Imperial College London
Department of Electrical and Electronic Engineering

Single and Multi-modal Image Super-resolution: From Sparse Modelling to Deep Neural Network

Xin Deng

Supervised by Professor Pier Luigi Dragotti

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Electrical and Electronic Engineering of Imperial College
London and the Diploma of Imperial College London, 2019

Declaration of Originality

I herewith declare that the content of this thesis is the product of my own work under the guidance of my PhD supervisor Prof. Pier Luigi Dragotti. All material that is not my own work has been properly acknowledged.

Xin Deng

Copyright Declaration

The copyright of this thesis resets with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Abstract

Single image super-resolution is a long studied inverse problem, which aims to infer a high-resolution (HR) image from a single low-resolution (LR) one. Sometimes, we can infer an HR image not only from its corresponding LR image, but also with the guidance of an image from a different modality, e.g., RGB guided depth image super-resolution. This is called multi-modal image super-resolution. In this thesis, we develop methods based on sparse modelling and deep neural network to tackle both the single and multi-modal image super-resolution problems. We also extend the applications to the general multi-modal image restoration and image fusion tasks.

Firstly, we present a method for single image super-resolution, which aims to achieve the best trade-off between the objective and perceptual image quality. In our method, we show that the objective and perceptual quality are influenced by different elements of an image, and we use stationary wavelet transform to separate these elements. A novel wavelet domain style transfer algorithm is proposed to achieve the best trade-off between the image distortion and perception.

Next, we develop a robust algorithm for RGB guided depth image super-resolution, through combining the finite rate of innovation (FRI) theory and a multi-modal dictionary learning algorithm. In addition, to speed up the super-resolution process, we introduce a projection-based rapid upscaling algorithm to pre-calculate the projections from the joint LR depth and HR intensity pairs to the HR depth. We demonstrate that this method only needs a fraction of training data but can achieve state-of-the-art performance and is resilient to noisy condition and unknown blurring kernels.

For the general multi-modal image super-resolution problems, we propose a deep coupled ISTA network based on joint sparse modelling across modalities. The network architecture is derived by unfolding the iterative shrinkage and thresholding algorithm (ISTA) used in the model. Moreover, since the network initialization plays an important role in deep network training, we further propose a layer-wise optimization algorithm to initialize the network parameters before running the back-propagation algorithm. We demonstrate that the new initialization algorithm is effective, and the coupled ISTA network consistently outperforms other methods both quantitatively and qualitatively for various multi-modal scenarios, including RGB/depth, RGB/multi-spectral and RGB/near-infrared images.

Finally, we extend the scope of our research from the multi-modal image super-resolution to the more general case of multi-modal image restoration (MIR) and multi-modal image fusion (MIF). We develop a novel deep Common and Unique information splitting network (CU-Net), whose network architecture is designed by drawing inspirations from a new proposed multi-modal convolutional sparse coding model. The CU-Net has good interpretability, and can automatically split the common information shared among different modalities, from the unique information owned by each single modality. We demonstrate the effectiveness of our method on a variety of MIR and MIF tasks, including RGB guided depth image super-resolution, flash guided non-flash image denoising, multi-focus and multi-exposure image fusion.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my PhD supervisor, Prof. Pier Luigi Dragotti. I really appreciate his meeting with me and also every member in our group every week. He can always give me helpful advices to solve my research confusions using his wisdom and experience. I have learned a lot from him, not only about how to do the academic research, but also how to be a good scholar. He also gave me many opportunities to attend conferences and summer school, and to co-supervise the Master and undergraduate students. I especially want to thank him for his support and encouragement whenever I feel anxious and unconfident about my research.

I would like to thank my Master supervisor, Prof. Mai Xu from Beihang University. He recommended me to my PhD supervisor, and we still keep in good contact in my PhD study. We also have cooperation in some academic research and published some papers.

I also want to thank our group members, Roxana, Su, Pingfan, Junjie, Vincent, Herman, Ecenaz, Maria and Marek. Special thanks to my friends, Maomao and Ying for sharing my every good and bad time in London; thanks to my flat mates, Pengchao and Qi, for accompany me going home and making delicious food together; thanks to all my PhD friends who share the long journey with me.

Finally, I would like to thank my family who give me endless love, support and understanding when I encounter difficulties in my life abroad, especially to thank my mom for giving me a video chat every day to comfort and support me. I also would like to thank the Electrical and Electronic Engineering department and Chinese Scholarship Council, for funding my PhD research for 3.5 years.

Abbreviations

SISR	Single image super-resolution
MSE	Mean squared error
LR	Low resolution
HR	High resolution
MISR	Multi-modal image super-resolution
FRI	Finite rate of innovation
SWT	Stationary wavelet transform
ISWT	Inverse stationary wavelet transform
DWT	Discrete wavelet transform
IDWT	Inverse discrete wavelet transform
DCT	Discrete cosine transform
ToF	Time-of-flight
CD	Common dictionary
UD	Unique dictionary
OMP	Orthogonal matching pursuit
MQ	Moderate quality
FDU	FRI-based depth upscaling
TDL	Triple dictionary learning
PRU	Projection-based rapid upscaling
RMSE	Root mean square error
PSNR	Peak signal-to-noise ratio
SSIM	Structural SIMilarity
JMDL	Joint multi-modal dictionary learning
ISTA	Iterative shrinkage and thresholding algorithm
LOA	Layer-wise optimization algorithm
MS	Multi-spectral
NIR	Near-infrared
MCSC	multi-modal convolutional sparse coding
CNN	Convolutional neural network

GAN	Generative adversarial network
PD	Perception-distortion
WDST	Wavelet domain style transfer
JBF	Joint bilateral filter
GF	Guided filter
WMF	Weighted median filter
RGF	Rolling guided filter
DJF	Deep joint filtering
MRI	Magnetic resonance imaging
AMP	Approximate message passing
ADMM	Alternating direction method of multipliers
PGD	Projected gradient descent
MIR	Multi-modal image restoration
MIF	Multi-modal image fusion
UFEM	Unique feature extraction module
CFPM	Common feature preservation module
IRM	Image reconstruction module
LCSC	Learned convolutional sparse coding

Notations

$\|\cdot\|_F^2$	Frobenius norm
$\|\cdot\|_0$	ℓ_0 norm
$\|\cdot\|_1$	ℓ_1 norm
$\|\cdot\|_2$	ℓ_2 norm
$S_\lambda(\cdot)$	Soft-thresholding operation with threshold as λ
$*$	Convolution operation
$\boldsymbol{A}_p$	Super-resolved image with high perceptual quality
$\boldsymbol{A}_o$	Super-resolved image with high objective quality
$\boldsymbol{A}_r$	Combined image using WDST algorithm
$\boldsymbol{F}^l(\cdot)$	Feature map of layer l
$\boldsymbol{G}^l(\cdot)$	Gram matrix of layer l
$\boldsymbol{X}$	Training samples for low-resolution images
$\boldsymbol{Y}$	Training samples for guidance images
$\boldsymbol{Z}$	Training samples for high-resolution images
$\boldsymbol{\Psi}_c^l$	Common dictionary for low-resolution images
$\boldsymbol{\Psi}_u^l$	Unique dictionary for low-resolution images
$\boldsymbol{\Psi}_c^h$	Common dictionary for high-resolution images
$\boldsymbol{\Psi}_u^h$	Unique dictionary for high-resolution images
$\boldsymbol{\Phi}_c^h$	Common dictionary for guidance images
$\boldsymbol{\Phi}_u^h$	Unique dictionary for guidance images
$\boldsymbol{C}$	Common sparse representations among $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$
$\boldsymbol{U}$	Unique sparse representations of $\boldsymbol{X}$ and $\boldsymbol{Z}$
$\boldsymbol{V}$	Unique sparse representations of $\boldsymbol{Y}$
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{D}_x$	Dictionary for $\boldsymbol{X}$
$\boldsymbol{D}_y$	Dictionary for $\boldsymbol{Y}$
$\boldsymbol{D}_z$	Dictionary for $\boldsymbol{Z}$
$\boldsymbol{\Gamma}_x$	Sparse representations for $\boldsymbol{X}$
$\boldsymbol{\Gamma}_y$	Sparse representations for $\boldsymbol{Y}$
$\boldsymbol{\Gamma}_z$	Sparse representations for $\boldsymbol{Z}$
$\boldsymbol{W}_x$, $\boldsymbol{W}_y$	Transform matrices to map $\boldsymbol{\Gamma}_x$ and $\boldsymbol{\Gamma}_y$ to $\boldsymbol{\Gamma}_z$

$\boldsymbol{x}$	The input distorted image
$\boldsymbol{y}$	The guidance image
$\boldsymbol{z}$	The target image
$\boldsymbol{d}_k^c$	The k-th common filter to represent $\boldsymbol{x}$
$\boldsymbol{d}_k^u$	The k-th unique filter to represent $\boldsymbol{x}$
$\boldsymbol{h}_k^c$	The k-th common filter to represent $\boldsymbol{y}$
$\boldsymbol{h}_k^u$	The k-th unique filter to represent $\boldsymbol{y}$
$\boldsymbol{g}_k^c$	The k-th common filter to represent $\boldsymbol{z}$
$\boldsymbol{g}_k^u$	The k-th unique filter to represent $\boldsymbol{z}$
$\boldsymbol{c}_k$	the k-th common feature response of $\boldsymbol{x}$ and $\boldsymbol{y}$
$\boldsymbol{u}_k$	the k-th unique feature response of $\boldsymbol{x}$
$\boldsymbol{v}_k$	the k-th unique feature response of $\boldsymbol{y}$

Contents

Declaration of Originality i

Abstract v

Acknowledgements vii

Abbreviations vii

Notations xi

1 Introduction 1

1.1 Motivation and Objectives 1

1.2 Original contribution and outline of thesis 3

1.3 Publications 5

2 Background 7

2.1 Single image super-resolution 7

2.1.1 Objective quality driven SISR 7

2.1.2 Perceptual quality driven SISR 10

2.1.3 Perception-distortion trade-off in SISR 12

2.2 Multi-modal image restoration and image fusion 14
2.2.1 Multi-modal image restoration 14
2.2.2 Multi-modal image fusion 16
2.3 Iterative unfolding strategy 17

3 Perception-distortion Tradeoff in Single Image Super-Resolution 19

3.1 Introduction 19
3.2 Proposed Method 21
3.2.1 Stationary wavelet transform 21
3.2.2 Motivation 21
3.2.3 Low-frequency sub-band enhancement 24
3.2.4 Wavelet domain style transfer 25
3.3 Numerical results 27
3.3.1 Experimental setup 27
3.3.2 Effectiveness of wavelet domain style transfer 28
3.3.3 Perception-distortion (PD) performance 29
3.3.4 Ablation study 30
3.4 Summary 32

4 RGB guided depth image super-resolution 34

4.1 Introduction 34
4.2 Proposed Method 36
4.2.1 FRI-based Depth Upscaling 36

4.2.2 Triple Dictionary Learning 40
4.2.3 Projection-based Rapid Upscaling 43
4.3 Numerical results 47
4.3.1 Datasets 47
4.3.2 Experimental setup 49
4.3.3 Performance Analysis of Our Method 50
4.3.4 Numerical Comparison Against Other Methods 52
4.3.5 Qualitative results 54
4.3.6 Noisy case 60
4.4 Summary 62

5 Coupled ISTA Network for Multi-modal Image Super-Resolution 63

5.1 Introduction 63
5.2 Proposed Method 64
5.2.1 Joint multi-modal dictionary learning 64
5.2.2 Deep coupled ISTA network 68
5.2.3 Layer-wise optimization algorithm 70
5.3 Numerical results 74
5.3.1 Experimental setup 74
5.3.2 Performance analysis of our method 75
5.3.3 Quantitative comparison against other methods 79
5.3.4 Qualitative comparison against other methods 82
5.3.5 Noisy case 86

5.4 Summary 89

6 Deep CNN for Multi-modal Image Restoration and Fusion 90

6.1 Introduction 90

6.2 Multi-modal convolutional sparse coding 91

6.2.1 The MCSC model 92

6.2.2 Convolutional sparse coding 94

6.3 Common and unique information splitting network (CU-Net) 96

6.3.1 Unique feature extraction module 96

6.3.2 Common feature preservation module 97

6.3.3 Image reconstruction module 98

6.3.4 Discussion about the CU-Net architecture 99

6.4 Numerical results 99

6.4.1 Experimental setup 100

6.4.2 MIR related tasks 100

6.4.3 MIF related tasks 105

6.4.4 Common and unique reconstruction visualization 108

6.4.5 Ablation study 110

6.4.6 Running speed 113

6.5 Summary 113

7 Conclusion 114

7.1 Summary of Thesis Achievements 114

7.2 Future Work 117

7.2.1 Fast wavelet domains style transfer 117

7.2.2 Perceptual based multi-modal image restoration 118

7.2.3 Extension from image to video processing 118

7.2.4 Joint multi-modal image compression 119

A Chapter 4 **120**

A.1 FRI reconstruction on 1-D signal 120

B Chapter 6 **123**

B.1 Learned Convolutional Sparse Coding [1] 123

Bibliography **124**

List of Tables

3.1 PSNR and NRQM scores on the BSD100 dataset. 23

3.2 Benchmark comparisons for 4× upscaling, with the best results bold and the second bests underlined. 31

3.3 Effects of wavelet filter on Set 14 dataset. 31

3.4 Ablation study of WDST on each sub-band. 31

4.1 Notation list. 37

4.2 The proposed projection-based rapid upscaling algorithm 48

4.3 Testing datasets 48

4.4 Individual contributions of each element on the four noise-free datasets. . . 51

4.5 RMSE comparison with and without the guidance of intensity images at different scaling factors on the four noise-free datasets. 51

4.6 OMP versus PRU for both clustering and non-clustering cases on Middlebury dataset. 51

4.7 Results on Middlebury dataset for 2× and 4× upscaling. The best results are in bold and the second bests are underlined. 55

4.8 Results on the NT, Sintel and NYU datasets for 4× upscaling. The best results are in bold and the second bests are underlined. 56

4.9 Results on the Middlebury dataset for 8× and 16× upscaling. The best results are in bold and the second bests are underlined. 57

4.10 Performance of Our and other methods for 4× upscaling, with arbitrary blurring kernels. 57

4.11 Running time in seconds of our method on Middlebury dataset. 58

4.12 Quantitative results of RMSE on noisy Middlebury dataset. 58

5.1 Comparisons of RMSE and PSNR among different initialization methods on RGB/depth dataset. 79

5.2 Influence of number of layers. K=4 is the default number of layers. 79

5.3 Results in terms of PSNR (dB) on the RGB/depth, RGB/MS and RGB/NIR datasets for 4× upscaling, with the best results in bold and the second best results underlined. 81

5.4 Results in terms of PSNR (dB) on the RGB/depth, RGB/MS and RGB/NIR datasets datasets for 8× upscaling, with the best results in bold and the second best results underlined. 83

5.5 Results in terms of PSNR (dB) on the RGB/depth, RGB/MS and RGB/NIR datasets datasets for 16× upscaling, with the best results in bold and the second best results underlined. 84

5.6 The running time in seconds of our method with the corresponding PSNR values for RGB/MS dataset. 84

5.7 The PSNR results on the noisy RGB/depth, RGB/MS and RGB/NIR datasets for 4× upscaling, with the best results in bold. Our* indicates our method with re-trained models using the noisy dataset. 88

6.1 Summary of notations 92

6.2 Results on the RGB/depth dataset for 4× upscaling, with the best results in bold and the second best results underlined. Note that RADAR and CoISTA are the methods introduced in Chapters 4 and 5, respectively, in this thesis. 101

6.3 Results on the RGB/MS dataset for 4× upscaling, with the best results in bold and the second best results underlined. Note that CoISTA is the method introduced in Chapter 5 in this thesis. 103

6.4 Results on the Flash/non-flash dataset for image denoising, with the best results in bold and the second best results underlined. 103

6.5 Importance of the residual module architecture. 112

6.6 Effects of the filter size. 112

6.7 Effects of the network depth in each module. 112

6.8 The average running time (in seconds) of our method on various tasks. . . . 112

List of Figures

2.1 The network architecture of FSRCNN [2]. 9

2.2 The network architecture of SRGAN [3]. 11

2.3 Perception-distortion performance of different SISR methods. 12

3.1 Perception-distortion performance of different SISR methods. The blue points represent methods aiming for high objective quality, the green points represent methods aiming for good perceptual quality, and the orange points represent methods aiming for a trade-off between perception and distortion. The higher PSNR value indicates better objective quality and the higher perceptual score indicates better perceptual quality. The bottom left corner is the best. Our method achieves the best trade-off among all the "orang" methods. 20

3.2 Illustration of two level 2D stationary wavelet transform (SWT) of image X, with H_0 and G_0 as the low-pass and high-pass filters, respectively. . . . 22

3.3 The first row shows the histograms of different sub-bands of A_p which has high perceptual quality but low objective quality. The second row shows the histograms of different sub-bands of A_o which has high objective quality but low perceptual quality. The third row shows the ground-truth histograms. All the histograms are drawn in RGB colour space. 23

3.4 (a) shows the framework of our method, (b) illustrates the wavelet domain style transfer (WDST) algorithm, and (c) shows the low-frequency sub-band enhancement (LSE) network. 24

3.5 The first row shows the restored images of *Zebra* in Set 14 using EDSR, CX and our method, with the red values indicating the PSNR/NRQM values. The second row visualizes the HL sub-bands of the images in the first row, together with the histograms. 28

3.6 Visual comparisons of image from BSD100 for 4× upscaling. The red numbers indicate the PSNR and NRQM values. 29

3.7 Visual comparisons of image from Urban100 for 4× upscaling. The red numbers indicate the PSNR and NRQM values. 30

3.8 The perception-distortion (PD) curves of EDSR and CX, RCAN and CX, SRResNet-MSE and SRGAN-vgg54. 32

4.1 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. 35

4.2 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. 36

4.3 (a) shows the decomposition of one image using 2D DWT, and (b) visualizes the four sub-bands of image *Cones*. 38

4.4 FDU algorithm with clean LR depth image. 39

4.5 FDU algorithm with noisy LR depth image. 39

4.6 (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 J of each subspace. The results are from the validation dataset, which is composed of 10 images from the New Tsukuba dataset [4]. 49

4.7 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. 50

4.8 Visual comparison of *Tsukuba* in Middlebury dataset with upscaling factor = 4. (a) Ground truth. (b) Bicubic. (c) Ferstl *et al.* [5]. (d) Xie *et al.* [6]. (e) Lu *et al.* [7]. (f) Gu *et al.* [8]. (g) Song *et al.* [9]. (h) Our method. . . . 58

4.9 Visual comparison of *Bamboo* in Sintel dataset with upscaling factor = 4. (a) Ground truth. (b) Bicubic. (c) Xie *et al.* [6]. (d) Timofte *et al.* [10]. (e) Lu *et al.* [7]. (f) Ferstl *et al.* [5]. (g) Gu *et al.* [8]. (h) Our method. . . 59

4.10 Visual comparison of *Cones* in Middlebury dataset with upscaling factor = 8. (a) Ground truth. (b) Bicubic. (c) Timofte *et al.* [10]. (d) Xie *et al.* [6]. (e) Lu *et al.* [7]. (f) Ferstl *et al.* [5]. (g) Song *et al.* [9]. (h) Our method. 59

4.11 Results on noisy Middlebury dataset for 2× upscaling, with the variance of noise $\sigma^2 = 65$. (a) Ground truth, (b) Wang *et al.* [11], (c) Ferstl *et al.* [5], (d) Our method with noise-free version, (e) Our method with noisy version. The RMSE value is averaged among four images. 61

4.12 Qualitative results on the real-world ToFMark [5] dataset for 2× upscaling. (a) Ground truth, (b) bicubic interpolation, (c) Wang *et al.* [11], (d) Ferstl *et al.* [5], (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.* [12]. 61

4.13 PSNR versus noise variance σ^2 when applying noise-free and noisy versions of our algorithm on the noisy images. 62

5.1 The architecture of proposed deep coupled ISTA network. The dashed line indicates that the mean value of patch $\boldsymbol{x}$ is added to the output patch $\boldsymbol{z}$. . 69

5.2 A single layer obtained by unfolding Eq. (5.19). 70

5.3 The convergence behaviour of our JMDL algorithm. The images above the curve are examples of the reconstructed results at different iterations. . . . 76

5.4 The change of MSE with the iterations in the first and second layers. 77

5.5 The training loss across 50 epochs with different initialization methods for different settings of dictionary size m and number of layers K. 78

5.6 Visual comparisons of $Urban_3$ in RGB/NIR dataset with upscaling factor = 4. The first row shows the error maps between reconstructed images and the ground-truth images using different methods. The second row shows the corresponding reconstructed images, with the numbers in red indicating the PSNR values. 85

5.7 Visual comparisons of $Cave$ in Sintel dataset with upscaling factor = 8 using different methods. (a) Ground truth, (b) Bicubic, (c) Park *et al.* [13], (d) Lu *et al.* [7], (e) Gu *et al.* [8], (f) Ferstl *et al.* [14], (g) SCN [15], (h) VDSR [16], (i) Ours. The numbers in red indicate the RMSE values. . . . 85

5.8 Visual comparisons of $Egyptian$ in RGB/MS dataset with upscaling factor = 8. The first row shows the error maps between reconstructed images and the ground-truth images using different methods. The second row enlarges some regions in the reconstructed images for better comparison. The numbers in red indicate3 the PSNR values. 87

5.9 Visual comparisons of Art in Middlebury dataset with upscaling factor = 16. Some regions are enlarged for better viewing. The red numbers inside indicate the RMSE values with different methods. 87

5.10 Visual comparison of 4× upscaling results in noisy case with $\sigma^2 = 2$. Some regions are enlarged for better comparison. The symbol * indicates that the results are obtained by re-training the models using the noisy datasets. 89

6.1 Examples of different multi-modal image restoration and fusion tasks. The first row shows the MIR related applications, including RGB guided depth image SR, RGB guided multi-spectral image SR, and flash guided non-flash image denoising. The second row shows the MIF related applications, including multi-exposure image fusion, multi-focus image fusion, and medical image fusion. 91

6.2 Network Architecture of the proposed CU-Net. For the MIR related tasks, the final reconstruction (Point 4) is composed of the common reconstruction (Point 1) and the unique reconstruction (Point 2). For the MIF related tasks, the final reconstruction is composed of the common reconstruction (Point 1) and the two unique reconstructions (Point 2 and Point 3). 95

6.3 The network architectures of UFEM and CFPM. 95

6.4 Visual comparisons of RGB guided depth SR results for 4× upscaling: (a) Bicubic, (b) Lu *et al.* [7], (c) Xie *et al.* [6], (d) Gu *et al.* [8], (e) SCN [15], (f) DGF [17], (g) DJFR [18], (h) RADAR [19], (i) Our CU-Net, (j) Ground truth. 104

6.5 Visual comparisons of RGB guided multi-spectral image SR results for 4× upscaling. The first row shows the super-resolved images using different methods, and the second row shows the error maps between the images in the first row and the ground-truth. 104

6.6 Visual comparisons of flash guided non-flash image denoising results with $\sigma^2 = 75$: (a) Noisy non-flash input image, (b) MuGIF [20], (c) CBM3D [21], (d) DnCNN [22], (e) Our CU-Net. 106

6.7 Visual comparisons of multi-exposure image fusion results: (a) input under-exposed image, (b) input over-exposed image, (c) SPD-MEF [23], (d) MEF-OPT [24], (e) our CU-Net. The last row shows the enlarged regions in the corresponding images. Better view in electronic version. 108

6.8 Visual comparisons of multi-focus image fusion results. (a) The near-focus image, (b) The far-focus image, (c) CSR [25], (d) Deepfuse network [26], (e) Densefuse network [27], (f) our CU-Net. Better view in electronic version. 109

6.9 Visual comparisons of medical image fusion results. (a) T1-weighted MR image, (b) T2-weighted MR image, (c) CSR [25], (d) Deepfuse network [26], (e) Densefuse network [27], (f) our CU-Net. Better view in electronic version. 110

6.10 Visualization of the reconstructed image of different parts in the CU-Net for the RGB guided depth SR task. (b) is the common reconstruction which corresponds to Point 1 in Fig. 6.2, (c) is the unique reconstruction, which corresponds to Point 2 in Fig. 6.2, and (d) is the final reconstruction, which is the sum of (b) and (c), corresponding to the Point 4 in Fig. 6.2. Better view in electronic version. 110

6.11 Visualization of the reconstructed image of different parts in the CU-Net for the multi-exposure image fusion task. (c) is the common reconstruction which corresponds to Point 1 in Fig. 6.2, (d) and (e) are the unique reconstructions, which correspond to Point 2 and 3 in Fig. 6.2, respectively, and (f) is the final reconstruction, which is the sum of (c), (d) and (e). Better view in electronic version. 111

Chapter 1

Introduction

1.1 Motivation and Objectives

Recent years have seen many learning based methods for image restoration problems, such as image denoising, image super-resolution and image enhancement, which often achieve better results than traditional model-based or feature-based methods. Despite the good performance, there still exist some challenging problems. Firstly, most of the deep learning based approaches are purely data-driven, and they often rely on a huge amount of training data to improve the performance. In the cases when the data available is limited or the data quality is not good enough, the performance can be significantly affected. Secondly, deep neural networks are often not easy to interpret, and are sometimes treated as a "black box". In contrast, model-based methods are more stable, with good interpretability and do not require large training data. Thus, many research works now focus on combining prior knowledge in the model based methods with deep learning approaches.

In this thesis, we solve a variety of image restoration problems by incorporating domain knowledge into the deep neural networks, in order to leverage both the advantages of model-based and deep learning based approaches. The image restoration problems solved in this thesis range from the single image super-resolution to multi-modal image super-resolution as well as the general case of multi-modal image restoration and fusion problems. The challenge here is how to successfully enrich the learning framework with domain

knowledge. In this thesis, we explore several ways to achieve this for different image restoration tasks.

Single image super-resolution (SISR) is a typical problem in image processing and computer vision, which aims to infer a high-resolution (HR) image from a single low-resolution (LR) image. Many methods use deep neural networks to learn the effective mappings from the LR to HR images. However, most of these methods either focus on improving the objective image quality, by minimizing the mean square error (MSE) between the LR and HR images, or focus on enhancing the perceptual image quality, by minimizing the perceptual loss using adversarial training. The former methods can achieve low MSE distortion but with poor perceptual quality, while the latter methods can generate photo-realistic images but with large MSE distortion. In this thesis, we will present a method that uses style transfer in the wavelet domain to combine the advantages of these two kinds of methods, in order to generate both accurate and photo-realistic super-resolved images. We first use the wavelet transform to split an image into two parts: the low-frequency sub-band affecting the perceptual quality and the high-frequency sub-band affecting the objective quality. Then, for the low-frequency sub-band, we improve its objective quality and for high-frequency sub-band, we use wavelet domain style transfer to improve its perceptual quality. By recombining these sub-bands, we can obtain an image with good perception-distortion tradeoff.

Different from SISR, in multi-modal image super-resolution (MISR), one aims to infer a HR image, not only from the LR image but also with the guidance of an image from a different modality. In real-life, one scene is often captured using different imaging sensors and this allows the scene to be represented more accurately. For example, in 3D model representation, both RGB and depth images are captured. However, due to the limitations of the sensing mechanism, the depth images are often captured with very low resolution. We find that the columns and rows in depth images can be well modelled as piece-wise polynomial signals, and thus they can be restored using the finite rate of innovation (FRI) sampling and reconstruction theory [28, 29]. In this thesis, we will introduce a method which combines the FRI theory with a multi-modal dictionary learning algorithm, to upscale the depth image with the guidance of its corresponding RGB image. The combined

use of a model-based algorithm followed by a data-driven algorithm has several advantages. We only need a fraction of training data to achieve very good performance, and we are resilient to mismatches between training and testing datasets.

