RADAR: Robust Algorithm for Depth Image Super Resolution Based on FRI Theory and Multimodal Dictionary Learning

Xin Deng, *Student Member, IEEE*, Pingfan Song, *Student Member, IEEE*, Miguel R. D. Rodrigues, *Senior Member, IEEE*, Pier Luigi Dragotti, *Fellow, IEEE*

***Abstract*—Depth image super-resolution is a challenging problem, since normally high upscaling factors are required (e.g., $16\times$), and depth images are often noisy. In order to achieve large upscaling factors and resilience to noise, we propose a Robust Algorithm for Depth imAge super Resolution (RADAR) that combines the power of finite rate of innovation (FRI) theory with multimodal dictionary learning. Given a low-resolution (LR) depth image, we first model its rows and columns as piecewise polynomials and propose a FRI-based depth upscaling (FDU) algorithm to super-resolve the image. Then, the upscaled moderate quality (MQ) depth image is further enhanced with the guidance of a registered high-resolution (HR) intensity image. This is achieved by learning multimodal mappings from the joint MQ depth and HR intensity pairs to the HR depth, through a recently proposed triple dictionary learning (TDL) algorithm. Moreover, to speed up the super-resolution process, we introduce a new projection-based rapid upscaling (PRU) technique that pre-calculates the projections from the joint MQ depth and HR intensity pairs to the HR depth. Compared with state-of-the-art deep learning based methods, our approach has two distinct advantages: we need a fraction of training data but can achieve the best performance, and we are resilient to mismatches between training and testing datasets. Extensive numerical results show that the proposed method outperforms other state-of-the-art methods on either noise-free or noisy datasets with large upscaling factors up to $16\times$ and can handle unknown blurring kernels well.**

***Index Terms*—Depth image super-resolution, finite rate of innovation, multimodal image processing.**

I. Introduction

High quality depth images play an important role in many computer vision applications, such as image segmentation [1], [2], 3D object reconstruction [3] and gesture recognition [4]. However, the fast acquisition of accurate and dense depth data is, in practice, difficult to achieve. Recently, time-of-flight (ToF) cameras have become popular in both academic and industrial communities, because they can work in real-time by capturing the depth data through measuring the phase-delay of reflected infrared light [5]. Despite this merit, ToF camera can only provide low resolution depth images, and this fact affects the progress of many related research areas.

Xin Deng and Pier Luigi Dragotti are with the Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, London, the United Kingdom. (e-mail: x.deng16@imperial.ac.uk; p.dragotti@imperial.ac.uk). Pingfan Song and Migual R. D. Rodrigues are with the department of Electronic and Electrical Engineering, University College London, WC1E 6BT, London, the United Kingdom. (e-mail: pingfan.song.14@ucl.ac.uk, m.rodrigues@ucl.ac.uk). Xin Deng is supported by the CSC Imperial Scholarship.

Depth image super-resolution aims to recover a high-resolution (HR) depth image from a low-resolution (LR) one. This is a problem similar to the more studied one of color image super-resolution. However, compared with the color image case, the depth image super-resolution problem has distinct challenges. First of all, the depth images captured by cameras like ToF are usually at very low resolution, e.g., less than 1/4 that of the color image, which means a large upscaling factor of, e.g., $8\times$ and $16\times$, is required, and when the upscaling factor increases, the mapping between the LR and HR counterpart becomes more difficult. Secondly, the depth images are usually more affected by noise than color images due to the capturing process. Thirdly, unlike color images, it is sometimes more difficult to find a large training dataset for depth images. Without sufficient training data, the ability of most learning based methods [6], [7] can be significantly affected. Finally, for intensity guided depth super-resolution, a new challenge comes in detecting the correlations between the two modalities, i.e., depth and intensity.

Ideally, a good depth image super-resolution method should be able to solve all the above problems. However, most of the existing methods only address some of them. For example, Xie *et al.* [8] proposed to enhance the depth image resolution by inferring the HR edges, but it can handle only clean inputs with small upscaling factors, i.e., less than $4\times$. Riegler *et al.* [6] proposed ATGV-Net combining deep convolutional network with a variational method to learn the mappings from LR depths to HR counterparts, which can handle noisy depth images but still struggles with upscaling factors larger than $4\times$. Recently, Song *et al.* [7] proposed a progressive deep convolutional neural network (CNN) structure to reconstruct HR depth images through gradually learning the high frequencies with the color image as guidance. This work can cope with a larger upscaling factor, i.e., $8\times$, but is not resilient to noise. Besides, both [6] and [7] require a large dataset for network training.

In this paper, we propose a Robust Algorithm for Depth imAge super Resolution (RADAR) that combines the power of finite rate of innovation (FRI) theory [9], [10] and multimodal dictionary learning [11], to achieve depth image super-resolution at very high upscaling factors with a small training dataset. Moreover, the proposed method can handle noisy depth images and is resilient to mismatches between training and testing datasets. The method is composed of two stages. In the first stage, we use FRI theory to upscale the

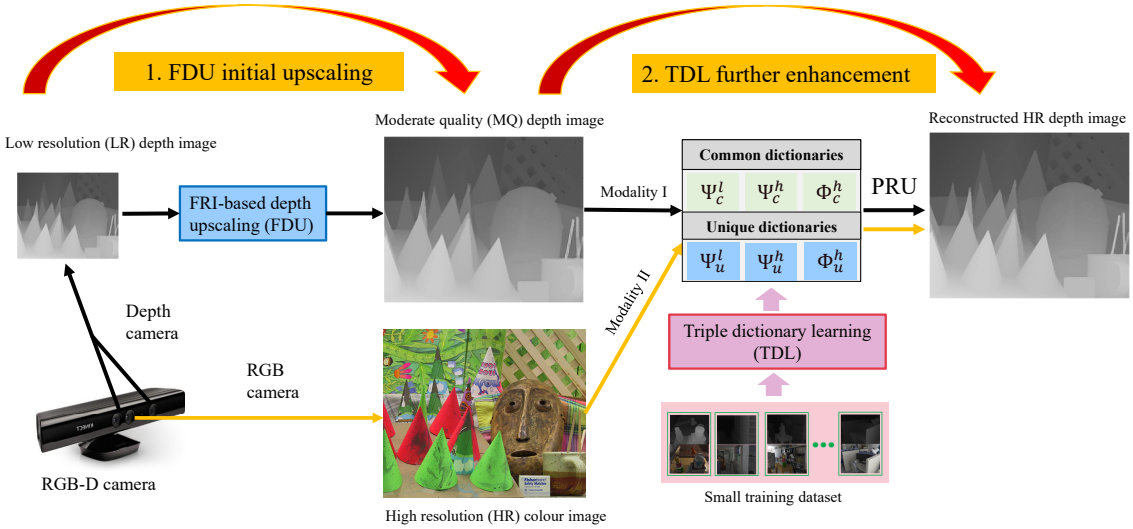


Fig. 1. The cascaded framework of the proposed RADAR approach. The first stage is the FRI based upscaling (FDU) algorithm which uses FRI theory to do initial upscaling, and the second stage is projection-based rapid upscaling (PRU) algorithm based on triple dictionary learning (TDL) model, which extracts the useful information of colour/intensity image to further enhance the depth image.

LR depth image to a moderate quality (MQ) depth image. This stage does not require intensity image. Then, in the second stage, we use HR intensity image as guidance to further improve the quality of the MQ depth image. This is achieved by proposing a projection-based rapid upscaling (PRU) algorithm based on the triple dictionary learning (TDL) model. The cascaded framework is shown in Fig. 1.

The main contributions of this paper are as follows:

- We propose to use FRI theory to upscale the depth image with no external training dataset. Based on the observation that rows and columns of depth images can be approximately modeled as piece-wise polynomials and FRI theory can perfectly reconstruct this kind of signals, we propose a FRI-based depth upscaling (FDU) algorithm to super-resolve the depth image. The proposed method can handle both noise-free and noisy depth images. Besides, leveraging the ability of FRI theory to handle any blurring kernel, we can upscale depth images blurred with arbitrary blurring functions.
- Leveraging from the triple dictionary learning (TDL) model we proposed in [11], we develop a projection-based rapid upscaling (PRU) algorithm, to speed up the multimodal depth image super-resolution process. In particular, the multimodal training samples are classified into different subsets, and for each subset, we learn a projection from the joint MQ depth and HR intensity pairs to the HR depth. With the projections pre-calculated and stored, the sparse coding process can be skipped so that the upscaling process is accelerated.
- We test the effectiveness of our approach on various datasets, including both synthetic and real-world datasets with both noise-free and noisy scenerios. The robustness of our approach is also demonstrated with different blurring kernels. Numerical results show that our method outperforms other state-of-the-art methods, especially at large upscaling factors, e.g., $8\times$, $16\times$.

The combined use of a model-based algorithm (i.e., FDU) followed by a data-driven algorithm (i.e., TDL) has several advantages. The FDU algorithm can elevate the depth image quality to a higher level, so that the mapping ambiguity between LR and HR versions can be drastically reduced, and this allows the TDL algorithm to operate properly for large upscaling factors as well. Moreover, since parts of high frequency details have already been recovered by FDU, only a small training dataset is required in the TDL training process and we do not need complex deep neural networks to achieve state-of-the-art performance. All this, combined with the proposed PRU algorithm, leads to a fast, robust and competitive method for depth image super-resolution.

The remainder of this paper is organized as follows. Section II reviews the related work about depth image super-resolution. The proposed method is introduced in Section III. Section IV presents the experimental results and finally Section V concludes this paper.

II. Related Work

Depth image super-resolution (DISR) approaches can be broadly classified into three categories, single DISR approaches [6], [8], [12]–[16] which require only LR depth images, depth prediction approaches [17]–[21] which require only HR intensity images, and intensity guided DISR approaches [22]–[26], [26]–[29] which use both the LR depth and HR intensity images. Many ideas in single DISR are based on the success of single color image super-resolution (CISR) algorithms which we now briefly review.

Single CISR approaches aim to recover a HR color image from a single LR color image. The most successful algorithms are based on some forms of learning, and consist of two stages: training and upscaling. In the training stage, the mappings between LR and HR counterparts are learned from the training LR-HR pairs, and the upscaling stage uses the learned mappings to super-resolve the LR images. Representative

works include sparse coding [30], K-SVD [31], A+ [32], SelfEx [33], and random forests [34]. The representative works when deep learning is used include SRCNN [35], CSCN [36], VDSR [37], MSCN [38], and ESPCN [39]. The deep learning based methods usually show better reconstruction results, but they need huge training datasets, and are not always resilient to mismatches between training and testing datasets.

Single DISR approaches are usually derived from the single CISR methods with some variations. For example, Aodha *et al.* [12] extended the work of [40] to the depth image by using a Markov random field (MRF) labeling model. The LR depth image is divided into parts, and the HR depth image is assembled using the corresponding HR counterparts selected from an external dataset. The work of [13] further extended [12] by adding geometric constraints from self-similar structures. Considering that no texture is contained in depth images, Xie *et al.* [8] proposed an edge guided depth image super-resolution approach to reconstruct an HR edge map through a MRF optimization. The HR depth image was obtained using a bilateral filter with the HR edge map as guidance. In addition to the direct patch synthesis approaches, [14] and [15] learned coupled dictionaries for DISR under the assumption that LR and HR depth patches share the same reconstruction coefficients. More recently, Riegler *et al.* [6] combined deep convolutional networks with a variational model to super-resolve depth images. One big problem for the single DISR approaches is that they may struggle with large upscaling factors, e.g., $8\times$ and $16\times$, because the fine details of HR images may not be evident in the LR versions.

Depth prediction approaches aim to infer a HR depth image from a single HR color/intensity image. This is a highly ill-posed problem, because the intensity image contains no distance information which the depth image needs. Liu *et al.* [17] proposed to model the depth prediction problem as a discrete-continuous optimization problem. For a specific intensity image, several similar images are gathered with known depth information, which are used to predict the depth information for the target intensity image. Eigen *et al.* [18] proposed a multi-scale convolutional network which integrates the coarse-scale depth prediction with the fine-scale prediction. Li *et al.* [19] proposed to estimate the depth at the super-pixel level with a trained CNN network, and then use conditional random fields (CRF) to refine the depth at the pixel level. These methods are all based on supervised learning, which needs a large training dataset. Recently, Kuznietsov *et al.* [21] proposed to use semi-supervised learning, which employed a deep residual network in an encoder-decoder architecture. Compared to the single DISR approaches, the depth prediction approaches have an obvious disadvantage, since the intensity image cannot provide specific depth cues, and this makes the depth prediction accuracy far from satisfactory.

Intensity guided DISR approaches use a registered HR intensity image to assist the super-resolution of a depth image. Intensity and depth images are two modalities of the same scene, so they should be correlated. Different approaches have been proposed to exploit this correlation. For example, with the assumption that pixels around a region with similar colors tend to have similar depth values, Yang *et al.* [22] used joint bilateral filtering to iteratively interpolate depth values in HR depth image. Chan *et al.* [23] extended this work by introducing a noise-aware bilateral filter that alternates between standard upsampling and joint bilateral filtering based on the local statistics. Later, He *et al.* [25] proposed a color image guided filter to preserve the fine edges in the reconstructed depth image, based on an assumption that joint occurrence exists between depth discontinuities and color image edges. Lu *et al.* [27] also adopted this assumption. They first split the HR color image into parts using image segmentation and then independently predicted the depth values of each part using depth smoothing methods.

