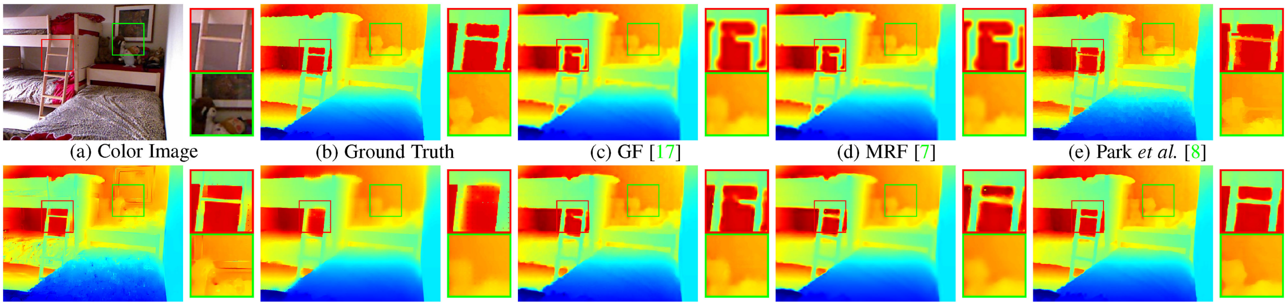
8 Realistic Guided Depth Reconstruction

In this section, we provide some experimental results for other depth map restoration problems. We evaluate the proposed methods on two datasets. The first dataset is a synthetic dataset proposed by Lu *et al.* [11]. In order to mimic real low-quality depth images, Lu *et al.* [11] add zero mean additive Gaussian noise to the depth images, and then manually set 13 percent of pixels in the depth map as missing values to simulate the depth map acquired from consumer level depth sensors. Moreover, the second dataset is a real sensor dataset provided by [9]. A Time of Flight (Tof) and a CMOS camera are used to obtain low resolution depth maps and intensity images, and the ground truth depth images are generated by a structured light scanner. The detailed experimental setting will be introduced in the following subsections.

8.1 Experimental Results on Synthetic Dataset [11]

In [11], Lu *et al.* propose a synthetic dataset to evaluate guided depth reconstruction methods. 30 depth and RGB image pairs in the Middlebury database [55] are included in the dataset. The size of all the images have been normalized to the same height of 370 pixels. To compare with previous algorithms, we utilized the cross-validation method to obtain the reconstruction results on all the 30 images. Concretely, we divide the 30 images into 10 groups, and utilize 9 groups to train models to estimate the depth maps in the remainder group. We compare our method with other methods designed for this task, which include a low rank based method [11] and the recently proposed mutual-structure joint filtering method [18].

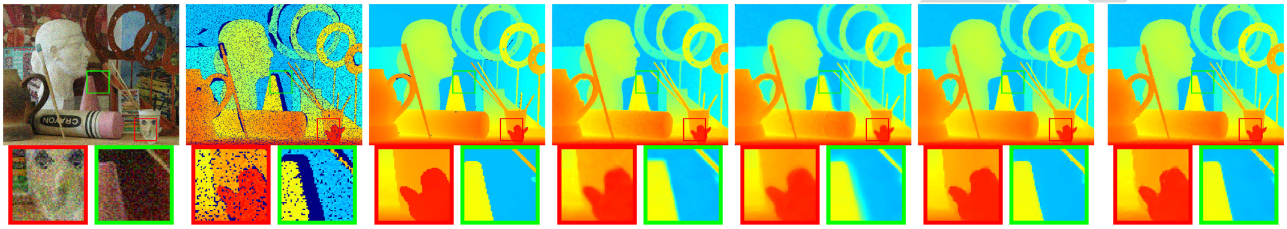
Since our proposed method does not consider the noise in the RGB image, for fair comparison, we pre-process the RGB image by a state-of-the-art denoising method [63], [64] and use the denoised image to guide the restoration of the depth map. Such a method has been utilized in the original paper [11] to compare with other depth restoration methods. In addition, since the missing values in the depth map



(a) Color Image (b) Ground Truth (c) GF [17] (d) MRF [7] (e) Park *et al.* [8]

(f) TGV [9] (g) SDF [61] (h) DJF [21] (i) DG-RBF (j) DG-CNN

Fig. 6. Depth SR results by different methods for a testing image in the NYU dataset [4].