The aforementioned RGB guided depth image super-resolution is only a specific case of the MISR problem. For solving the general MISR problem using deep learning, two important elements need to be considered: how to properly model the dependencies among different modalities and how to design a neural network with good interpretability. In this thesis, we will propose a joint multi-modal dictionary learning model to model the dependencies across different modalities using sparse coding, and then we design an interpretable deep network architecture by unrolling the iterative shrinkage and thresholding algorithm (ISTA). We will also introduce a layer-wise optimization algorithm to initialize the network parameters before back-propagation.

Beyond the MISR problem, multi-modal image restoration and image fusion are more general tasks. Specifically, the MISR problem belongs to the multi-modal image restoration task, which aims to restore a better version of a distorted image with the guidance of another image from a different modality. In contrast, the multi-modal image fusion task aims to fuse two images from different modalities, in order to obtain a more comprehensive image which contains the useful information of both source images. The multi-focus and multi-exposure image fusion are examples of this task. In this thesis, we will present a deep Common and Unique information splitting network (CU-Net), to provide an effective and flexible network architecture to solve both the multi-modal image restoration and fusion problems. The CU-Net is designed by drawing inspirations from a multi-modal convolutional sparse coding model using again the unfolding approach, in order to achieve good performance and better interpretablity.

1.2 Original contribution and outline of thesis

The remainder of this thesis is organised as follows

In Chapter 2, we review state-of-the-art works on single image super-solution (SISR),

multi-modal image restoration and image fusion, and the iterative unfolding strategy.

In Chapter 3, we present our SISR method which aims to achieve a good trade-off between the objective and perceptual image quality in super-resolved images. We propose to use 2D stationary wavelet transform (SWT) to decompose one image into low-frequency and high-frequency sub-bands. For the low-frequency sub-band, we improve its objective quality through an enhancement network. For the high-frequency sub-band, we propose to use wavelet domain style transfer (WDST) to effectively improve perceptual quality. By the perfect reconstruction property of wavelets, these sub-bands can be re-combined to obtain an image which has simultaneously high objective and perceptual quality.

In Chapter 4, we introduce our RGB guided depth image super-resolution method which combines the power of finite rate of innovation (FRI) theory with multi-modal dictionary learning. Given a LR depth image, we first model its rows and columns as piecewise polynomial signals and propose a FRI-based depth upscaling algorithm to super-resolve it. Then, the upscaled moderate quality (MQ) depth image is further enhanced with the guidance of a HR intensity image. This is achieved by learning multi-modal mappings from the joint MQ depth and HR intensity pairs to the HR depth, through a recently proposed triple dictionary learning algorithm [30]. Moreover, to speed up the super-resolution process, we introduce a projection-based rapid upscaling technique that pre-calculates the projections from the joint MQ depth and HR intensity pairs to the HR depth. We demonstrate the effectiveness of this method on various RGB/depth datasets and the robutness of this method on noisy data and unknown blurring kernels.

In Chapter 5, we consider how to solve the general MISR task. We first introduce a novel joint multi-modal dictionary learning (JMDL) algorithm to model cross-modality dependency. In JMDL, we simultaneously learn three dictionaries and two transform matrices to correlate the modalities. Then, by unfolding the iterative shrinkage and thresholding algorithm (ISTA), we turn the JMDL model into a deep neural network, called deep coupled ISTA network. Since the network initialization plays an important role in deep network training, we further propose a layer-wise optimization algorithm (LOA) to initialize the parameters of the network before running back-propagation strategy. Specifically,

we model the network initialization as a multi-layer dictionary learning problem, and solve it through convex optimization. The proposed LOA is demonstrated to effectively decrease the training loss and increase the reconstruction accuracy. The experimental results validate the superiority of our method over other state-of-the-art MISR approaches in both reconstruction accuracy and running time, for various multi-modal cases.

In Chapter 6, we move from the MISR task to the more general multi-modal image restoration and image fusion tasks. We propose a novel deep Common and Unique information splitting network (CU-Net), whose architecture is designed by drawing inspirations from a multi-modal convolutional sparse coding (MCSC) model. We demonstrate the effectiveness of the CU-Net on a variety of multi-modal image restoration and image fusion tasks, including RGB guided depth image super-resolution, flash guided non-flash image denoising, multi-focus and multi-exposure image fusion.

Finally, in Chapter 7, we conclude the thesis by summarising the contributions and discuss possible directions for future study.

1.3 Publications

The material presented in this thesis has led to the following publications:

Submitted

- X. Deng and P. L. Dragotti, "Deep Convolutional Neural Network for Multi-modal Image Restoration and Fusion", submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Peer-reviewed Journals

- X. Deng and P. L. Dragotti, "Deep Coupled ISTA Network for Multi-modal Image Super-Resolution", *IEEE Transactions on Image Processing*, to appear, 2019.

- X. Deng, P. Song, Miguel R. D. Rodrigues and P. L. Dragotti, "RADAR: Robust Algorithm for Depth Image Super-Resolution Based on FRI Theory and Multi-modal Dictionary Learning", *IEEE Transactions on Circuits and Systems for Video Technology*, to appear, 2019.

Peer-reviewed Conferences

- X. Deng, R. Yang, M. Xu and P. L. Dragotti, "Wavelet Domain Style Transfer for an Effective Perception-distortion Tradeoff in Single Image Super-Resolution", *IEEE International Conference on Computer Vision (ICCV)*, oral presentation, 2019.
- X. Deng and P. L. Dragotti, "Coupled ISTA network for Multi-modal Image Super-Resolution", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.
- X. Deng, J. Huang, M. Liu and P. L. Dragotti, "U-FRESH: An FRI-Based Single Image Super Resolution Algorithm and An Application in Image Compression", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

Chapter 2

Background

2.1 Single image super-resolution

Single image super-resolution is a typical problem in computer vision and image processing, which aims to infer a high-resolution (HR) image from a single low-resolution (LR) one. It has attracted a lot of interest from both academic and industrial communities and has wide applications in fields like medical imaging, infrared imaging, and face recognition. Many methods have been proposed to tackle the SISR problem, which can be broadly classified into three categories based on the optimization target: methods for improving the objective quality, methods for improving the perceptual quality, and methods for a good trade-off between objective and perceptual quality. Next, we will review the recent development on these three research directions.

2.1.1 Objective quality driven SISR

To improve the objective quality, the most common strategy is to minimize the mean squared error (MSE) between the restored image and the ground-truth. We will review both more traditional methods which are normally based on dictionary learning and the state-of-the-art methods which use deep learning approaches.

Dictionary learning for SISR. Many of the traditional SISR methods rely on dictionary learning and sparse representations to restore the HR image. Representative algorithms include ScSR [31], Zeyde *et al.* [32], CDL [33], A+ [34] and CSC [35]. Specifically, Zeyde *et al.* [32] used K-SVD algorithm [36] to learn over-completed dictionaries for both LR and HR images, with the assumption that the LR and HR counterpart share the same sparse coefficients. Given the LR training patches $\{\boldsymbol{x}_i\}_{i=1}^{P}$, they solve the following optimization to obtain both the dictionary and sparse representations for the LR patches:

$$\min_{\boldsymbol{D}_l,\{\boldsymbol{\alpha}_i\}_{i=1}^{P}} \sum_{i=1}^{P} \|\boldsymbol{x}_i - \boldsymbol{D}_l\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_0 \quad s.t. \quad \|\boldsymbol{\alpha}_i\|_0 \leq L, \quad \forall i, \tag{2.1}$$

where $\boldsymbol{\alpha}_i$ is the sparse representation of $\boldsymbol{x}_i$, and $\boldsymbol{D}_l \in \mathbb{R}^{M\times Q}$ $(M < Q)$ is the over-complete dictionary for the LR image. The ℓ_0 norm here indicates the number of non-zero elements in the vector, and L is the constrained sparsity level. The K-SVD dictionary learning algorithm [36] can be used to solve this problem easily. After that, the HR dictionary is constructed by assuming that the LR and HR patches share the same sparse representations, as follows:

$$\boldsymbol{D}_h = \min_{\boldsymbol{D}_h} \sum_{i=1}^{P} \|\boldsymbol{y}_i - \boldsymbol{D}_h\boldsymbol{\alpha}_i\|_2^2 . \tag{2.2}$$

However, the calculation of sparse coefficients in (2.1) is very time consuming. To handle this problem, A+ [34] proposed to classify the training patches into different clusters and pre-calculate the projections from LR to HR patches for each cluster. Then, the sparse coefficients calculation can be avoided, leading to a significant computational speed-up. Since the traditional sparse coding is patch based, the restored image is obtained by assembling all the patches using the over-lapping strategy. However, in this case, the neighbourhood information among adjacent patches is not exploited and this may decrease the reconstruction accuracy. To solve this problem, CSC [35] proposed to use convolutional dictionary learning to solve the SISR problem, in which the sparse representations are calculated in the

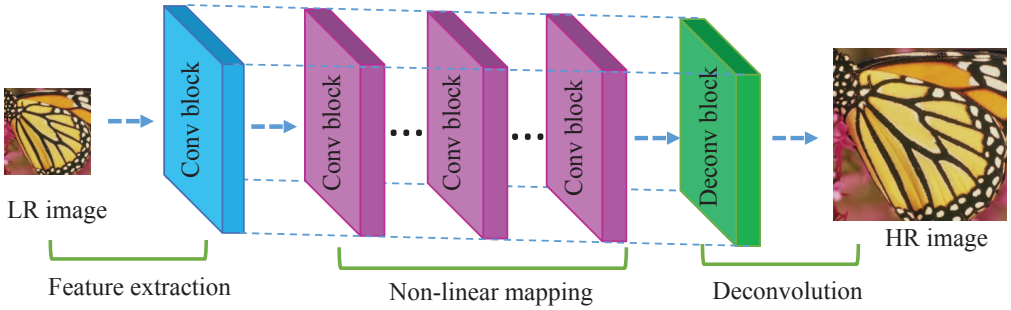


Figure 2.1: The network architecture of FSRCNN [2].

image level.

Deep learning for SISR. Recently, many SISR methods based on deep neural networks have appeared and show state-of-the-art performance in SISR. The representative networks include SRCNN [37], SCN [11], VDSR [16], DRCN [38], SRResNet [39], MemNet [40], EDSR [41], D-DPBN [42], RCAN [43], to name a few. The SRCNN [11] is one of the first convolutional neural network (CNN) based methods for SISR, which has shown the effectiveness of CNN in solving the SISR problem and inspired many follow-up works. In SRCNN, the LR image is firstly upscaled to the same size as the HR image using bicubic interpolation, and then the network is trained to minimize the mean squared error (MSE) between the ground truth and the bicubic interpolated image. The architecture of the SRCNN is very shallow with only three layers. In VDSR [16], the depth of network is increased, i.e., with 20 layers, by learning the residual maps between the ground-truth and the input. Like the SRCNN, the VDSR also increased the LR image size to the same of the HR image before the network, which may restrict the receptive field, increase the training complexity, and decrease the super-resolution performance. To overcome these drawbacks, FSRCNN [2] was proposed to directly learn from the original LR image to the HR image without interpolation. Fig. 2.1 shows the network architecture of the FSRCNN, which is composed of three parts: feature extraction module to extract useful features from the LR image, non-linear mapping module to map the LR features to the HR features, and the deconvolutional module to transform the HR features to the HR image. Most of the follow-up works, e.g., SRResNet [39],

MemNet [40], EDSR [41], RCAN [43], adopted this framework to achieve SISR, but with different design of the inner convolutional block structures. For example, SRResNet [39] used a residual block, MemNet [40] used a memory block, and RCAN [43] used a residual in residual (RIR) block.

2.1.2 Perceptual quality driven SISR

As pointed out by many works [3,44], the MSE loss cannot measure well the perceptual similarity between two images, and can also lead to blurred and over-smooth images. To overcome this drawback, Johnson *et.al* [45] proposed to use the perceptual loss to replace the traditional MSE loss. As we know, the MSE loss is defined by the Euclidean distance between the estimated image $\boldsymbol{I}_{est}$ and the ground truth image $\boldsymbol{I}_{gt}$ in the pixel space. Different from MSE loss, the perceptual loss is calculated in the feature space, which is defined as follows [45]:

$$\mathcal{L}_P = \|\boldsymbol{\phi}(\boldsymbol{I}_{est}) - \boldsymbol{\phi}(\boldsymbol{I}_{gt})\|_F^2 , \tag{2.3}$$

where $\boldsymbol{\phi}(\cdot)$ indicates the feature map of $\boldsymbol{I}_{est}$ and $\boldsymbol{I}_{gt}$ through a pre-trained network. Usually, we use the VGG network [46], which is pre-trained on ImageNet [47] for image recognition, to extract the feature maps. In [45], the feature maps are extracted using the $relu2_2$ layer in VGG network. Through training the network using the perceptual loss defined above, the network is encouraged to produce I_{est} to have similar feature representations as the ground truth I_{gt}, and thus the estimated image can look photo-realistic.

Later, Ledig *et al.* [3] proposed a generative adversarial network (GAN) for single image super-resolution, named SRGAN. Instead of only using the perceptual loss in (2.3), the SRGAN introduces an adversarial loss. Thus, the total loss is defined as a weighted sum of the VGG loss in (2.3) and adversarial loss. The VGG loss is good at representing the perceptual similarity between two images, and the adversarial loss can make the restored image look realistic. Fig. 2.2 shows the framework of

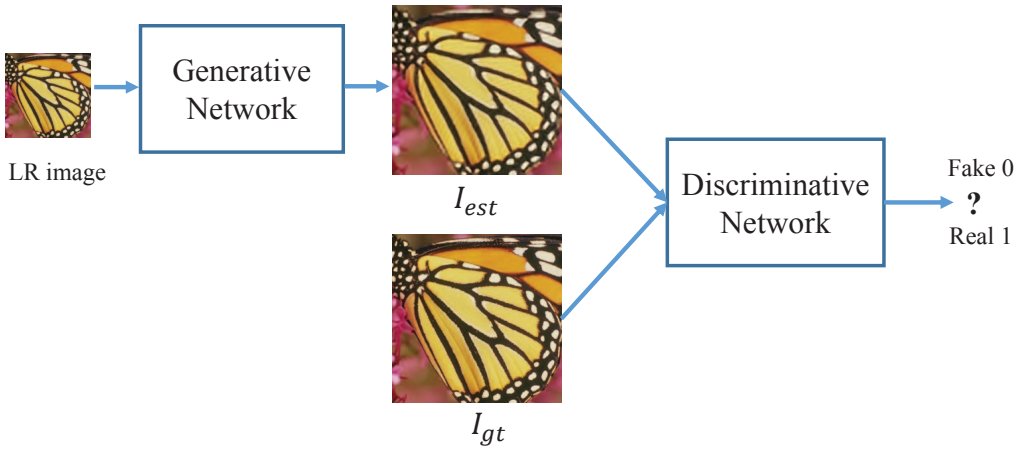


Figure 2.2: The network architecture of SRGAN [3].

SRGAN training network. Given a LR image, the generative network is trained to minimize the perceptual loss between the estimated and ground-truth images. Then, the discriminative network is trained to distinguish the estimated image from the ground-truth image, i.e., the discriminative network is trying to output real (= 1) when the input is the ground-truth image $\boldsymbol{I}_{gt}$, and output fake (= 0) when the input is the estimated image $\boldsymbol{I}_{est}$. The final aim of SRGAN is to train a generative network to produce images which can fool the discriminative network. The generated image should be extremely similar to the real image, so that the discriminative network cannot distinguish these two images. On the basis of SRGAN, Saggadi *et.al* [48] proposed an ENet which added another loss called texture matching loss to the VGG loss and adversarial loss. The texture matching loss is defined as follows:

$$\mathcal{L}_T = \|\boldsymbol{G}(\boldsymbol{\phi}(\boldsymbol{I}_{est})) - \boldsymbol{G}(\boldsymbol{\phi}(\boldsymbol{I}_{gt}))\|_F^2 . \tag{2.4}$$

Here, the $\boldsymbol{G}(\cdot)$ is the Gram matrix, $\boldsymbol{G}(\boldsymbol{F}) = \boldsymbol{F}\boldsymbol{F}^T$. In [48], the texture matching loss is calculated patch by patch. This is done to prioritize features between $\boldsymbol{I}_{est}$ and $\boldsymbol{I}_{gt}$ which are locally similar. Later, Wang *et al.* proposed ESRGAN [49], which is also developed based on the SRGAN. It did not add any new loss, but changed the network structure to make it work better. Specifically, it removed the batch normalization function in SRGAN, calculated the perceptual loss before activation,

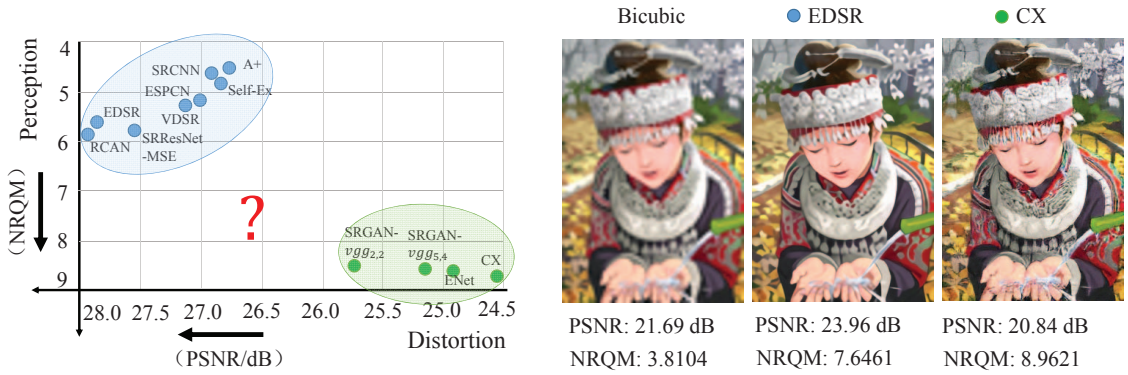


Figure 2.3: Perception-distortion performance of different SISR methods.

used a relativistic discriminator and a residual in residual dense block (RRDB). As a consequence, the ESRGAN overcomes many drawbacks in SRGAN, and can generate very clean images with sufficient details and rich textures. Recently, instead of the perceptual loss, Mechrez *et.al* [50] proposed a method called CX which uses the contextual loss to make the internal statistics of the restored image similar to the ground-truth, leading to more realistic images. Zhang *et.al* [51] proposed a RankSRGAN to optimize the perceptual quality through training a ranker to learn the behavior of the perceptual metrics.

2.1.3 Perception-distortion trade-off in SISR

Given a LR image, the final aim of SISR is to find a HR image which is both accurate and photo-realistic. However, as pointed in [52], the objective and perceptual quality are not compatible, which means there is a fundamental trade-off between them. In Fig. 2.3, we show the performance of different SISR methods in terms of objective and perceptual quality. Here, the objective quality is measured by PSNR, and the perceptual quality is measured by NRQM [53]. The larger value of PSNR indicates better objective quality and the larger value of NRQM indicates better perceptual quality. In this figure, the blue points represent the SISR methods which aim for high objective quality, the green points represent the methods aiming for high perceptual quality. We can see that the SISR methods that optimize the objective quality achieve high PSNR but with low perceptual quality, while the SISR methods that

optimize the perceptual quality achieve high perceptual quality but with low PSNR values. To better demonstrate this observation, we show in Fig. 2.3 the images super-resolved by methods EDSR [41] and CX [50], which are two state-of-the-art SISR methods aiming for objective and perceptual quality, respectively. We can see that the image restored by EDSR has high PSNR value, but it looks excessively smooth, while the image generated by CX looks visually pleasing with sufficient details, but it has low objective quality.

Recently, some methods are proposed in order to achieve a good trade-off between the perceptual and objective quality in SISR. Specifically, in [3], the SRGAN-MSE method is proposed to minimize the combined loss of MSE and adversarial loss. However, the training process is extremely unstable for the following reasons. On the one hand, the adversarial loss encourages the synthesis of high-frequency details in the results. On the other hand, since these high-frequency details are not in the right places, the MSE distortion increases. In [48], the authors first propose to add a texture matching loss to the MSE and adversarial loss to make the training process more stable. However, their results still suffer from blocking effects and noisy artifacts. Choi *et.al* [54] train a multi-scale super-resolution model with a discriminator network and two qualitative score predictors, which achieves high perceptual quality while preserving the objective quality. Most recently, the ESRGAN [49] approach was proposed to train two networks which aim to enhance the objective and perceptual quality, respectively, and then these two networks are interpolated to achieve a trade-off between the objective and perceptual quality. However, the network interpolation method requires that the two networks should have exactly the same architectures, which might limits performance.

2.2 Multi-modal image restoration and image fusion

2.2.1 Multi-modal image restoration

The multi-modal image restoration task aims to restore a high-quality version of a distorted image with the assistance of another image from a different modality, e.g., the RGB guided depth image super-resolution and flash guided non-flash image denoising. The related approaches can be broadly classified into the two categories: the traditional methods based on joint image filtering and recent methods based on deep learning.

The basic idea of joint image filtering is to transfer the salient structures in the guidance image, e.g., edges and textures, to the target image through constructing some joint filters. According to the filtering type, the joint image filtering methods can be further classified into two categories: the statically guided [55–59] and dynamically guided methods [20, 60–64]. The statically guided methods assume that the guided image is fixed, and thus they are only effective for the scenarios where the guided image provides sufficiently reliable details. Representative algorithms include the joint bilateral filter (JBF) [55], guided filter (GF) [56], and weighted median filter (WMF) [57]. They may struggle when there are structural inconsistencies between the two images, e.g., RGB and depth images. For the dynamically guided methods, they consider both the original guided image and the filtered target image as guidance, which makes them more resilient to inconsistency between the two images. Specifically, Zhang *et al.* [61] proposed a rolling guided filter (RGF), which iteratively used the filtered image as guidance for small structure and edge recovery. Ham *et al.* [64] proposed a static/dynamic (SD) filter in which the static filter convolves the input image with a weight function calculated from only the static guidance image, and the dynamic filter uses the weight function repeatedly calculated from the filtered output image. To avoid the inconsistent structure transfer between guidance and target images, Shen *et al.* [62] defined three different types of structures: mutual structure, inconsistent structure and smooth regions. They further defined

a normalized cross-correlation (NCC) metric to find the mutual structures between two images and used them to guide the filtering. However, the NCC metric is defined at patch level, and thus it can only measure the patch similarity and this fact sometimes leads to halo artefacts. To solve this problem, Guo *et al.* [20] recently proposed a mutually guided image filter (muGIF) which defines a relative structure to measure the similarity between two images at image level.

Recently, some authors have proposed using deep neural networks to solve the MIR problem. The related methods can be classified into two categories: methods aiming to solve the general MIR problem and methods that are tailored to solve the MIR problem for specific pairs of modalities (e.g., RGB and depth). For the general MIR task, Li *et. al* [65] proposed a deep joint filtering (DJF) network, which used two independent convolutional neural networks (CNN) to extract the structural information from the guidance and input images, respectively, and then used a third CNN to predict the target image. Later, this work is further improved by adding a skip connection between the input and target images, which leads to DJFR [18]. Kim *et. al* [66] proposed a deformable kernel network (DKN) for joint image filtering, which employs a similar network architecture as DJF [18], but with a new added weight and offset learning module to learn the neighbourhood system for each pixel. Wu *et. al* [17] turned the traditional guided filtering (GF) algorithm into a differentiable block, namely guided filtering layer, and then plug it into a convolutional neural network, so that it can be trained end-to-end. Recently, Pan *et. al* [67] proposed a spatially variant linear representation model, in which the target image is linearly represented by the guidance and input images. They then use a deep CNN to estimate the representation coefficients to restore the target image. Very recently, the authors in [68] extent the traditional patch-based sparse coding to the convolutional sparse coding, and then unfold it into a deep convolutional neural network. However, it only targets the multi-modal image super-resolution problem. For the specific MIR task, methods [9, 69–71] aim to upscale the depth image with guidance from the RGB image using deep neural networks. Methods [72–74] aim

to improve the resolution of the multi-spectral image with the assistance of either RGB or panchromatic images. Note that the methods aiming for a specific MIR task may not perform well on other tasks, because either the filters or the network architectures have been tailored to the specific characteristics of image modalities under consideration.

2.2.2 Multi-modal image fusion

Multi-modal image fusion aims to integrate the complementary information contained in different source images to generate a fused image. Since the fused image can provide more comprehensive information about the scene, it is more helpful for human or machine perception in different applications. The source images are usually captured by different sensors, e.g., Magnetic resonance (MR) and computed tomography (CT) medical images, but sometimes they are obtained using the same sensor with different imaging parameters, e.g., multi-focus and multi-exposure images.

Most traditional image fusion methods follow a three-step fusion procedure [75]. Firstly, the source images are mapped into a specific transform domain, e.g., wavelet transform. Then, the transform coefficients are fused based on a certain fusion rule, and finally the fused coefficients are transformed back into the image domain to obtain the final fused image. There are two elements here which play a critical role in the fusion performance: the selection of transform domain and the fusion rule. Many works have studied the fusion performance in different transform domains, including discrete wavelet transform (DWT) [76], discrete cosine transform (DCT) [77], non-subsampled contourlet transform [78], to name a few. There are also some works based on sparse coding [25, 79, 80], which fuse the sparse representations in the sparse domain. The most widely used fusion rules are choose-max [81] and weighted average [82]. Usually, in order to avoid coefficient inconsistency, a neighbourhood morphological processing step is used after the choose-max strategy [75]. However, the transform domain is manually selected and the fusion rule is hand-crafted, which

may influence the fusion results.

To overcome the aforementioned drawbacks, some authors have recently proposed to use deep neural networks to solve the MIF problems. Specifically, Liu *et al.* [83] proposed a simple CNN to predict the decision map for multi-focus image fusion. Prabhakar *et al.* [26] proposed a CNN based unsupervised image fusion method to fuse one under-exposed image with an over-exposed one. Li *et al.* [27] proposed a CNN network with the dense block structure to solve the infrared and visible image fusion problem. To improve the perceptual quality of the fused image, Ma [84] proposed a generative adversarial network (GAN), called FusionGAN, for infrared and visual image fusion.

2.3 Iterative unfolding strategy

The iterative unfolding strategy has been widely used to design deep neural networks which can incorporate domain knowledge. As we know, many model-based parameter estimation algorithms [85–87] update the parameters through iterations. Usually, each iteration consists of a linear operation followed by a non-linear thresholding operation, which is similar to a single layer in a deep neural network. These iterative algorithms include, for example, iterative shrinkage and thresholding algorithm (ISTA) [85] for sparse estimation, the approximate message passing (AMP) [86] for compressive sensing, the alternating direction method of multipliers (ADMM) [87] for generic inverse problem. Intuitively, we can turn the traditional iterative algorithms into deep networks by unfolding each iteration. Some papers [88–98] that use this unfolding perspective have appeared recently to solve problems in different fields. For example, for the task of sparse code estimation, the papers [88–91] unroll the ISTA algorithm to be a deep network and [93] turns the AMP algorithm into a deep network to predict the sparse codes. Sun *et.al* [94] proposed to unfold the ADMM algorithm for compressive sensing magnetic resonance imaging (MRI), and Metzler *et.al* [95] proposed a learned denoising-based AMP network for image de-

noising. Recently, Bertocchi *et.al* [96] proposed to unfold a proximal interior point method to a deep network to solve image deblurring problem, and Lohit [74] *et al.* proposed to unroll the iterations of the projected gradient descent (PGD) algorithm to obtain a deep network to solve the panchromatic guided multi-spectral image super-resolution problem.

The recipe for designing the network using an unfolding strategy is as follows: firstly, we model the given task as a traditional optimization problem; secondly, we try to solve this problem using an iterative algorithm; finally, we unfold each iteration in the iterative algorithms into a deep network. In this thesis, we employ the unfolding strategy in designing deep neural networks for multi-modal image restoration and image fusion tasks, and this strategy leads to neural network architectures that achieve state-of-the-art results.