Different from the above model based approaches, the learning based approaches aim to learn the correlation through training. For example, Tosic *et al.* [41] proposed to learn joint over-complete dictionaries for intensity and depth modalities and used a joint basis pursuit algorithm to find the sparse coefficients. Kwon *et al.* [28] also used dictionary learning to upscale the depth image. They realized that an edge in a color image sometimes does not necessarily lead to a depth discontinuity and thus proposed a RGB-D structure similarity measure to predict the consistency between edges and discontinuities. More recently, Gu *et al.* [29] proposed a stage-by-stage intensity guided depth upscaling algorithm, based on a weighted analysis representation model. Dynamic guidance is learned for each stage through a task-driven learning strategy. Song *et al.* [7] proposed to use convolutional neural network to upscale a depth image gradually, with the statistics of intensity image as guidance.

In this paper, we also use dictionary learning to explore the dependency between depth and intensity modalities. However, different from [28], [41] and in line with [11], [42], we represent each modality by two dictionaries which are learned through a proposed multimodal dictionary learning algorithm. In particular, one dictionary represents the information which is common between depth and intensity, while the other represents the information that is unique to each modality. In this way, the unrelated information in intensity images cannot affect the depth image upscaling, which can help improve the depth reconstruction accuracy.

III. Proposed methods

In this section, the main elements of our approach are introduced. In Section III-A, the FRI-based depth upscaling (FDU) algorithm is introduced, and in Section III-B, we review the triple dictionary learning (TDL) model. Then, based on the learned dictionaries, a novel projection-based rapid upscaling (PRU) algorithm is introduced in Section III-C.

A. FRI-based Depth Upscaling (FDU)

In this section, we introduce a model-based method benefiting from the finite rate of innovation (FRI) theory to upscale a LR depth image. This method does not require the use of external datasets. FRI theory has shown that it is possible to reconstruct perfectly piecewise polynomial signals from samples obtained with an arbitrary blurring kernel that might also be the scaling function in a wavelet decomposition

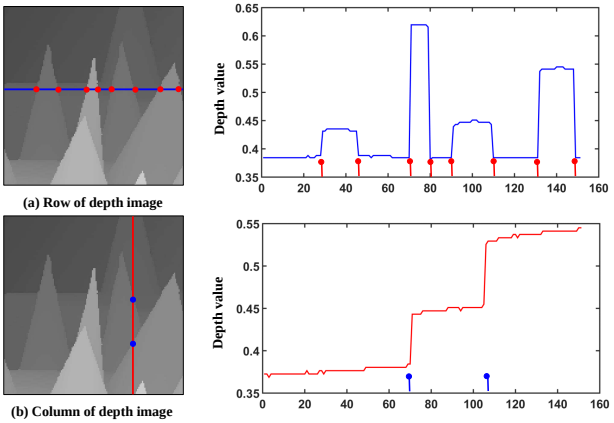


Fig. 2. Illustrations of the rows and columns in a depth image. Each row and column can be approximately modeled as a piece-wise polynomial function with switch points indicated by red and blue dots.

[10]. As observed from Fig. 2, the rows and columns of depth images are quite close to a 1-D piece-wise polynomial function. Thus, for each row and column, given its low-pass version, we can reconstruct the original line using FRI theory [9], [43]. For more details of FRI reconstruction, please see Appendix. In the case of depth image super-resolution, the key insight is that we treat the LR image as the lowpass version of a wavelet decomposition of the HR image, and use FRI theory to infer the missing wavelet coefficients. As shown in Fig. 3, the HR image can be decomposed into four sub-bands using 2D Discrete Wavelet Transform (DWT) [44], i.e., LL (smooth approximation), LH (horizontal details), HL (vertical details), and HH (diagonal details). We assume the LR image is the LL sub-band, and then the super-resolution process is equivalent to retrieving the other three sub-bands, and this can be achieved using FRI theory.

For retrieving the LH subband which contains the horizontal details, we first perform linear interpolation horizontally on the LR image, and then perform FRI reconstruction vertically column by column, to obtain a HR image with horizontal details which we denote as HR_h. The LH sub-band, LH^h, can then be extracted from HR_h using 2D DWT[1]. Likewise, we reconstruct HR_v by doing linear interpolation vertically and then FRI reconstruction horizontally. The HL sub-band, HL^v, can be extracted from HR_v using 2D DWT. Since both HR_h and HR_v contain the diagonal details, we can perform 2D DWT on either of them, or use the average of them to retrieve the HH sub-band. We find that there is no noticeable performance change among these three solutions. In this paper, we simply use HH^v, i.e., the 2D DWT on HR_v, to get the diagonal details. Fig. 4 illustrates this process for 2 $\times$ upcaling. When the upscaling factor is larger than 2, e.g., 4$\times$, we perform a 2$\times$ upscaling followed by another 2$\times$ upscaling. The same cascaded strategy applies for 8 $\times$ and 16 $\times$ upscaling.

Noise-free case. With the retrieved sub-bands (i.e., LH^h, HL^v, HH^v) and the clean LR depth image, we use 2D inverse DWT (IDWT) to combine them to obtain a super-resolved HR image. Since the depth line is not exactly a piece-wise polynomial signal, the estimated discontinuity locations may contain some errors. In order to achieve a better quality, following [45], we employ an internal self-learning algorithm proposed in [46] to correct these errors caused by FRI reconstruction. The insight is to establish an internal LR-HR dictionary through a pyramid of recursively downscaled and FDU upscaled images, and then learn a linear mapping from LR to HR patches. For more details about internal self-learning, we refer to [45], [46].

[1]Here, since only parts of the sub-bands are required, partial 2D DWT is enough if considering the computational complexity.

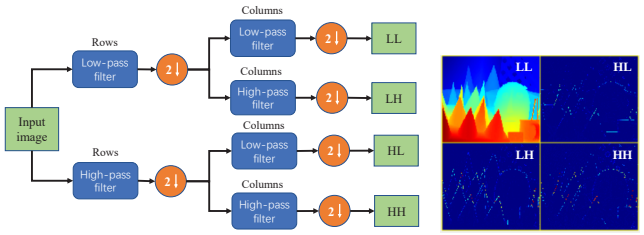


Fig. 3. (a) shows the decomposition of one image using 2D DWT, and (b) visualizes the four sub-bands of image *Cones*.

TABLE I
NOTATION LIST.

LL:	Input noise-free/noisy LR image
HR_h:	Restored HR image with horizontal details with LL as input
HR_v:	Restored HR image with vertical details with LL as input
LL^h:	LL subband extracted from HR_h using 2D DWT
LH^h:	LH subband extracted from HR_h using 2D DWT
LL^v:	LL subband extracted from HR_v using 2D DWT
HL^v:	HL subband extracted from HR_v using 2D DWT
HR_h^h:	Restored HR image with horizontal details with LL^h as input
HR_v^h:	Restored HR image with vertical details with LL^h as input
LH_h^h:	LH subband extracted from HR_h^h using 2D DWT
LL_v^h:	LL subband extracted from HR_v^h using 2D DWT
HL_v^h:	HL subband extracted from HR_v^h using 2D DWT
HR_h^v:	Restored HR image with horizontal details with LL^v as input
HR_v^v:	Restored HR image with vertical details with LL^v as input
LL_h^v:	LL subband extracted from HR_h^v using 2D DWT
LH_h^v:	LH subband extracted from HR_h^v using 2D DWT
HL_v^v:	HL subband extracted from HR_v^v using 2D DWT

Noisy case. In the case of noisy LR images, the prediction algorithm for the three high-frequency sub-bands is the same as that in the noise-free case, and we denote it as the basic FDU unit. However, since the LR image is noisy, combining it with the other three sub-bands would lead to a noisy HR output. The intuitive solution is to find a denoised version of the noisy LR input and we have two choices: the LL^h extracted from HR_h or the LL^v extracted from HR_v. The problem is that LL^h is only denoised along horizontal lines but still noisy along vertical lines, and conversely, LL^v is only denoised along vertical lines but still noisy along horizontal lines. To tackle this issue, our solution is to reconstruct the sub-bands with LL^h and LL^v as the LR inputs, respectively. Then, we average these sub-bands and perform 2D IDWT on the averaged sub-bands to obtain the final HR image. For clarity, the notations we use are listed in Table I and the algorithm is shown in Fig. 5.

As it can be seen from the figure, the algorithm consists of

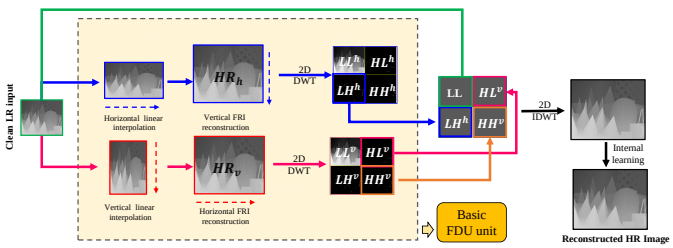


Fig. 4. FDU algorithm with clean LR depth image.

two branches: the upper branch reconstructs four sub-bands with LL^h as input and the lower branch reconstructs four sub-bands with LL^v as input. We now focus on the upper branch to explain how it works. Given a noisy LR input, we reconstruct the HR_h image using the same approach as in the noise-free case, and extract LL^h from HR_h using 2D DWT. The LL^h is then fed into the basic FDU unit which leads to two reconstructed images: HR_h^h and HR_v^h. Similar to the noise-free case, we extract the LH subband LH_h^h from HR_h^h, the HL subband HL_v^h from HR_v^h, and HH subband from either of them. Since the LL^h subband is noisy vertically, we choose to use the LL_v^h extracted from HR_v^h instead, as shown in Fig. 5. In the lower branch, we use a similar approach to obtain four sub-bands with LL^v as input, i.e., LL_h^v, HL_v^v, LH_h^v and HH_v^v. The four sub-bands obtained by the upper and lower branches are averaged per sub-band, and then we perform 2D IDWT on the four averaged sub-bands to reconstruct the final HR image. Note that in the noisy case, the internal self-learning algorithm is not employed, because the ground-truth LR image is not available.

B. Triple Dictionary Learning (TDL)

The aforementioned FDU algorithm allows us to upscale the LR depth image to a moderate quality (MQ) version, without the guidance of intensity image. Next, to further enhance the MQ depth image, we use a recently proposed multi-modal dictionary learning algorithm [11] to fully use the guidance of HR intensity images.

The basic insight in our model [11] is that the MQ depth image, HR depth image, and the corresponding HR intensity image are from the same scene. Therefore, they share some latent features. However, only part of depth information is related to part of intensity information, while other elements are unrelated. For example, the edges in intensity images normally correspond to depth discontinuities, while the texture in intensity images has no relationship with the depth images. If they are not properly separated, texture elements may occur in the reconstructed depth, leading to texture copying artifacts. To avoid that, we model each modality with two dictionaries, one common dictionary (CD) and one unique dictionary (UD). Specifically, we denote by $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ the MQ depth patch, HR depth patch and HR intensity patch, respectively, and assume they share some common sparse features but also have unique elements. Suppose that $\boldsymbol{\Psi}_c^l$ and $\boldsymbol{\Psi}_u^l$ are the CD and UD of MQ depth modality, $\boldsymbol{\Psi}_c^h$ and $\boldsymbol{\Psi}_u^h$ are the CD and UD of HR depth modality, and $\boldsymbol{\Phi}_c^h$ and $\boldsymbol{\Phi}_u^h$ are the CD and UD of HR intensity modality, we then assume they have the following relationship:

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Psi}_c^l & \boldsymbol{\Psi}_u^l & \mathbf{0} \\ \boldsymbol{\Psi}_c^h & \boldsymbol{\Psi}_u^h & \mathbf{0} \\ \boldsymbol{\Phi}_c^h & \mathbf{0} & \boldsymbol{\Phi}_u^h \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \end{bmatrix}, \tag{1}$$

where $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ are the sparse coefficients. For $\boldsymbol{y}$ and $\boldsymbol{z}$, their CDs share the same sparse coefficient $\mathbf{u}$, while the UDs have distinct sparse coefficients $\mathbf{v}$ and $\mathbf{w}$. This is because only the common information in $\boldsymbol{z}$ is useful for the reconstruction of $\boldsymbol{y}$. For $\boldsymbol{x}$ and $\boldsymbol{y}$, both the CDs and UDs share the same sparse coefficients, because all the information in $\boldsymbol{x}$ is useful for the reconstruction of $\boldsymbol{y}$.