(a) Color Image (b) Input (c) Ground Truth (d) Lu *et al.* [11] (e) Shen *et al.* [19] (f) **DG-RBF (ours)** (f) **DG-CNN (ours)**

Fig. 7. Depth restoration results of different methods.

are represented as zeros which may be considered as very sharp edges in the depth map, we use a simple masked joint bilateral filtering [65] method to generate initialization values for the unknown points in the depth map.

The restoration results by different methods are shown in Table 11. For both the DG-RBF and DG-CNN model, the hyper-parameters are the same as used for the super-resolution experiment with zooming factor 4. The results of [11] and [18] are downloaded from the websites of the respective authors. Both proposed DG-RBF and DG-CNN methods outperform the competing methods. Interestingly, different from our experimental results for the guided super-resolution task, the results by the DG-CNN approach are just comparable to the results by DG-RBF. The main reason is the very limited training data, the 27 low-resolution images are insufficient to train the complex DG-CNN model for best performance. In contrast, the DG-RBF model can still achieve good performance with a small training dataset because its number of parameters is much lower than that of DG-CNN. Some visual examples of the restoration results of different algorithms have been provided in Fig. 7.

8.2 Experimental Results on Real Sensor Data

In addition to synthetic data, we also evaluate the proposed method on a real sensor dataset [9]. We utilize the same 46 images from the Middlebury dataset [55] as training images. As for our experiment on the synthetic dataset, we also utilized the joint bilateral filtering [65] method to generate initialization values for the unknown points in the depth map. For both the DG-RBF and DG-CNN model, the hyper-parameters are the same as for the noise-free Middleburry super-resolution experiment with zooming factor 4. We compare our methods with other classic or state-of-art methods. The guided reconstruction results are shown in Table 12. Our methods get the best results in terms of the mean absolute error (MAE). From Fig. 8 it is easy to see that our methods are capable of generating clean estimations, whereas the results by other methods copy irrelevant textures from the intensity image.

TABLE 11
Experimental Results (RMSE) on the 30 Test Images in [11]

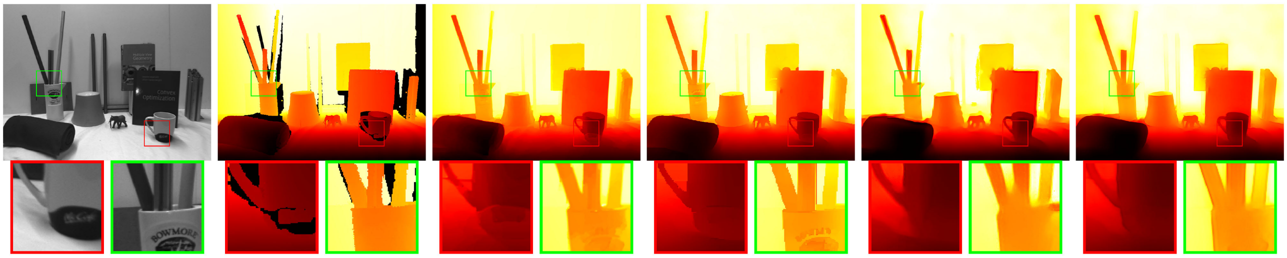
Lu *et al.* [11]	Shen *et al.* [19]	DG-RBF (ours)	DG-CNN (ours)
2.59	2.64	2.30	2.27

9 Discussion

By analyzing previous optimization-based methods, we proposed a WASR model for the task of guided depth reconstruction. Instead of solving the optimization problem of the WASR model, we proposed to utilize different parameters in the optimization process and conduct the depth reconstruction with a dynamic guidance strategy. In particular, we unfolded the optimization process of WASR and got the