Chapter 3

Perception-distortion Tradeoff in Single Image Super-Resolution

3.1 Introduction

In single image super-resolution, the methods driven by objective quality can achieve low mean squared error (MSE) but with poor perceptual quality, while the other approaches can generate photo-realistic images but with large MSE distortion. Ideally, we wish to obtain a super-resolved image which looks photo-realistic and at the same time has high objective quality. However, as pointed out in [52], there exists a fundamental tradeoff between the ability to achieve low MSE and high perceptual quality, i.e., the increase of perceptual image quality is at the expense of its objective quality, and vice versa. We find that the objective and perceptual quality are affected by different elements in an image. Thus, in order to achieve better trade-off between objective and perceptual quality, we need to seperate the elements affecting the objective quality from those affecting the perceptual qualtiy, and then optimize each of them separately.

In this chapter, we propose to use wavelet transforms to achieve this separation. As we know, wavelet transforms can split an image into one low-frequency and

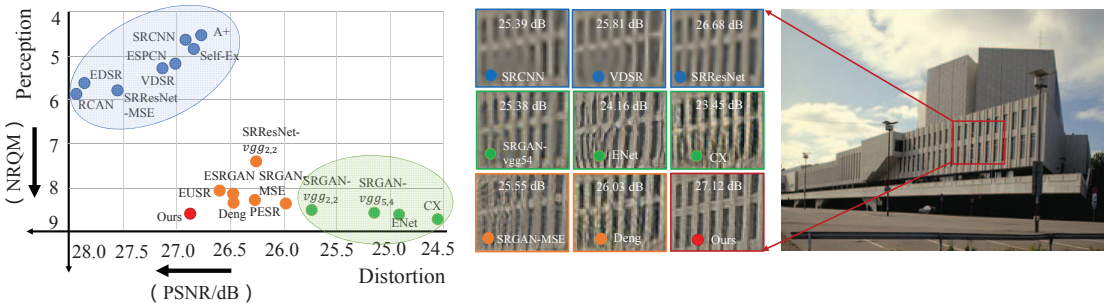


Figure 3.1: Perception-distortion performance of different SISR methods. The blue points represent methods aiming for high objective quality, the green points represent methods aiming for good perceptual quality, and the orange points represent methods aiming for a trade-off between perception and distortion. The higher PSNR value indicates better objective quality and the higher perceptual score indicates better perceptual quality. The bottom left corner is the best. Our method achieves the best trade-off among all the "orang" methods.

several high-frequency sub-bands. As we will introduce in Section 3.2.2, we find that the low-frequency sub-band plays an important role in the objective quality, while the high-frequency sub-bands can affect the perceptual quality significantly. After separation, to obtain the best tradeoff, we use an enhancement network to improve the objective quality of the low-frequency sub-band, and wavelet domain style transfer to improve the perceptual quality of the high-frequency sub-bands.

Note that in this chapter, we are not aiming for a new SISR method towards high perceptual or objective image quality, which has been extensively explored recently. Instead, we propose a novel image fusion method which combines two images to achieve the best tradeoff between the perception and distortion, as shown in Fig. 3.1. Our method overcomes many drawbacks of the existing methods. For example, compared with SRGAN-MSE [3], we do not need to train a deep network, and thus we have no concerns on the stability of training. Compared with ESRGAN [49], we are more flexible with the choice of the network architecture, which gives us more freedom to achieve the best perception-distortion (PD) trade-off. All these contribute to higher reconstruction performance and a better PD tradeoff.

3.2 Proposed Method

3.2.1 Stationary wavelet transform

The wavelet transform allows multi-resolution analysis of images [99]. The classical discrete wavelet transform (DWT) has a drawback, i.e., it is not shift-invariant. The stationary wavelet transform (SWT), also known as undecimated wavelet transform, overcomes this drawback by removing the downsampling operation in DWT [100]. Fig. 3.2 illustrates the 2D SWT process for 2 level decomposition. Suppose that H_0 and G_0 are the low-pass and high-pass filters of a standard 1D wavelet decomposition, we can obtain the z transform of LL, LH, HL, and HH sub-bands at the i-th level ($i = 1, \cdots, N$) through the following formulations:

$$LL_i(z_x, z_y) = H_0(z_y^{2^{i-1}})H_0(z_x^{2^{i-1}})LL_{i-1}(z_x, z_y), \tag{3.1}$$

$$LH_i(z_x, z_y) = G_0(z_y^{2^{i-1}})H_0(z_x^{2^{i-1}})LL_{i-1}(z_x, z_y), \tag{3.2}$$

$$HL_i(z_x, z_y) = H_0(z_y^{2^{i-1}})G_0(z_x^{2^{i-1}})LL_{i-1}(z_x, z_y), \tag{3.3}$$

$$HH_i(z_x, z_y) = G_0(z_y^{2^{i-1}})G_0(z_x^{2^{i-1}})LL_{i-1}(z_x, z_y), \tag{3.4}$$

where the LL_{i-1} is the LL sub-band at the $(i-1)$-th level, with LL_0 as the input image X. After the N-th level decomposition, we obtain $(3N+1)$ wavelet sub-bands with the same size as the input image, i.e., $LL_N, \{LH_i\}_{i=1}^N, \{HL_i\}_{i=1}^N, \{HH_i\}_{i=1}^N$, where LL_N contains the low-frequency information at the N-th level, LH_i, HL_i and HH_i contain the horizontal, vertical and diagonal details at the i-th level, respectively.

3.2.2 Motivation

The 2D SWT can decompose an image into multiple sub-bands, including one low-frequency and several high-frequency sub-bands. Our key insight here is that the low-frequency sub-band has a significant effect on the objective quality of an image,

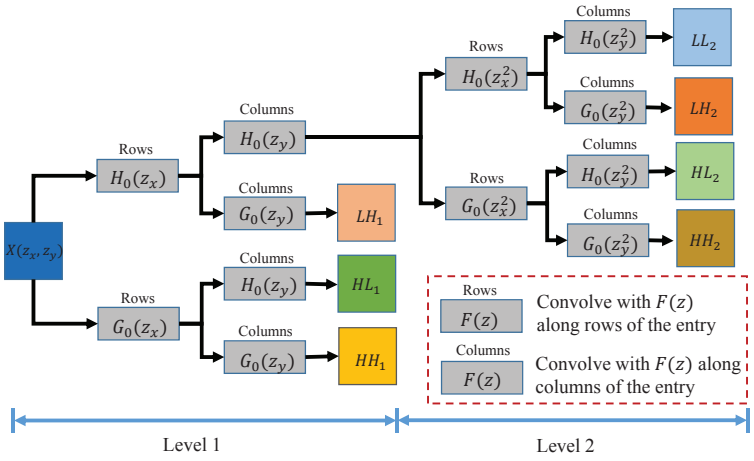


Figure 3.2: Illustration of two level 2D stationary wavelet transform (SWT) of image X, with H_0 and G_0 as the low-pass and high-pass filters, respectively.

while the high-frequency sub-bands affect the perceptual quality significantly. To verify that, we consider two super-resolved images: A_p with high perceptual quality but low objective quality, and A_o with high objective quality but low perceptual quality. Fig. 3.3 shows these two images, together with the histograms of their sub-bands after one level 2D SWT. Here, A_p and A_o are obtained using the existing SISR methods CX [50] and EDSR [41], respectively. We use peak signal-to-noise ratio (PSNR) to measure the objective quality, and NRQM [53] to measure the perceptual quality following [50]. Note that larger PSNR and NRQM values indicate better objective and perceptual quality, respectively. As shown in Fig. 3.3, the high-frequency sub-bands (i.e., LH, HL, HH) of A_p have quite similar histogram distributions as the ground-truth, but this is not the case for A_o. Since the high-frequency sub-bands contain the detail information, this can explain why A_p has high perceptual quality. For the LL sub-band, A_o has a more similar histogram as the ground-truth than A_p, which is one of the reasons why A_o has high objective quality.

In order to further verify our observation, a simple experiment is performed as follows. We replace the low-frequency sub-band of A_p with that of A_o, and keep all its high-frequency sub-bands. These sub-bands are then combined via 2D inverse SWT (ISWT) to obtain a reconstructed image $\tilde{A}_p$. Likewise, we replace the low-

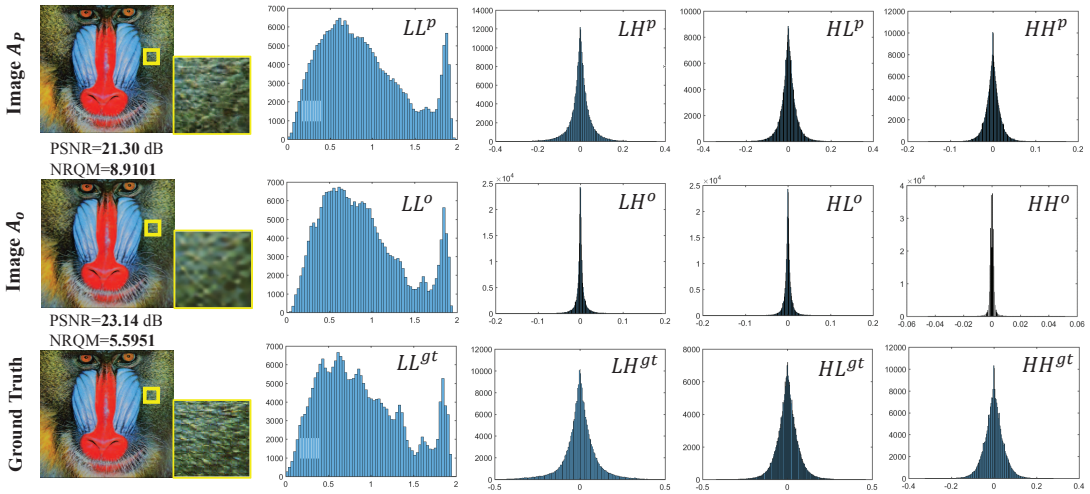


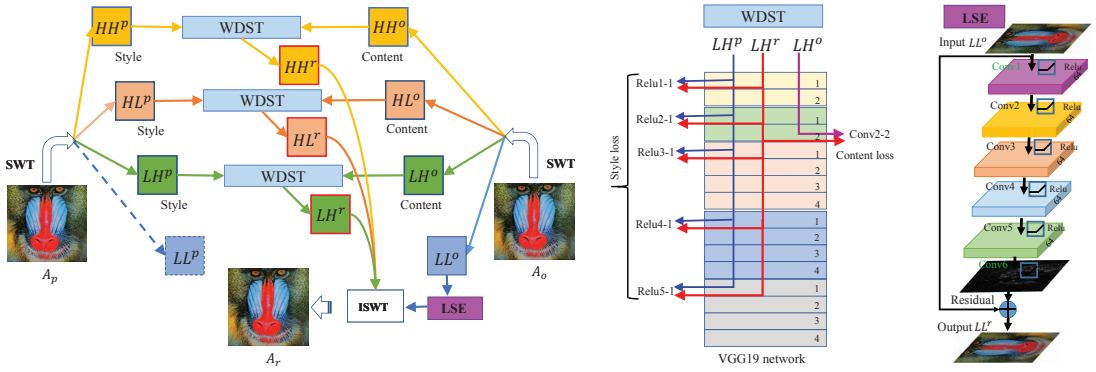
Figure 3.3: The first row shows the histograms of different sub-bands of A_p which has high perceptual quality but low objective quality. The second row shows the histograms of different sub-bands of A_o which has high objective quality but low perceptual quality. The third row shows the ground-truth histograms. All the histograms are drawn in RGB colour space.

Methods	A_p	$\tilde{A}_p$	A_o	$\tilde{A}_o$
PSNR	24.58	25.68	27.80	26.57
NRQM	8.8007	8.7775	5.7159	5.8864

Table 3.1: PSNR and NRQM scores on the BSD100 dataset.

frequency sub-band of A_o with that of A_p to obtain a reconstructed image $\tilde{A}_o$. Table 3.1 shows the PSNR and NRQM results on the BSD100 dataset. As can be seen, the PSNR of $\tilde{A}_p$ improves more than 1dB over A_p while the NRQM score does not change too much. Similar phenomenon can be observed between $\tilde{A}_o$ and A_o. The reason why the objective quality is significantly affected is that the low-frequency sub-band is changed. In contrast, the perceptual quality is not particularly influenced because we preserve the high-frequency sub-bands. Thus, in order to obtain an image with a good PD tradeoff, one possible solution is to pursue high objective quality of its low-frequency sub-band and high perceptual quality of its high-frequency sub-bands.

Fig. 3.4 (a) shows the framework of our method. Given one image A_p with high perceptual quality and another image A_o with high objective quality, we first perform 2D SWT on these two images, so that each image is decomposed into one



(a) (b) (c)

Figure 3.4: (a) shows the framework of our method, (b) illustrates the wavelet domain style transfer (WDST) algorithm, and (c) shows the low-frequency sub-band enhancement (LSE) network.

low-frequency and several high-frequency sub-bands. Take the decomposition with one level for example, A_p is decomposed into $\{LL^p, LH^p, HL^p, HH^p\}$, and A_o is decomposed into $\{LL^o, LH^o, HL^o, HH^o\}$. For LL_o, we use the low-frequency sub-band enhancement network to enhance its objective quality. For high-frequency sub-bands pairs, e.g., LH_p and LH_o, we use wavelet domain style transfer (WDST) to fuse them to a new sub-band. Finally, all fused sub-bands and enhanced LL_o are synthesised by ISWT to obtain image A_r.

3.2.3 Low-frequency sub-band enhancement

For the low-frequency sub-band LL^o, we aim to further improve its objective quality. Here, we employ the basic network structure of VDSR [16] to achieve this goal, as shown in Fig. 3.4 (c). The network is composed of 6 convolutional layers with a rectified linear unit (Relu) after each layer. For each layer, the filter size is 3×3 and the number of filters is 64. The input to the network is the low-frequency sub-band LL^o from the image A_o, and the target is the LL^{gt} from the ground-truth image A_{gt}. To speed up the training process, we also use the residual learning strategy which learns the difference between target LL^{gt} and the input LL^o. The training goal is to minimize the ℓ_2 norm between the predicted outputs LL^r and the ground truth

LL^{gt}:

$$\mathcal{L} = \sum_{i=1}^{N} \|LL^{gt}(i) - LL^{r}(i)\|_F^2, \tag{3.5}$$

where LL^r is the sum of LL^o and the learned residual map. Here, the wavelet sub-bands are all in RGB colour space.

3.2.4 Wavelet domain style transfer

For the high-frequency sub-bands, we propose a wavelet domain style transfer (WDST) algorithm to improve their perceptual quality. Take the sub-band pair LH^p and LH^o for example, as shown in Fig. 3.3, the wavelet coefficients in LH^p are richer than those in LH^o, i.e., LH^p contains more non-zero wavelet coefficients than LH^o. However, the LH^p often has twisty structures. For example, the stripes in the zebra leg as shown in Fig. 3.5 are not smooth. In contrast, the LH^o has clear structures, despite lack of sufficient details. We wish to transfer the detailed wavelet coefficients in LH^p to LH^o, so that LH^o can have higher perceptual quality and at the same time keep its accurate structures. Thus, we regard LH^p as the style input and LH^o as the content input to generate an output sub-band LH^r using the wavelet domain style transfer. Different from the conventional style transfer algorithm where the inputs are pixel values, we use the wavelet coefficients as inputs in the WDST. Since the wavelet coefficients can be negative or larger than 1, a pre-processing step is required to normalize them between 0 and 1. Note that after the WDST, the results should be converted back to the right wavelet coefficients.

After normalization, for each high-frequency sub-band pair, the WDST algorithm is performed by minimizing a loss function that combines the content loss L_c, style loss L_s [101] and a ℓ_1 norm loss. The ℓ_1 norm loss is specifically added to preserve the sparsity of wavelet coefficients. Specifically, take the LH sub-band for example, the total loss function is defined as:

$$\mathcal{L}_{LH} = \alpha L_c(LH^r, LH^o) + \beta L_s(LH^r, LH^p) + \gamma \|LH^r\|_1, \tag{3.6}$$

where α, β and γ are the weights for the content, style and ℓ_1 norm loss, respectively.

The content loss is defined as the MSE between the feature maps of the content input and the generated output at a specific layer L of a pre-trained VGG network [46]:

$$L_c(LH^r, LH^o) = \frac{1}{2\sqrt{N_L M_L}} \sum_{i,j} (F^L_{ij}(LH^r) - F^L_{ij}(LH^o))^2. \tag{3.7}$$

Here, $F^L(LH^r)$ and $F^L(LH^o)$ are the feature maps of a pre-trained VGG network [46] at layer L with LH^r and LH^o as inputs, respectively. In addition, N_L is the number of feature maps at layer L, and M_L is the product of the width and height of the feature map. Different from the content loss which is calculated between the content input LH^o and LH^r, the style loss is calculated between the style input LH^p and LH^r. Moreover, unlike the content loss calculated at a single layer, the total style loss is defined by a weighed sum of the style loss at different layers:

$$L_s(LH^r, LH^p) = \sum_l w_l L^l_s(LH^r, LH^p), \tag{3.8}$$

where w_l is the weight for the style loss at the l-th layer. The $L^l_s(LH^r, LH^p)$ is calculated as the MSE between the Gram matrices of feature maps at the l-th layer in the pre-trained VGG network with LH^r and LH^p as inputs, respectively. Mathematically, the style loss at layer l is defined as:

$$L^l_s(LH^r, LH^p) = \frac{1}{4N_l^2 M_l^2} \sum_{ij} (G^l_{ij}(LH^r) - G^l_{ij}(LH^p))^2, \tag{3.9}$$

where $G^l(LH^r)$ and $G^l(LH^p)$ are the Gram matrices of LH^r and LH^p at the l-th layer, respectively. We have $G^l(LH^r) = F^l(LH^r)^T F^l(LH^r)$ and $G^l(LH^p) = F^l(LH^p)^T F^l(LH^p)$. In contrast to the artistic style transfer [101] which uses conv 4-2 (the second convolutional layer in the fourth block) to calculate the content loss, we use the layer conv 2-2 (the second convolutional layer in the second block) in VGG network [46] to calculate the content loss. Layers Relu1-1, Relu2-1, Relu3-1, Relu4-1, and Relu5-1 are used to calculate the style loss. With all loss defined, following Gatys *et al.* [101], we use the L-BFGS algorithm [102] to obtain LH^r in

(3.6). For the other high-frequency sub-bands, we use the similar approach to obtain the combined sub-band HL^r and HH^r.

After obtaining high-frequency sub-bands LH^r, HL^r, and HH^r, we need to de-normalize them. Then, we can reconstruct image A_r by performing 2D ISWT on these high-frequency sub-bands together with the low-frequency sub-band LL^r using the synthesis low-pass and high-pass filters H_1 and G_1. Here, for perfect reconstruction, H_1 and G_1 are the synthesis wavelet filters related to the analysis filters H_0 and G_0 used in the decomposition [103].

3.3 Numerical results

3.3.1 Experimental setup

For the 2D SWT, we use the LeGall 5/3 wavelet filters (*bior*2.2 family in Matlab) as the default wavelet filter. The wavelet decomposition level is 2, which means we have one low-frequency sub-band and six high-frequency sub-bands (see Fig. 3.2). In the LSE process, the loss function is minimized using stochastic gradient descent (SGD) with backpropagation. The batch size is 64, the basic learning rate is 0.01 and the momentum is 0.9. In the WDST process, the ratio between the content loss and the style loss is 10^{-3}, the ratio between the content loss and the ℓ_1 norm loss is 10^{-5}, and the weight of each layer when calculating the style loss is 0.2. The maximum iteration number is 5000 and 1000 for the first and second level decompositions, respectively. We use EDSR method [41] to obtain A_o, and CX method [50] to obtain A_p. The networks in EDSR and CX are trained using the DIV2K dataset [104]. Following [50], the perceptual score is calculated using NRQM [53]. We evaluate the performance of our method on various datasets, including Set5 [105], Set14 [32], BSD100 [106], Urban100 [107], and PIRM [108].

We compare our approach with many state-of-the-art SISR methods, which can be classified into three categories: methods that aim to improve the objective image

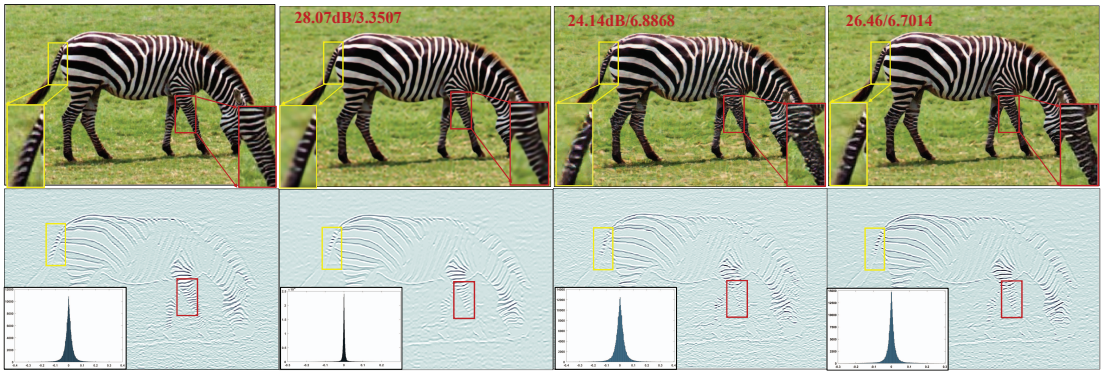
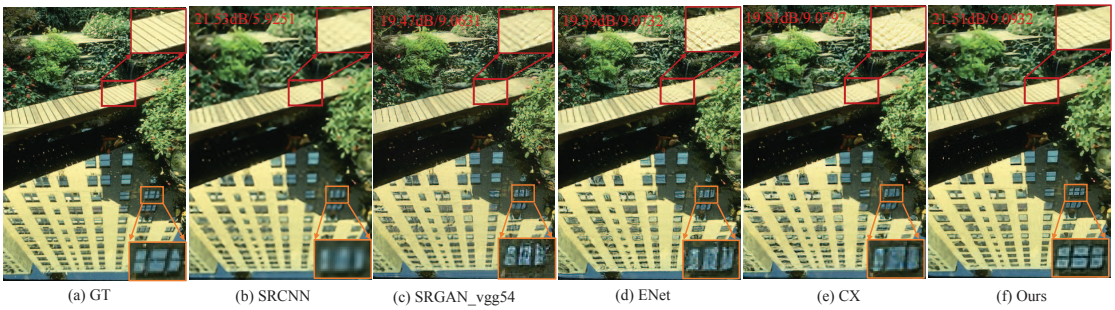


Figure 3.5: The first row shows the restored images of *Zebra* in Set 14 using EDSR, CX and our method, with the red values indicating the PSNR/NRQM values. The second row visualizes the HL sub-bands of the images in the first row, together with the histograms.

quality including A+ [34], Self-Ex [107], SRCNN [37], ESPCN [109], SRResNet-MSE [3], VDSR [16], EDSR [41], and RCAN [43]; methods that aim to improve the perceptual image quality including SRGAN-vgg54 [3], SRGAN-vgg22 [3], ENet [48], and CX [50]; and methods that aim to achieve a good tradeoff between the objective and perceptual quality including SRGAN-MSE [3], G–MGBP [110], PESR [111], EUSR [54], Deng [112] and ESRGAN [49].

3.3.2 Effectiveness of wavelet domain style transfer

In order to visually show the effectiveness of our WDST algorithm, we present in Fig. 3.5 the input content and style sub-bands, as well as the output sub-band after the WDST algorithm. As can be seen, the content sub-band lacks many high-frequency details and the style sub-band has messy structures, e.g, the zebra leg and tail. After the WDST, the output sub-band overcomes these drawbacks, which is now abundant in high-frequency details and has clear textures and structures. To some extent, the output sub-band corrects the wrong information in the style sub-band, with the guidance of the content sub-band. We also show the histogram distributions of the wavelet coefficients in the sub-bands in Fig. 3.5. It can be seen that, compared to the content input, our histogram is closer to the ground-truth, which is the reason why we have higher perceptual quality.



(a) GT (b) SRCNN (c) SRGAN_vgg54 (d) ENet (e) CX (f) Ours

Figure 3.6: Visual comparisons of image from BSD100 for 4× upscaling. The red numbers indicate the PSNR and NRQM values.

3.3.3 Perception-distortion (PD) performance

Fig. 3.1 compares the PD performance of different methods in the PSNR and NRQM plane. As we can see from this figure, methods A+, Self-Ex, SRCNN, ESPCN, SRResNet-MSE, VDSR, EDSR, RCAN occupy the upper left region which means that they have high objective quality but low perceptual quality. In contrast, methods SRGAN-vgg54, SRGAN-vgg22, ENet, and CX take up the bottom right region, which indicates that they have high perceptual quality but low objective quality. Other methods like SRGAN-MSE, PESR, Deng, and ESRGAN stand in the middle region, which are all trying to achieve a good tradeoff between distortion and perceptual quality. Among all these methods, our method is the closest to the bottom left corner, which means that we achieve the best trade-off between the objective and perceptual quality. Table 3.2 compares the numerical results of our method with SRGAN-MSE [3], G–MGBP [110] , PESR [111], Deng [112] and ESRGAN [49] (with α =0.8), which all aim to improve both the perceptual and objective quality. As we can see, in most cases, our method outperforms others in terms of both perceptual and objective quality.

Visual comparison. Figs. 3.6 and 3.7 visualize the images of our and other methods. We can see from Fig. 3.6 that our method can restore correctly the texture of the bridge and the structure of the window, while others either distort the texture or struggle to restore the structure. From Fig. 3.7, we can see that our method can restore the wall and lights clearly, while others fail to do so.

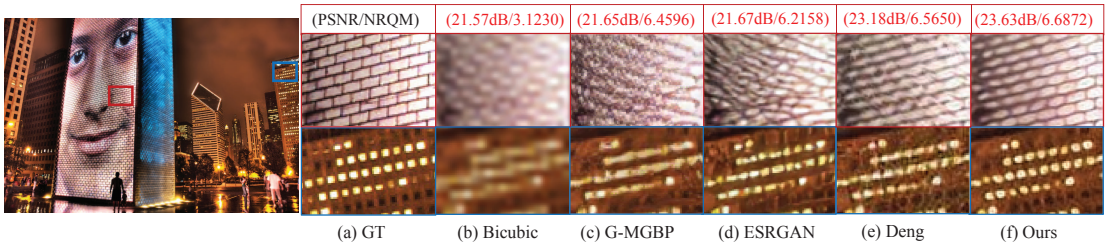


Figure 3.7: Visual comparisons of image from Urban100 for 4× upscaling. The red numbers indicate the PSNR and NRQM values.

3.3.4 Ablation study

Wavelet filter sensitivity. In our algorithm, we use wavelet filters to decompose each image into various sub-bands. In order to investigate the effects of wavelet filters on the performance of our algorithm, we present in Table 3.3 the PSNR, SSIM and NRQM results when using different wavelet filters. These filters include *haar*, *db*2 and *db*4 from Daubechies wavelet family, *bior*2.2 and *bior*4.4 from Biorthogonal wavelet family, *rbio*2.2 from Reverse biorthogonal wavelet family, and $coif2$ from Coifman wavelet family. From Table 3.3, we can see that the choice of the wavelet filter indeed has some effects on the results. Specifically, the *haar* filter has the highest SSIM value, the *db*2 filter performs best in the PSNR metric and the *bior*4.4 filter has the best perceptual quality. However, we notice that the difference among different filters is not very significant.