The triple dictionary training problem can then be formulated as [11]

$$\begin{aligned} \{\boldsymbol{C}, \boldsymbol{D}\} = \underset{\boldsymbol{C},\boldsymbol{D}}{\operatorname{argmin}} & \left\| \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Psi}_c^l & \boldsymbol{\Psi}_u^l & \mathbf{0} \\ \boldsymbol{\Psi}_c^h & \boldsymbol{\Psi}_u^h & \mathbf{0} \\ \boldsymbol{\Phi}_c^h & \mathbf{0} & \boldsymbol{\Phi}_u^h \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \\ \mathbf{W} \end{bmatrix} \right\|_F^2, \\ s.t. \quad & \|\boldsymbol{u}_k\|_0 + \|\boldsymbol{v}_k\|_0 + \|\boldsymbol{w}_k\|_0 \le L, \quad \forall k, \end{aligned} \tag{2}$$

where $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z} \in \mathbb{R}^{M\times P}$ are the training MQ depth features, HR depth features and HR intensity features respectively collected from the training dataset. $\boldsymbol{D}$ represents all the six dictionaries (each having size $\mathbb{R}^{M\times Q}$) and $\boldsymbol{C}$ represents all the sparse coefficients $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{W} \in \mathbb{R}^{Q\times P}$. Here, M is the size of each feature, Q is the number of atoms in each dictionary, and P is the number of training features. Note that in theory the common dictionaries and unique dictionaries can have different number of atoms, but in this paper we just set them to be the same. In (2), $\boldsymbol{u}_k$, $\boldsymbol{v}_k$, $\boldsymbol{w}_k$ are the k-th ($0 < k \le P$) columns of matrices $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{W}$ respectively, $\|\cdot\|_0$ is the ℓ_0 norm and L is the sparsity constraint. Since (2) is non-convex, it is difficult to get a solution for all the six dictionaries simultaneously. Thus, we first relax (2) to (3) by disregarding dictionaries $\boldsymbol{\Psi}_c^h$ and $\boldsymbol{\Psi}_u^h$ temporarily, and then we have

$$\begin{aligned} \{\boldsymbol{C}, \boldsymbol{D'}\} = \underset{\mathbf{C},\mathbf{D'}}{\operatorname{argmin}} & \left\| \begin{bmatrix} \mathbf{X} \\ \mathbf{Z} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Psi}_c^l & \boldsymbol{\Psi}_u^l & \mathbf{0} \\ \boldsymbol{\Phi}_c^h & \mathbf{0} & \boldsymbol{\Phi}_u^h \end{bmatrix} \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{V} \\ \boldsymbol{W} \end{bmatrix} \right\|_F^2, \\ s.t. \quad & \|\boldsymbol{u}_k\|_0 + \|\boldsymbol{v}_k\|_0 + \|\boldsymbol{w}_k\|_0 \le L, \quad \forall k, \end{aligned} \tag{3}$$

where $\boldsymbol{D'}$ represent the four dictionaries in (3). The four dictionaries are initialized using DCT frames. We use two steps to solve this problem. In the first step, we fix both the common and unique dictionaries and use orthogonal matching pursuit (OMP) algorithm [47] to calculate the sparse coefficients, and then by fixing the coefficients and the unique dictionaries, we use K-SVD [48] to update the common dictionaries. This process is repeated until convergence. In the second step, we fix the learned coefficients and the common dictionaries, and use K-SVD to update the unique dictionaries. This process is also repeated until convergence. Then, the first step and second step are carried out in an alternate way until all the dictionaries converge. Once we have obtained the dictionaries $\boldsymbol{\Psi}_c^l$, $\boldsymbol{\Psi}_u^l$, $\boldsymbol{\Phi}_c^h$, $\boldsymbol{\Phi}_u^h$ and the sparse coefficients $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{W}$, the next step is to learn the remaining dictionaries $\boldsymbol{\Psi}_c^h$ and $\boldsymbol{\Psi}_u^h$. From (2), we can observe that the HR depth dictionaries $\boldsymbol{\Psi}_c^h$ and $\boldsymbol{\Psi}_u^h$ share the same sparse coefficients,

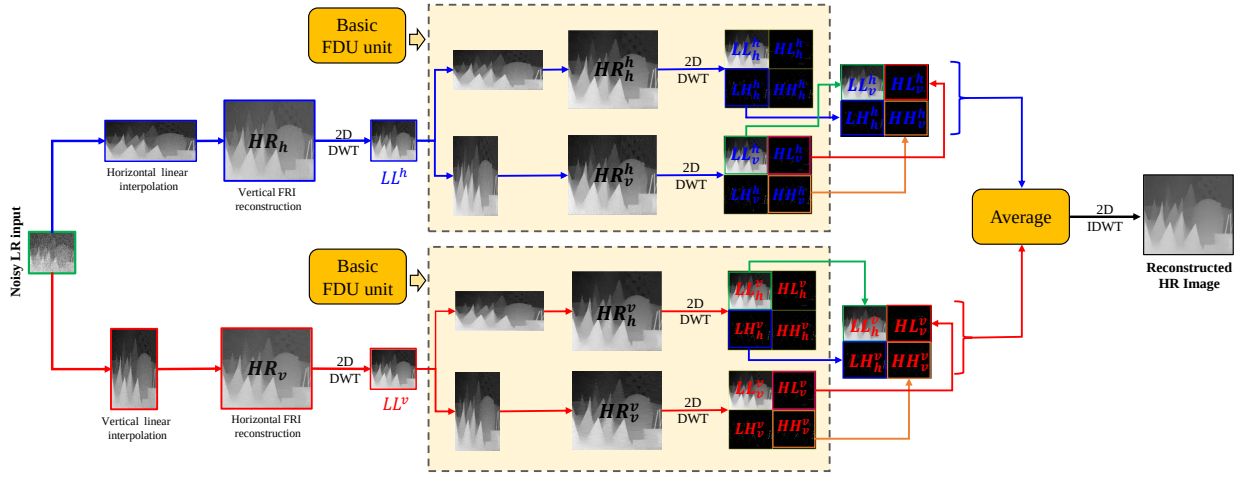


Fig. 5. FDU algorithm with noisy LR depth image.

i.e., $\boldsymbol{U}$ and $\boldsymbol{V}$, with the MQ depth dictionaries $\boldsymbol{\Psi}_c^l$ and $\boldsymbol{\Psi}_u^l$. With these sparse coefficients, we can obtain $\boldsymbol{\Psi}_c^h$ and $\boldsymbol{\Psi}_u^h$ by minimizing the total reconstruction error of all the HR depth training features $\boldsymbol{Y}$:

$$\{\boldsymbol{\Psi}_c^h, \boldsymbol{\Psi}_u^h\} = \underset{\boldsymbol{\Psi}_c^h, \boldsymbol{\Psi}_u^h}{\operatorname{argmin}} \left\| \boldsymbol{Y} - \begin{bmatrix} \boldsymbol{\Psi}_c^h & \boldsymbol{\Psi}_u^h \end{bmatrix} \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{V} \end{bmatrix} \right\|_F^2 + \lambda \left\| \begin{bmatrix} \boldsymbol{\Psi}_c^h & \boldsymbol{\Psi}_u^h \end{bmatrix} \right\|_F^2. \tag{4}$$

By solving this least square fitting problem, we have dictionaries $\boldsymbol{\Psi}_c^h$ and $\boldsymbol{\Psi}_u^h$ as follows,

$$\begin{bmatrix} \boldsymbol{\Psi}_c^h & \boldsymbol{\Psi}_u^h \end{bmatrix} = \boldsymbol{Y} \begin{bmatrix} \boldsymbol{U}^T & \boldsymbol{V}^T \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{V} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}^T & \boldsymbol{V}^T \end{bmatrix} + \lambda \boldsymbol{I} \right)^{-1}, \tag{5}$$

where λ is the regularization parameter and $\boldsymbol{I}$ is the identity matrix.

C. Projection-based Rapid Upscaling (PRU)

In the upscaling process, given a MQ depth feature $\boldsymbol{x}_t$ and the corresponding HR intensity feature $\boldsymbol{z}_t$, we aim to retrieve the HR depth feature $\boldsymbol{y}_t$. To this end, we need to calculate the sparse coefficients first by solving the following optimization,

$$\underset{\mathbf{u},\mathbf{v},\mathbf{w}}{\operatorname{argmin}} \|\mathbf{u}\|_0 + \|\mathbf{v}\|_0 + \|\mathbf{w}\|_0, s.t., \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{z}_t \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Psi}_c^l & \boldsymbol{\Psi}_u^l & \mathbf{0} \\ \boldsymbol{\Phi}_c^h & \mathbf{0} & \boldsymbol{\Phi}_u^h \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \end{bmatrix}. \tag{6}$$

In theory, one can solve (6) by OMP algorithm [47] to obtain the coefficients $\mathbf{u}, \mathbf{v}, \mathbf{w}$. After that, $\boldsymbol{y}_t$ can be recovered as

$$\boldsymbol{y}_t = \boldsymbol{\Psi}_c^h \mathbf{u} + \boldsymbol{\Psi}_{\mathbf{u}}^{\mathbf{h}} \mathbf{v}. \tag{7}$$

Unfortunately, the OMP algorithm is quite time-consuming. To overcome this drawback, we propose a projection-based rapid upscaling (PRU) algorithm. The key insight is to precalculate the projections from the joint MQ depth and HR intensity feature pairs to HR depth feature before upscaling. This algorithm is composed of two important parts: sub-dictionary learning and sub-projection learning.

Sub-dictionary learning. Suppose that $\{\mathbf{X}; \mathbf{Z}\}$ are the training data pairs related to MQ depth and HR intensity features, and $\{\mathbf{Y}\}$ are the training data with HR depth features. Firstly, we use k-means clustering algorithm [49] to split $\{\mathbf{X}; \mathbf{Z}\}$ into N clusters and store the centroids $\{\boldsymbol{O}_i\}_{i=1}^N$ for each cluster. Then, for each centroid, we use k-nearest neighbour (kNN) to search for its K nearest samples in $\{\mathbf{X}; \mathbf{Z}\}$ to form subsets $\{\boldsymbol{X}^{(i)}; \boldsymbol{Z}^{(i)}\}_{i=1}^N$. Note that the subsets may have some samples in common. For each sample in the i-th subset $\{\boldsymbol{X}^{(i)}; \boldsymbol{Z}^{(i)}\}$, its sparse coefficients with respect to the dictionaries learned before can be calculated by solving (6) using OMP algorithm. Specifically, for the k-th ($0 < k \leq K$) sample in $\{\boldsymbol{X}^{(i)}; \boldsymbol{Z}^{(i)}\}$, its sparse coefficients are denoted as $[\boldsymbol{u}_k^{(i)}; \boldsymbol{v}_k^{(i)}; \boldsymbol{w}_k^{(i)}]$. By stacking together the sparse coefficients of all samples in each subset separately, we obtain three coefficient matrices per subset as follows,

$$\begin{aligned} \boldsymbol{M}_u^{(i)} &= \left[\boldsymbol{u}_1^{(i)}, \boldsymbol{u}_2^{(i)}, \ldots, \boldsymbol{u}_K^{(i)}\right]; \\ \boldsymbol{M}_v^{(i)} &= \left[\boldsymbol{v}_1^{(i)}, \boldsymbol{v}_2^{(i)}, \ldots, \boldsymbol{v}_K^{(i)}\right]; \\ \boldsymbol{M}_w^{(i)} &= \left[\boldsymbol{w}_1^{(i)}, \boldsymbol{w}_2^{(i)}, \ldots, \boldsymbol{w}_K^{(i)}\right]. \end{aligned} \tag{8}$$

Based on the above coefficient matrices, we aim to compute the sub-dictionaries $\{\boldsymbol{\Psi}_c^{l(i)}, \boldsymbol{\Psi}_u^{l(i)}, \boldsymbol{\Psi}_c^{h(i)}, \boldsymbol{\Psi}_u^{h(i)}, \boldsymbol{\Phi}_c^{h(i)}, \boldsymbol{\Phi}_u^{h(i)}\}$ of the i-th subset.

For simplicity, we denote $\boldsymbol{\Psi}^{(i)} = \begin{bmatrix} \boldsymbol{\Psi}_c^{l(i)} & \boldsymbol{\Psi}_u^{l(i)} \\ \boldsymbol{\Psi}_c^{h(i)} & \boldsymbol{\Psi}_u^{h(i)} \end{bmatrix}$, and $\boldsymbol{\Phi}^{(i)} = \begin{bmatrix} \boldsymbol{\Phi}_c^{h(i)} & \boldsymbol{\Phi}_u^{h(i)} \end{bmatrix}$, and we can obtain the sub-dictionaries by solving the following optimizations:

$$\boldsymbol{\Psi}^{(i)} = \underset{\boldsymbol{\Psi}^{(i)}}{\operatorname{argmin}} \left\| \begin{bmatrix} \boldsymbol{X}^{(i)} \\ \boldsymbol{Y}^{(i)} \end{bmatrix} - \boldsymbol{\Psi}^{(i)} \begin{bmatrix} \boldsymbol{M}_u^{(i)} \\ \boldsymbol{M}_v^{(i)} \end{bmatrix} \right\|_F^2 + \lambda \left\| \boldsymbol{\Psi}^{(i)} \right\|_F^2, \tag{9}$$

and

$$\boldsymbol{\Phi}^{(i)} = \underset{\boldsymbol{\Phi}^{(i)}}{\operatorname{argmin}} \left\| \boldsymbol{Z}^{(i)} - \boldsymbol{\Phi}^{(i)} \begin{bmatrix} \boldsymbol{M}_u^{(i)} \\ \boldsymbol{M}_w^{(i)} \end{bmatrix} \right\|_F^2 + \lambda \left\| \boldsymbol{\Phi}^{(i)} \right\|_F^2. \tag{10}$$

The above optimization problems can be solved through least square fitting, with the solutions as follows,

$$\boldsymbol{\Psi}^{(i)} = \begin{bmatrix} \boldsymbol{X}^{(i)} \\ \boldsymbol{Y}^{(i)} \end{bmatrix} \left[\boldsymbol{M}_u^{(i)T} \boldsymbol{M}_v^{(i)T} \right] \left(\begin{bmatrix} \boldsymbol{M}_u^{(i)} \\ \boldsymbol{M}_v^{(i)} \end{bmatrix} \left[\boldsymbol{M}_u^{(i)T} \boldsymbol{M}_v^{(i)T} \right] + \lambda \boldsymbol{I} \right)^{-1},$$

and

$$\boldsymbol{\Phi}^{(i)} = \boldsymbol{Z}^{(i)} \left[\boldsymbol{M}_u^{(i)T} \boldsymbol{M}_w^{(i)T} \right] \left(\begin{bmatrix} \boldsymbol{M}_u^{(i)} \\ \boldsymbol{M}_w^{(i)} \end{bmatrix} \left[\boldsymbol{M}_u^{(i)T} \boldsymbol{M}_w^{(i)T} \right] + \lambda \boldsymbol{I} \right)^{-1}, \tag{11}$$

where λ is the regularization parameter. Finally, after we calculate $\boldsymbol{\Psi}^{(i)}$ and $\boldsymbol{\Phi}^{(i)}$, the sub-dictionaries for the i-th subset $\{\boldsymbol{\Psi}_c^{l(i)}, \boldsymbol{\Psi}_u^{l(i)}, \boldsymbol{\Psi}_c^{h(i)}, \boldsymbol{\Psi}_u^{h(i)}, \boldsymbol{\Phi}_c^{h(i)}, \boldsymbol{\Phi}_u^{h(i)}\}$ can be easily recovered from $\boldsymbol{\Psi}^{(i)}$ and $\boldsymbol{\Phi}^{(i)}$.