TABLE 12
Real Data Results (MAE) on the 3 Test Images in [9]

	Books	Shark	Devil	Average
Nearest Neighbor	18.21	21.83	19.36	19.80
Bilinear	17.10	20.17	18.66	18.64
Kopf [16]	16.03	18.79	27.57	20.80
He [17]	15.74	18.21	27.04	20.33
FBS [62]	13.42	17.07	16.10	15.53
SDF [61]	13.47	16.75	16.36	15.53
TGV [9]	12.36	15.29	14.68	14.11
Yang [60]	12.25	14.71	13.83	13.60
DG-RBF (ours)	12.18	14.48	13.79	13.48
DG-CNN (ours)	12.14	14.46	13.11	13.24



(a) Intensity Image (b) Ground Truth (c) TGV [9] (d) SDF [61] (e) **DG-RBF (ours)** (f) **DG-CNN (ours)**

Fig. 8. Depth reconstruction results of different methods based on real data (Books).

formula of stage-wise operation for guided depth reconstruction. Based on the stage-wise formula Eq. (7), we introduced two networks which parameterize the stage-wise operation with RBF kernels (DG-RBF) or convolutional neural networks (DG-CNN). Experimentally, we have shown that both the DG-RBF and DG-CNN models are able to generate good depth reconstruction results. In this section, we discuss the respective merits and drawbacks of the two models.

DG-RBF follows the unfolded optimization process of WASR strictly and parameterizes the nonlinear penalty functions with Gauss RBF kernels. In comparison, the DG-CNN model approximates the stage-wise operation in a lose way; we decompose the stage-wise operation as an intensity encoder, a depth encoder and a joint decoder, and use several layers of CNN to parameterize these sub-components. Although both methods benefit from the domain knowledge of previous researches as well as training data, they adopt different trade-offs between the two merits. The DG-RBF method strictly follows the unfolded optimization process of WASR. It is more related to previous optimization-based approaches. This prior knowledge about the guided depth reconstruction problem enables the proposed DG-RBF method to capture the relationship between the guidance and the depth image in a more economic way. As a result, the DG-RBF method can be trained on small datasets and its generalization capacity is better than that of DG-CNN in general. On the synthetic dataset provided by Lu *et al.* [11], which only has 27 small training images, DG-RBF model achieved comparable results to the DG-CNN model with much less parameters. Yet, following the unfolded WASR model strictly limits the flexibility of DG-RBF on datasets with large amounts of training data. The results generated by the DG-RBF are not as good as those of some learning-based approaches. In comparison to DG-RBF, DG-CNN benefits from the overall structure of the unfolded WASR model. The stage-wise formula provides useful hints on the design of the DG-CNN, while the advances in deep learning enable DG-CNN to take full advantage of training data. Consequently, the DG-CNN achieved stage-of-the-art performance on different datasets.

Another difference between DG-RBF and DG-CNN resides in the training. Different from CNNs, where one can use the back-propagation algorithm for gradient calculation, the computation of the parameter gradients for the DG-RBF model is time consuming. In addition, the L-BFGS method [49] used to train DG-RBF requires to calculate parameter gradients for all the training samples. We have also tried to train DG-RBF with stochastic algorithms, such as stochastic gradient descent (SGD) [66] and its ADAM variation [54]. L-BFGS always generates better models which can generate high quality depth reconstruction results. The limited performance achieved by the SGD trained DG-RBF model may be due to our parameterization scheme. Studies [67] in the deep learning literature have found that components in the network can greatly affect the training of the network. Inappropriate activation functions in the network may lead to the vanishing gradient problem and can render the network hard to train. The complex parameterization scheme adopted in our DG-RBF model did not take the training performance into consideration. Stochastic algorithms with heuristic learning rates may not be able to deliver a good model. L-BFGS computes accurate gradients on the whole training set and utilizes a line search method to determine the step length in each step. It has been utilized to train optimization-inspired networks in many previous works [26], [28].