Content and Style inputs sensitivity. To show the position of our method more clearly, we draw in Fig. 3.8 the perception-distortion (PD) curve of EDSR and CX, which are the two default methods to generate A_o and A_p in our work. The curve is drawn by interpolating the pixel values of A_o and A_p with a parameter $\mu \in [0, 1]$, as follows:

$$A_r = \mu A_p + (1 - \mu) A_o. \tag{3.10}$$

Obviously, when μ increases, the perceptual quality increases while the objective quality decreases. As we can see from Fig. 3.8, our method is far lower than the PD curve, which means that we are much better than the simple interpolation of A_o and

Set5	Bicubic	EDSR	CX	SRGAN-MSE	G–MGBP	PESR	Deng	ESRGAN	Ours
PSNR	28.42	32.63	29.10	30.66	30.87	30.76	31.14	31.11	**31.46**
SSIM	0.8245	0.9117	0.8523	0.8758	0.8807	0.8915	0.8917	0.8839	**0.8929**
NRQM	3.7624	5.2106	7.9566	7.3082	7.3115	7.1344	7.0022	7.0724	**7.5180**
Set14	Bicubic	EDSR	CX	SRGAN-MSE	G–MGBP	PESR	Deng	ESRGAN	Ours
PSNR	26.10	28.95	26.01	27.01	27.56	27.57	27.77	27.53	**28.07**
SSIM	0.7850	0.8583	0.7839	0.8033	0.8206	0.8322	0.8325	0.8228	**0.8356**
NRQM	3.6598	5.3788	7.9423	**7.8770**	7.5042	7.5301	7.5575	7.5936	7.6827
BSD100	Bicubic	EDSR	CX	SRGAN-MSE	G–MGBP	PESR	Deng	ESRGAN	Ours
PSNR	25.96	27.80	24.58	25.98	26.59	26.33	26.46	26.44	**26.82**
SSIM	0.6675	0.7432	0.6432	0.6429	0.6926	0.6980	0.7048	0.7002	**0.7058**
NRQM	3.7207	5.7159	8.8007	8.4276	8.1790	8.3298	8.4452	8.3034	**8.5948**
Urban100	Bicubic	EDSR	CX	SRGAN-MSE	G–MGBP	PESR	Deng	ESRGAN	Ours
PSNR	23.14	26.86	24.00	-	25.15	25.88	25.96	26.08	**26.26**
SSIM	0.9011	0.9679	0.9313	-	0.9495	0.9610	0.9620	0.9624	**0.9649**
NRQM	3.4412	5.3365	6.7982	-	6.2190	6.3190	6.4317	6.1762	**6.4556**
PIRM	Bicubic	EDSR	CX	SRGAN-MSE	G–MGBP	PESR	Deng	ESRGAN	Ours
PSNR	26.51	28.72	25.41	-	27.17	27.11	27.48	26.66	**27.63**
SSIM	0.8232	0.8930	0.8177	-	0.8524	0.8649	0.8728	0.8529	**0.8755**
NRQM	3.8376	5.7116	8.5746	-	8.0556	8.2172	8.1665	8.2445	**8.3692**

Table 3.2: Benchmark comparisons for 4× upscaling, with the best results bold and the second bests underlined.

Filter	*haar*	*db*2	*bior*2.2	*rbior*2.2	*coif*2	*db*4	*bior*4.4
PSNR	28.06	**28.08**	28.07	27.96	28.05	28.06	28.05
SSIM	**0.8379**	0.8369	0.8356	0.8336	0.8344	0.8348	0.8343
NRQM	7.5109	7.6103	7.6827	7.6403	7.7101	7.6928	**7.7442**

Table 3.3: Effects of wavelet filter on Set 14 dataset.

A_p. To investigate our sensitivity to the content and style inputs, we also draw the PD curves of RCAN [43] and CX, SRResNet-MSE and SRGAN-vgg54 [3], together with our corresponding results. We can see that, even in the worst case (with SRResNet-MSE and SRGAN-vgg54 as inputs), our algorithm still achieves better PD trade-off (i.e., PSNR/NRQM=26.56 dB/8.5005) than Deng (26.46 dB/8.4452) and ESRGAN (26.44 dB/8.3034).

Sub-band	*LH*	*HL*	*HH*	PSNR	SSIM	NRQM
WDST	N	Y	Y	27.19	0.7195	7.8490
	Y	N	Y	27.28	0.7227	7.8343
	Y	Y	N	26.96	0.7105	8.0542
	Y	Y	Y	26.82	0.7058	8.5948

Table 3.4: Ablation study of WDST on each sub-band.

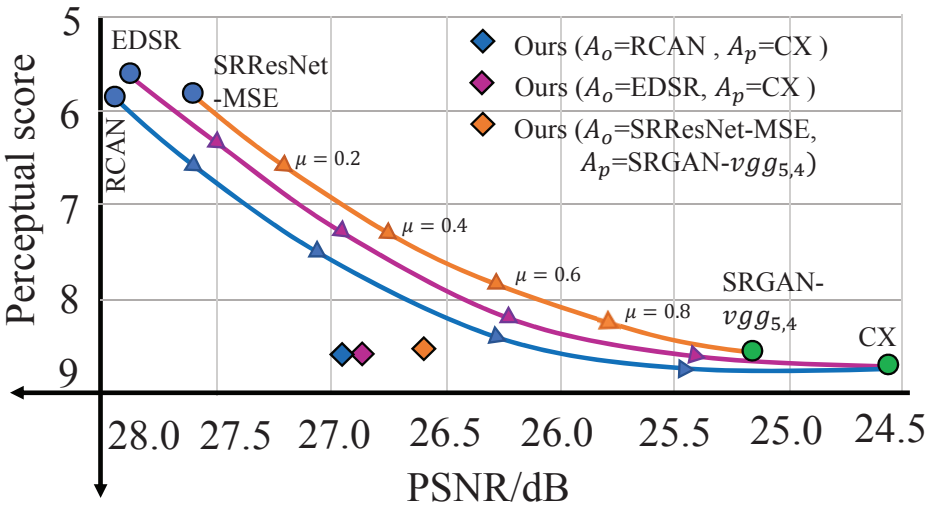


Figure 3.8: The perception-distortion (PD) curves of EDSR and CX, RCAN and CX, SRResNet-MSE and SRGAN-vgg54.

In order to study the effects of each high-frequency sub-band on the perception-distortion performance, we show in Table 3.4 the results when WDST is not performed on one of the sub-bands. From this table, we can see that each sub-band contributes to the perception-distortion performance. When WDST is not performed on any of them, the perceptual quality (NRQM) decreases significantly. However, compared with LH and HL sub-bands, the influence of HH sub-band is not very significant. This is because the HH sub-band contains the diagnonal information, which is often not as much as the horizontal and vertical information contained in the LH and HL sub-bands.

3.4 Summary

In this chapter, we presented a novel SISR method based on wavelet domain style transfer (WDST), which achieves a better perception-distortion (PD) tradeoff than the GAN based methods. Specifically, we propose to use 2D stationary wavelet transform (SWT) to decompose the image into one low-frequency and several high-frequency sub-bands. For the low-frequency sub-band, we improve its objective quality through an enhancement network. For the high-frequency sub-band, we

propose to use WDST to effectively improve its perceptual quality. By feat of the perfect reconstruction property of wavelets, these sub-bands can be re-combined to obtain an image which has simultaneously high objective and perceptual quality. The numerical results on various datasets show that our method achieves the best trade-off between the distortion and perceptual quality among the existing state-of-the-art SISR methods.

Chapter 4

RGB guided depth image super-resolution

4.1 Introduction

High quality depth images play an important role in many computer vision applications, such as image segmentation [113, 114], 3D object reconstruction [115] and gesture recognition [116]. 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 [117]. Despite this merit, ToF camera can only provide low resolution depth images, and this fact affects the progress of many related research areas. 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× and 16×, is required, and when the upscaling factor increases, the mapping

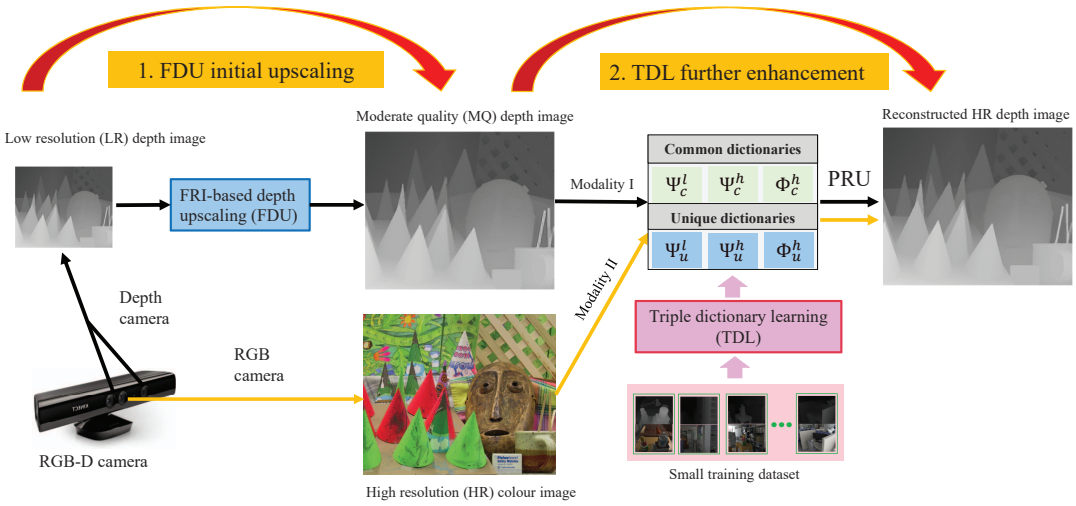
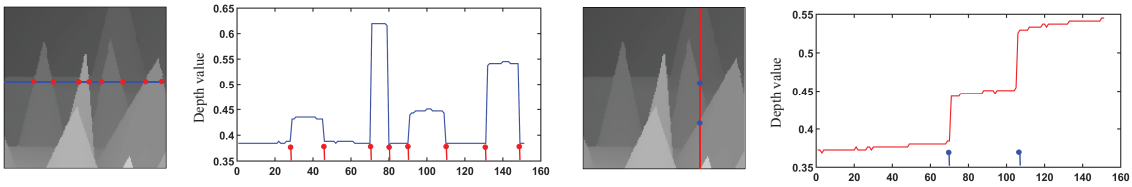


Figure 4.1: The cascaded framework of the proposed RADAR approach. The first stage is the FRI based upscaling (FDU) algorithm which uses FRI theory to do initial upscaling, and the second stage is projection-based rapid upscaling (PRU) algorithm based on triple dictionary learning (TDL) model, which extracts the useful information of colour/intensity image to further enhance the depth image.

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.

In this chapter, we propose a Robust Algorithm for Depth imAge super Resolution (RADAR) that combines the power of finite rate of innovation (FRI) theory [28, 29] and multimodal dictionary learning [30], 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 LR depth image to a moderate quality (MQ) 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 signal, we propose a FRI-based depth upscaling (FDU) algorithm to super-resolve the depth 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



(a) Row of depth image (b) Column of depth image

Figure 4.2: 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.

rapid upscaling (PRU) algorithm based on a triple dictionary learning (TDL) model. The cascaded framework is shown in Fig. 4.1. 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 detail 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.

4.2 Proposed Method

4.2.1 FRI-based Depth Upscaling

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 [29]. As observed from Fig. 4.2, the rows and columns of depth images are quite close

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

Table 4.1: Notation list.

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 [28,118]. For more details of the 1-D FRI reconstruction, please see Appendix A. 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. 4.3, the HR image can be decomposed into four sub-bands using 2D Discrete Wavelet Transform (DWT) [119], 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. Likewise, we reconstruct HR_v by doing linear interpolation vertically and then FRI reconstruction horizontally. The HL sub-band, HL^v, can be extracted

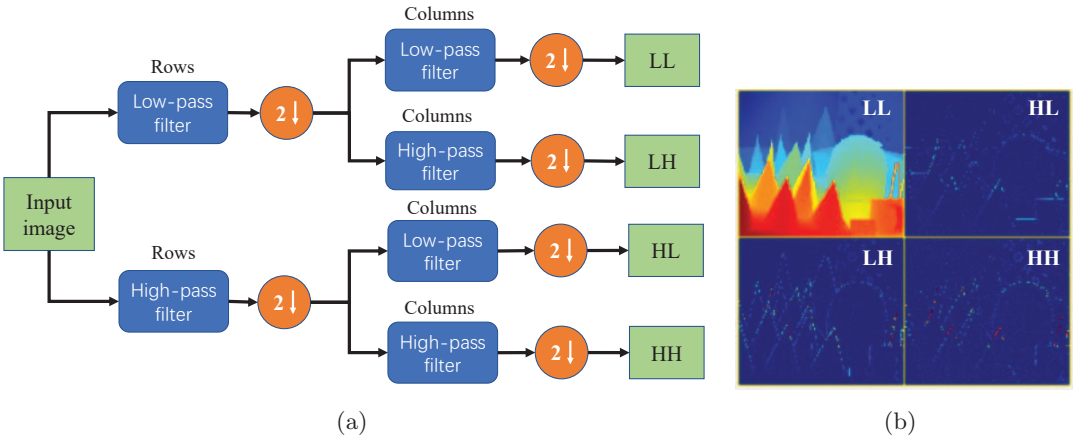


Figure 4.3: (a) shows the decomposition of one image using 2D DWT, and (b) visualizes the four sub-bands of image *Cones*.

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. 4.4 illustrates this process for 2 $\times$ upcaling.

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 [120], we employ an internal self-learning algorithm proposed in [121] to correct the 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 [120, 121].

Noisy case. In the case of noisy LR depth 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 directly with the three high-frequency sub-bands would lead to a noisy output. The intuitive solution is to find a denoised version of the noisy LR input to combine with

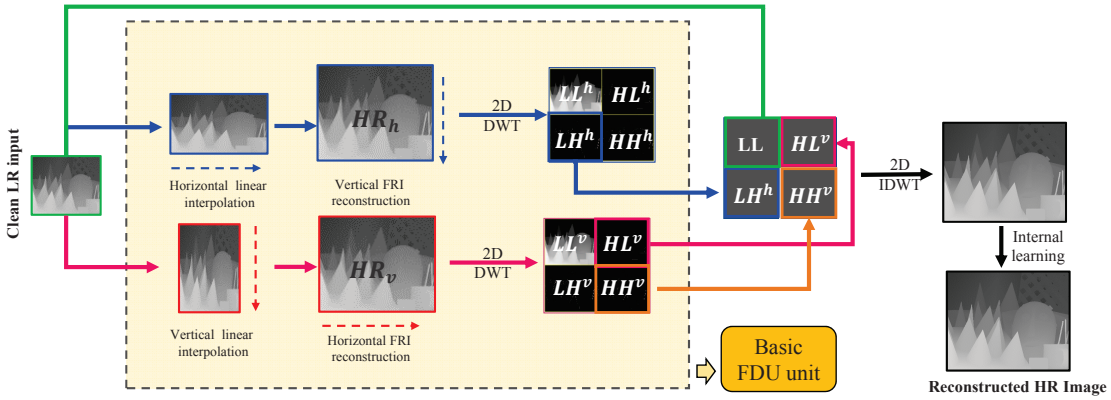


Figure 4.4: FDU algorithm with clean LR depth image.

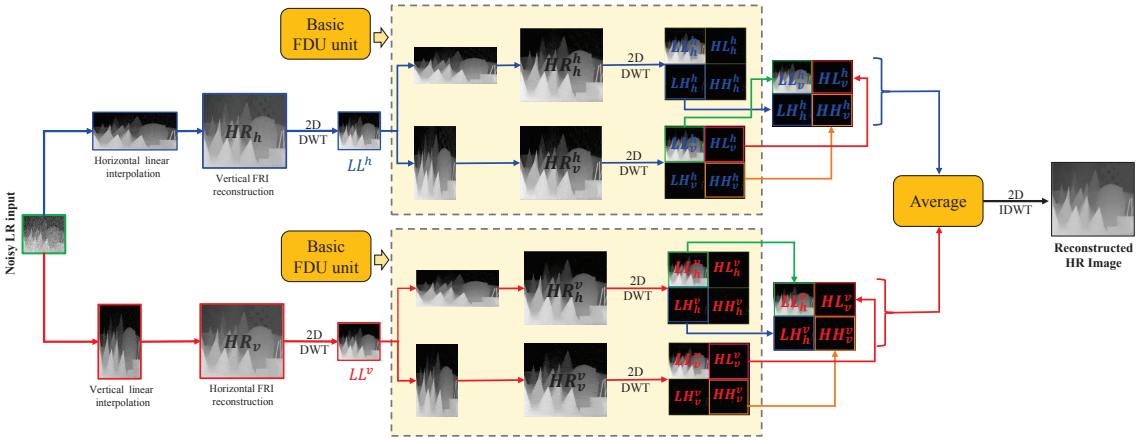


Figure 4.5: FDU algorithm with noisy LR depth image.

the other three sub-bands. Here, we have two options: the LL^h extracted from HR_h or the LL^v extracted from HR_v. However, the problem is that LL^h is only denoised along the horizontal lines but still noisy along the vertical lines, and conversely, LL^v is only denoised along the vertical lines but still noisy along the 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 4.1 and the algorithm is shown in Fig. 4.5.

As it can be seen from the figure, the algorithm consists of 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. 4.5. 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.

4.2.2 Triple Dictionary Learning

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 [30] to fully use the guidance of HR intensity images.

The basic insight in the TDL model [30] is that the MQ depth image, HR depth image, and the corresponding HR intensity image are from the same scene. Therefore, they share some common 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:

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Psi}_c^l & \boldsymbol{\Psi}_u^l & \mathbf{0} \\ \boldsymbol{\Psi}_c^h & \boldsymbol{\Psi}_u^h & \mathbf{0} \\ \boldsymbol{\Phi}_c^h & \mathbf{0} & \boldsymbol{\Phi}_u^h \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{u} \\ \mathbf{v} \end{bmatrix}, \tag{4.1}$$

where $\mathbf{c}$, $\mathbf{u}$ and $\mathbf{v}$ are the sparse coefficients. For $\boldsymbol{y}$ and $\boldsymbol{z}$, their CDs share the same sparse coefficient $\mathbf{c}$, while the UDs have distinct sparse coefficients $\mathbf{u}$ and $\mathbf{v}$. 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 [30]

$$\min_{\substack{\boldsymbol{\Psi}_c^l,\boldsymbol{\Psi}_c^h,\boldsymbol{\Phi}_c^h,\\ \{\boldsymbol{\Psi}_u^l,\boldsymbol{\Psi}_u^h,\boldsymbol{\Phi}_u^h,\}\\ \mathbf{C},\mathbf{U},\mathbf{V}}} \left\| \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Psi}_c^l & \boldsymbol{\Psi}_u^l & \mathbf{0} \\ \boldsymbol{\Psi}_c^h & \boldsymbol{\Psi}_u^h & \mathbf{0} \\ \boldsymbol{\Phi}_c^h & \mathbf{0} & \boldsymbol{\Phi}_u^h \end{bmatrix} \begin{bmatrix} \mathbf{C} \\ \mathbf{U} \\ \mathbf{V} \end{bmatrix} \right\|_F^2, \quad s.t. \quad \|\boldsymbol{c}_k\|_0 + \|\boldsymbol{u}_k\|_0 + \|\boldsymbol{v}_k\|_0 \leq L, \quad \forall k, \tag{4.2}$$

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. Each dictionary has size $\mathbb{R}^{M\times Q}$, and the sparse coefficients $\boldsymbol{C}$, $\boldsymbol{U}$ and $\boldsymbol{V} \in \mathbb{R}^{Q\times P}$. 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 (4.2), $\boldsymbol{c}_k$, $\boldsymbol{u}_k$, $\boldsymbol{v}_k$ are the k-th ($0 < k \leq P$) columns of matrices $\boldsymbol{C}$, $\boldsymbol{U}$ and $\boldsymbol{V}$ respectively, $\|\cdot\|_0$ is the ℓ_0 norm and L is the sparsity constraint. Since (4.2) is non-convex, it is difficult to get a solution for all the six dictionaries simultaneously. Thus, we first relax (4.2) to (4.3) by disregarding dictionaries $\boldsymbol{\Psi}_c^h$ and $\boldsymbol{\Psi}_u^h$ temporarily, and then we have

$$\min_{\substack{\{\boldsymbol{\Psi}_c^l,\boldsymbol{\Phi}_c^h,\boldsymbol{\Psi}_u^l,\boldsymbol{\Phi}_u^h,\}\\ \mathbf{C},\mathbf{U},\mathbf{V}}} \left\| \begin{bmatrix} \mathbf{X} \\ \mathbf{Z} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Psi}_c^l & \boldsymbol{\Psi}_u^l & \mathbf{0} \\ \boldsymbol{\Phi}_c^h & \mathbf{0} & \boldsymbol{\Phi}_u^h \end{bmatrix} \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{U} \\ \boldsymbol{V} \end{bmatrix} \right\|_F^2, \quad s.t. \quad \|\boldsymbol{c}_k\|_0 + \|\boldsymbol{u}_k\|_0 + \|\boldsymbol{v}_k\|_0 \leq L, \quad \forall k, \tag{4.3}$$

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 [122] to calculate the sparse coefficients, and then by fixing the coefficients and the unique dictionaries, we use K-SVD [123] to update the common dictionaries. This process is repeated until convergence, i.e., the change of loss in (4.3) is below a pre-set threshold. 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 $\boldsymbol{C}$, $\boldsymbol{U}$ and $\boldsymbol{V}$, the next step is to learn the remaining dictionaries $\boldsymbol{\Psi}_c^h$ and $\boldsymbol{\Psi}_u^h$. From (4.2), we can observe that the HR depth dictionaries $\boldsymbol{\Psi}_c^h$ and $\boldsymbol{\Psi}_u^h$ share the same sparse coefficients, i.e., $\boldsymbol{C}$ and $\boldsymbol{U}$, 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 $\boldsymbol{Y}$:

$$\{\boldsymbol{\Psi}_c^h, \boldsymbol{\Psi}_u^h\} = \underset{\boldsymbol{\Psi}_c^h, \boldsymbol{\Psi}_u^h}{\operatorname{argmin}} \left\| \boldsymbol{Y} - \begin{bmatrix} \boldsymbol{\Psi}_c^h & \boldsymbol{\Psi}_u^h \end{bmatrix} \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{U} \end{bmatrix} \right\|_F^2 + \lambda \left\| \begin{bmatrix} \boldsymbol{\Psi}_c^h & \boldsymbol{\Psi}_u^h \end{bmatrix} \right\|_F^2. \tag{4.4}$$

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

$$\begin{bmatrix} \boldsymbol{\Psi}_c^h & \boldsymbol{\Psi}_u^h \end{bmatrix} = \boldsymbol{Y} \begin{bmatrix} \boldsymbol{C}^T & \boldsymbol{U}^T \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{U} \end{bmatrix} \begin{bmatrix} \boldsymbol{C}^T & \boldsymbol{U}^T \end{bmatrix} + \lambda \boldsymbol{I} \right)^{-1}, \tag{4.5}$$

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

4.2.3 Projection-based Rapid Upscaling

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,

$$\underset{\mathbf{c},\mathbf{u},\mathbf{v}}{\operatorname{argmin}} \|\mathbf{c}\|_0 + \|\mathbf{u}\|_0 + \|\mathbf{v}\|_0, s.t., \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{z}_t \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Psi}_c^l & \boldsymbol{\Psi}_u^l & \mathbf{0} \\ \boldsymbol{\Phi}_c^h & \mathbf{0} & \boldsymbol{\Phi}_u^h \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{u} \\ \mathbf{v} \end{bmatrix}. \tag{4.6}$$

In theory, one can solve (4.6) by OMP algorithm [122] to obtain the coefficients $\mathbf{c}, \mathbf{u}, \mathbf{v}$. After that, $\boldsymbol{y}_t$ can be recovered as

$$\boldsymbol{y}_t = \boldsymbol{\Psi}_c^h \mathbf{c} + \boldsymbol{\Psi}_{\mathbf{u}}^{\mathbf{h}} \mathbf{u}. \tag{4.7}$$

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 [124] 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 J 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 (4.6) using OMP algorithm. Specifically, for the j-th $(0 < j \leq J)$ sample in $\{\boldsymbol{X}^{(i)};\boldsymbol{Z}^{(i)}\}$, its sparse coefficients are denoted as $[\boldsymbol{c}_j^{(i)};\boldsymbol{u}_j^{(i)};\boldsymbol{v}_j^{(i)}]$. By stacking together the sparse coefficients of all samples in each subset separately, we obtain three coefficient matrices per subset as follows,

$$\begin{aligned}
\boldsymbol{M}_c^{(i)} &= \left[\boldsymbol{c}_1^{(i)}, \boldsymbol{c}_2^{(i)}, \ldots, \boldsymbol{c}_J^{(i)}\right]; \\
\boldsymbol{M}_u^{(i)} &= \left[\boldsymbol{u}_1^{(i)}, \boldsymbol{u}_2^{(i)}, \ldots, \boldsymbol{u}_J^{(i)}\right]; \\
\boldsymbol{M}_v^{(i)} &= \left[\boldsymbol{v}_1^{(i)}, \boldsymbol{v}_2^{(i)}, \ldots, \boldsymbol{v}_J^{(i)}\right].
\end{aligned} \tag{4.8}$$

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:

$$\boldsymbol{\Psi}^{(i)} = \underset{\boldsymbol{\Psi}^{(i)}}{\operatorname{argmin}} \left\| \begin{bmatrix} \boldsymbol{X}^{(i)} \\ \boldsymbol{Y}^{(i)} \end{bmatrix} - \boldsymbol{\Psi}^{(i)} \begin{bmatrix} \boldsymbol{M}_c^{(i)} \\ \boldsymbol{M}_u^{(i)} \end{bmatrix} \right\|_F^2 + \lambda \left\| \boldsymbol{\Psi}^{(i)} \right\|_F^2, \tag{4.9}$$

and

$$\boldsymbol{\Phi}^{(i)} = \underset{\boldsymbol{\Phi}^{(i)}}{\operatorname{argmin}} \left\| \boldsymbol{Z}^{(i)} - \boldsymbol{\Phi}^{(i)} \begin{bmatrix} \boldsymbol{M}_c^{(i)} \\ \boldsymbol{M}_v^{(i)} \end{bmatrix} \right\|_F^2 + \lambda \left\| \boldsymbol{\Phi}^{(i)} \right\|_F^2 . \tag{4.10}$$

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

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

and

$$\boldsymbol{\Phi}^{(i)} = \boldsymbol{Z}^{(i)} \left[\boldsymbol{M}_c^{(i)T} \boldsymbol{M}_v^{(i)T} \right] \left(\begin{bmatrix} \boldsymbol{M}_c^{(i)} \\ \boldsymbol{M}_v^{(i)} \end{bmatrix} \left[\boldsymbol{M}_c^{(i)T} \boldsymbol{M}_v^{(i)T} \right] + \lambda \boldsymbol{I} \right)^{-1}, \tag{4.11}$$

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 (4.6) with the sub-dictionaries learned in (4.11). Then, suppose that $\boldsymbol{\beta} = [\mathbf{c}, \mathbf{u}, \mathbf{v}]^{\mathbf{T}} \in \mathbb{R}^{\mathbf{3Q}}$, $\boldsymbol{D}_l^{(i)} = \begin{bmatrix} \boldsymbol{\Psi}_c^{l(i)} & \boldsymbol{\Psi}_u^{l(i)} & \mathbf{0} \\ \boldsymbol{\Phi}_c^{h(i)} & \mathbf{0} & \boldsymbol{\Phi}_u^{h(i)} \end{bmatrix} \in \mathbb{R}^{2M \times 3Q}$ and relax ℓ_0 norm to ℓ_2 norm, (4.6) can be rewritten as

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\| \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{z}_t \end{bmatrix} - \boldsymbol{D}_l^{(i)} \boldsymbol{\beta} \right\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2, \tag{4.12}$$

where λ is a regularization parameter. The reason why we relax the $\ell 0$ norm with $\ell 2$ norm is to make Eq. (4.12) a ridge regression problem, so that it can have a closed-form solution, which helps to speed up the upscaling process. The solution

of (4.12) is as follows,

$$\boldsymbol{\beta} = (\boldsymbol{D}_l^{(i)T}\boldsymbol{D}_l^{(i)} + \lambda\boldsymbol{I})^{-1}\boldsymbol{D}_l^{(i)T}\begin{bmatrix}\boldsymbol{x}_t\\ \boldsymbol{z}_t\end{bmatrix}, \tag{4.13}$$

where $\boldsymbol{D}_l^{(i)T} \in \mathbb{R}^{3Q\times 2M}$ is the transpose of $\boldsymbol{D}_l^{(i)} \in \mathbb{R}^{2M\times 3Q}$, and $\boldsymbol{I} \in \mathbb{R}^{3Q\times 3Q}$ is an identity matrix. With coefficients $\boldsymbol{\beta}$, the HR depth feature $\boldsymbol{y}_t$ can be reconstructed by

$$\boldsymbol{y}_t = \boldsymbol{D}_h^{(i)}\boldsymbol{\beta} = \boldsymbol{D}_h^{(i)}(\boldsymbol{D}_l^{(i)T}\boldsymbol{D}_l^{(i)} + \lambda\boldsymbol{I})^{-1}\boldsymbol{D}_l^{(i)T}\begin{bmatrix}\boldsymbol{x}_t\\ \boldsymbol{z}_t\end{bmatrix}, \tag{4.14}$$

where $\boldsymbol{D}_h^{(i)} = \begin{bmatrix}\boldsymbol{\Psi}_c^{h(i)} & \boldsymbol{\Psi}_u^{h(i)} & \boldsymbol{0}\end{bmatrix} \in \mathbb{R}^{M\times 3Q}$.