Sub-projection learning. Given the sub-dictionaries learned in each subset, we aim to find a projection for each subset which can directly project the joint MQ depth and HR intensity features to the HR depth feature. To this end, we first replace the dictionaries in (6) with the sub-dictionaries learned in (11). Then, suppose that $\boldsymbol{\beta} = [\mathbf{u}, \mathbf{v}, \mathbf{w}]^{\mathbf{T}}$, $\boldsymbol{D}_l^{(i)} = \begin{bmatrix} \boldsymbol{\Psi}_c^{l(i)} & \boldsymbol{\Psi}_u^{l(i)} & \mathbf{0} \\ \boldsymbol{\Phi}_c^{h(i)} & \mathbf{0} & \boldsymbol{\Phi}_u^{h(i)} \end{bmatrix}$ and relax ℓ_0 norm to ℓ_2 norm, (6) can be rewritten as

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\| \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{z}_t \end{bmatrix} - \boldsymbol{D}_l^{(i)} \boldsymbol{\beta} \right\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2, \tag{12}$$

where λ is a regularization parameter. The reason why we relax the $\ell 0$ norm with $\ell 2$ norm is to make Eq. (12) a ridge regression problem, so that it can have a closed-form solution, which helps to speed up the upscaling process. The solution of (12) is as follows,

$$\boldsymbol{\beta} = (\boldsymbol{D}_l^{(i)T} \boldsymbol{D}_l^{(i)} + \lambda \boldsymbol{I})^{-1} \boldsymbol{D}_l^{(i)T} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{z}_t \end{bmatrix}, \tag{13}$$

where $\boldsymbol{D}_l^{(i)T}$ is the transpose of $\boldsymbol{D}_l^{(i)}$, and $\boldsymbol{I}$ is an identity matrix. With coefficients $\boldsymbol{\beta}$, the HR depth feature $\boldsymbol{y}_t$ can be reconstructed by

$$\boldsymbol{y}_t = \boldsymbol{D}_h^{(i)} \boldsymbol{\beta} = \boldsymbol{D}_h^{(i)} (\boldsymbol{D}_l^{(i)T} \boldsymbol{D}_l^{(i)} + \lambda \boldsymbol{I})^{-1} \boldsymbol{D}_l^{(i)T} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{z}_t \end{bmatrix}, \tag{14}$$

where $\boldsymbol{D}_h^{(i)} = \begin{bmatrix} \boldsymbol{\Psi}_c^{h(i)} & \boldsymbol{\Psi}_u^{h(i)} & \mathbf{0} \end{bmatrix}$.

For this reason, the sub-projection matrix $\boldsymbol{P}^{(i)}$ for the i-th subset is given by

$$\boldsymbol{P}^{(i)} = \boldsymbol{D}_h^{(i)} (\boldsymbol{D}_l^{(i)T} \boldsymbol{D}_l^{(i)} + \lambda \boldsymbol{I})^{-1} \boldsymbol{D}_l^{(i)T}. \tag{15}$$

Moreover, the sub-projections $\{\boldsymbol{P}^{(i)}\}_{i=1}^N$ can be pre-stored to speed up the upscaling process. In upscaling, for a joint input of MQ depth and HR intensity patches $[\boldsymbol{x}_t; \boldsymbol{z}_t]$, we just need to search for its nearest centroid in $\{\boldsymbol{O}_i\}_{i=1}^N$, and multiply the joint input $[\boldsymbol{x}_t; \boldsymbol{z}_t]$ by the sub-projection matrix corresponding to this centroid to get the HR depth patch $\boldsymbol{y}_t$. We also explored the possibility of using more than one neighbour and use their weighted sub-projections to reconstruct the HR image. However, the results are worse than those obtained using only the nearest neighbour. The full procedure to calculate the sub-projections is as follows: firstly, we learn the global dictionaries through Eq. (2), and then we learn the sub-dictionaries through Eqs. (9)-(11), and finally the sub-projections are obtained by using Eq. (15). Table II shows the summarized procedure for PRU algorithm.

In the upscaling process, given a LR depth image, we first use FDU algorithm to get a MQ depth image without any training dataset. Then, with the assistance of the corresponding intensity image, this image is further enhanced by the learned projections from MQ depth and HR intensity pairs to the HR depth.

TABLE II
THE PROPOSED PROJECTION-BASED RAPID UPSCALING ALGORITHM

Training process

- **Input:** Training features $\{\mathbf{X}; \mathbf{Z}\}$ and $\{\mathbf{Y}\}$, global dictionaries $\boldsymbol{\Psi}_c^l$, $\boldsymbol{\Psi}_u^l$, $\boldsymbol{\Psi}_c^h$, $\boldsymbol{\Psi}_u^h$, $\boldsymbol{\Phi}_c^h$, $\boldsymbol{\Phi}_u^h$.
- **Output:** Cluster centroids $\{\boldsymbol{O}^{(i)}\}_{i=1}^N$, Sub-projections $\{\boldsymbol{P}^{(i)}\}_{i=1}^N$.
- Split the training features $\{\mathbf{X}; \mathbf{Z}\}$ into N clusters with centroids $\{\boldsymbol{O}^{(i)}\}_{i=1}^N$ using K-means clustering.
- **For** each cluster centroid
  1. Search for K nearest samples to the cluster centroid using k-NN. These K samples form a subset.
  2. Calculate the sparse coefficients for the K samples in the subset using (6).
  3. Learn the sub-dictionaries for each subset using (9), (10), and (11).
  4. Learn the sub-projections $\{\boldsymbol{P}^{(i)}\}_{i=1}^N$ for each subset using (15).

  End
- Save centroids $\{\boldsymbol{O}^{(i)}\}_{i=1}^N$ and sub-projections $\{\boldsymbol{P}^{(i)}\}_{i=1}^N$.

Upscaling process

- **Input:** Joint input feature $[\boldsymbol{x}_t; \boldsymbol{z}_t]$, centroids $\{\boldsymbol{O}^{(i)}\}_{i=1}^N$ and sub-projections $\{\boldsymbol{P}^{(i)}\}_{i=1}^N$.
- **Output:** Reconstructed feature $\boldsymbol{y}_t$.
- Search for the centroid that is nearest to the joint input feature.
- Multiply the corresponding sub-projection by the input feature, to reconstruct $\boldsymbol{y}_t$.
- Save $\boldsymbol{y}_t$.

IV. Experimental Results

In this section, we quantitatively analyse the performance of both our and other state-of-the-art approaches, using the root mean square error (RMSE) and structural similarity (SSIM) [50] metrics. In addition to the quantitative comparison, we also analyse the visual quality of the reconstructed images and consider both noise-free and noisy cases. All the results can be downloaded through the link http://www.commsp.ee.ic.ac.uk/~xindeng/RADAR_results.zip.

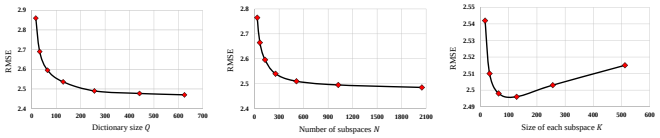
A. Datasets

For the training dataset, our approach needs both depth images and their corresponding intensity images for dictionary learning. Compared with the deep learning based methods [6], [7], we use a very small training dataset, which is composed of only 15 depth images with their registered colour images selected from the New Tsukuba dataset provided by [51]. The resolution of training images is 640×480. The colour images are changed from RGB to YCbCr format and only the luminance channel is used for training.

For the testing dataset, we evaluate the performance of our approach on five publicly available datasets, including Middlebury stereo dataset [52], Sintel stereo dataset [53], New Tsukuba (NT) dataset [51], NYU indoor scene dataset [54], and ToFMark dataset [26]. The first four datasets are used for the noise-free case while the ToFMark dataset and the noisy Middlebury dataset are used for the noisy case. From each dataset, we use four randomly selected image pairs of depth image and the corresponding intensity image (ToFMark only has three) as testing images. The testing images vary in resolutions, and cover most of the scenarios we may encounter, including intensity images of real and synthetic scenes and

TABLE III
TESTING DATASETS

Dataset name	Resolution	Real-scene intensity	Noisy depth	Texture detail	Edge detail
Middlebury	low: 448×368	Y	N	Strong	Middle
NT dataset	high: 640×480	Y	N	Middle	Strong
Sintel dataset	very high: 1024×432	N	N	Middle	Strong
NYU dataset	mid-high: 560×424	Y	N	Middle	Middle
ToFMark dataset	high: 810×610	Y	Y	Low	Strong



(a) (b) (c)

Fig. 6. (a) RMSE as a function of the size Q of dictionaries, (b) RMSE as a function of the number N of subspaces, (c) RMSE as a function of the size K of each subspace

depth images with or without noise. For example, the images in the Middlebury dataset are at low resolution, while those in NT dataset are at high resolution. The Sintel dataset is a synthetic dataset in which both the intensity and depth images are synthesized by computer. The NYU and ToFMark datasets are both real-world datasets, but the former is collected using Kinect and the latter using ToF camera. For the specific difference, please refer to Table III.

B. Performance Analysis of Our Method

Parameter settings. In the TDL process, there are some parameters which can affect the performance, like the dictionary size Q, the sparsity constraint L and the size of training patches. Fig. 6 (a) plots the RMSE change with dictionary size Q, from which we can see that larger Q leads to better reconstruction results. However, it is at the expense of training complexity. In this paper, we set $Q = 256$ to achieve a good trade-off between the training complexity and reconstruction performance. Moreover, the sparsity level L is set to 3. The training patch sizes for $2\times$, $4\times$, $8\times$, and $16\times$ upscaling are 4×4, 8×8, 12×12, and 24×24, respectively. We use the mean-removed MQ depth patch as MQ depth feature, the mean-removed HR intensity patch as HR intensity feature, and the HR depth feature is obtained from the HR depth patch by subtracting the mean value of its corresponding MQ depth patch. The regularization parameter λ, which is used in Eqs. (4), (9), (10), (12), (15), is set to 0.1 for all cases. In the PRU process, as shown in Fig. 6 (b) and (c), we find that a large number of clusters N can improve the reconstruction accuracy but increases also the computational time. The number of samples K in each cluster has little influence on the performance. Based on our experiments, we set $N = 1024$ and $K = 32$. Note that the multiplication of K and N is not required to be equal to the number of training patches, since clusters may overlap, i.e., different clusters may have some common samples. We just need to satisfy that the number of clusters is equal to N and the samples inside each cluster is equal to K.

TABLE IV
INDIVIDUAL CONTRIBUTIONS OF EACH ELEMENT ON THE FOUR NOISE-FREE DATASETS.

RMSE	2×	4×	8×	16×
Bicubic	2.60	3.96	5.85	8.64
Only FDU	1.86	2.70	4.01	6.85
Only PRU	1.58	2.99	4.55	6.93
FDU+PRU	**1.14**	**2.33**	**3.81**	**6.45**
FDU+TDL(OMP)	1.21	2.39	3.87	6.48
SSIM	2×	4×	8×	16×
Bicubic	0.9878	0.9692	0.9425	0.9067
Only FDU	0.9935	0.9847	0.9649	0.9349
Only PRU	0.9954	0.9833	0.9574	0.9315
FDU+PRU	**0.9974**	**0.9887**	**0.9712**	**0.9409**
FDU+TDL(OMP)	0.9951	0.9872	0.9666	0.9381

TABLE V
RMSE COMPARISON WITH AND WITHOUT THE GUIDANCE OF INTENSITY IMAGES AT DIFFERENT SCALING FACTORS ON THE FOUR NOISE-FREE DATASETS.

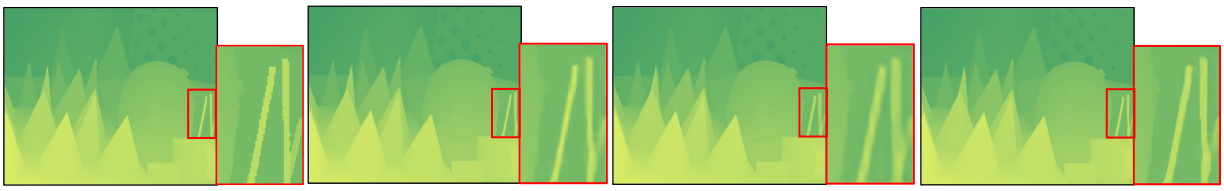
RMSE	2×	4×	8×	16×
Without intensity	1.42	2.53	3.96	6.73
With intensity	**1.14**	**2.33**	**3.81**	**6.45**

TABLE VI
OMP VERSUS PRU FOR BOTH CLUSTERING AND NON-CLUSTERING CASES ON MIDDLEBURY DATASET.

RMSE		*Cones*	*Teddy*	*Tsukuba*	*Venus*	Average
$N = 1$	OMP	2.56	1.85	5.08	0.86	2.59
	PRU	2.68	1.94	5.35	1.02	2.75
$N = 1024$	OMP	2.48	1.81	4.97	0.83	2.52
	PRU	2.26	1.63	4.90	0.73	**2.38**

Individual contributions. Table IV shows the individual contributions of FDU and PRU elements to the performance of our method, in which the results are averaged over the four noise-free datasets. Here, the results in the "only PRU" case are obtained with the bicubic interpolated images as inputs. The RMSE results are averaged among the four noise-free testing datasets. As can be seen, when FDU and PRU are used as stand-alone methods, while they can enhance the image quality, their performance is not comparable to the enhancement achieved when they are used in cascade. This is because when cascaded, the FDU process can lift the image quality to a higher level, decreasing the mapping ambiguity in TDL training. Moreover, compared with small upscaling factors, $2\times$ and $4\times$, FDU plays a more important role in large factors, i.e., $8\times$ and $16\times$. As can be seen, with only the FDU, the RMSE value improves on average by nearly 1.8 for $8\times$ and $16\times$ upscaling, compared with 1.0 for $2\times$ and $4\times$ upscaling. To make the individual contribution of each part more evident, we show in Fig. 7 the reconstruction results with only FDU algorithm, with only PRU algorithm and with both of them together. As can be seen, when combining these two algorithms, the reconstructed image is sharper and clearer than that with only one of them. Moreover, we also compare the performance with our PRU algorithm and the OMP algorithm (i.e., we use OMP to solve Eq. (6)). As shown in Table IV, the PRU algorithm performs better in both RMSE and SSIM.