10 Conclusion

To model the dependency between the guiding intensity image and the depth image we proposed a weighted analysis sparse representation (WASR) model for guided depth reconstruction. An intensity weighting term and an analysis representation regularization term are combined to model the complex relationship between the depth image and RGB image. We unfold the optimization process of the WASR model as a series of stage-wise operations. Two models, DG-RBF and DG-CNN, have been introduced to parameterize the stage-wise operation with Gaussian RBF kernels and CNNs, respectively, and we learn their model parameters in a task-driven manner. Both models generate high quality depth estimation in just a couple of stages. We experimentally validated their effectiveness for guided depth super-resolution and realistic depth reconstruction tasks using standard benchmarks. To the best of our knowledge, our proposed DG-RBF and DG-CNN methods achieve the best quantitative results (RMSE) to date and better visual quality than the compared state-of-the-art approaches.

Acknowledgments

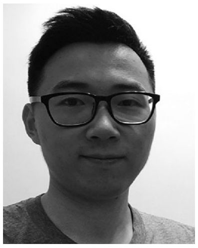
This work was supported in part by the ETH Zurich OK Fund, by Huawei and by Nvidia through a GPU Grant.

References

[1] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, “Efficient regression of general-activity human poses from depth images,” in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 415–422.
[2] C. Wan, T. Probst, L. Van Gool, and A. Yao, “Dense 3D regression for hand pose estimation,” in *Proc. IEEE Conf. Comput. Visi. Pattern Recognit.*, 2018, pp. 5147–5156.