For this reason, the sub-projection matrix $\boldsymbol{P}^{(i)} \in \mathbb{R}^{M\times 2M}$ for the i-th subset is given by

$$\boldsymbol{P}^{(i)} = \boldsymbol{D}_h^{(i)}(\boldsymbol{D}_l^{(i)T}\boldsymbol{D}_l^{(i)} + \lambda\boldsymbol{I})^{-1}\boldsymbol{D}_l^{(i)T}. \tag{4.15}$$

Moreover, the sub-projections $\{\boldsymbol{P}^{(i)}\}_{i=1}^N$ can be pre-stored to speed up the upscaling process. In the upscaling process, with 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 then multiply the joint input $[\boldsymbol{x}_t; \boldsymbol{z}_t]$ by the sub-projection matrix $\boldsymbol{P}^{(i)}$ corresponding to this centroid to get the HR depth patch $\boldsymbol{y}_t$. We also explored the possibility of searching for more than one nearest neighbour, i.e., we use the weighted sub-projections of several nearest neighbours to reconstruct the HR image. However, the results are worse than those obtained using only the nearest neighbour.

In the PRU algorithm, the full procedure to calculate the sub-projections in each cluster is as follows: firstly, for the K samples in each cluster, we calculate their sparse representations through Eq. (4.6), and then we learn the sub-dictionaries through Eqs. (4.9)-(4.11), and finally the sub-projections are obtained by using Eq. (4.15). Table 4.2 shows the summarized procedure for PRU algorithm. In summary, in the upscaling process, given a LR depth image, we first use FDU algorithm to get a MQ depth image. 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.

4.3 Numerical results

4.3.1 Datasets

For the training dataset, our approach needs both depth images and their corresponding RGB images for dictionary learning. Compared with the deep learning based methods [9, 125], we use a very small training dataset, which is composed of only 15 depth images with their registered RGB images selected from the New Tsukuba dataset provided by [4]. The resolution of training images is 640×480. Since the RGB and depth image patch should share the same common sparse representations as in Eq. (4.1), they should have the same number of channels. Following [8, 9], we convert the RGB images to YCbCr format and only use the luminance channel for training.

For the testing dataset, we evaluate the performance of our approach on five publicly available datasets, including Middlebury stereo dataset [126], Sintel stereo dataset [127], New Tsukuba (NT) dataset [4], NYU indoor scene dataset [128], and ToFMark dataset [5]. 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 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 4.3.

Training process

– **Input:** Training features $\{\mathbf{X};\mathbf{Z}\}$ and $\{\mathbf{Y}\}$, global dictionaries $\mathbf{\Psi}_c^l$, $\mathbf{\Psi}_u^l$, $\mathbf{\Psi}_c^h$, $\mathbf{\Psi}_u^h$, $\mathbf{\Phi}_c^h$, $\mathbf{\Phi}_u^h$.

– **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 J nearest samples to the cluster centroid using k-NN. These J samples form a subset.

  2 Calculate the sparse coefficients for the J samples in the subset using (4.6).

  3 Learn the sub-dictionaries for each subset using (4.9), (4.10), and (4.11).

  4 Learn the sub-projections $\{\boldsymbol{P}^{(i)}\}_{i=1}^N$ for each subset using (4.15).

  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 $\{\boldsymbol{P}^{(i)}\}_{i=1}^N$.

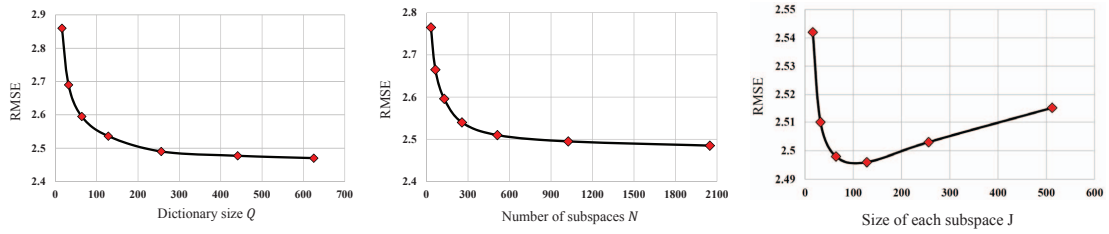
– **Output:** Reconstructed feature $\boldsymbol{y}_t$.

- 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 $\boldsymbol{y}_t$.

Table 4.2: The proposed projection-based rapid upscaling algorithm

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

Table 4.3: Testing datasets

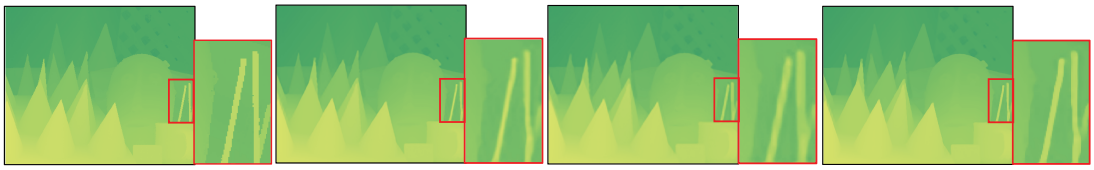


(a) (b) (c)

Figure 4.6: (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 J of each subspace. The results are from the validation dataset, which is composed of 10 images from the New Tsukuba dataset [4].

4.3.2 Experimental setup

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. 4.6 (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×, 4×, 8×, and 16× 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 is set to 0.1 for all cases. In the PRU process, as shown in Fig. 4.6 (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 J 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.



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

Figure 4.7: 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.

4.3.3 Performance Analysis of Our Method

Individual contributions. Table 4.4 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× and 4×, FDU plays a more important role in large factors, i.e., 8× and 16×. As can be seen, with only the FDU, the RMSE value improves on average by nearly 1.8 for 8× and 16× upscaling, compared with 1.0 for 2× and 4× upscaling. To make the individual contribution of each part more evident, we show in Fig. 4.7 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. (4.6)). As shown in Table 4.4, the PRU algorithm performs better in both RMSE and SSIM.

Use of intensity images. In order to test the contribution of the intensity image

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 4.4: Individual contributions of each element 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 4.5: RMSE comparison with and without the guidance of intensity images at different scaling factors on the four noise-free datasets.

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 (4.2), the MQ depth and HR depth. Table 4.5 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 4.6 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,

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**

Table 4.6: OMP versus PRU for both clustering and non-clustering cases on Middlebury dataset.

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).

4.3.4 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.* [31], Timofte *et al.* [10], Huang *et al.* [107], Dong *et al.* [37], Wang *et al.* [11], and Kim *et al.* [16]. 2) State-of-the-art single depth image super-resolution methods, including Aodha *et al.* [129], Ferstl *et al.* [14], Xie *et al.* [6], and Riegler *et al.* [125]. 3) State-of-the-art intensity guided depth image super-resolution methods, including Park *et al.* [13], Ferstl *et al.* [5], Lu *et al.* [7], Song *et al.* [9], and Gu *et al.* [8]. Note that [9,11,16,37,125] are all deep learning based methods, which use a large dataset for training. The numerical results of these methods are obtained either by implementing the source code, or emailing the authors for the upscaling results. 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 that the fast RADAR version removes the self-learning part in the FDU algorithm, which is slightly time-consuming.

Small upscaling factors. Table 4.7 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 others [9,11,16,37], 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 4.8. 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 4.9 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 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 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 4.10 shows the performance of our and the deep learning based methods [11,16,37] 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 4.11 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×, i.e., 4.69s per image on average, which is faster than that of Ferstl *et al.* [5] (135s per image) and Lu *et al.* [7] (286s per image). However, it is still slower than that of the deep learning based method, for example, Wang *et al.* [11] (0.93s per image) and Kim *et al.* [16] (2.65s per image).

4.3.5 Qualitative results

Figs. 4.8 and 4.9 visualize the 4× upscaling results of *Tsukuba* and *Bamboo* from Middlebury and Sintel 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 [8, 10] or diffused edges [5] [7].

In Fig. 4.10, we compare the 8× 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 [5] and [7].

Methods	Scaling factor=2							
	Cones		Teddy		Tsukuba		Venus	
	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
Aodha *et al.* [129]	4.32	0.9606	3.28	0.9690	9.11	0.9364	2.20	0.9874
Ferstl *et al.* [14]	2.21	0.9866	1.72	0.9884	5.33	0.9766	1.12	0.9963
Xie *et al.* [6]	2.73	0.9633	2.49	0.9625	6.35	0.9464	1.64	0.9852
Timofte *et al.* [10]	2.13	0.9871	1.72	0.9882	4.98	0.9776	1.16	0.9960
Huang *et al.* [107]	2.63	0.9846	2.43	0.9853	6.08	0.9544	2.17	0.9705
Park *et al.* [13]	2.85	0.9699	2.18	0.9767	6.89	0.9320	1.26	0.9910
Ferstl *et al.* [5]	3.81	0.9788	2.93	0.9795	6.89	0.9576	1.39	0.9938
Lu *et al.* [7]	3.28	0.9875	2.07	0.9895	7.35	0.9659	1.30	0.9952
Gu *et al.* [8]	1.75	0.9858	1.65	0.9841	3.41	0.9848	0.59	0.9981
Dong *et al.* [37]	1.62	0.9900	1.34	0.9911	3.68	0.9798	0.63	0.9971
Wang *et al.* [11]	1.86	0.9900	1.37	0.9918	3.89	0.9858	0.83	0.9979
Kim *et al.* [16]	1.45	0.9932	1.15	0.9940	3.03	0.9922	0.70	0.9984
Song *et al.* [9]	1.44	0.9915	1.20	0.9920	3.02	0.9898	0.56	0.9989
Riegler *et al.* [125]	**1.00**	-	0.82	-	2.38	-	**0.20**	-
RADAR	1.01	**0.9964**	**0.80**	**0.9959**	2.24	**0.9959**	0.23	**0.9997**
RADAR (fast)	1.15	0.9952	0.86	0.9954	**2.04**	0.9956	0.24	0.9997
Methods	Scaling factor = 4							
	Cones		Teddy		Tsukuba		Venus	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
Bicubic	3.71	0.9620	2.84	0.9688	8.32	0.9321	1.92	0.9876
Aodha *et al.* [129]	3.69	0.9392	4.11	0.9520	7.69	0.9080	2.65	0.9822
Ferstl *et al.* [14]	3.57	0.9645	2.65	0.9716	7.54	0.9413	1.78	0.9893
Xie *et al.* [6]	4.41	0.9319	3.28	0.9331	9.78	0.8822	2.37	0.9730
Timofte *et al.* [10]	3.41	0.9709	2.43	0.9750	7.35	0.9507	1.59	0.9916
Huang *et al.* [107]	5.29	0.9533	4.23	0.9508	9.76	0.9312	3.06	0.9749
Park *et al.* [13]	6.54	0.9420	4.35	0.9553	12.12	0.8981	2.36	0.9862
Ferstl *et al.* [5]	4.66	0.9625	3.67	0.9707	9.91	0.9245	1.60	0.9900
Lu *et al.* [7]	4.02	0.9697	2.73	0.9782	10.20	0.9212	1.77	0.9888
Gu *et al.* [8]	3.93	0.9627	2.86	0.9694	8.36	0.9346	1.59	0.9922
Dong *et al.* [37]	3.55	0.9712	2.56	0.9813	7.88	0.9552	1.26	0.9883
Wang *et al.* [11]	3.07	0.9756	2.03	0.9811	6.29	0.9677	1.20	0.9952
Kim *et al.* [16]	2.49	0.9837	1.72	0.9859	5.21	0.9762	0.92	0.9971
Song *et al.* [9]	2.80	0.9802	1.80	0.9837	6.21	0.9641	0.88	0.9972
Riegler *et al.* [125]	2.93	-	**1.50**	-	6.63	-	**0.38**	-
RADAR	**2.26**	**0.9851**	1.63	**0.9858**	**4.90**	**0.9777**	0.73	**0.9981**
RADAR (fast)	2.67	0.9813	1.83	0.9836	5.51	0.9745	0.72	0.9980

Table 4.7: Results on Middlebury dataset for 2× and 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
Xie *et al.* [6]	1.42	0.9767	2.62	0.9711	1.84	0.9686	2.33	0.9699
Timofte *et al.* [10]	0.84	0.9935	1.47	0.9898	1.01	0.9928	1.40	0.9897
Huang *et al.* [107]	0.96	0.9885	1.65	0.9866	1.44	0.9884	1.65	0.9864
Park *et al.* [13]	0.96	0.9885	1.78	0.9855	1.37	0.9892	1.86	0.9823
Ferstl *et al.* [5]	1.02	0.9875	1.88	0.9844	1.24	0.9914	1.82	0.9831
Lu *et al.* [7]	1.00	0.9882	1.83	0.9849	1.71	0.9721	2.06	0.9735
Gu *et al.* [8]	1.28	0.9889	1.79	0.9856	1.40	0.9897	1.77	0.9850
Dong *et al.* [37]	0.79	0.9945	1.28	0.9923	0.92	0.9946	1.22	0.9913
Wang *et al.* [11]	1.45	0.9929	1.21	0.9924	0.90	0.9954	1.16	0.9925
Kim *et al.* [16]	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
Xie *et al.* [6]	8.79	0.9438	19.02	0.8301	9.14	0.9221	12.21	0.8869
Timofte *et al.* [10]	5.05	0.9756	12.06	0.8910	5.34	0.9681	7.03	0.9462
Huang *et al.* [107]	5.32	0.9720	12.45	0.8784	5.69	0.9616	7.30	0.9412
Park *et al.* [13]	6.03	0.9678	12.05	0.8910	7.13	0.9379	9.45	0.9067
Ferstl *et al.* [5]	5.99	0.9701	11.54	0.8950	6.40	0.9563	8.01	0.9298
Lu *et al.* [7]	5.53	0.9712	10.61	0.9028	6.10	0.9610	8.31	0.9266
Gu *et al.* [8]	6.04	0.9766	13.35	0.9001	6.15	0.9613	8.10	0.9470
Dong *et al.* [37]	5.02	0.9761	11.89	0.8934	5.32	0.9684	6.87	0.9487
Wang *et al.* [11]	4.29	0.9850	9.63	0.9389	4.37	0.9769	5.94	0.9664
Kim *et al.* [16]	**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
Xie *et al.* [6]	1.77	0.9889	1.92	0.9897	2.83	0.9768	1.99	0.9855
Timofte *et al.* [10]	0.91	0.9942	0.94	0.9936	1.44	0.9891	1.14	0.9932
Huang *et al.* [107]	1.12	0.9934	1.22	0.9926	1.88	0.9834	1.47	0.9898
Park *et al.* [13]	1.53	0.9903	1.37	0.9921	2.14	0.9823	1.64	0.9885
Ferstl *et al.* [5]	1.78	0.9889	1.26	0.9924	2.25	0.9810	1.78	0.9867
Lu *et al.* [7]	1.38	0.9920	1.40	0.9916	2.19	0.9802	1.46	0.9898
Gu *et al.* [8]	1.24	0.9914	1.22	0.9935	1.96	0.9844	1.41	0.9905
Dong *et al.* [37]	0.91	0.9942	0.92	0.9941	1.42	0.9901	1.13	0.9927
Wang *et al.* [11]	**0.79**	0.9953	0.74	0.9966	1.28	0.9909	0.96	0.9943
Kim *et al.* [16]	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 4.8: Results on the NT, Sintel and NYU datasets for 4× 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.* [31]	5.33	0.9448	3.69	0.9530	11.74	0.9002	2.21	0.9822
Timofte *et al.* [10]	5.09	0.9452	3.48	0.9572	11.33	0.9023	1.87	0.9892
Park *et al.* [13]	7.81	0.9389	5.52	0.9456	17.60	0.8498	3.35	0.9642
Ferstl *et al.* [5]	6.68	0.9422	5.36	0.9351	16.30	0.8575	2.42	0.9800
Lu *et al.* [7]	5.54	0.9442	3.76	0.9561	13.32	0.8852	2.20	0.9870
Gu *et al.* [8]	5.59	0.9373	4.05	0.9489	12.72	0.8761	2.61	0.9806
Dong *et al.* [37]	5.28	0.9449	3.55	0.9567	11.32	0.9023	1.98	0.9889
Wang *et al.* [11]	4.75	0.9483	3.14	0.9597	10.59	0.8864	1.94	0.9872
Kim *et al.* [16]	5.55	0.9322	4.04	0.9451	12.08	0.8693	2.79	0.9766
Song *et al.* [9]	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.* [31]	7.45	0.9033	5.42	0.9287	16.53	0.8291	2.88	0.9687
Timofte *et al.* [10]	6.99	0.9096	4.95	0.9312	16.20	0.8301	2.63	0.9709
Park *et al.* [13]	10.23	0.8903	8.36	0.9124	19.36	0.8122	5.16	0.9547
Ferstl *et al.* [5]	11.84	0.8768	8.13	0.8977	26.54	0.8026	4.94	0.9607
Lu *et al.* [7]	10.96	0.8922	8.01	0.9135	16.76	0.8232	3.26	0.9655
Gu *et al.* [8]	7.40	0.9189	5.81	0.9316	17.59	0.8272	3.77	0.9723
Dong *et al.* [37]	7.18	0.9076	5.26	0.9379	16.25	0.8311	2.65	0.9710
Wang *et al.* [11]	7.11	0.9145	5.45	0.9256	18.82	0.7812	3.76	0.9664
Kim *et al.* [16]	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 4.9: Results on the Middlebury dataset for 8× and 16× upscaling. The best results are in bold and the second bests are underlined.

Blurring function	*bior2.4*		*bior6.8*		*rbio2.8*		*linear spline*	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
Dong *et al.* [37]	5.68	0.9250	4.89	0.9376	4.59	0.9469	5.64	0.9294
Wang *et al.* [11]	4.57	0.9670	3.14	0.9766	3.41	0.9725	4.37	0.9692
Kim *et al.* [16]	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**

Table 4.10: Performance of Our and other methods for 4× upscaling, with arbitrary blurring kernels.

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 4.11: Running time in seconds of our method on Middlebury dataset.

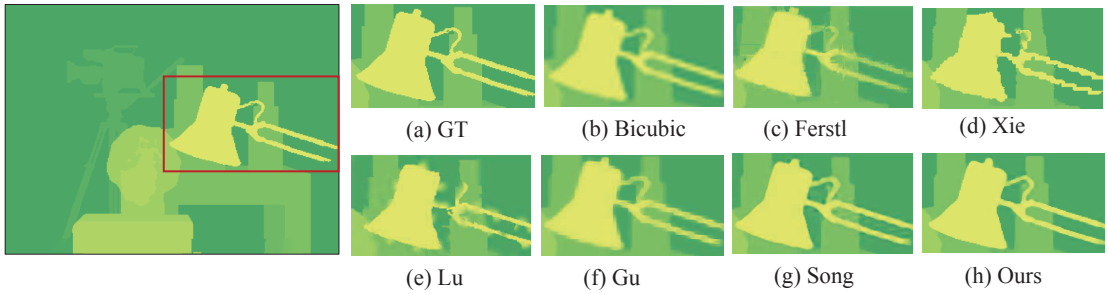


Figure 4.8: Visual comparison of *Tsukuba* in Middlebury dataset with upscaling factor = 4. (a) Ground truth. (b) Bicubic. (c) Ferstl *et al.* [5]. (d) Xie *et al.* [6]. (e) Lu *et al.* [7]. (f) Gu *et al.* [8]. (g) Song *et al.* [9]. (h) Our method.

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

Table 4.12: Quantitative results of RMSE on noisy Middlebury dataset.

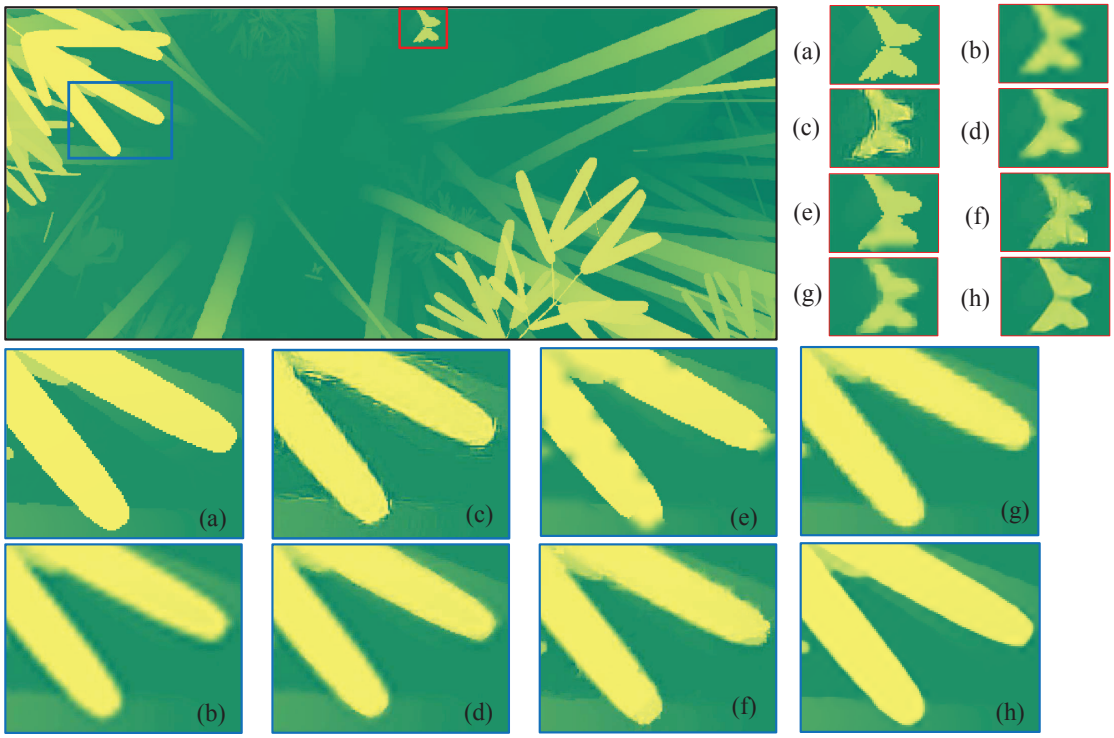


Figure 4.9: Visual comparison of *Bamboo* in Sintel dataset with upscaling factor = 4. (a) Ground truth. (b) Bicubic. (c) Xie *et al.* [6]. (d) Timofte *et al.* [10]. (e) Lu *et al.* [7]. (f) Ferstl *et al.* [5]. (g) Gu *et al.* [8]. (h) Our method.

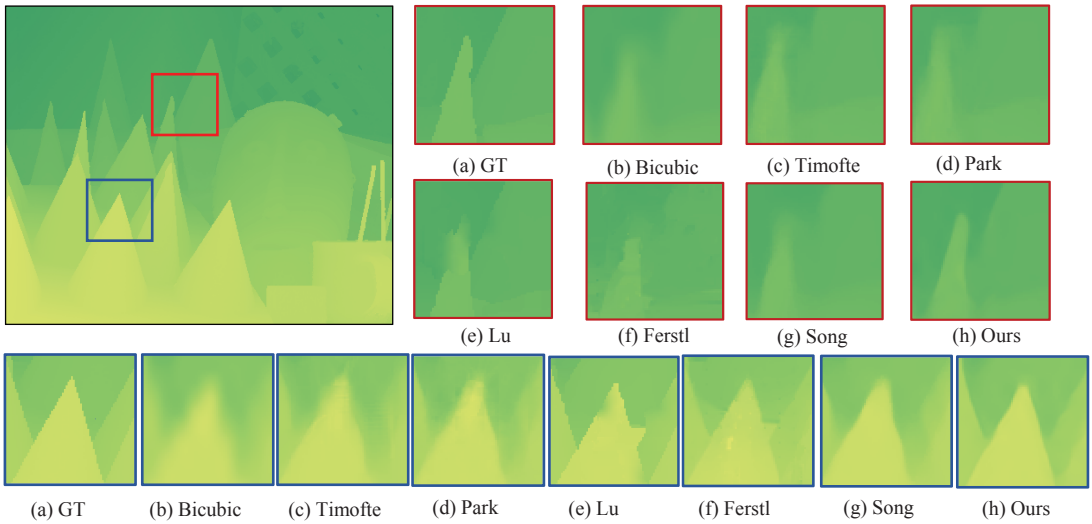


Figure 4.10: Visual comparison of *Cones* in Middlebury dataset with upscaling factor = 8. (a) Ground truth. (b) Bicubic. (c) Timofte *et al.* [10]. (d) Xie *et al.* [6]. (e) Lu *et al.* [7]. (f) Ferstl *et al.* [5]. (g) Song *et al.* [9]. (h) Our method.

4.3.6 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 [5] 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. 4.11 and 4.12 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 4.12 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. 4.13 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. 4.13, 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 the 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. 4.13, we can set $\theta = 2.5$, and we empirically set $\delta = 1.5$.

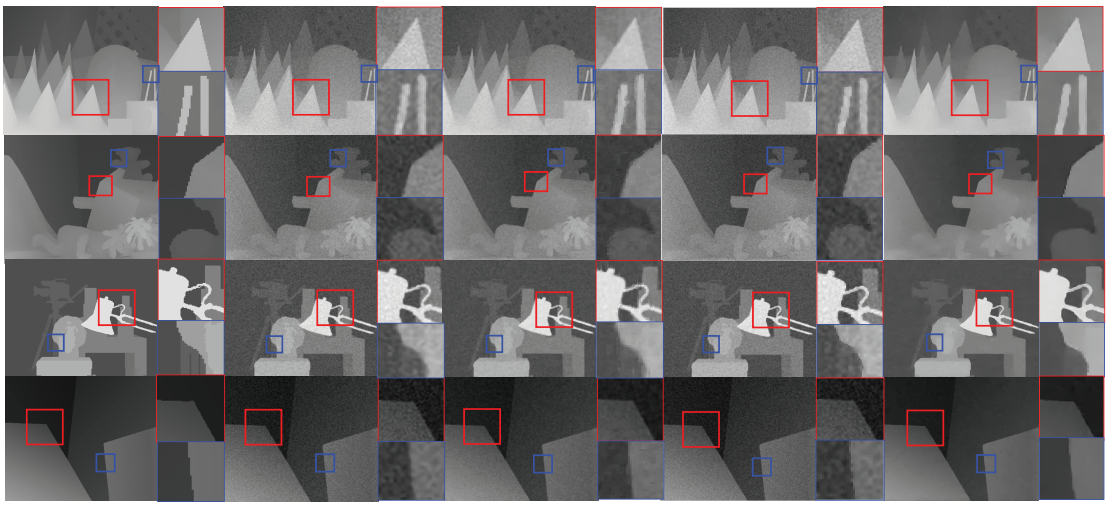


Figure 4.11: Results on noisy Middlebury dataset for 2× upscaling, with the variance of noise $\sigma^2 = 65$. (a) Ground truth, (b) Wang *et al.* [11], (c) Ferstl *et al.* [5], (d) Our method with noise-free version, (e) Our method with noisy version. The RMSE value is averaged among four images.