Use of intensity images to improve performance. In order



(a) GT (b) FDU (2.74/0.9799) (c) PRU (3.25/0.9735) (d) FDU+PRU (2.26/0.9851)

Fig. 7. Visual comparison of *Cones* in Middlebury dataset with upscaling factor $= 4$. The values in the bracket are RMSE/SSIM values. (a) Ground truth. (b) Result only with FDU algorithm. (c) Result only with PRU algorithm. (d) Result with both FDU and PRU algorithms.

TABLE VII
RESULTS ON MIDDLEBURY DATASET FOR 2× AND 4× UPSCALING. THE BEST RESULTS ARE IN BOLD AND THE SECOND BESTS ARE UNDERLINED.

Methods	Scaling factor=2								Scaling factor = 4							
	Cones		Teddy		Tsukuba		Venus		Cones		Teddy		Tsukuba		Venus	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
Bicubic	2.47	0.9837	1.88	0.9869	5.56	0.9723	1.28	0.9951	3.71	0.9620	2.84	0.9688	8.32	0.9321	1.92	0.9876
Kuznietsov *et al.* [21]	69.92	0.5568	62.10	0.6058	93.64	0.4866	58.26	0.7243	69.92	0.5568	62.10	0.6058	93.64	0.4866	58.26	0.7243
Aodha *et al.* [12]	4.32	0.9606	3.28	0.9690	9.11	0.9364	2.20	0.9874	3.69	0.9392	4.11	0.9520	7.69	0.9080	2.65	0.9822
Ferstl *et al.* [15]	2.21	0.9866	1.72	0.9884	5.33	0.9766	1.12	0.9963	3.57	0.9645	2.65	0.9716	7.54	0.9413	1.78	0.9893
Xie *et al.* [8]	2.73	0.9633	2.49	0.9625	6.35	0.9464	1.64	0.9852	4.41	0.9319	3.28	0.9331	9.78	0.8822	2.37	0.9730
Timofte *et al.* [32]	2.13	0.9871	1.72	0.9882	4.98	0.9776	1.16	0.9960	3.41	0.9709	2.43	0.9750	7.35	0.9507	1.59	0.9916
Huang *et al.* [33]	2.63	0.9846	2.43	0.9853	6.08	0.9544	2.17	0.9705	5.29	0.9533	4.23	0.9508	9.76	0.9312	3.06	0.9749
Park *et al.* [24]	2.85	0.9699	2.18	0.9767	6.89	0.9320	1.26	0.9910	6.54	0.9420	4.35	0.9553	12.12	0.8981	2.36	0.9862
Ferstl *et al.* [26]	3.81	0.9788	2.93	0.9795	6.89	0.9576	1.39	0.9938	4.66	0.9625	3.67	0.9707	9.91	0.9245	1.60	0.9900
Lu *et al.* [27]	3.28	0.9875	2.07	0.9895	7.35	0.9659	1.30	0.9952	4.02	0.9697	2.73	0.9782	10.20	0.9212	1.77	0.9888
Gu *et al.* [29]	1.75	0.9858	1.65	0.9841	3.41	0.9848	0.59	0.9981	3.93	0.9627	2.86	0.9694	8.36	0.9346	1.59	0.9922
Dong *et al.* [35]	1.62	0.9900	1.34	0.9911	3.68	0.9798	0.63	0.9971	3.55	0.9712	2.56	0.9813	7.88	0.9552	1.26	0.9883
Wang *et al.* [36]	1.86	0.9900	1.37	0.9918	3.89	0.9858	0.83	0.9979	3.07	0.9756	2.03	0.9811	6.29	0.9677	1.20	0.9952
Kim *et al.* [37]	1.45	0.9932	1.15	0.9940	3.03	0.9922	0.70	0.9984	2.49	0.9837	1.72	0.9859	5.21	0.9762	0.92	0.9971
Song *et al.* [7]	1.44	0.9915	1.20	0.9920	3.02	0.9898	0.56	0.9989	2.80	0.9802	1.80	0.9837	6.21	0.9641	0.88	0.9972
Riegler *et al.* [6]	**1.00**	-	0.82	-	2.38	-	**0.20**	-	2.93	-	**1.50**	-	6.63	-	**0.38**	-
RADAR	1.01	**0.9964**	**0.80**	**0.9959**	2.24	**0.9959**	0.23	**0.9997**	**2.26**	**0.9851**	1.63	**0.9858**	**4.90**	**0.9777**	0.73	**0.9981**
RADAR (fast)	1.15	0.9952	0.86	0.9954	**2.04**	0.9956	0.24	0.9997	2.67	0.9813	1.83	0.9836	5.51	0.9745	0.72	0.9980

to test the contribution of the intensity image to recovering the depth image, we just remove the guidance of intensity image in the TDL process. That is, there are only two modalities in (2), the MQ depth and HR depth. Table V presents the averaged RMSE results with and without intensity guidance with scaling factors ranging from 2× to 16× averaged over the four noise-free datasets. We can see that the intensity image indeed helps to super-resolve the depth image.

Effects of clustering. In order to show the effectiveness of clustering, we present in Table VI the RMSE results when the number of clusters is $N = 1$ (which indicates no clustering is used) and $N = 1024$ on Middlebury dataset. In addition, we compare our algorithm with the case in which the OMP algorithm is used after clustering. As we can see, whether for our PRU or the OMP algorithms, the reconstruction accuracy is improved with more clusters. When there is no clustering, the OMP algorithm has higher reconstruction accuracy (RMSE=2.59) than our PRU (RMSE=2.75). This is because in this case we only have one global projection. When the sub-projection is used with $N = 1024$ clusters, our PRU algorithm (RMSE=2.38) performs better than the OMP (RMSE=2.52).

C. Numerical Comparison Against Other Methods

Benchmarks. We compare our method with the following benchmarks which can be classified into three categories. 1) Single color image super-resolution methods, including Yang *et al.* [30], Timofte *et al.* [32], Huang *et al.* [33], Dong *et al.* [35], Wang *et al.* [36], and Kim *et al.* [37]. 2) State-of-the-art single depth image super-resolution methods, including Aodha *et al.* [12], Ferstl *et al.* [15], Xie *et al.* [8], and Riegler *et al.* [6]. 3) State-of-the-art intensity guided depth image super-resolution methods, including Park *et al.* [24], Ferstl *et al.* [26], Lu *et al.* [27], Song *et al.* [7], and Gu *et al.* [29]. Note that [6], [7], [35]–[37] are all deep learning based methods, which use a large dataset for training. To make the comparison more exhaustive, we also compare with the state-of-the-art depth prediction method proposed by Kuznietsov *et al.* [21]. The numerical results of these methods are obtained either by implementing the source code, or emailing the authors for the upscaling results[2]. Since some methods cannot do upscaling with large factors, e.g., 8× and 16×, we just ignore them when doing the comparison for those specific settings. For our approach, we present the results of two versions, a normal RADAR and a fast RADAR version. The only difference is

[2]For the deep learning based methods [6], [7], [35]–[37], we directly use their already trained models. Since we have no access to the code of [6], we just use the results in the paper, which only provides the RMSE results.

TABLE VIII
RESULTS ON THE NT, SINTEL AND NYU DATASETS FOR 4× UPSCALING. THE BEST RESULTS ARE IN BOLD AND THE SECOND BESTS ARE UNDERLINED.

NT dataset	NT0200		NT0350		NT1400		NT1525	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
Bicubic	0.99	0.9905	1.80	0.9839	1.38	0.9898	1.66	0.9858
Kuznietsov *et al.* [21]	74.32	0.4821	66.39	0.5782	84.63	0.4957	76.29	0.5023
Xie *et al.* [8]	1.42	0.9767	2.62	0.9711	1.84	0.9686	2.33	0.9699
Timofte *et al.* [32]	0.84	0.9935	1.47	0.9898	1.01	0.9928	1.40	0.9897
Huang *et al.* [33]	0.96	0.9885	1.65	0.9866	1.44	0.9884	1.65	0.9864
Park *et al.* [24]	0.96	0.9885	1.78	0.9855	1.37	0.9892	1.86	0.9823
Ferstl *et al.* [26]	1.02	0.9875	1.88	0.9844	1.24	0.9914	1.82	0.9831
Lu *et al.* [27]	1.00	0.9882	1.83	0.9849	1.71	0.9721	2.06	0.9735
Gu *et al.* [29]	1.28	0.9889	1.79	0.9856	1.40	0.9897	1.77	0.9850
Dong *et al.* [35]	0.79	0.9945	1.28	0.9923	0.92	0.9946	1.22	0.9913
Wang *et al.* [36]	1.45	0.9929	1.21	0.9924	0.90	0.9954	1.16	0.9925
Kim *et al.* [37]	0.72	0.9948	1.05	0.9937	0.94	0.9939	1.20	0.9922
RADAR	**0.65**	**0.9956**	0.94	0.9950	**0.66**	**0.9967**	**0.96**	**0.9943**
RADAR (fast)	0.65	0.9956	**0.92**	**0.9950**	0.70	0.9962	1.03	0.9935
Sintel dataset	**Ambush**		**Bamboo**		**Cave**		**Market**	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
Bicubic	6.22	0.9704	13.76	0.8845	6.54	0.9524	8.74	0.9341
Kuznietsov *et al.* [21]	81.35	0.4552	68.87	0.5435	69.50	0.7337	95.68	0.3865
Xie *et al.* [8]	8.79	0.9438	19.02	0.8301	9.14	0.9221	12.21	0.8869
Timofte *et al.* [32]	5.05	0.9756	12.06	0.8910	5.34	0.9681	7.03	0.9462
Huang *et al.* [33]	5.32	0.9720	12.45	0.8784	5.69	0.9616	7.30	0.9412
Park *et al.* [24]	6.03	0.9678	12.05	0.8910	7.13	0.9379	9.45	0.9067
Ferstl *et al.* [26]	5.99	0.9701	11.54	0.8950	6.40	0.9563	8.01	0.9298
Lu *et al.* [27]	5.53	0.9712	10.61	0.9028	6.10	0.9610	8.31	0.9266
Gu *et al.* [29]	6.04	0.9766	13.35	0.9001	6.15	0.9613	8.10	0.9470
Dong *et al.* [35]	5.02	0.9761	11.89	0.8934	5.32	0.9684	6.87	0.9487
Wang *et al.* [36]	4.29	0.9850	9.63	0.9389	4.37	0.9769	5.94	0.9664
Kim *et al.* [37]	**3.32**	0.9896	**8.69**	0.9463	3.97	0.9839	6.21	0.9677
RADAR	3.43	**0.9919**	8.74	**0.9551**	**3.57**	**0.9857**	**5.13**	**0.9782**
RADAR (fast)	3.52	0.9911	9.04	0.9505	3.73	0.9831	5.32	0.9755
NYU dataset	**Image-1**		**Image-2**		**Image-3**		**Image-4**	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
Bicubic	1.20	0.9935	1.27	0.9936	1.63	0.9873	1.34	0.9915
Kuznietsov *et al.* [21]	86.87	0.4352	85.39	0.4568	78.29	0.5087	66.37	0.5345
Xie *et al.* [8]	1.77	0.9889	1.92	0.9897	2.83	0.9768	1.99	0.9855
Timofte *et al.* [32]	0.91	0.9942	0.94	0.9936	1.44	0.9891	1.14	0.9932
Huang *et al.* [33]	1.12	0.9934	1.22	0.9926	1.88	0.9834	1.47	0.9898
Park *et al.* [24]	1.53	0.9903	1.37	0.9921	2.14	0.9823	1.64	0.9885
Ferstl *et al.* [26]	1.78	0.9889	1.26	0.9924	2.25	0.9810	1.78	0.9867
Lu *et al.* [27]	1.38	0.9920	1.40	0.9916	2.19	0.9802	1.46	0.9898
Gu *et al.* [29]	1.24	0.9914	1.22	0.9935	1.96	0.9844	1.41	0.9905
Dong *et al.* [35]	0.91	0.9942	0.92	0.9941	1.42	0.9901	1.13	0.9927
Wang *et al.* [36]	**0.79**	0.9953	0.74	0.9966	1.28	0.9909	0.96	0.9943
Kim *et al.* [37]	0.83	0.9952	**0.70**	0.9967	1.24	0.9918	0.94	0.9948
RADAR	0.88	**0.9954**	0.71	**0.9970**	**1.17**	**0.9927**	**0.92**	**0.9949**
RADAR (fast)	1.06	0.9943	0.82	0.9964	1.44	0.9909	1.19	0.9934

TABLE IX
RESULTS ON THE MIDDLEBURY DATASET FOR 8× AND 16× UPSCALING. THE BEST RESULTS ARE IN BOLD AND THE SECOND BESTS ARE UNDERLINED.