[3] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," *Comput. Vis. Winter Workshop*, pp. 1–10, 2015.
[4] P. K. N. Silberman, D. Hoiem, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp 746–760.
[5] J. Zhu, L. Wang, R. Yang, J. E. Davis, and Z. Pan, "Reliability fusion of time-of-flight depth and stereo geometry for high quality depth maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1400–1414, Jul. 2011.
[6] S. Izadi *et al.*,"KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 559–568.
[7] J. Diebel and S. Thrun, "An application of Markov random fields to range sensing," in *Proc. 18th Int. Conf. Neural Inf. Process. Syst.*, 2005, pp. 291–298.
[8] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon, "High quality depth map upsampling for 3D-TOF cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1623–1630.
[9] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 993–1000.
[10] H. Kwon, Y.-W. Tai, and S. Lin, "Data-driven depth map refinement via multi-scale sparse representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 159–167.
[11] S. Lu, X. Ren, and F. Liu, "Depth enhancement via low-rank matrix completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3390–3397.
[12] D. Chan, H. Buisman, C. Theobalt, and S. Thrun, "A noise-aware filter for real-time depth upsampling," in *Proc. Workshop Multi-Camera Multi-Modal Sens. Fusion Algorithms Appl.*, A. Cavallaro and H. Aghajan, Eds. Marseille, France. Inria-00326784, Oct. 2008.
[13] J. Lu, H. Benko, and A. D. Wilson, "Hybrid HFR depth: Fusing commodity depth and color cameras to achieve high frame rate, low latency depth camera interactions," in *Proc. ACM CHI Conf. Human Factors Comput. Syst.*, 2017, pp. 5966–5975.
[14] J. Zhu, L. Wang, R. Yang, and J. Davis, "Fusion of time-of-flight depth and stereo for high accuracy depth maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[15] D. L. J. Kopf, M. F. Cohen, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Trans. Graphics*, vol. 26, no. 3, 2007, Art. no. 96.
[16] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp 1–14.
[17] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.
[18] X. Shen, C. Zhou, L. Xu, and J. Jia, "Mutual-structure for joint filtering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3406–3414.
[19] M. Kiechle, S. Hawe, and M. Kleinsteuber, "A joint intensity and depth co-sparse analysis model for depth map super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1545–1552.
[20] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep joint image filtering," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp 154–169.
[21] T.-W. Hui, C. C. Loy, and X. Tang, "Depth map super-resolution by deep multi-scale guidance," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 353–369.
[22] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[23] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1646–1654.
[24] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1529–1537.
[25] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2774–2781.
[26] Y. Chen, W. Yu, and T. Pock, "On learning optimized reaction diffusion processes for effective image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5261–5269.
[27] J. Domke, "Learning graphical model parameters with approximate marginal inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2454–2467, Oct. 2013.
[28] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-net for compressive sensing MRI," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 10–18.
[29] G. Riegler, D. Ferstl, M. Rüther, and H. Bischof, "A deep primal-dual network for guided depth super-resolution," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 7.1–7.14.
[30] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Model. Simul.*, vol. 7, no. 3, pp. 1005–1028, 2008.
[31] B. Ham, M. Cho, and J. Ponce, "Robust image filtering using joint static and dynamic guidance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4823–4831.
[32] S. Gu, W. Zuo, S. Guo, Y. Chen, C. Chen, and L. Zhang, "Learning dynamic guidance for depth image enhancement," *Analysis*, vol. 10, no. y2, 2017, Art. no. 2.
[33] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenom.*, vol. 60, no. 1, pp. 259–268, 1992.
[34] J.-L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 670–684, Jun. 2002.
[35] S. Roth and M. J. Black, "Fields of experts," *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 205–229, 2009.
[36] W. Zuo, D. Ren, S. Gu, L. Lin, and L. Zhang, "Discriminative learning of iteration-wise priors for blind deconvolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3232–3240.
[37] R. Rubinstein, T. Peleg, and M. Elad, "Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model," *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 661–677, Feb. 2013.
[38] S. Gu, D. Meng, W. Zuo, and L. Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1708–1716.
[39] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, 2007, Art. no. 947.
[40] S. C. Zhu, Y. Wu, and D. Mumford, "Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling," *Int. J. Comput. Vis.*, vol. 27, no. 2, pp. 107–126, 1998.
[41] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 228–242, Feb. 2008.
[42] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.
[43] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2009, pp. 693–696.
[44] Y. Li and S. Osher, "Coordinate descent optimization for l1 minimization with application to compressed sensing; a greedy algorithm," *Inverse Problems Imag.*, vol. 3, no. 3, pp. 487–503, 2009.
[45] E. Kobler, T. Klatzer, K. Hammernik, and T. Pock, "Variational networks: Connecting variational methods and deep learning," in *Proc. German Conf. Pattern Recognit.*, 2017, pp. 281–293.
[46] R. Liu, S. Cheng, Y. He, X. Fan, Z. Lin, and Z. Luo, "On the convergence of learning-based iterative methods for nonconvex inverse problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2019.2920591.
[47] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.
[48] Y. Chen, R. Ranftl, and T. Pock, "Insights into analysis operator learning: From patch-based sparse models to higher order MRFs," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1060–1072, Mar. 2014.
[49] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, no. 1–3, pp. 503–528, 1989.
[50] M. Schmidt, "minfunc," 2013. [Online]. Available: http://mloss.org/software/view/529/
[51] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.
[52] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2013, Art. no. 3.
[53] A. Paszke *et al.*, "Automatic differentiation in pytorch," in *Proc. Int. Conf. Neural Inf. Process. Syst. Autodiff Workshop*, 2017.
[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[55] H. Hirschmüller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