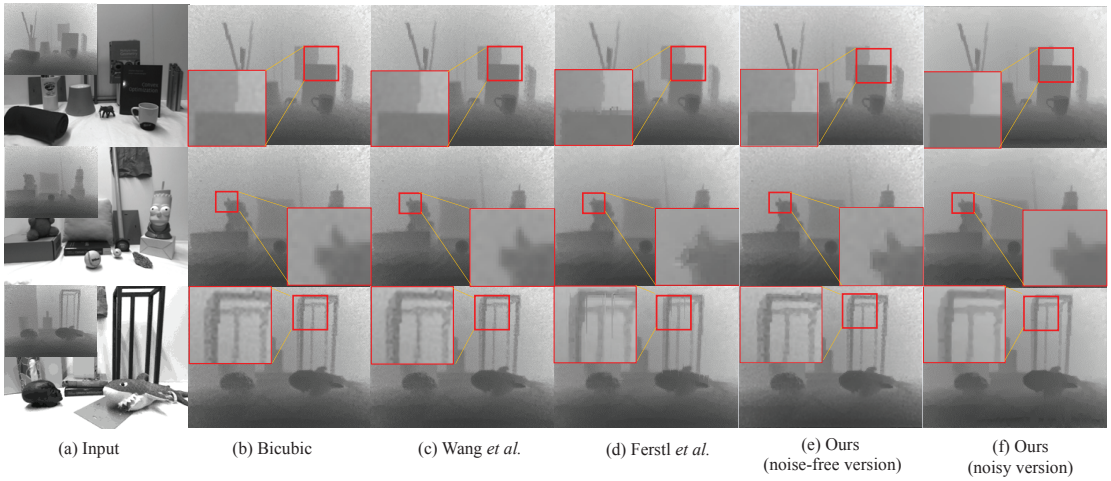


Figure 4.12: Qualitative results on the real-world ToFMark [5] dataset for 2× upscaling. (a) Ground truth, (b) bicubic interpolation, (c) Wang *et al.* [11], (d) Ferstl *et al.* [5], (e) Our method with noise-free version, (f) Our method with noisy version. The variances of noise are $\sigma^2 = 4.58$, 3.80 and 4.21 for input noisy depth images from top to down, which are estimated using method proposed by Liu *et al.* [12].

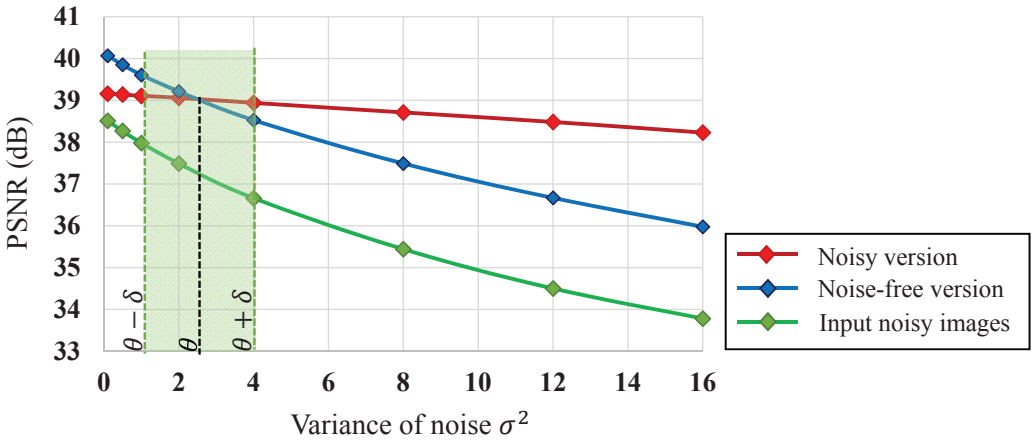


Figure 4.13: PSNR versus noise variance σ^2 when applying noise-free and noisy versions of our algorithm on the noisy images.

4.4 Summary

Depth image super-resolution is a challenging problem, since normally high upscaling factors are required (e.g., 16×), and depth images are often noisy. In order to achieve large upscaling factors and resilience to noise, we proposed 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 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 projection-based rapid upscaling (PRU) algorithm. Compared with the deep learning based methods, our approach has two distinct advantages: we only need a fraction of training data but can achieve state-of-the-art 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 and can handle unknown blurring kernels well.

Chapter 5

Coupled ISTA Network for Multi-modal Image Super-Resolution

5.1 Introduction

Multi-modal image super-resolution (MISR) aims to improve the resolution of a LR image from one modality with the guidance of an HR image from a different modality. Traditional methods mainly focus on joint image filtering [55, 56, 63, 64, 130] and dictionary learning [8, 131–135]. The key insight of joint image filtering based methods is to transfer the structure in the reference image to the LR image by learning some joint filters. The dictionary learning based methods try to learn different dictionaries for each modality with some prior constraints. Recently, some works based on deep neural networks [9, 65, 136] have achieved state-of-the-art performance in MISR.

In this and the following chapters, we draw inspirations from multi-modal dictionary learning to design new deep network architectures. This will be achieved by "unfolding" specific iterative algorithms. In Chapter 6, the focus is on convolutional

dictionaries whereas here we focus on unstructured dictionaries. In this chapter, we first propose a joint multi-modal dictionary learning (JMDL) model to solve the general MISR problem. This model jointly learns three dictionaries for the LR and HR versions of one modality and the HR version of the guided modality, respectively. In addition, two transform matrices are learned to correlate the three dictionaries in the sparse representation space. Based on the JMDL model, we design a new deep coupled neural network for solving the MISR problem, by unfolding the iterative shrinkage and thresholding algorithm (ISTA). Since the network initialization plays an important role in deep network training, we further develop a new layer-wise optimization algorithm (LOA) to initialize the network parameters. The LOA initializes all layers one by one, and in each layer, the algorithm can converge in ten iterations. Compared to other widely used initialization algorithms, the LOA is demonstrated to optimize the training loss function better and to improve reconstruction accuracy.

5.2 Proposed Method

5.2.1 Joint multi-modal dictionary learning

The MISR task aims to find the HR patch $\boldsymbol{z}$ from the LR patch $\boldsymbol{x}$ with the guidance of another HR patch $\boldsymbol{y}$, where $\boldsymbol{x}$ and $\boldsymbol{z}$ are from the same modality and $\boldsymbol{y}$ is from a different modality. Assume that $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ admit a sparse representation in dictionaries $\boldsymbol{D}_x \in \mathbb{R}^{n\times m}$, $\boldsymbol{D}_y \in \mathbb{R}^{n\times m}$ and $\boldsymbol{D}_z \in \mathbb{R}^{n\times m}$, respectively, and that we can approximately represent $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ as the linear combinations of the atoms in dictionaries $\boldsymbol{D}_x$, $\boldsymbol{D}_y$ and $\boldsymbol{D}_z$, as follows:

$$\boldsymbol{x} \simeq \boldsymbol{D}_x\boldsymbol{a}, \tag{5.1}$$

$$\boldsymbol{y} \simeq \boldsymbol{D}_y\boldsymbol{b}, \tag{5.2}$$

$$\boldsymbol{z} \simeq \boldsymbol{D}_z\boldsymbol{c}, \tag{5.3}$$

where $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are the sparse representation vectors with only a small number of non-zero entries. Since $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ are from the same scene, we assume that their sparse representations are related, i.e., $\boldsymbol{c}$ can be inferred from $\boldsymbol{a}$ and $\boldsymbol{b}$. Specifically, we make the following modelling assumption:

$$\boldsymbol{c} = S_{\boldsymbol{\lambda}_c}(\boldsymbol{W}_x\boldsymbol{a} + \boldsymbol{W}_y\boldsymbol{b}), \tag{5.4}$$

where $\boldsymbol{W}_x \in \mathbb{R}^{m\times m}$ and $\boldsymbol{W}_y \in \mathbb{R}^{m\times m}$ are the transform matrices which can convert $\boldsymbol{a}$ and $\boldsymbol{b}$ to $\boldsymbol{c}$ through a soft-thresholding operator $S_{\boldsymbol{\lambda}_c}(\cdot)$. Here, $\boldsymbol{\lambda}_c$ is the threshold used to adjust the sparsity of the sparse coefficient $\boldsymbol{c}$.

Next, we focus on learning the dictionaries $\boldsymbol{D}_x$, $\boldsymbol{D}_y$ and $\boldsymbol{D}_z$ and the transform matrices $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$. Suppose that $\boldsymbol{X} \in \mathbb{R}^{n\times t}$, $\boldsymbol{Y} \in \mathbb{R}^{n\times t}$ and $\boldsymbol{Z} \in \mathbb{R}^{n\times t}$ are the gathered LR features, HR guided features, and HR target features extracted from the training dataset, we then formulate the JMDL problem as follows:

$$\begin{aligned}
\min_{\substack{\boldsymbol{D}_x,\,\boldsymbol{D}_y,\,\boldsymbol{D}_z,\\ \{\,\boldsymbol{\Gamma}_x,\boldsymbol{\Gamma}_y,\boldsymbol{\Gamma}_z,\,\}\\ \boldsymbol{W}_x,\,\boldsymbol{W}_y}} & \frac{1}{2}\left\|\boldsymbol{X} - \boldsymbol{D}_x\boldsymbol{\Gamma}_x\right\|_F^2 + \frac{1}{2}\left\|\boldsymbol{Y} - \boldsymbol{D}_y\boldsymbol{\Gamma}_y\right\|_F^2 \\
& + \frac{1}{2}\left\|\boldsymbol{Z} - \boldsymbol{D}_z\boldsymbol{\Gamma}_z\right\|_F^2 + \nu_x\left\|\boldsymbol{\Gamma}_x\right\|_1 + \nu_y\left\|\boldsymbol{\Gamma}_y\right\|_1 \\
& + \nu_z\left\|\boldsymbol{\Gamma}_z\right\|_1 + \mu_x\left\|\boldsymbol{W}_x\right\|_F^2 + \mu_y\left\|\boldsymbol{W}_y\right\|_F^2 \\
& + \alpha\left\|\boldsymbol{\Gamma}_z - \boldsymbol{W}_x\boldsymbol{\Gamma}_x - \boldsymbol{W}_y\boldsymbol{\Gamma}_y\right\|_F^2, \\
& s.t., \left\|\boldsymbol{d}_{x,i}\right\|_2^2 \le 1, \left\|\boldsymbol{d}_{y,i}\right\|_2^2 \le 1, \left\|\boldsymbol{d}_{z,i}\right\|_2^2 \le 1, \forall i,
\end{aligned} \tag{5.5}$$

where ν_x, ν_y and ν_z are the regularization parameters for the sparse representations $\boldsymbol{\Gamma}_x$, $\boldsymbol{\Gamma}_y$ and $\boldsymbol{\Gamma}_z$, respectively, μ_x and μ_y are the regularization parameters for the transform matrices, and α is the regularization parameter for the sparse representation mapping error. Also, $\boldsymbol{d}_{x,i}$, $\boldsymbol{d}_{y,i}$ and $\boldsymbol{d}_{z,i}$ are the i-th atoms of dictionaries $\boldsymbol{D}_x$, $\boldsymbol{D}_y$ and $\boldsymbol{D}_z$, respectively. Note that here there is no need to use soft-threshold $\boldsymbol{\lambda}_c$ to enforce the sparsity, because we already have a ℓ_1 norm constraint on the $\boldsymbol{\Gamma}_z$. We use ℓ_1 instead of ℓ_0 norm to impose sparsity for the following reasons. The ℓ_1 norm makes Eqs. (5.6)-(5.12) a LASSO problem, which is convex and easier to

solve. Moreover, with the ℓ_1 norm, Eq. (5.18) can be solved by using the iterative algorithm ISTA. As we will explain in Section III-B, we can then unfold the solution of Eq. (5.18) into a deep network. The two most related models with Eq. (5.5) are the SCDL model in [137] and the $SliM^2$ model in [138]. However, the SCDL model only correlates two images, i.e., LR and HR target images, with no consideration of the guided image. The $SliM^2$ model aims to solve the image retrieval problem, which can correlate different modalities. However, the correlation is established in a one-to-one manner, i.e., each modality is only related to another modality. In contrast, our JMDL model aims to solve the MISR problem and we establish the correlations in a two-to-one manner, i.e., the joint mapping from the LR and guided HR images to a target HR image.

The problem in Eq. (5.5) is not convex with regard to variables $\boldsymbol{D}_x$, $\boldsymbol{D}_y$, $\boldsymbol{D}_z$, $\boldsymbol{\Gamma}_x$, $\boldsymbol{\Gamma}_y$, $\boldsymbol{\Gamma}_z$, $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$. However, it is convex to one variable when the others are fixed. Thus, we can solve this problem using an alternating strategy, which is composed of the following three steps: In the first step, we fix the dictionaries and transform matrices to update the sparse representations. Here, the dictionaries can be initialized by DCT frames and the transform matrices can be initialized as $0.5\boldsymbol{I}$ where $\boldsymbol{I}$ is the identity matrix. Then, the sparse representations $\boldsymbol{\Gamma}_x$, $\boldsymbol{\Gamma}_y$, and $\boldsymbol{\Gamma}_z$ can be obtained through solving the following three problems, respectively:

$$\min_{\boldsymbol{\Gamma}_x} \frac{1}{2} \left\| \boldsymbol{X} - \boldsymbol{D}_x \boldsymbol{\Gamma}_x \right\|_F^2 + \alpha \left\| \boldsymbol{\Gamma}_z - \boldsymbol{W}_x \boldsymbol{\Gamma}_x - \boldsymbol{W}_y \boldsymbol{\Gamma}_y \right\|_F^2 + \nu_x \left\| \boldsymbol{\Gamma}_x \right\|_1, \tag{5.6}$$

$$\min_{\boldsymbol{\Gamma}_y} \frac{1}{2} \left\| \boldsymbol{Y} - \boldsymbol{D}_y \boldsymbol{\Gamma}_y \right\|_F^2 + \alpha \left\| \boldsymbol{\Gamma}_z - \boldsymbol{W}_x \boldsymbol{\Gamma}_x - \boldsymbol{W}_y \boldsymbol{\Gamma}_y \right\|_F^2 + \nu_y \left\| \boldsymbol{\Gamma}_y \right\|_1, \tag{5.7}$$

$$\min_{\boldsymbol{\Gamma}_z} \frac{1}{2} \left\| \boldsymbol{Z} - \boldsymbol{D}_z \boldsymbol{\Gamma}_z \right\|_F^2 + \alpha \left\| \boldsymbol{\Gamma}_z - \boldsymbol{W}_x \boldsymbol{\Gamma}_x - \boldsymbol{W}_y \boldsymbol{\Gamma}_y \right\|_F^2 + \nu_z \left\| \boldsymbol{\Gamma}_z \right\|_1. \tag{5.8}$$

These are all LASSO problems [139]. Take Eq. (5.6) for example, we can turn it into:

$$\min_{\boldsymbol{\Gamma}_x} \frac{1}{2} \left\| \begin{bmatrix} \boldsymbol{X} \\ \sqrt{2\alpha}(\boldsymbol{\Gamma}_z - \boldsymbol{W}_y \boldsymbol{\Gamma}_y) \end{bmatrix} - \begin{bmatrix} \boldsymbol{D}_x \\ \sqrt{2\alpha} \boldsymbol{W}_x \end{bmatrix} \boldsymbol{\Gamma}_x \right\|_F^2 + \nu_x \left\| \boldsymbol{\Gamma}_x \right\|_1. \tag{5.9}$$

By denoting $\hat{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{X} \\ \sqrt{2\alpha}(\boldsymbol{\Gamma}_z - \boldsymbol{W}_y\boldsymbol{\Gamma}_y) \end{bmatrix}$ and $\hat{\boldsymbol{D}}_x = \begin{bmatrix} \boldsymbol{D}_x \\ \sqrt{2\alpha}\boldsymbol{W}_x \end{bmatrix}$, we can simplify Eq. (5.9) to the following standard LASSO problem, which can be solved using the SPAMS toolbox[1],

$$\min_{\boldsymbol{\Gamma}_x} \frac{1}{2} \left\| \hat{\boldsymbol{X}} - \hat{\boldsymbol{D}}_x\boldsymbol{\Gamma}_x \right\|_F^2 + \nu_x \left\| \boldsymbol{\Gamma}_x \right\|_1 . \tag{5.10}$$

Similarly, we can turn Eq. (5.7) and Eq. (5.8) to be:

$$\min_{\boldsymbol{\Gamma}_y} \frac{1}{2} \left\| \hat{\boldsymbol{Y}} - \hat{\boldsymbol{D}}_y\boldsymbol{\Gamma}_y \right\|_F^2 + \nu_y \left\| \boldsymbol{\Gamma}_y \right\|_1 , \tag{5.11}$$

and

$$\min_{\boldsymbol{\Gamma}_z} \frac{1}{2} \left\| \hat{\boldsymbol{Z}} - \hat{\boldsymbol{D}}_z\boldsymbol{\Gamma}_z \right\|_F^2 + \nu_z \left\| \boldsymbol{\Gamma}_z \right\|_1 , \tag{5.12}$$

where $\hat{\boldsymbol{Y}} = \begin{bmatrix} \boldsymbol{Y} \\ \sqrt{2\alpha}(\boldsymbol{\Gamma}_z - \boldsymbol{W}_x\boldsymbol{\Gamma}_x) \end{bmatrix}$, $\hat{\boldsymbol{Z}} = \begin{bmatrix} \boldsymbol{Z} \\ \sqrt{2\alpha}(\boldsymbol{W}_x\boldsymbol{\Gamma}_x + \boldsymbol{W}_y\boldsymbol{\Gamma}_y) \end{bmatrix}$, $\hat{\boldsymbol{D}}_y = \begin{bmatrix} \boldsymbol{D}_y \\ \sqrt{2\alpha}\boldsymbol{W}_y \end{bmatrix}$ and $\hat{\boldsymbol{D}}_z = \begin{bmatrix} \boldsymbol{D}_z \\ \sqrt{2\alpha}\boldsymbol{I} \end{bmatrix}$.

In the second step, we fix the sparse representations and the transform matrices to update the dictionaries $\boldsymbol{D}_x$, $\boldsymbol{D}_y$, and $\boldsymbol{D}_z$. This leads to the following minimization problems:

$$\min_{\boldsymbol{D}_x} \ \tfrac{1}{2} \left\| \boldsymbol{X} - \boldsymbol{D}_x\boldsymbol{\Gamma}_x \right\|_F^2 , \quad s.t., \quad \left\| \boldsymbol{d}_{x,i} \right\|_2^2 \leq 1, \forall i, \tag{5.13}$$

$$\min_{\boldsymbol{D}_y} \ \tfrac{1}{2} \left\| \boldsymbol{Y} - \boldsymbol{D}_y\boldsymbol{\Gamma}_y \right\|_F^2 , \quad s.t., \quad \left\| \boldsymbol{d}_{y,i} \right\|_2^2 \leq 1, \forall i, \tag{5.14}$$

$$\min_{\boldsymbol{D}_z} \ \tfrac{1}{2} \left\| \boldsymbol{Z} - \boldsymbol{D}_z\boldsymbol{\Gamma}_z \right\|_F^2 , \quad s.t., \quad \left\| \boldsymbol{d}_{z,i} \right\|_2^2 \leq 1, \forall i. \tag{5.15}$$

These problems are standard dictionary learning problems. Following [140], we update each atom in the dictionary one by one while keeping the others fixed.

[1]The SPAMS toolbox can be downloaded from the website http://spams-devel.gforge.inria.fr/

In the third step, with the updated dictionaries and sparse representations, we can learn the transform matrices $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$, through solving the following optimization problem:

$$\min_{\{\boldsymbol{W}_x,\boldsymbol{W}_y\}} \alpha \left\|\boldsymbol{\Gamma}_z - \boldsymbol{W}_x\boldsymbol{\Gamma}_x - \boldsymbol{W}_y\boldsymbol{\Gamma}_y\right\|_F^2 + \mu_x \left\|\boldsymbol{W}_x\right\|_F^2 + \mu_y \left\|\boldsymbol{W}_y\right\|_F^2 . \tag{5.16}$$

This is a least square regression problem, which can be solved by Ridge Regression [141]. The closed-form solution of Eq. (5.16) can be obtained as follows,

$$\left[\sqrt{\tfrac{\mu_x}{\alpha}}\boldsymbol{W}_x \sqrt{\tfrac{\mu_y}{\alpha}}\boldsymbol{W}_y\right] = \boldsymbol{\Gamma}_z \left[\sqrt{\tfrac{\alpha}{\mu_x}}\boldsymbol{\Gamma}_x \sqrt{\tfrac{\alpha}{\mu_y}}\boldsymbol{\Gamma}_y\right]^T \left(\begin{bmatrix} \sqrt{\frac{\alpha}{\mu_x}}\boldsymbol{\Gamma}_x \\ \sqrt{\frac{\alpha}{\mu_y}}\boldsymbol{\Gamma}_y \end{bmatrix} \begin{bmatrix} \sqrt{\frac{\alpha}{\mu_x}}\boldsymbol{\Gamma}_x \\ \sqrt{\frac{\alpha}{\mu_y}}\boldsymbol{\Gamma}_y \end{bmatrix}^T + \boldsymbol{I} \right)^{-1}, \tag{5.17}$$

where $\boldsymbol{I}$ is the identity matrix.

The above three steps are iterated until convergence. In the reconstruction phase, given a LR patch $\boldsymbol{x}$ and a HR guidance patch $\boldsymbol{y}$, we first need to find the sparse coefficients with the learned dictionaries $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$:

$$\min_{\{\boldsymbol{a},\boldsymbol{b}\}} \frac{1}{2}\left\|\boldsymbol{x} - \boldsymbol{D}_x\boldsymbol{a}\right\|_2^2 + \frac{1}{2}\left\|\boldsymbol{y} - \boldsymbol{D}_y\boldsymbol{b}\right\|_2^2 + \nu_x \left\|\boldsymbol{a}\right\|_1 + \nu_y \left\|\boldsymbol{b}\right\|_1 . \tag{5.18}$$

The optimization over $\boldsymbol{a}$ and $\boldsymbol{b}$ in Eq. (5.18) can be decoupled, with each one being a LASSO problem. After obtaining $\boldsymbol{a}$ and $\boldsymbol{b}$, we can have sparse representation $\boldsymbol{c}$ using Eq. (5.4). Finally, the HR depth patch $\boldsymbol{z}$ can be calculated by multiplying $\boldsymbol{c}$ with the dictionary $\boldsymbol{D}_z$.

5.2.2 Deep coupled ISTA network

The aforementioned algorithm has two drawbacks. Firstly, the calculation of sparse representations $\boldsymbol{a}$ and $\boldsymbol{b}$ in Eq. (5.18) relies on the iterative algorithm, e.g., ISTA or FISTA, which can be time-consuming. Secondly, the training and synthesis phases are not fully coupled, since the ground-truth HR patch is not accessible in the synthesis phase, and this may affect the reconstruction accuracy. To overcome these

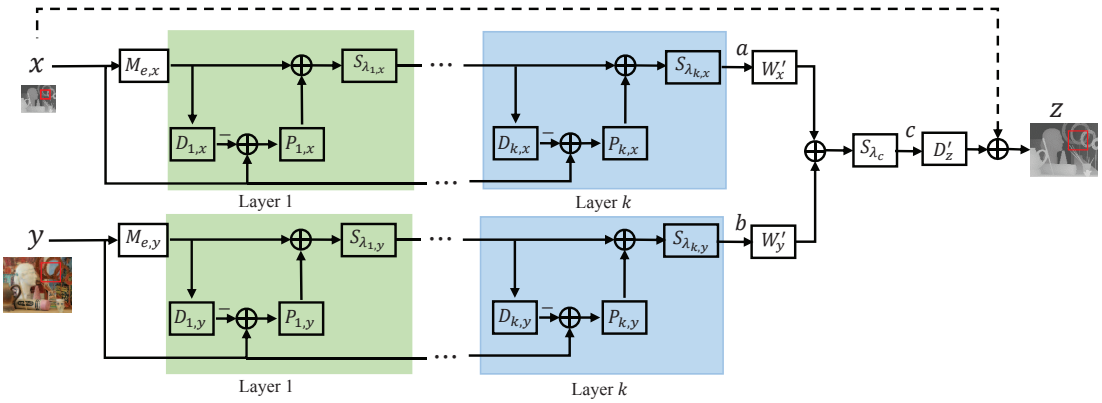


Figure 5.1: The architecture of proposed deep coupled ISTA network. The dashed line indicates that the mean value of patch $\boldsymbol{x}$ is added to the output patch $\boldsymbol{z}$.

drawbacks, we further propose a deep coupled ISTA network which is obtained by unfolding the ISTA algorithm. The architecture of the unfolded network is shown in Fig. 5.1. Specifically, the network is composed of two branches: the upper branch aims to infer the sparse representation $\boldsymbol{a}$ for the LR input patch $\boldsymbol{x}$, while the lower branch aims to infer the sparse representation $\boldsymbol{b}$ for the guided HR patch $\boldsymbol{y}$. Take the upper branch for example, the ISTA algorithm [85] works in iterations to obtain $\boldsymbol{a}$ using the following formulation:

$$\boldsymbol{a}_k = S_{\lambda_k}(\boldsymbol{a}_{k-1} + \boldsymbol{D}_x^T(\boldsymbol{x} - \boldsymbol{D}_x\boldsymbol{a}_{k-1})), \tag{5.19}$$

where $\boldsymbol{a}_k$ is the value of $\boldsymbol{a}$ at the k-th iteration. Through unfolding Eq. (5.19), we can have the basic layer as shown in Fig. 5.2. The procedure to use Eq. (5.19) to construct the diagram in Fig. 5.2 can be performed in three steps. Specifically, in the first step, given $\boldsymbol{a}_{k-1}$ which is the predicted sparse code of the previous $(k-1)$-th layer, we multiply it with $\boldsymbol{D}_x$, and then the result of that multiplication is subtracted from $\boldsymbol{x}$. This yields $(\boldsymbol{x}-\boldsymbol{D}_x\boldsymbol{a}_{k-1})$. Then in the second step, $(\boldsymbol{x}-\boldsymbol{D}_x\boldsymbol{a}_{k-1})$ is multiplied by $\boldsymbol{D}_x^T$, and the result of the multiplication is added to $\boldsymbol{a}_{k-1}$. This leads to $(\boldsymbol{a}_{k-1} + \boldsymbol{D}_x^T(\boldsymbol{x} - \boldsymbol{D}_x\boldsymbol{a}_{k-1}))$. Finally, we add the soft-thresholding operator S_{λ_k} to obtain $\boldsymbol{a}_k$. Here, we can see that Fig. 5.2 is unfolded following the exact format of ISTA iterations, e.g., the $\boldsymbol{D}_x$ and $\boldsymbol{D}_x^T$ are separated and treated as two independent variables, while in [15], $\boldsymbol{D}_x$ and $\boldsymbol{D}_x^T$ are treated as one variable.

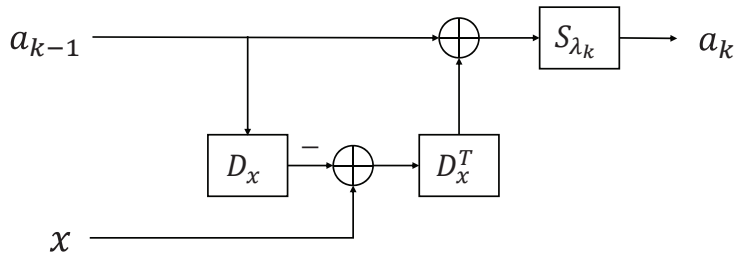


Figure 5.2: A single layer obtained by unfolding Eq. (5.19).