Methods	Scaling factor=8							
	Cones		Teddy		Tsukuba		Venus	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
Bicubic	5.77	0.9322	4.19	0.9442	12.61	0.8684	2.79	0.9771
Yang *et al.* [30]	5.33	0.9448	3.69	0.9530	11.74	0.9002	2.21	0.9822
Timofte *et al.* [32]	5.09	0.9452	3.48	0.9572	11.33	0.9023	1.87	0.9892
Park *et al.* [24]	7.81	0.9389	5.52	0.9456	17.60	0.8498	3.35	0.9642
Ferstl *et al.* [26]	6.68	0.9422	5.36	0.9351	16.30	0.8575	2.42	0.9800
Lu *et al.* [27]	5.54	0.9442	3.76	0.9561	13.32	0.8852	2.20	0.9870
Gu *et al.* [29]	5.59	0.9373	4.05	0.9489	12.72	0.8761	2.61	0.9806
Dong *et al.* [35]	5.28	0.9449	3.55	0.9567	11.32	0.9023	1.98	0.9889
Wang *et al.* [36]	4.75	0.9483	3.14	0.9597	10.59	0.8864	1.94	0.9872
Kim *et al.* [37]	5.55	0.9322	4.04	0.9451	12.08	0.8693	2.79	0.9766
Song *et al.* [7]	4.59	0.9510	2.93	0.9682	11.79	0.8942	1.74	0.9897
RADAR	**4.45**	**0.9589**	**2.49**	**0.9726**	**9.84**	**0.9275**	**1.18**	**0.9949**
RADAR (fast)	4.58	0.9533	2.85	0.9680	10.05	0.9208	1.45	0.9923
Methods	Scaling factor=16							
	Cones		Teddy		Tsukuba		Venus	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
Bicubic	8.43	0.8996	6.68	0.9162	16.77	0.8232	4.31	0.9628
Yang *et al.* [30]	7.45	0.9033	5.42	0.9287	16.53	0.8291	2.88	0.9687
Timofte *et al.* [32]	6.99	0.9096	4.95	0.9312	16.20	0.8301	2.63	0.9709
Park *et al.* [24]	10.23	0.8903	8.36	0.9124	19.36	0.8122	5.16	0.9547
Ferstl *et al.* [26]	11.84	0.8768	8.13	0.8977	26.54	0.8026	4.94	0.9607
Lu *et al.* [27]	10.96	0.8922	8.01	0.9135	16.76	0.8232	3.26	0.9655
Gu *et al.* [29]	7.40	0.9189	5.81	0.9316	17.59	0.8272	3.77	0.9723
Dong *et al.* [35]	7.18	0.9076	5.26	0.9379	16.25	0.8311	2.65	0.9710
Wang *et al.* [36]	7.11	0.9145	5.45	0.9256	18.82	0.7812	3.76	0.9664
Kim *et al.* [37]	7.41	0.9159	5.89	0.9290	17.68	0.8229	4.00	0.9688
RADAR	**6.18**	**0.9310**	**4.00**	**0.9500**	**14.98**	**0.8625**	**2.11**	**0.9863**
RADAR (fast)	6.39	0.9278	4.37	0.9449	15.25	0.8547	2.46	0.9822

TABLE X
PERFORMANCE OF OUR AND OTHER METHODS FOR 4× UPSCALING, WITH ARBITRARY BLURRING KERNELS.

Blurring function	*bior2.4*		*bior6.8*		*rbio2.8*		*linear spline*	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
Dong *et al.* [35]	5.68	0.9250	4.89	0.9376	4.59	0.9469	5.64	0.9294
Wang *et al.* [36]	4.57	0.9670	3.14	0.9766	3.41	0.9725	4.37	0.9692
Kim *et al.* [37]	5.00	0.9572	3.16	0.9780	3.11	0.9815	4.66	0.9633
RADAR	**2.40**	**0.9889**	**2.34**	**0.9894**	**2.38**	**0.9892**	**2.43**	**0.9879**

that the fast RADAR version removes the self-learning part in the FDU algorithm, which is slightly time-consuming.

Small upscaling factors. Table VII presents the RMSE and SSIM results of our method and the benchmarks for 2× and 4× upscaling in the Middlebury stereo dataset. As can be seen, our method outperforms others in both RMSE and SSIM in most of the cases. In particular, even with a fraction of the training dataset, our method consistently performs better than [7], [35]–[37], which are all state-of-the-art deep learning based methods. The reason is probably because our method fully exploits the intrinsic property of depth images and the correlation between depth and intensity images. The property of depth images, i.e., the rows/columns can be approximated as piece-wise polynomials, is exploited in the FDU process, and the relationship between modalities is fully exploited in the TDL process. In order to further demonstrate the strengths of our approach, we show results on the other datasets in Table VIII. Again, we can see that our method achieves state-of-the-art results.

Large upscaling factors. It is more challenging to do depth image super-resolution at large upscaling factors, e.g., 8× and 16×, because more than 90% of information is lost. This is one of the reasons why many methods perform well with small upscaling factors but fail with large factors. For large upscaling factors, the guidance of intensity images becomes important, because the information provided by the LR depth is very limited. Table IX shows the reconstruction results for 8× and 16× upscaling, from which we can see that our method performs consistently better than others. There are two reasons why we can achieve such good performance in large factor upscaling. Firstly, our FDU algorithm can enhance the LR depth to a higher quality, which decreases the coupling ambiguity in the TDL process. Secondly, the structured dictionaries learned in the TDL algorithm are able to fully use the guidance of HR intensity image.

Resilience to uncertainties on the blurring function. The image capturing process can be regarded as a blurring

TABLE XI
RUNNING TIME IN SECONDS OF OUR METHOD ON MIDDLEBURY DATASET.

RADAR		*Cones*	*Teddy*	*Tsukuba*	*Venus*	Average
2×	FDU	14.00	13.90	8.54	13.67	12.53
	PRU	2.06	2.00	1.38	2.01	1.86
4×	FDU	17.05	17.70	10.74	16.85	15.58
	PRU	2.12	2.05	1.52	2.03	1.93
8×	FDU	17.54	17.53	11.19	16.97	15.81
	PRU	4.17	5.14	3.06	4.16	4.13
RADAR (fast)		*Cones*	*Teddy*	*Tsukuba*	*Venus*	Average
2×	FDU	2.13	2.08	1.00	2.10	1.83
	PRU	2.06	2.00	1.38	2.01	1.86
4×	FDU	2.46	2.43	1.31	2.35	2.14
	PRU	2.12	2.05	1.52	2.03	1.93
8×	FDU	2.50	2.46	1.38	2.36	2.18
	PRU	4.17	5.14	3.06	4.16	4.13

TABLE XII
RUNNING TIME IN SECONDS OF OMP AND OUR PRU ALGORITHMS.

		Cones	*Teddy*	*Tsukuba*	*Venus*	Average
2×	OMP	485.56	483.74	462.13	486.21	479.42
	PRU	2.06	2.00	1.38	2.01	1.86
4×	OMP	628.79	621.88	603.63	636.62	622.73
	PRU	2.12	2.05	1.52	2.03	1.93
8×	OMP	1065.28	1054.43	1033.57	1074.82	1057.10
	PRU	4.17	5.14	3.06	4.16	4.13

and downsampling operation on the true image, where the blurring function is often unknown. Thus, for image upscaling, it is important to develop algorithms that are resilient to uncertainties on the blurring function (blurring kernel). Table X shows the performance of our and the deep learning based methods [35]–[37] using different blurring functions for testing without changing the blurring function used in the training. We can see that for different blurring functions, the RMSE and SSIM values of our method only have small variations, while those of deep learning based methods have significant fluctuations. This is because our FDU algorithm takes advantage of the intrinsic characteristic of depth images to do upscaling, i.e., the depth lines can be modeled as piece-wise polynomial signals, and the change of extrinsic settings, such as the blurring kernels, does not affect this intrinsic characteristic. In contrast, since the deep learning based methods solely rely on the training dataset, a mismatch between the training and testing datasets can have a significant impact on the performance.

Running time. Table XI presents the running time of our approach as well as the fast version, for different upscaling factors. In the upscaling process, the running time is composed of two parts: the FDU and PRU algorithms. Thus, we present the running time breakdown between FDU and PRU. Recall that the only difference between the complete and fast versions is that the fast version does not include the self-learning part in the FDU algorithm. Here, the time is tested on a Windows PC with 3.4GHz Inter(R) Core i7 CPU and 16GB RAM. As we can see from this table, the running time of the fast version is relatively good for upscaling factors up to $8\times$, i.e., 4.69s per image on average, which is faster than that of Ferstl *et al.* [26] (135s per image) and Lu *et al.* [27] (286s per image). However, it is still slower than that of the deep learning based method, for example, Wang *et al.* [36] (0.93s per image) and Kim *et al.* [37] (2.65s per image). As we mentioned before, our PRU algorithm can significantly speed up the upscaling process compared to the conventional OMP algorithm. To verify this, we compare in Table XII the running time of the OMP and our PRU algorithms, respectively. As we can see, our PRU algorithm is hundreds of times faster than the OMP which takes more than 1000s per image for $8\times$ upscaling.

TABLE XIII
QUANTITATIVE RESULTS OF RMSE ON NOISY MIDDLEBURY DATASET.

2×		*Cones*	*Teddy*	*Tsukuba*	*Venus*	Average
σ^2=65	Bicubic	6.97	6.77	8.73	6.06	7.13
	Ours	3.11	2.82	4.96	1.85	3.18
σ^2=130	Bicubic	9.54	9.45	10.85	9.30	9.79
	Ours	3.82	3.39	5.56	2.41	3.80
σ^2=260	Bicubic	13.21	13.20	14.20	13.07	13.42
	Ours	4.62	4.25	6.58	3.09	4.63
4×		*Cones*	*Teddy*	*Tsukuba*	*Venus*	Average
σ^2=65	Bicubic	7.62	7.29	10.79	6.86	8.14
	Ours	5.24	3.99	9.52	2.72	5.37
σ^2=130	Bicubic	10.03	9.74	12.58	9.43	10.44
	Ours	5.77	4.59	9.88	3.28	5.88
σ^2=260	Bicubic	13.73	13.46	15.45	13.26	13.98
	Ours	6.45	5.38	10.47	4.22	6.63

D. Qualitative Results

Figures 8, 9 and 10 visualize the $4\times$ upscaling results of *Tsukuba*, *Bamboo*, and *Image-1* from Middlebury, Sintel, and NYU datasets, respectively. As can be seen, our method can reconstruct depth images with the visual quality quite close to the ground truth, while others result in either blurred [29], [32] or diffused edges [26] [27].

In Figure 11, we compare the $8\times$ upscaling results of *Cones* from Middlebury dataset. It can be easily observed that our method reconstructs clearer and sharper edges compared to others, and at the same time our method avoids the depth diffusion problem of [26] and [27].

E. Noisy Case

Real-world depth images are usually corrupted by noise. For this reason, we evaluate the performance of our approach on the noisy Middlebury dataset and the real-world ToFMark [26] dataset. Here, in the Middlebury dataset, Gaussian white noise is added to the LR versions of the images, via the MATLAB imnoise function with zero mean, and three different variances $\sigma^2 = 65$, 130, and 260. The ToFMark dataset provides a LR noisy depth image for each scene, thus we do not need to add noise. Note that here we assume that the intensity images which are used as guidance are always noiseless. In addition, the dictionaries need to be re-trained in the noisy case. Figs. 12 and 13 show the visual results on the noisy Middlebury and ToFMark datasets. We can see that most of the noise is eliminated using our method and the boundaries/edges corrupted by noise are nicely reconstructed. Table XIII shows quantitative results in terms of RMSE on the noisy Middlebury dataset for $2\times$ and $4\times$ upscaling with three different noise levels. From this table, we can see that our algorithm is able to eliminate the noise with larger variances more effectively. In practical scenarios, to determine whether to use noise-free or noisy version of our algorithm, we have run both

versions of the method on noisy testing images. Fig. 14 shows the changes of peak signal-to-noise ratio (PSNR) values with respect to the noise variance σ^2 for both the noise-free and noisy versions of our algorithm. As shown in Fig. 14, we can see that the noise-free version is more effective for noise variance $\sigma^2 < \theta$, whereas the noisy version performs better for $\sigma^2 > \theta$. Considering the diversity of the depth images in real world, we allow an interval $[\theta-\delta, \theta+\delta]$ in which either noise-free or noisy version can be used. When the noise variance is larger than $\theta+\delta$, it is more appropriate use the noisy version, while for the noise variance smaller than $\theta-\delta$, we prefer to use the noise-free version. According to Fig. 14, we can set $\theta = 2.5$, and we empirically set $\delta = 1.5$.

V. Conclusions

This paper proposes a novel depth image super-resolution approach, called RADAR, which combines the FRI reconstruction theory with multi-modal dictionary learning. The main strength of this method is that it achieves upscaling of depth images at very high upscaling factors, but with a small training dataset. Moreover, it does not require to train a complex deep neural network. Also, it is robust to noisy depth images and unknown blurring kernels. Given a LR depth image, we firstly approximate its rows and columns as piece-wise polynomial signals and propose a 2-D FRI reconstruction algorithm to upscale it. The initially upscaled image is further enhanced through a fast, projection based upscaling process in which the HR intensity image guides the super-resolution. Extensive experimental results show that our method outperforms other state-of-the-art methods on both synthetic and real-world datasets with upscaling factor of up to 16×.

Appendix

We model the pixels in a row or column of the LR depth image as follows:

$$y_n = \langle x(t), \varphi(t-n)\rangle, \quad n \in \{0,1,2,...N\}, \tag{16}$$

where $x(t)$ is a piecewise polynomial function and $\varphi(t)$ models the distortion due to the lenses in the camera. We call $\varphi(t)$ the "blurring kernel". It is possible to show that the scaling functions in a wavelet transform model well the function $\varphi(t)$ [56]. Moreover, it is possible to show that there exist coefficients $c_{m,n}$ such that:

$$\sum_{n\in\mathbb{Z}} c_{m,n}\varphi(t-n) \simeq e^{jw_0 mt}. \tag{17}$$

Under some mild assumptions, the approximation in Eq. (17) is exact, otherwise the error is small [10]. Finally, it is possible to show that the reconstruction of the piecewise polynomial signal $x(t)$ from the pixels y_n is equivalent to the reconstruction of a stream of Diracs [43]. We therefore focus on this second case.