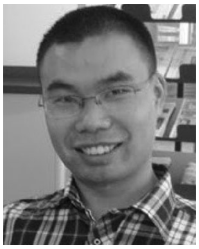
[56] R. Timofte, R. Rothe, and L. Van Gool, "Seven ways to improve example-based single image super resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1865–1873.
[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[58] Q. Yang, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[59] J. Yang, X. Ye, K. Li, C. Hou, and Y. Wang, "Color-guided depth recovery from RGB-D data using an adaptive auto-regressive model," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3443–3458, Aug. 2014.
[60] B. Ham, M. Cho, and J. Ponce, "Robust image filtering using joint static and dynamic guidance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4823–4831.
[61] J. T. Barron and B. Poole, "The fast bilateral solver," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 617–632.
[62] C.-Y. Yang, C. Ma, and M.-H. Yang, "Single-image super-resolution: A benchmark," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 372–386.
[63] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2862–2869.
[64] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," *Int. J. Comput. Vis.*, vol. 121, pp. 183–208, 2017.
[65] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 664–672, 2004.
[66] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Symp. Comput. Statist.*, 2010, pp. 177–186.
[67] S. Hochreiter *et al.*, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," 2001.



Shuhang Gu received the BE degree from the School of Astronautics, Beihang University, Beijing, China, in 2010, the ME degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 2013, and the PhD degree from the Department of Computing, Hong Kong Polytechnic University, in 2017. He currently holds a postdoctoral position with ETH Zurich, Switzerland.

Shi Guo received the BE degree from the School of Astronautics, Harbin Institute of Technology, Harbin, China, in 2017. He is currently working toward the graduate degree in the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China.

Wangmeng Zuo received the PhD degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in 2007. He is currently a professor with the School of Computer Science and Technology, Harbin Institute of Technology. His current research interests include image enhancement and restoration, image and face editing, object detection, visual tracking, and image classification. He has published more than 70 papers in toptier academic journals and conferences. He has served as a tutorial organizer in ECCV 2016, an associate editor of the *IET Biometrics* and *Journal of Electronic Imaging*.



Yunjin Chen received the BSc degree in applied physics from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2007, the MSc degree in optical engineering from the National University of Defense Technology, Haigang, China, in 2010, and the PhD degree in computer science from the Graz University of Technology, Graz, Austria, in 2015. His current research interests include learning image prior model for low-level computer vision problems and convex optimization.

Radu Timofte received the PhD degree in electrical engineering from the KU Leuven, Leuven, Belgium, in 2013. He is currently the group leader and lecturer with ETH Zurich, Switzerland. He serves as a reviewer for top journals and conferences, is area editor of the *Elsevier's Computer Vision and Image Understanding journal* and served as area chair for ACCV 2018 and ICCV 2019. His work received several awards. He is a co-founder of Merantix and a co-organizer of NTIRE, CLIC, PIRM, and AIM events. His current research interests include deep learning, implicit models, compression, image restoration, and enhancement.

Luc Van Gool received the degree in electromechanical engineering from the Katholieke Universiteit Leuven, Leuven, Belgium, in 1981. Currently, he is a professor with the Katholieke Universiteit Leuven in Belgium and the ETH in Zurich, Switzerland. He leads computer vision research at both places, where he also teaches computer vision. He has authored more than 200 papers in this field. He has been a program committee member of several major computer vision conferences. His main interests include 3D reconstruction and modeling, object recognition, tracking, and gesture analysis. He received several best paper awards. He is a co-founder of five spin-off companies.

Lei Zhang (M04–SM14–F18) received the PhD degrees in control theory and engineering from Northwestern Polytechnical University, Xi'an, P.R. China, in 2001. In 2006, he joined the Department of Computing, The Hong Kong Polytechnic University, as an assistant professor. Since July 2017, he has been a chair professor with the same department. His research interests include computer vision, image and video analysis, pattern recognition, biometrics, etc. He has published more than 200 papers in these areas. As of 2019, his publications have been cited more than 46,000 times in literature. He is a senior associate editor of the *IEEE Transactions on Image Processing*, and is/was an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *SIAM Journal of Imaging Sciences*, *IEEE Transactions on Circuits and Systems for Video Technology*, and *Image and Vision Computing*, etc. He is a Clarivate Analytics Highly Cited Researcher from 2015 to 2019. He is a fellow of the IEEE. More information can be found in his homepage http://www4.comp.polyu.edu.hk/ cslzhang/.