By stacking all layers together, we can have the upper branch in Fig. 5.1. In order to make the network more flexible, we make three relaxations with respect to the original ISTA algorithm. Firstly, the dictionary $\boldsymbol{D}_x$ is not required to be the same across different layers, i.e., we have a set of synthesis dictionaries $\{\boldsymbol{D}_{1,x}, \cdots, \boldsymbol{D}_{k,x}\}$ as shown in Fig. 5.1. Secondly, the relationship between $\boldsymbol{D}_x^T$ and $\boldsymbol{D}_x$ is relaxed and we replace $\boldsymbol{D}_x^T$ with a set of analysis dictionaries $\{\boldsymbol{P}_{1,x}, \cdots, \boldsymbol{P}_{k,x}\}$. Thirdly, the soft threshold is allowed to change across layers, and we use a vector threshold instead of a constant scalar. As shown in Fig. 5.1, we have different vector thresholds $\{\boldsymbol{\lambda}_{1,x}, \cdots, \boldsymbol{\lambda}_{k,x}\}$ for each layer.

The lower branch can be obtained using the similar method as the upper branch. Then, the outputs of these two branches are combined by the transform matrices to obtain $\boldsymbol{c}$. Finally, the sparse representation $\boldsymbol{c}$ is multiplied by the reconstruction matrix to reconstruct the HR patch $\boldsymbol{z}$.

5.2.3 Layer-wise optimization algorithm

Given the network architecture of Fig. 5.1, we now propose a layer-wise optimization algorithm (LOA) to initialize the network parameters, before running the traditional back-propagation strategy. Let us focus on the upper branch first. We aim to minimize the mean squared error (MSE) between the predicted sparse codes and the target $\boldsymbol{\Gamma}_x$ obtained by solving (5.5). Since the parameters of the upper and lower branches can be initialized using the same algorithm, we just ignore the subscripts x and y to make the notations simpler. Specifically, in the k-th layer, we have the

following optimization target:

$$\min_{\boldsymbol{P}_k,\boldsymbol{D}_k,\boldsymbol{\lambda}_k} \|\boldsymbol{\Gamma} - S_{\boldsymbol{\lambda}_k}(\boldsymbol{B}_{k-1} + \boldsymbol{P}_k(\boldsymbol{X} - \boldsymbol{D}_k\boldsymbol{B}_{k-1}))\|_F^2 , \tag{5.20}$$

where $\boldsymbol{\Gamma}$ is the ground-truth sparse code and $\boldsymbol{B}_{k-1}$ is the predicted sparse code of the previous $(k-1)$-th layer. Here, $\boldsymbol{P}_k \in \mathbb{R}^{m\times n}(m > n)$ can be regarded as an analysis dictionary, $\boldsymbol{D}_k \in \mathbb{R}^{n\times m}$ is a synthesis dictionary, and $\boldsymbol{\lambda}_k \in \mathbb{R}^m$ is a threshold vector of the k-th layer. For initialization, $\boldsymbol{B}_0$ is obtained by minimizing the reconstruction error of training samples $\boldsymbol{X}$ using the original dictionary $\boldsymbol{D}$ (equal to $\boldsymbol{D}_x$ for the upper branch and $\boldsymbol{D}_y$ for the lower branch):

$$\boldsymbol{B}_0 = \underset{\boldsymbol{B}_0}{\operatorname{argmin}} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{B}_0\|_F^2 + \mu \|\boldsymbol{B}_0\|_F^2 , \tag{5.21}$$

which yields the solution $\boldsymbol{B}_0 = (\boldsymbol{D}^T\boldsymbol{D} + \mu\boldsymbol{I})^{-1}\boldsymbol{D}^T\boldsymbol{X}$, where $\boldsymbol{I}$ is the identity matrix. To make the network structure simple, we define a mapping matrix $\boldsymbol{M}_e = (\boldsymbol{D}^T\boldsymbol{D} + \mu\boldsymbol{I})^{-1}\boldsymbol{D}^T$ with $\boldsymbol{M}_e \in \mathbb{R}^{m\times n}$. This matrix directly maps $\boldsymbol{X}$ to $\boldsymbol{B}_0$ (see Fig. 5.1). Next, we focus on solving the optimization problem in (5.20). Since we have three variables to optimize, we update them iteratively.

Step 1. We fix $\boldsymbol{D}_k$ and $\boldsymbol{\lambda}_k$ to update the analysis dictionary $\boldsymbol{P}_k$. In this case, since $(\boldsymbol{X} - \boldsymbol{D}_k\boldsymbol{B}_{k-1})$ does not change, we denote it by $\boldsymbol{A} \in \mathbb{R}^{n\times t}$, and then $\boldsymbol{P}_k$ can be obtained through solving the following optimization problem:

$$\boldsymbol{P}_k = \underset{\boldsymbol{P}_k}{\operatorname{argmin}} \|\boldsymbol{\Gamma} - S_{\boldsymbol{\lambda}_k}(\boldsymbol{B}_{k-1} + \boldsymbol{P}_k\boldsymbol{A})\|_F^2 . \tag{5.22}$$

We update the atoms of $\boldsymbol{P}_k$ row by row. When updating the j-th atom, the other atoms remain fixed. Specifically, the j-th atom of $\boldsymbol{P}_k$ is updated as follows:

$$\boldsymbol{p}_j^T = \underset{\boldsymbol{p}_j^T}{\operatorname{argmin}} \left\| \boldsymbol{g}_j^T - S_{\lambda_k^j}(\boldsymbol{b}_j^T + \boldsymbol{p}_j^T\boldsymbol{A}) \right\|_2^2 , \tag{5.23}$$

where $\boldsymbol{g}_j^T$, $\boldsymbol{b}_j^T$, $\boldsymbol{p}_j^T$ are the j-th $(0 < j \leq m)$ row of $\boldsymbol{\Gamma}$, $\boldsymbol{B}_{k-1}$, and $\boldsymbol{P}_k$, respectively. λ_k^j is the j-th element of the vector $\boldsymbol{\lambda}_k$. The difficulty here is that the soft-thresholding

Algorithm 1 The layer-wise optimization algorithm (LOA)

Require: Training samples $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$; dictionaries $\boldsymbol{D}_x$, $\boldsymbol{D}_y$; target sparse codes $\boldsymbol{\Gamma}_x$, $\boldsymbol{\Gamma}_y$; transform matrices $\boldsymbol{W}_x$, $\boldsymbol{W}_y$;

Ensure: Initialized dictionaries and thresholds of all K layers of upper branch$\{\boldsymbol{P}_{k,x}, \boldsymbol{D}_{k,x}, \boldsymbol{\lambda}_{k,x}\}_{k=1}^{K}$, and lower branch $\{\boldsymbol{P}_{k,y}, \boldsymbol{D}_{k,y}, \boldsymbol{\lambda}_{k,y}\}_{k=1}^{K}$; initialized transform matrices $\boldsymbol{W}_x'$, $\boldsymbol{W}_y'$ and reconstruction matrix $\boldsymbol{D}_z'$.

```
for each branch do
    Initialize B_0 by (5.21);
    for k = 1 → K do
        for iter = 1 → max_iter do
            Fix D_k and λ_k, and update P_k through (5.22) to (5.26);
            Fix P_k and λ_k, and update D_k through (5.27) and (5.28);
            Fix P_k and D_k, and update λ_k by (5.29);
        end for
        Update B_k by (5.30).
    end for
end for
Calculate the transform matrices W_x' and W_y' by (5.31) and (5.32);
Calculate the reconstruction matrix D_z' by (5.33);
return {P_{k,x}, D_{k,x}, λ_{k,x}}_{k=1}^K, {P_{k,y}, D_{k,y}, λ_{k,y}}_{k=1}^K, W_x', W_y' and D_z'.
```

operator is non-linear. Inspired by [142], we divide the non-linear operation into two linear operations. Specifically, the soft-thresholding operator splits the signals $(\boldsymbol{b}_j^T + \boldsymbol{p}_j^T \boldsymbol{A})$ into two sets: one set with all zeros after soft-thresholding and the other set with non-zero values. Suppose $\boldsymbol{J}$ denotes the indices of the non-zero samples and $\hat{\boldsymbol{J}}$ denotes the indices of samples that are set to zeros, we can split $\boldsymbol{A}$ into $\boldsymbol{A}^J \in \mathbb{R}^{n \times J}$ and $\boldsymbol{A}^{\hat{J}} \in \mathbb{R}^{n \times \hat{J}}$. Likewise, we can split $\boldsymbol{g}_j^T$ into $\boldsymbol{g}_j^J$ and $\boldsymbol{g}_j^{\hat{J}}$, $\boldsymbol{b}_j^T$ into $\boldsymbol{b}_j^J$ and $\boldsymbol{b}_j^{\hat{J}}$. Then, (5.23) can be written as follows,

$$\boldsymbol{p}_j^T = \underset{\boldsymbol{p}_j^T}{\operatorname{argmin}} \left\| \boldsymbol{g}_j^J - (\boldsymbol{b}_j^J + \boldsymbol{p}_j^T \boldsymbol{A}^J \pm \lambda_k^j) \right\|_2^2 + \left\| \boldsymbol{g}_j^{\hat{J}} \right\|_2^2. \tag{5.24}$$

Here, we assume that $\left\| \boldsymbol{g}_j^{\hat{J}} \right\|_2^2$ is constant when the threshold is fixed, and we can further simplify (5.24) to be

$$\boldsymbol{p}_j^T = \underset{\boldsymbol{p}_j^T}{\operatorname{argmin}} \left\| \boldsymbol{g}_j^J - (\boldsymbol{b}_j^J + \boldsymbol{p}_j^T \boldsymbol{A}^J \pm \lambda_k^j) \right\|_2^2. \tag{5.25}$$

This can be solved through least square fitting, which yields the following $\boldsymbol{p}_j^T$:

$$\boldsymbol{p}_j^T = (\boldsymbol{g}_j^J - \boldsymbol{b}_j^J \pm \lambda_k^j)(\boldsymbol{A}^J)^T(\boldsymbol{A}^J(\boldsymbol{A}^J)^T + \mu\boldsymbol{I})^{-1}, \tag{5.26}$$

where μ is a regularization parameter applied when $\boldsymbol{A}^J$ is not full rank.

Step 2. We fix $\boldsymbol{P}_k$ and $\boldsymbol{\lambda}_k$ to update $\boldsymbol{D}_k$, through solving the following optimization problem:

$$\min_{\boldsymbol{D}_k} \|\boldsymbol{\Gamma} - S_{\boldsymbol{\lambda}_k}(\boldsymbol{B}_{k-1} + \boldsymbol{P}_k\boldsymbol{X} - \boldsymbol{P}_k\boldsymbol{D}_k\boldsymbol{B}_{k-1}))\|_F^2 . \tag{5.27}$$

To solve this problem, we first denote $\boldsymbol{V}_k = \boldsymbol{P}_k\boldsymbol{D}_k$ and update $\boldsymbol{V}_k$ using the same procedure as in Step 1. Then, since we already have the updated version of $\boldsymbol{P}_k$, we can extract $\boldsymbol{D}_k$ from $\boldsymbol{V}_k$ by minimizing $\|\boldsymbol{V}_k - \boldsymbol{P}_k\boldsymbol{D}_k\|_F^2$, and this yields

$$\boldsymbol{D}_k = (\boldsymbol{P}_k^T\boldsymbol{P}_k + \mu\boldsymbol{I})^{-1}\boldsymbol{P}_k^T\boldsymbol{V}_k. \tag{5.28}$$

Step 3. We fix $\boldsymbol{P}_k$ and $\boldsymbol{D}_k$ to update the thresholds $\boldsymbol{\lambda}_k$. Here, $\boldsymbol{\lambda}_k$ is a vector, and we update each of its elements independently. The updating of the j-th element is through the following optimization:

$$\lambda_k^j = \underset{\lambda_k^j}{\operatorname{argmin}} \left\| \boldsymbol{g}_j^T - S_{\lambda_k^j}(\boldsymbol{b}_j^T + \boldsymbol{p}_j^T\boldsymbol{A}) \right\|_2^2 . \tag{5.29}$$

Suppose that $\boldsymbol{q}^T = \boldsymbol{b}_j^T + \boldsymbol{p}_j^T\boldsymbol{A}$, and the elements in $\boldsymbol{q}^T$ are sorted such that $|q_1| \leq |q_2| \leq \ldots \leq |q_t|$. The candidates for λ_k^j can be selected from the list $\{0, (|q_1| + |q_2|)/2, (|q_2| + |q_3|)/2, \ldots, |q_t|\}$. The value which minimizes the loss in (5.29) is the new updated value of λ_k^j.

These three steps are iterated until reaching the maximum target iteration time. After we obtain $\boldsymbol{P}_k$, $\boldsymbol{D}_k$, and $\boldsymbol{\lambda}_k$, we can update the new sparse codes $\boldsymbol{B}_k$ through:

$$\boldsymbol{B}_k = S_{\boldsymbol{\lambda}_k}(\boldsymbol{B}_{k-1} + \boldsymbol{P}_k(\boldsymbol{X} - \boldsymbol{D}_k\boldsymbol{B}_{k-1})). \tag{5.30}$$

Then, optimization of the next layer can be launched. Finally, after K layer opti-

mizations, we obtain the final predicted sparse codes $\boldsymbol{B}_{K,x}$ for the upper branch and $\boldsymbol{B}_{K,y}$ for the lower branch, and the transform matrices are initialized by:

$$\boldsymbol{W}_x' = \boldsymbol{W}_x \boldsymbol{\Gamma}_x \boldsymbol{B}_{K,x}^T (\boldsymbol{B}_{K,x} \boldsymbol{B}_{K,x}^T + \mu \boldsymbol{I})^{-1}, \tag{5.31}$$

$$\boldsymbol{W}_y' = \boldsymbol{W}_y \boldsymbol{\Gamma}_y \boldsymbol{B}_{K,y}^T (\boldsymbol{B}_{K,y} \boldsymbol{B}_{K,y}^T + \mu \boldsymbol{I})^{-1}, \tag{5.32}$$

where $\boldsymbol{W}_x$, $\boldsymbol{W}_y$, $\boldsymbol{\Gamma}_x$, and $\boldsymbol{\Gamma}_y$ are obtained by solving (5.5). By initializing $\boldsymbol{\lambda}_c$ as zeros, we can denote $\boldsymbol{B}_{K,z} = \boldsymbol{W}_x' \boldsymbol{B}_{K,x} + \boldsymbol{W}_y' \boldsymbol{B}_{K,y}$ according to (5.4). Then, the reconstruction matrix $\boldsymbol{D}_z'$ can be obtained by:

$$\boldsymbol{D}_z' = \boldsymbol{Z} \boldsymbol{B}_{K,z}^T (\boldsymbol{B}_{K,z} \boldsymbol{B}_{K,z}^T + \mu \boldsymbol{I})^{-1}. \tag{5.33}$$

Algorithm 1 presents the procedure for the LOA algorithm. After LOA, we refine the optimization of the network in Fig. 5.1 using the standard back-propagation algorithm. Note that we add a soft-threshold operation $S_{\boldsymbol{\lambda}_c}$ after combining $\boldsymbol{a}$ and $\boldsymbol{b}$ (see Fig. 5.1), in order to ensure that $\boldsymbol{\Gamma}_z$ is still sparse after back-propagation.

5.3 Numerical results

5.3.1 Experimental setup

Datasets. We test the performance of our method in three different multi-modal scenerios: RGB/multi-spectral (MS), RGB/near-infrared (NIR), and RGB/depth. For the training datasets, we use the RGB/MS dataset from the Columbia multi-spectral database[2], RGB/NIR dataset from the EPFL RGB-NIR Scene database[3], and RGB/depth dataset from [136]. For the testing datasets, we randomly select 7 image pairs from the RGB/MS and RGB/NIR for testing, which are removed from the training images. For the RGB/depth testing images, we use the standard

[2]http://www.cs.columbia.edu/CAVE/databases/multispectral/
[3]http://ivrl.epfl.ch/supplementary_material/cvpr11/

Middlebury dataset[4] and MPI Sintel dataset[5]. In all the three cases, the RGB images are transformed into YCbCr domain and only the intensity channel is used as guidance to super resolve the other modality.

For the comparison metrics, we use peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) to measure the quality of the reconstructed images for RGB/MS and RGB/NIR cases. For RGB/depth case, in line with the other methods [8, 14, 143], we use root mean squared error (RMSE) and SSIM. Note that for PSNR and SSIM, larger values indicate better quality, while for RMSE, smaller value means better quality.

Training parameters. To obtain the LR training images, we first downsample the HR target images by a factor, e.g., 4×, and then use bicubic interpolation to upscale them by the same factor. The training patch size is 8×8, 12×12 and 16×16 for 4×, 8× and 16× upscaling factors, respectively. We use the mean-removed LR patch as the LR feature, the mean-removed HR guidance patch as the HR guidance feature, and the HR target feature is obtained from HR target patch by subtracting the mean value of the LR patch. In the JMDL algorithm, the dictionary size m is set to 512, the regularization parameters ν_x, ν_y, ν_z, μ_x, μ_y and α are all set to 0.01, and the maximum number of iteration time is 50. In the network training process, the number of layers K is set to be 4. We use Adam optimizer to train the network with a basic learning rate of 0.0001 and it is decayed by 0.9 every 50 epochs. The mini-batch size is set to 64 and the total epoch number is 300.

5.3.2 Performance analysis of our method

Effectiveness of JMDL algorithm: To demonstrate the effectiveness of our JMDL algorithm, we show in Fig. 5.3 the convergence performance of the JMDL algorithm during the training phase. It can be seen that the training loss decreases rapidly with the number of iterations, and nearly converges within 40 iterations. The

[4]http://vision.middlebury.edu/stereo/data/scenes2005/
[5]http://sintel.is.tue.mpg.de/

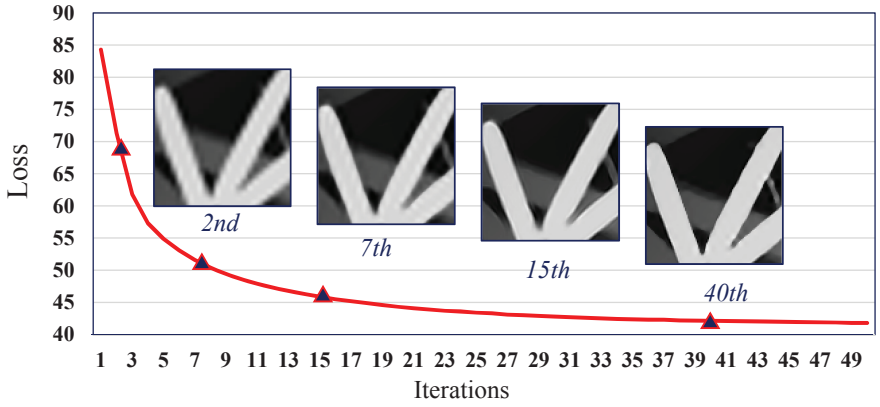


Figure 5.3: The convergence behaviour of our JMDL algorithm. The images above the curve are examples of the reconstructed results at different iterations.

reconstructed images are also visualized with different iterations, and we can observe that the reconstructed image becomes clearer and sharper after more iterations.

Effectiveness of LOA: In this paper, we have proposed a LOA algorithm for the network initialization, which solves a multi-layer dictionary learning problem to obtain a series of analysis-synthesis dictionaries and nonlinear thresholds. Here, in order to validate the effectiveness of the LOA algorithm, we first analyse it and then compare it with other widely used network initialization algorithms.

In the LOA algorithm, we calculate several layers of analysis-synthesis dictionaries and nonlinear thresholds to make the predicted sparse codes approach the ground-truth sparse codes. For each layer, we use 10 iterations to update the dictionaries and thresholds. After each iteration, we can get a predicted sparse code which is expected to approach the ground-truth. Fig. 5.4 shows the change of mean squared error (MSE) between the predicted and the ground-truth sparse codes for the first two layers. From this figure, we can see that in each layer, the MSE value keeps decreasing with the number of iterations, and it has a fast convergence. In addition, from the first to the second layer, we can observe a big drop in MSE. This verifies the effectiveness of the first layer initialization, which makes the predicted sparse codes much closer to the ground-truth.

To further verify the effectiveness of our LOA algorithm, we compare it with other

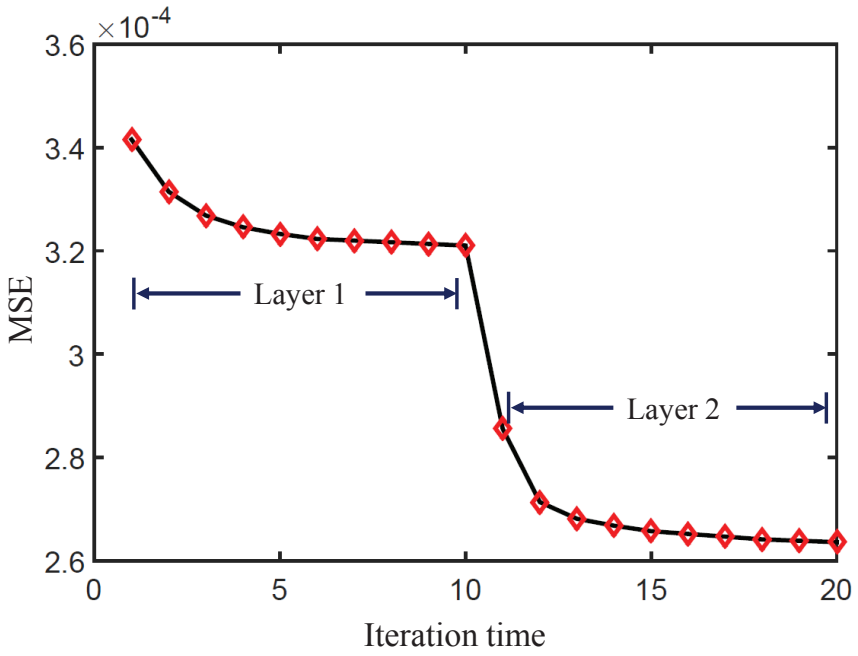
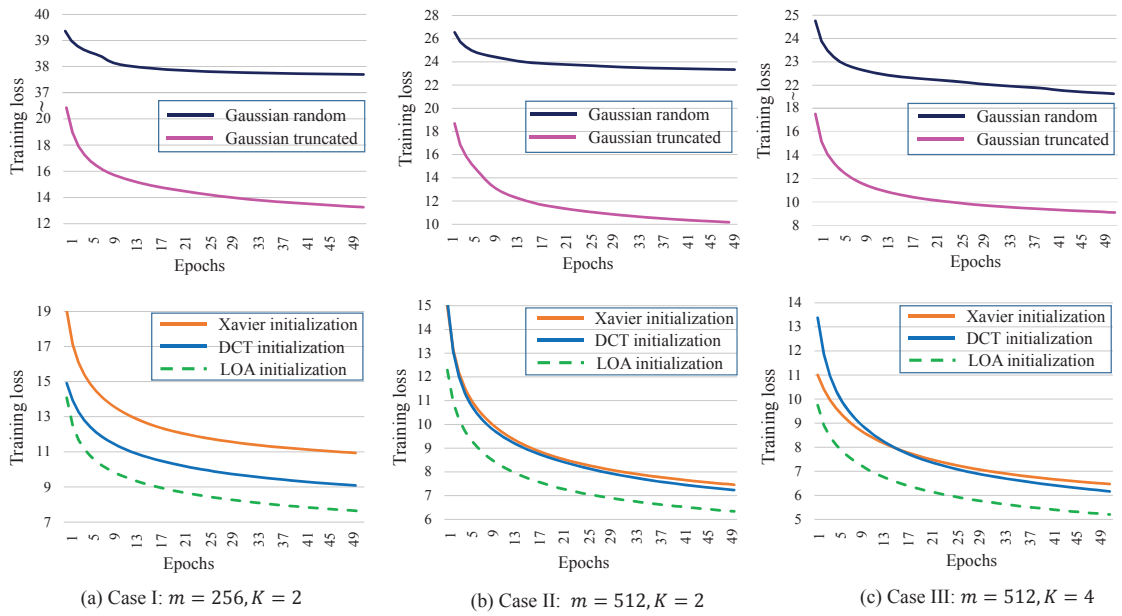


Figure 5.4: The change of MSE with the iterations in the first and second layers.

four network initialization algorithms, including three random initialization algorithms: the Gaussian_normal, Gaussian_truncated, and Xavier [144], and one Discrete Cosine Transform (DCT) initialization algorithm. The Gaussian_normal initialization algorithm initializes the network parameters with a normal distribution, and the Gaussian_truncated algorithm initializes the network with a truncated normal distribution. The Xavier [144] initializer is designed to keep the scale of the gradients roughly the same in all layers. The DCT initializer initializes all the dictionaries using DCT frames. The comparison is performed in three cases with different network capacities. Specifically, in Case I, the number of layers K is set to 2 and in each layer the dictionary size m is set to 256. In Case II, the number of layers is kept to $K = 2$ but the network capacity is increased by using a larger dictionary size $m = 512$. In Case III, the network capacity is further increased with the number of layers $K = 4$ and dictionary size $m = 512$.

Table 5.1 compares the reconstruction performance using different initialization methods. We can see that with the increase of the network capacity, the performance of all methods improves, and our LOA method consistently outperforms the others in all the three cases. For the other four initialization methods, the DCT per-



(a) Case I: $m = 256, K = 2$ (b) Case II: $m = 512, K = 2$ (c) Case III: $m = 512, K = 4$

Figure 5.5: The training loss across 50 epochs with different initialization methods for different settings of dictionary size m and number of layers K.

forms better than the Xavier, Gaussian_truncated, followed by the Gaussian_normal. Comparing our LOA with the second best, i.e., DCT, we can see that the PSNR value using DCT in Case II is similar to ours in Case I, which means we save half the network capacity to achieve nearly the same reconstruction accuracy. The possible reason is as follows: since a deep network leads to a highly non-convex problem, the conventional initialization methods may find it difficult to get an accurate solution with a small number of network parameters. However, our LOA initialization acts as a good initial solution to this problem, and thus with this good start, we can rapidly get to the accurate solution with simple back-propagation strategy.

Fig. 5.5 shows the curves of the training loss with epochs for our LOA and the other four initialization methods, for the above three cases, respectively. As we know, the essence of network initialization is to make the objective function easy to be optimized. In this respect, our LOA method performs better than the other four initialization methods. As shown in Fig. 5.5, the training loss of our LOA decreases faster, and we reach a lower training loss, regardless of the training epochs. This can also explain why our method outperforms other initialization methods in the

Methods	Case I		Case II		Case III	
	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR
Gaussian_normal	5.40	35.66	4.53	37.29	4.49	37.65
Gaussian_truncated	3.69	39.31	3.57	39.67	3.52	39.86
Xavier	3.57	39.65	3.36	40.22	3.18	40.99
DCT	3.47	40.01	3.34	40.41	3.17	41.04
LOA	**3.35**	**40.32**	**3.18**	**40.80**	**2.93**	**41.78**

Table 5.1: Comparisons of RMSE and PSNR among different initialization methods on RGB/depth dataset.

PSNR	Ambush	Bamboo	Cave	Market	Art	Books	Moebius	Average
K=1	38.29	29.35	37.40	33.71	42.72	50.21	49.84	40.22
K=2	38.82	29.78	37.98	34.39	43.15	50.63	50.86	40.80
K=3	39.20	30.29	38.42	34.76	43.55	52.15	51.21	41.37
K=4	**39.47**	**30.55**	**38.75**	**35.22**	**43.97**	**52.67**	**51.82**	**41.78**
K=5	39.56	30.58	38.91	35.35	44.09	52.97	51.93	41.91

Table 5.2: Influence of number of layers. K=4 is the default number of layers.

reconstruction accuracy as shown in Table 5.1.