Consider a stream of Diracs signal $x(t) = \sum_{k=0}^{K-1} a_k \delta(t - t_k)$, we begin by linearly combining the samples y_n with the coefficients $c_{m,n}$ of Eq. (17)

$$s_m = \sum_n c_{m,n} y_n, \quad m \in \{0,1,2,...L\}. \tag{18}$$

Taking Eq. (16) in to Eq. (18), we obtain

$$\begin{aligned} s_m &\overset{(a)}{=} \langle x(t), \sum_n c_{m,n}\varphi(t-n)\rangle \\ &\overset{(b)}{=} \int_{-\infty}^{+\infty} \sum_{k=0}^{K-1} a_k \delta(t-t_k) e^{jw_0 mt} dt \\ &\overset{(c)}{=} \sum_{k=0}^{K-1} a_k u_k^m, \quad m \in \{0,1,2,...L\}, \end{aligned} \tag{19}$$

where (a) follows from linearity of the inner product, (b) from the fact that $x(t) = \sum_{k=0}^{K-1} a_k \delta(t-t_k)$ and Eq. (17), and (c) by setting $u_k = e^{jw_0 t_k}$.

We now have the values of s_m from Eq. (18) and we hope to figure out the values of a_k and u_k in Eq. (19). To this end, following the Prony's method [57], we first need to construct an annihilating filter. Suppose the filter coefficients are $\{h_m\}_{m=0}^K$, the z-transform $H(z)$ of $\{h_m\}_{m=0}^K$ should satisfy the following condition,

$$H(z) = \sum_{m=0}^{K} h_m z^{-m} = \prod_{k=0}^{K-1} (1 - u_k z^{-1}). \tag{20}$$

In other words, the roots of $H(z)$ are the $u_k s$ which contain the information of the location t_k of the Diracs. The convolution of h_m and s_m is equal to zero, as the following proof shows,

$$\begin{aligned} h_m * s_m &= \sum_{l=0}^{K} h_l s_{m-l} \\ &= \sum_{l=0}^{K} h_l \sum_{k=0}^{K-1} a_k u_k^{m-l} \\ &= \underbrace{\sum_{l=0}^{K} h_l u_k^{-l}}_{H(u_k)=0} \sum_{k=0}^{K-1} a_k u_k^m = 0 \end{aligned} \tag{21}$$

If we have s_m for m =0, 1, ..., $2K$-1, $h_m * s_m = 0$ can be written as a square Toeplitz matrix form by imposing $h_0 = 1$,

$$\begin{bmatrix} s_{K-1} & s_{K-2} & \dots & s_0 \\ s_K & s_{K-1} & \dots & s_1 \\ \vdots & \vdots & \ddots & \vdots \\ s_{2K-2} & s_{2K-3} & \dots & s_{K-1} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_K \end{bmatrix} = - \begin{bmatrix} s_K \\ s_{K+1} \\ \vdots \\ s_{2K-1} \end{bmatrix}. \tag{22}$$

Eq. (22) is a classic Yule-Walker system, and through solving it we can get the filter coefficients $\{h_m\}_{m=0}^K$ and their z-transform $H(z)$. By calculating the roots of $H(z)$, we obtain the value of u_k and also the location t_k since $u_k = e^{jw_0 t_k}$. Recall from Eq. (19) that $s_m = \sum_{k=0}^{K-1} a_k u_k^m$, therefore the amplitude a_k can be obtained through solving the following system of equations,

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ u_0 & u_1 & \dots & u_{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_0^{K-1} & u_1^{K-1} & \dots & u_{K-1}^{K-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{K-1} \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{K-1} \end{bmatrix}. \tag{23}$$

Eq. (23) is a Vandermonde system which yields a unique solution if the $u_k s$ are distinct. In the case of a piecewise polynomial signal, the algorithm would retrieve the locations t_k of the discontinuities first and then the coefficients of the polynomials.

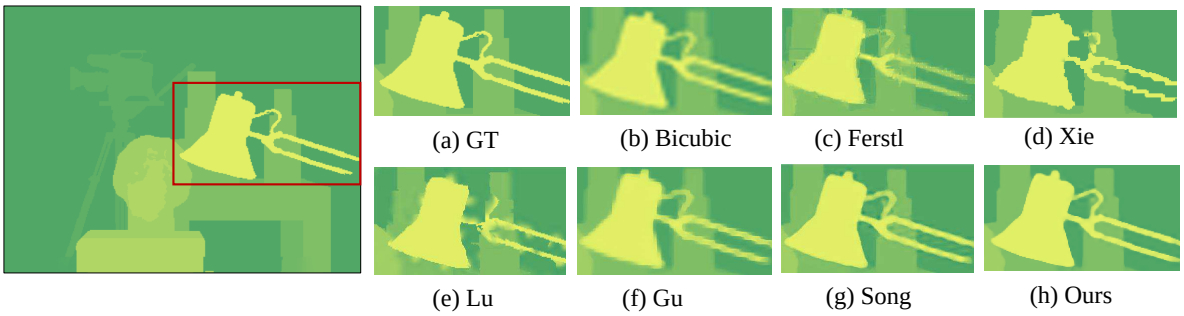


Fig. 8. Visual comparison of *Tsukuba* in Middlebury dataset with upscaling factor = 4. (a) Ground truth. (b) Bicubic. (c) Ferstl *et al.* [26]. (d) Xie *et al.* [8]. (e) Lu *et al.* [27]. (f) Gu *et al.* [29]. (g) Song *et al.* [7]. (h) Our method.

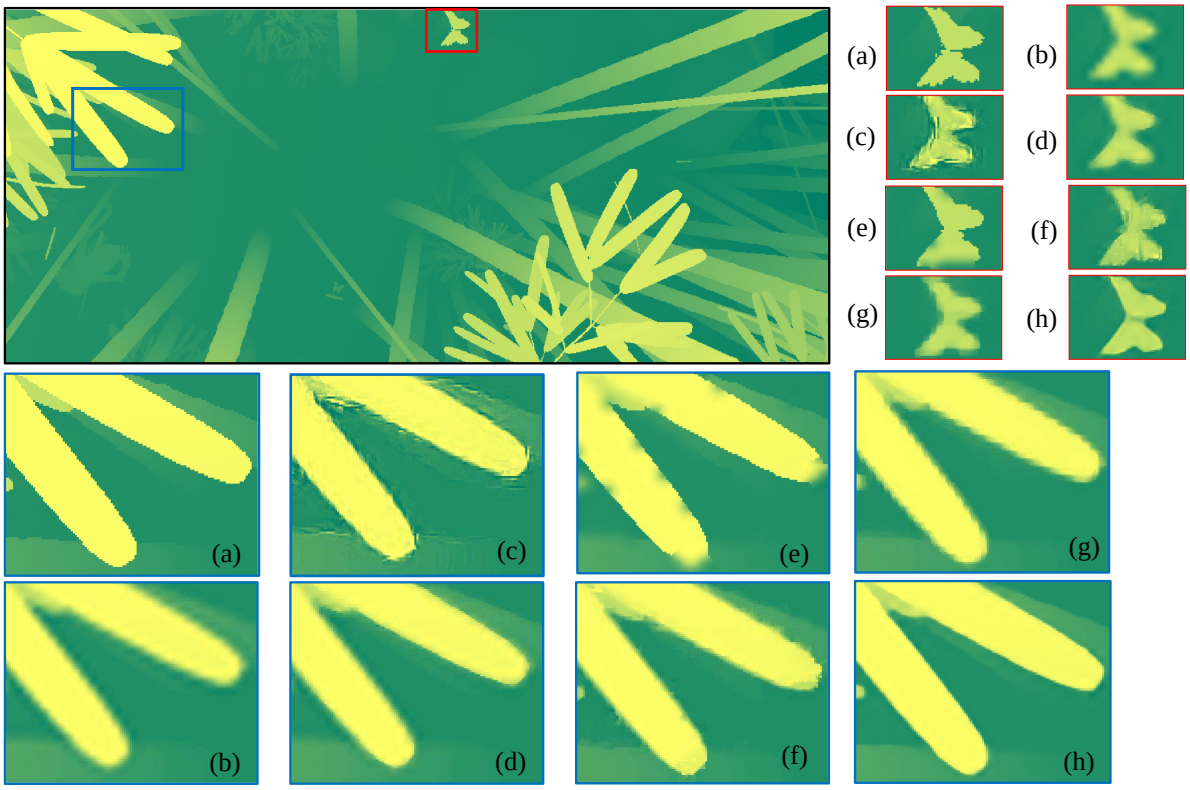


Fig. 9. Visual comparison of *Bamboo* in Sintel dataset with upscaling factor = 4. (a) Ground truth. (b) Bicubic. (c) Xie *et al.* [8]. (d) Timofte *et al.* [32]. (e) Lu *et al.* [27]. (f) Ferstl *et al.* [26]. (g) Gu *et al.* [29]. (h) Our method.

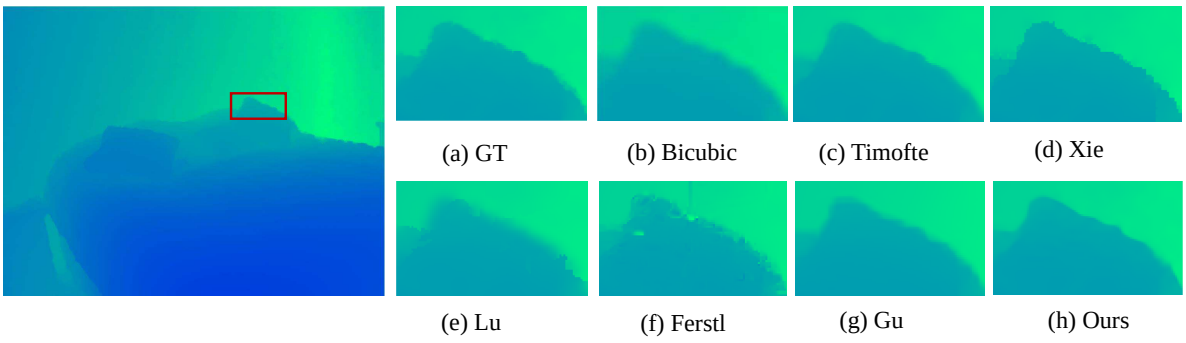


Fig. 10. Visual comparison of *Image1* in NYU dataset with upscaling factor = 4. (a) Ground truth. (b) Bicubic. (c) Timofte *et al.* [32]. (d) Xie *et al.* [8]. (e) Lu *et al.* [27]. (f) Ferstl *et al.* [26]. (g) Gu *et al.* [29]. (h) Our method.

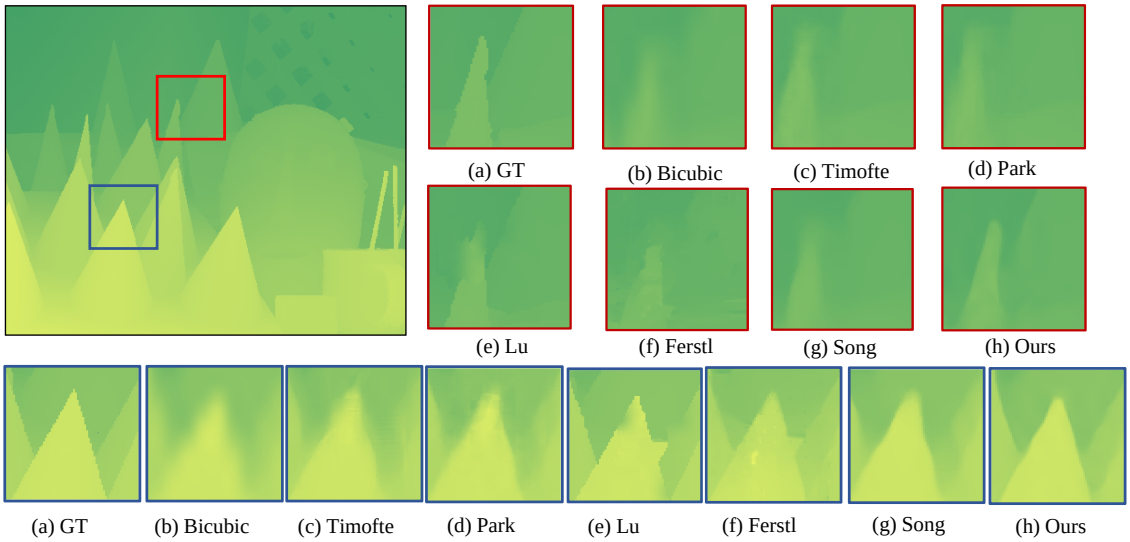


Fig. 11. Visual comparison of *Cones* in Middlebury dataset with upscaling factor $= 8$. (a) Ground truth. (b) Bicubic. (c) Timofte *et al.* [32]. (d) Xie *et al.* [8]. (e) Lu *et al.* [27]. (f) Ferstl *et al.* [26]. (g) Song *et al.* [7]. (h) Our method.

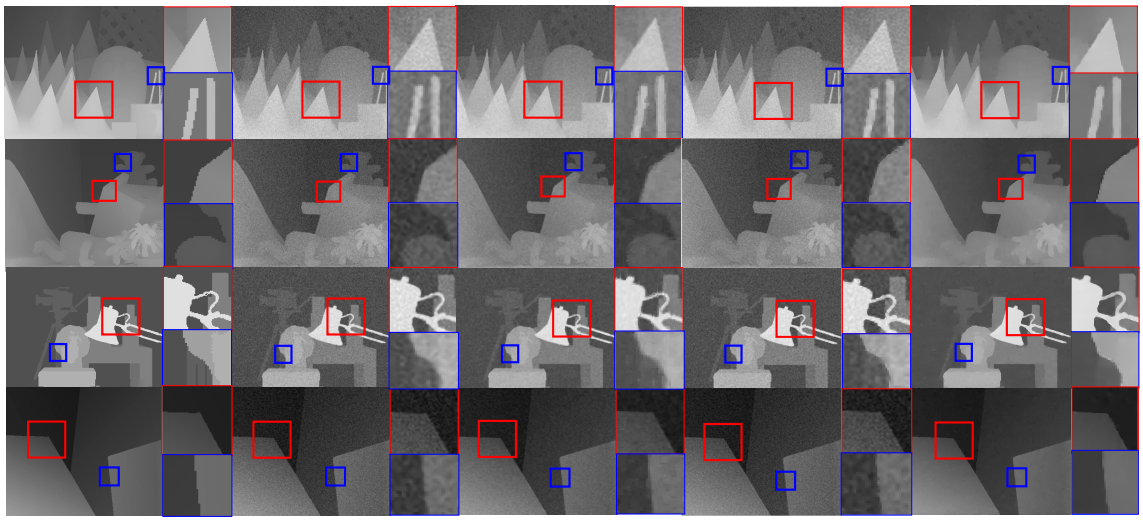


Fig. 12. Results on noisy Middlebury dataset for $2\times$ upscaling, with the variance of noise $\sigma^2 = 65$. (a) Ground truth, (b) Wang *et al.* [36], (c) Ferstl *et al.* [26], (d) Our method with noise-free version, (e) Our method with noisy version. The RMSE value is averaged among four images.

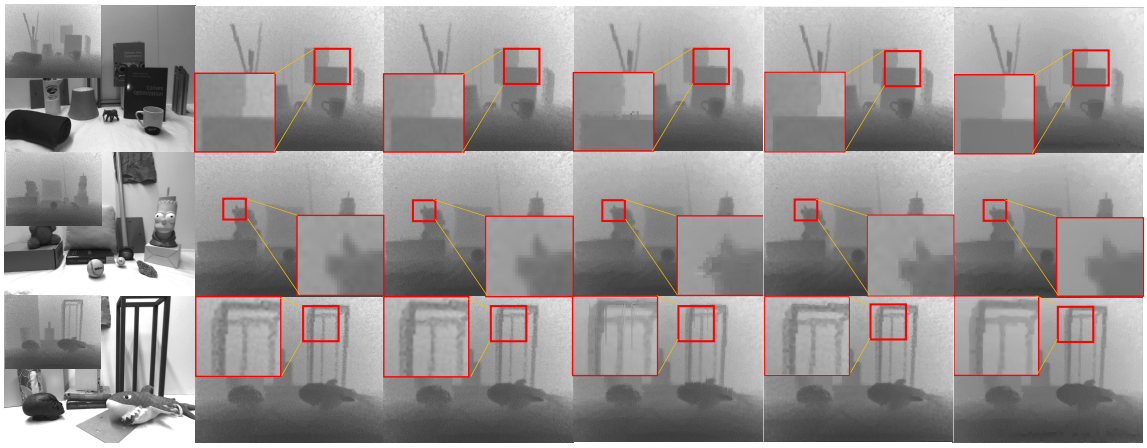


Fig. 13. Qualitative results on the real-world ToFMark [26] dataset for $2\times$ upscaling. (a) Ground truth, (b) bicubic interpolation, (c) Wang *et al.* [36], (d) Ferstl *et al.* [26], (e) Our method with noise-free version, (f) Our method with noisy version. The variances of noise are $\sigma^2 = 4.58$, 3.80 and 4.21 for input noisy depth images from top to down, which are estimated using method proposed by Liu *et al.* [55].

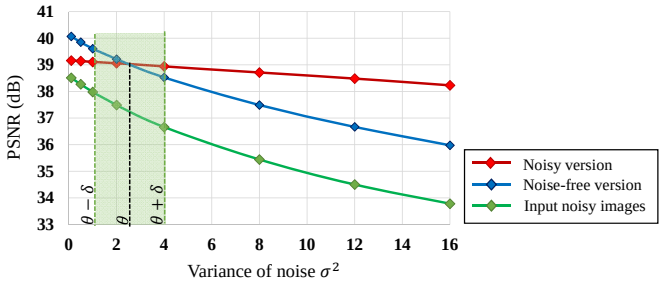


Fig. 14. PSNR versus noise variance σ^2 when applying noise-free and noisy versions of our algorithm on the noisy images.

REFERENCES

[1] R. Yonetani, A. Kimura, H. Sakano, and K. Fukuchi, "Single image segmentation with estimated depth." in *BMVC*. Citeseer, 2012, pp. 1–11.

[2] M. J. Dahan, N. Chen, A. Shamir, and D. Cohen-Or, "Combining color and depth for enhanced image segmentation and retargeting," *The Visual Computer*, vol. 28, no. 12, pp. 1181–1193, 2012.

[3] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt *et al.*, "Real-time non-rigid reconstruction using an RGB-D camera," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 156, 2014.

[4] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 1–7.

[5] M. Hansard, S. Lee, O. Choi, and R. P. Horaud, *Time-of-flight cameras: principles, methods and applications*. Springer Science & Business Media, 2012.

[6] G. Riegler, M. Rüther, and H. Bischof, "ATGV-net: Accurate depth super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 268–284.

[7] X. Song, Y. Dai, and X. Qin, "Deep depth super-resolution: Learning depth super-resolution using deep convolutional neural network," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 360–376.

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

[9] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, 2002.

[10] J. A. Urigüen, T. Blu, and P. L. Dragotti, "FRI sampling with arbitrary kernels," *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5310–5323, 2013.

[11] P. Song, X. Deng, M. Joao, D. Nikos, P. L. Dragotti, and M. Rodrigues, "Multimodal image super-resolution via joint sparse representations induced by coupled dictionaries," *IEEE Transactions on Image Processing, under review. (preprint available at https://arxiv.org/abs/1709.08680.)*.

[12] O. Mac Aodha, N. D. Campbell, A. Nair, and G. J. Brostow, "Patch based synthesis for single depth image super-resolution," in *European Conference on Computer Vision*. Springer, 2012, pp. 71–84.

[13] J. Li, Z. Lu, G. Zeng, R. Gan, and H. Zha, "Similarity-aware patchwork assembly for depth image super-resolution," in *the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3374–3381.

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

[15] D. Ferstl, M. Ruther, and H. Bischof, "Variational depth superresolution using example-based edge representations," in *the IEEE International Conference on Computer Vision*, 2015, pp. 513–521.

[16] M. Hornácek, C. Rhemann, M. Gelautz, and C. Rother, "Depth super resolution by rigid body self-similarity in 3d," in *the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1123–1130.

[17] M. Liu, M. Salzmann, and X. He, "Discrete-continuous depth estimation from a single image," in *the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 716–723.

[18] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *the IEEE International Conference on Computer Vision*, 2015, pp. 2650–2658.

[19] B. Li, C. Shen, Y. Dai, A. Van Den Hengel, and M. He, "Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs," in *the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1119–1127.

[20] A. Roy and S. Todorovic, "Monocular depth estimation using neural regression forest," in *the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5506–5514.

[21] Y. Kuznietsov, J. Stuckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6647–6655.

[22] Q. Yang, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," in *Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[23] D. Chan, H. Buisman, C. Theobalt, and S. Thrun, "A noise-aware filter for real-time depth upsampling," in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 2008*, 2008.

[24] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon, "High quality depth map upsampling for 3D-TOF cameras," in *the IEEE International Confernce on Computer Vision*. IEEE, 2011, pp. 1623–1630.

[25] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

[26] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *the IEEE International Conference on Computer Vision*, 2013, pp. 993–1000.

[27] J. Lu and D. Forsyth, "Sparse depth super resolution," in *the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2245–2253.

[28] H. Kwon, Y.-W. Tai, and S. Lin, "Data-driven depth map refinement via multi-scale sparse representation," in *the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 159–167.

[29] S. Gu, W. Zuo, S. Guo, Y. Chen, C. Chen, and L. Zhang, "Learning dynamic guidance for depth image enhancement," *the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 10, no. y2, p. 2, 2017.

[30] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[31] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730.

[32] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 111–126.

[33] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206.

[34] J.-J. Huang, W.-C. Siu, and T.-R. Liu, "Fast image interpolation via random forests," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3232–3245, 2015.

[35] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*. Springer, 2014, pp. 184–199.

[36] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *the IEEE International Conference on Computer Vision*, 2015, pp. 370–378.

[37] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[38] D. Liu, Z. Wang, N. Nasrabadi, and T. Huang, "Learning a mixture of deep networks for single image super-resolution," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 145–156.

[39] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[40] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[41] I. Tosic and S. Drewes, "Learning joint intensity-depth sparse representations," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2122–2132, 2014.

[42] P. Song, J. F. Mota, N. Deligiannis, and M. R. D. Rodrigues, "Coupled dictionary learning for multimodal image super-resolution," in *IEEE*

Global Conference on Signal and Information Processing (GlobalSIP). IEEE, 2016, pp. 162–166.
[43] P. L. Dragotti, M. Vetterli, and T. Blu, "Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang–Fix," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1741–1757, 2007.
[44] S. Mallat, *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.
[45] X. Wei and P. L. Dragotti, "FRESH: FRI-based single-image super-resolution algorithm," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3723–3735, 2016.
[46] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Single-image super-resolution via linear mapping of interpolated self-examples," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5334–5347, 2014.
[47] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4680–4688, 2011.
[48] M. Aharon, M. Elad, and A. Bruckstein, "rmk-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
[49] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.
[50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
[51] S. Martull, M. Peris, and K. Fukui, "Realistic CG stereo image dataset with ground truth disparity maps," in *ICPR workshop*, vol. 111, no. 430, 2012, pp. 117–118.
[52] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *the IEEE Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2003, pp. I–I.
[53] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conf. on Computer Vision (ECCV)*, ser. Part IV, LNCS 7577, A. Fitzgibbon et al. (Eds.), Ed. Springer-Verlag, Oct. 2012, pp. 611–625.
[54] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," *European Conference on Computer Vision*, pp. 746–760, 2012.
[55] X. Liu, M. Tanaka, and M. Okutomi, "Single-image noise level estimation for blind denoising," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5226–5237, 2013.
[56] L. Baboulaz and P. L. Dragotti, "Exact feature extraction using finite rate of innovation principles with an application to image super-resolution," *IEEE Transactions on Image Processing*, vol. 18, no. 2, pp. 281–298, 2009.
[57] R. Prony, "Essai expmental et analytique sur les lois de la dilatabilit fluides stiques et sur celles de la force expansive de la vapeur de leau et de la vapeur de lalkool, ffntes temptures,," in *Journal de Ecole Polytechnique*, 1975, pp. 24–76.

Xin Deng (S'14) received the Bachelor and Master degree in electronic engineering from Beihang University, Beijing, China, in 2013 and 2016. She is currently pursuing PhD degree in Department of Electrical and Electronic Engineering, Imperial College London, UK. Her research interests include sparse coding with applications in image and video processing, machine learning, and multimodal signal processing. Ms. Deng received the National Scholarship from the Chinese College Students and the Outstanding Graduate Student Award from Beihang University. She also received the 2014 IEEE Circuits and Systems Society Student Travel Award.

Pingfan Song is a research associate at Imperial College London (ICL). He obtained the Ph.D. degree, supported by Overseas Research Scholarships (ORS), at University College London (UCL), the master and bachelor degree both at Harbin Institute of Technology (HIT). His research interests lie in signal/image processing, machine learning, dictionary learning, etc. with applications on multimodal image denoising, super-resolution, reconstruction for a variety of image modalities.

Miguel Rodrigues is a Reader in Information Theory and Processing with the Department of Electronic and Electrical Engineering, University College London, and a Faculty Fellow with the Turing Institute. He was previously with the CS Dept., Univ. Porto, Portugal, rising through the ranks from Assistant to Associate Professor. He also held research positions at Princeton U. Cambridge U., and Duke U. His research interests – which lie in the general areas of information theory and processing – have led to nearly 200 publications in leading journals and conferences in the field, including the prestigious IEEE Communications and Information Theory Societies Joint Paper Award 2011. He is co-author of a book on "Information- Theoretic Methods in Data Science" to be published by Cambridge Univ Press.

Pier Luigi Dragotti is Professor of Signal Processing in the Electrical and Electronic Engineering Department at Imperial College London. He received the Laurea Degree (summa cum laude) in Electronic Engineering from the University Federico II, Naples, Italy, in 1997; the Master degree in Communications Systems from the Swiss Federal Institute of Technology of Lausanne (EPFL), Switzerland in1998; and PhD degree from EPFL, Switzerland, in April 2002. He has held several visiting positions. In particular, he was a visiting student at Stanford University, Stanford, CA in 1996, a summer researcher in the Mathematics of Communications Department at Bell Labs, Lucent Technologies, Murray Hill,NJ in 2000 and a visiting scientist at Massachusetts Institute of Technology(MIT) in 2011. Before joining Imperial College in November 2002, he was a senior researcher at EPFL working on distributed signal processing for sensor networks for the Swiss National Competence Center in Research on Mobile Information and Communication Systems. Dr Dragotti was Technical Co-Chair for the European Signal Processing Conference in 2012, Associate Editor of the IEEE Transactions on Image Processing from 2006 to 2009, a member of the IEEE Image, Video and Multidimensional Signal Processing Technical Committee and a member of the IEEE Signal Processing Theory and Methods Technical Committee. He was also the recipient of an ERC investigator award. Currently, he is Editor-in-Chief of the IEEE Transactions on Signal Processing, and a member of the IEEE Computational Imaging Technical Committee. His research interests include sampling theory, wavelet theory and its applications, sparsity-driven signal processing with application in image super-resolution, neuroscience and computational imaging.