Influence of number of layers: Theoretically, our network can contain any number of layers. However, when the number of layers increases, the training and testing complexity also increases. To investigate the influence of the number of layers K on the super-resolution performance, we gradually increase it from 1 to 5, and calculate the RMSE values of the reconstructed images. Table 5.2 presents the RMSE results for 4× upscaling with the RGB/depth dataset. As expected, we can see that the average RMSE value gradually decreases when the number of layers increases. However, with the increase of the number of layers, the decrease of RMSE value becomes less and less significant. This is reasonable because at this time the capacity of the network is large enough to accomplish this task. For this reason, in the later numerical comparisons, we use the number of layers $K = 4$.

5.3.3 Quantitative comparison against other methods

Benchmarks: We compare our method with other state-of-the-art methods in multi-modal image super-resolution. For the RGB/MS and RGB/NIR datasets, we

compare our method with JBF [55], GF [56], SDF [64], JFSM [63], VDSR [16], EDSR [41], SRFBN [145], DJF [65] and DGF [17]. For the RGB/depth dataset, we compare our approach against methods including Xie *et al.* [6], Park *et al.* [13], Ferstl *et al.* [14], Lu *et al.* [7], Gu *et al.* [8], SCN [15], VDSR [16], EDSR [41], SRFBN [145], RCGD [146] and DepthSR [69]. The results of these approaches are obtained by re-running the software codes they provided online. Three upscaling factors are considered for comparison: 4×, 8× and 16×.

Comparison results: Table 5.3 presents the 4× upscaling results with different methods, for the RGB/depth, RGB/MS and RGB/NIR datasets, respectively. We can see that our method achieves state-of-the-art performances in all the three multi-modal cases. Specifically, for the RGB/depth dataset, our method constantly outperforms the other two deep learning based methods: EDSR [41] and SRFBN [145]. For the RGB/MS and RGB/NIR datasets, our method also performs better than the second best method DJF [65] for most of the testing images. From this table, we can see that the SISR methods, e.g., SCN and EDSR, achieve good results in the RGB/depth case, but do not perform well on the RGB/MS and RGB/NIR cases. This is because the depth image usually has no complicated texture and details. Thus, even without the assistance of the guided images, these SISR methods still achieve good super-resolved results. However, this is not the case for the MS and NIR images which contain more texture and fine details. For these images, the information in the LR image is very limited and thus the assistance of the guidance image becomes quite essential. That is the reason why multi-modal image super-resolution methods like ours or e.g. DJF perform better than the SISR methods. Table 5.4 compares the 8× upscaling results between our and other state-of-the-art methods. It can be seen that our method again outperforms others for different multi-modal cases.

In addition to the 4× and 8× upscaling results, we also compare our method with the others at the very large upscaling factor 16×. This is a very challenging task, because less than 0.4% information is provided in the LR image, i.e., inferring a

RGB/depth	Ambush	Bamboo	Cave	Market	Art	Books	Moebius
Bicubic	32.02	25.15	31.73	29.21	36.37	44.05	45.72
Xie *et al.* [6]	29.25	22.55	28.91	26.40	36.56	43.89	45.65
Park *et al.* [13]	32.52	26.51	31.07	28.62	36.63	43.73	45.09
Ferstl *et al.* [14]	32.58	26.89	32.01	30.06	36.70	43.78	45.02
Lu *et al.* [7]	33.28	27.62	32.42	29.74	35.88	41.36	44.27
Gu *et al.* [8]	32.51	25.62	32.35	29.96	37.20	44.21	46.33
SCN [15]	35.48	28.46	35.32	32.66	39.86	47.46	48.76
RCGD [146]	33.14	25.71	32.68	30.49	38.42	45.15	46.62
VDSR [16]	38.08	28.87	37.13	33.29	42.69	50.63	49.34
DJF [65]	37.66	28.80	36.38	32.82	41.05	50.29	50.98
DGF [17]	23.74	16.35	22.98	22.71	27.52	33.73	34.69
EDSR [41]	<u>38.88</u>	30.15	<u>38.30</u>	34.80	<u>42.93</u>	<u>52.01</u>	51.48
SRFBN [145]	38.70	<u>30.23</u>	38.22	<u>34.97</u>	42.87	51.56	51.21
DepthSR [69]	-	-	-	-	39.25	33.69	**51.87**
Ours	**39.47**	**30.55**	**38.75**	**35.22**	**43.97**	**52.67**	<u>51.82</u>
RGB/MS	Chart toy	Egyptian	Feathers	Glass tiles	Jelly beans	Oil painting	Paints
Bicubic	28.94	36.57	30.80	26.65	27.81	31.67	29.29
JBF [55]	32.56	38.73	33.60	27.52	30.29	32.77	31.94
GF [56]	34.09	40.24	33.60	29.46	30.90	35.03	31.73
SDF [64]	31.87	39.43	33.45	28.22	30.32	32.86	31.96
JFSM [63]	32.98	40.39	33.89	28.98	31.18	35.91	32.76
VDSR [16]	31.98	37.74	32.08	28.19	28.99	32.42	31.03
DJF [65]	<u>35.84</u>	<u>43.88</u>	<u>38.68</u>	**34.34**	<u>35.39</u>	<u>37.09</u>	**38.61**
EDSR [41]	33.45	40.03	35.55	29.75	32.81	32.69	37.28
SRFBN [145]	33.43	40.04	35.53	29.53	32.97	32.68	36.06
DGF [17]	34.19	37.81	31.22	29.93	28.94	36.10	31.72
Ours	**36.58**	**45.91**	**39.62**	<u>33.99</u>	**38.92**	**37.26**	<u>38.40</u>
RGB/NIR	Urban_1	Urban_2	Urban_3	Urban_4	Urban_5	Urban_6	Urban_7
Bicubic	25.93	30.92	25.24	28.04	26.19	26.55	26.72
JBF [55]	28.47	32.10	27.59	30.67	26.82	27.58	27.32
GF [56]	28.75	32.66	27.70	30.69	26.89	27.59	27.35
SDF [64]	26.82	30.60	26.09	29.09	26.61	27.21	27.07
JFSM [63]	30.86	32.86	30.80	32.61	28.97	30.56	27.58
VDSR [16]	27.79	32.65	27.58	30.25	28.06	28.62	28.36
DJF [65]	<u>34.68</u>	<u>38.26</u>	32.33	<u>37.75</u>	**32.29**	<u>34.72</u>	<u>29.64</u>
EDSR [41]	29.02	33.61	28.77	31.28	29.10	29.16	29.47
SRFBN [145]	29.01	33.74	28.85	31.46	29.26	29.49	29.84
DGF [17]	34.53	32.03	<u>33.91</u>	35.43	27.23	28.66	26.56
Ours	**35.57**	**39.18**	**34.19**	**39.26**	<u>31.66</u>	**35.24**	**30.24**

Table 5.3: Results in terms of PSNR (dB) on the RGB/depth, RGB/MS and RGB/NIR datasets for 4× upscaling, with the best results in bold and the second best results underlined.

16×16 patch from only one pixel. The comparison results are shown in Table 5.5. It can be seen that our method achieves higher reconstruction accuracy than other methods for most of the testing images.

Running time. In the testing phase, we need to split the image into several blocks and then upscale each block using the trained network. Thus, the number of overlapping pixels can have an influence on the running speed. Usually, the more overlapping pixels we use, the higher PSNR value we can obtain, but the longer the running time. Table 5.6 shows how the running time and PSNR value change with the number of overlapping pixels, for 4×, 8× and 16 × upscaling factors, respectively. Here, the running time is computed using an Intel(R) i7-6700 CPU @3.4GHz. Recall that the patch sizes we use for 4×, 8× and 16 × upscaling are 8×8, 12×12 and 16×16, which means the largest number of overlapping pixels for them are 7, 11 and 15, respectively. From this table, we can see that the running time of our approach is acceptable even with the largest overlapping regions. In this paper, in order to show the best results, we use the largest number of overlapping pixels to generate the quantitative results in Tables 4.8, 5.4 and 5.5. However, as we can see in Table 5.6, there is actually no big difference between the PSNR values in the last two columns, even though their running speed varies significantly. Thus, in the case where faster running speed is required, we can decrease the number of overlapping pixels and this has small influence on the PSNR values.

5.3.4 Qualitative comparison against other methods

For 4× upscaling, Fig. 5.6 compares the reconstructed images of *Urban_3* from the RGB/NIR dataset using our and other methods. As shown in this figure, our method is able to reconstruct both the texture region and the edges well, while JBF [55] and GF [56] both fail to restore the textures, JFSM [63] struggles to restore clear edges, and DJF [65] over-emphasizes edges.

For 8× upscaling, Figs. 5.7 and 5.8 show the reconstructed images from the Sintel

RGB/depth	Ambush	Bamboo	Cave	Market	Art	Books	Moebius
Bicubic	29.06	22.20	28.65	25.99	33.35	40.82	42.33
Park *et al.* [13]	29.38	22.92	29.17	26.53	33.76	41.85	43.22
Ferstl *et al.* [14]	30.29	20.99	28.10	26.30	31.28	41.24	40.90
Lu *et al.* [7]	29.78	22.13	29.91	25.68	33.61	39.93	41.40
Gu *et al.* [8]	29.18	22.22	28.75	26.07	33.53	41.05	42.74
SCN [15]	31.78	24.60	31.53	28.37	36.22	43.89	45.15
RCGD [146]	30.51	22.81	29.62	27.82	35.85	42.56	43.62
VDSR [16]	32.61	24.12	31.54	27.92	36.44	43.22	44.49
DJF	32.41	24.43	31.28	27.67	36.24	43.12	43.68
DGF [17]	20.21	15.25	20.32	20.51	24.69	30.21	31.37
DepthSR [69]	-	-	-	-	36.84	32.88	**50.07**
EDSR [41]	33.69	25.67	33.36	29.65	38.28	45.72	47.30
Ours	**34.28**	**25.70**	**33.46**	**30.27**	**38.57**	**45.82**	46.00
RGB/MS	Chart toy	Egyptian	Feathers	Glass tiles	Jelly beans	Oil painting	Paints
Bicubic	25.00	33.12	25.59	22.56	23.04	30.56	26.40
JBF [55]	28.00	35.92	26.48	23.06	24.74	31.90	28.21
GF [56]	29.77	36.52	27.85	25.24	25.05	33.82	29.25
SDF [64]	27.89	34.22	28.25	24.66	24.88	31.61	28.54
JFSM [63]	30.28	38.31	28.24	25.31	26.59	32.55	29.97
VDSR [16]	26.23	34.56	26.65	23.22	24.35	31.77	27.79
DJF [65]	32.89	41.58	31.50	**29.53**	30.14	35.12	31.86
DGF [17]	28.39	34.98	26.83	25.46	25.97	33.25	27.89
EDSR [41]	27.04	35.62	28.38	23.53	24.73	31.95	30.33
Ours	**33.18**	**43.46**	**32.04**	27.96	**30.69**	**35.99**	**33.05**
RGB/NIR	Urban_1	Urban_2	Urban_3	Urban_4	Urban_5	Urban_6	Urban_7
Bicubic	22.71	27.03	21.97	24.62	23.87	24.40	24.46
JBF [55]	24.43	28.50	23.57	26.37	24.89	25.41	25.60
GF [56]	27.00	30.74	26.25	28.62	25.81	26.61	27.35
SDF [64]	26.44	30.20	25.82	28.26	25.27	26.12	26.95
JFSM [63]	29.19	32.91	29.28	30.89	27.43	28.90	27.08
VDSR [16]	23.68	28.09	23.11	25.79	24.75	25.21	25.70
DJF [65]	30.96	34.23	30.84	33.24	**29.86**	31.22	28.32
DGF [17]	32.04	31.47	32.32	34.57	25.91	26.26	25.66
EDSR [41]	25.01	28.96	24.13	26.96	25.91	25.61	26.52
Ours	**32.33**	**35.74**	**32.56**	**35.86**	29.51	**32.00**	**28.64**

Table 5.4: Results in terms of PSNR (dB) on the RGB/depth, RGB/MS and RGB/NIR datasets datasets for 8× upscaling, with the best results in bold and the second best results underlined.

RGB/depth	Ambush	Bamboo	Cave	Market	Art	Books	Moebius
Bicubic	26.15	19.25	25.42	22.83	29.78	37.76	39.34
Park *et al.* [13]	25.71	18.62	24.28	21.83	28.77	36.65	36.35
Ferstl *et al.* [14]	24.16	17.87	17.98	21.62	26.66	36.79	34.47
Lu *et al.* [7]	27.87	18.76	27.75	22.03	31.32	37.20	39.60
Gu *et al.* [8]	26.23	19.23	25.45	22.84	29.89	37.97	39.57
SCN [15]	28.67	20.64	27.58	24.94	32.42	40.42	41.77
RCGD [146]	27.38	19.64	26.21	24.06	31.61	39.10	40.90
DJF	28.74	20.47	27.07	24.56	32.20	38.28	40.38
DGF [17]	17.33	14.58	18.07	17.19	22.16	27.19	28.99
DepthSR [69]	-	-	-	-	33.67	32.26	**44.44**
Ours	**31.12**	**21.08**	**28.98**	**25.84**	**33.90**	**41.17**	41.98
RGB/MS	Chart toy	Egyptian	Feathers	Glass tiles	Jelly beans	Oil painting	Paints
Bicubic	22.36	30.32	23.96	21.75	18.70	27.66	20.95
JBF [55]	24.07	32.43	25.66	22.34	19.97	28.87	23.14
GF [56]	25.80	33.71	27.58	24.54	21.41	31.87	25.54
SDF [64]	24.98	33.05	26.87	23.69	20.89	29.69	23.96
JFSM [63]	27.54	36.01	27.79	25.04	24.05	31.76	27.38
DJF [65]	**30.03**	38.82	31.03	**29.08**	25.43	33.04	30.58
DGF [17]	26.83	33.64	25.25	24.16	23.23	31.84	26.16
Ours	29.85	**39.18**	**31.07**	27.82	**26.66**	**33.95**	**31.06**
RGB/NIR	Urban_1	Urban_2	Urban_3	Urban_4	Urban_5	Urban_6	Urban_7
Bicubic	20.17	24.33	19.56	22.32	22.19	22.18	21.33
JBF [55]	21.46	25.70	20.74	23.73	23.89	23.15	23.35
GF [56]	24.78	27.85	22.86	25.99	24.54	24.74	24.66
SDF [64]	22.31	26.58	21.30	24.59	24.06	23.54	23.98
JFSM [63]	27.67	30.00	27.73	29.20	26.21	26.50	24.72
DJF [65]	28.22	31.15	28.68	31.47	27.14	27.79	25.69
DGF [17]	28.26	29.13	29.32	32.71	24.96	24.57	24.55
Ours	**28.80**	**32.07**	**29.55**	**33.06**	**27.85**	**28.86**	**26.31**

Table 5.5: Results in terms of PSNR (dB) on the RGB/depth, RGB/MS and RGB/NIR datasets datasets for 16× upscaling, with the best results in bold and the second best results underlined.

4× upscaling						
Overlapping pixels	2	3	4	5	6	7
Time/s	0.35	0.50	0.85	1.52	3.50	14.10
PSNR/dB	37.87	37.96	38.41	38.44	38.58	38.67
8× upscaling						
Overlapping pixels	2	4	6	8	10	11
Time/s	0.21	0.37	0.62	1.50	6.05	24.23
PSNR/dB	32.82	33.08	33.15	33.46	33.70	33.77
16× upscaling						
Overlapping pixels	6	8	10	12	14	15
Time/s	0.18	0.29	0.52	1.10	4.61	19.02
PSNR/dB	30.73	30.91	31.17	31.26	31.35	31.37

Table 5.6: The running time in seconds of our method with the corresponding PSNR values for RGB/MS dataset.

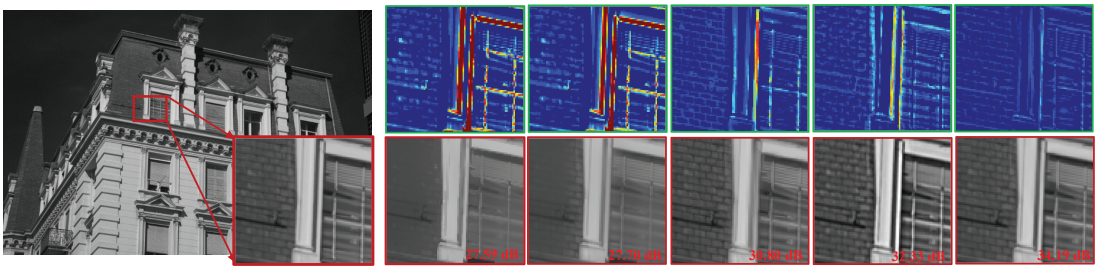


Figure 5.6: Visual comparisons of *Urban_3* in RGB/NIR dataset with upscaling factor = 4. The first row shows the error maps between reconstructed images and the ground-truth images using different methods. The second row shows the corresponding reconstructed images, with the numbers in red indicating the PSNR values.

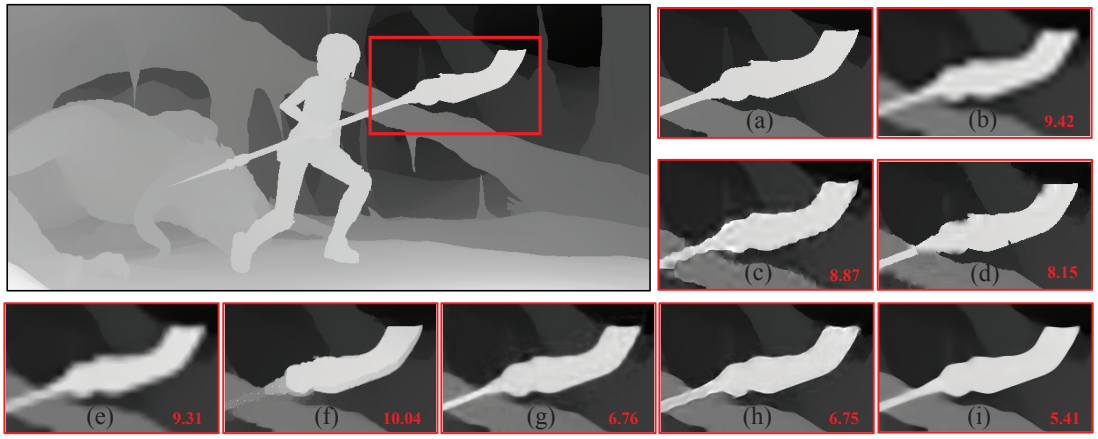


Figure 5.7: Visual comparisons of *Cave* in Sintel dataset with upscaling factor = 8 using different methods. (a) Ground truth, (b) Bicubic, (c) Park *et al.* [13], (d) Lu *et al.* [7], (e) Gu *et al.* [8], (f) Ferstl *et al.* [14], (g) SCN [15], (h) VDSR [16], (i) Ours. The numbers in red indicate the RMSE values.

dataset and the RGB/MS dataset, respectively. From Fig. 5.7, we can see that our method is able to restore clear and accurate edges. In contrast, the edges of Lu *et al.* [7] and Ferstl *et al.* [14] are relatively clear but not accurate, and the edges of SCN [15] and VDSR [16] are accurate but blurred. From the error maps in Fig. 5.8, we can see that the super-resolved image of our method is the closest to the ground-truth image. For the other methods, as indicated in the red box, the reconstructed results of JBF, GF and JFSM methods are strongly affected by the guided RGB image, and the DJF method fails to restore the detailed digits.

We also visualize in Fig. 5.9 the reconstructed depth image *Art* for 16× upscaling

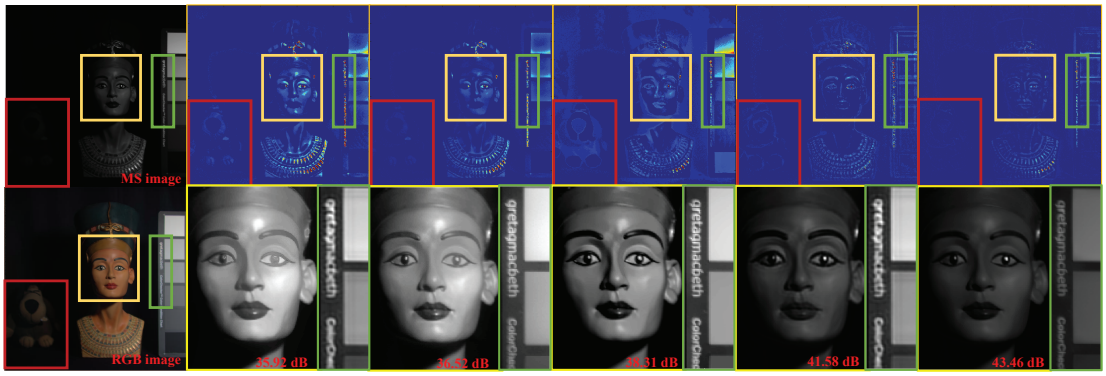
using various methods. Since the upscaling factor is very high, the restored depth images of most methods [8, 13, 15] are with highly blurred edges. Lu *et al.* [7] and Ferstl *et al.* [14] have severe depth diffusion problems. Compared to these methods, our approach is able to restore the depth edges more accurately.

5.3.5 Noisy case

In the multi-modal noisy case, we have the noisy LR input and noise-free guidance input, and we wish to generate the noise-free HR output. Here, the image super-resolution process implicitly include both upscaling and denoising operations. To generate the noisy datasets, we add Gaussian white noise to the LR inputs with different noise variance $\sigma^2 = 2$ and $\sigma^2 = 5$ using the following way:

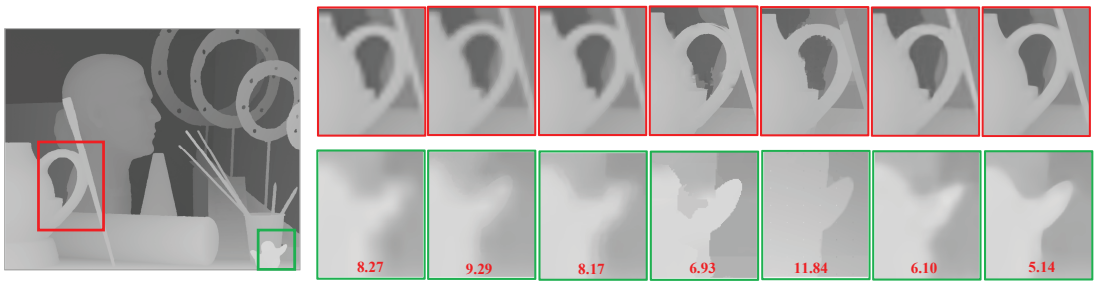
$$\boldsymbol{x}_n = \boldsymbol{x} + \sigma^2/255 * randn(size(\boldsymbol{x})) \tag{5.34}$$

To better evaluate the performance of our approach in the noisy case, we have the following two settings. In the first setting, we simply use the network trained with clean inputs to upscale the noisy images, which means there is a mismatch between the training and testing images. In the second setting, we re-train the network using noisy images and then use the re-trained model to upscale the noisy images. The results are shown in Table 5.7, where the methods with star symbol indicate that we used the re-trained model. From this table, we can see that our approach performs better than other methods whether the network is re-trained or not. This verifies that our method is effective also in noisy scenarios. Fig. 5.10 visualizes the super-resolved images using different methods, and we can see that our method can eliminate the noise and provide cleaner super-resolved images.



(a) GT (b) JBF (c) GF (d) JFSM (e) DJF (f) Ours

Figure 5.8: Visual comparisons of *Egyptian* in RGB/MS dataset with upscaling factor = 8. The first row shows the error maps between reconstructed images and the ground-truth images using different methods. The second row enlarges some regions in the reconstructed images for better comparison. The numbers in red indicate3 the PSNR values.



(a) Bicubic (b) Park (c) Gu (d) Lu (e) Ferstl (f) SCN (g) Ours

Figure 5.9: Visual comparisons of *Art* in Middlebury dataset with upscaling factor = 16. Some regions are enlarged for better viewing. The red numbers inside indicate the RMSE values with different methods.

Table 5.7: The PSNR results on the noisy RGB/depth, RGB/MS and RGB/NIR datasets for 4× upscaling, with the best results in bold. Our* indicates our method with re-trained models using the noisy dataset.

RGB/depth (PSNR/dB)								
σ^2=2	Ambush	Bamboo	Cave	Market	Art	Books	Moebius	Average
Bicubic	31.74	25.10	31.47	29.07	35.66	40.97	41.69	33.67
VDSR [16]	35.03	29.19	34.67	31.25	37.05	38.94	38.97	35.02
DJF [65]	29.91	22.87	28.55	25.88	33.95	39.67	40.49	31.62
EDSR [41]	35.22	29.83	35.09	33.00	37.37	39.38	39.41	35.61
Ours	**37.60**	**29.93**	**36.82**	**34.04**	**40.00**	**43.27**	**43.17**	**37.83**
Ours*	38.14	29.90	37.25	34.24	41.34	47.06	46.49	39.20
σ^2=5	Ambush	Bamboo	Cave	Market	Art	Books	Moebius	Average
Bicubic	30.55	24.81	30.32	28.35	33.15	35.32	35.50	31.14
VDSR [16]	30.00	26.80	30.08	28.39	31.00	31.66	31.71	29.95
DJF [65]	29.34	22.75	28.14	25.65	32.60	36.05	36.42	30.13
EDSR [41]	28.86	27.19	29.38	28.56	30.00	30.45	30.79	29.32
Ours	**33.61**	**28.47**	**33.11**	**31.55**	**35.01**	**36.51**	**36.22**	**33.50**
Ours*	36.50	28.91	35.27	33.10	38.69	43.58	43.10	37.02
RGB/Multi-spectral (PSNR/dB)								
σ^2=2	Chart toy	Egyptian	Feathers	Glass tiles	Jelly beans	Oil painting	Paints	Average
Bicubic	28.80	35.81	30.59	26.57	27.71	31.42	29.14	30.01
GF [56]	30.37	37.59	31.99	27.33	29.55	32.21	30.67	31.39
EDSR [41]	31.80	37.36	33.66	28.88	31.02	31.84	34.02	32.65
DJF [65]	33.56	39.78	35.57	30.08	32.94	33.37	35.65	34.42
Ours	**36.00**	**42.95**	**38.46**	**33.65**	**37.78**	**36.64**	**37.47**	**37.57**
Ours*	36.40	44.46	38.76	33.80	37.98	36.97	37.68	38.01
σ^2=5	Chart toy	Egyptian	Feathers	Glass tiles	Jelly beans	Oil painting	Paints	Average
Bicubic	28.16	33.24	29.63	26.17	27.21	30.30	27.42	28.86
GF [56]	30.14	36.94	31.64	27.18	29.37	31.83	30.51	31.09
EDSR [41]	28.80	31.62	29.68	26.73	27.84	28.74	29.47	29.98
DJF [65]	29.69	34.74	30.88	28.62	31.79	32.55	31.64	31.42
Ours	**33.88**	**37.47**	**35.14**	**32.22**	**34.47**	**34.30**	**34.49**	**34.57**
Ours*	35.20	41.95	37.35	32.68	35.71	35.76	36.15	36.40
RGB/NIR (PSNR/dB)								
σ^2=2	Urban_1	Urban_2	Urban_3	Urban_4	Urban_5	Urban_6	Urban_7	Average
Bicubic	25.89	30.66	25.12	27.91	26.18	26.45	26.57	26.97
GF [56]	27.27	31.46	26.32	29.39	26.70	27.20	27.19	27.93
EDSR [41]	28.38	32.31	27.95	30.29	28.19	28.29	28.80	29.17
DJF [65]	30.46	34.16	29.39	31.59	29.07	29.99	29.03	30.53
Ours	**34.10**	**37.19**	**33.55**	**37.64**	**31.02**	**33.98**	**29.91**	**33.91**
Ours*	34.54	38.40	34.39	38.19	31.31	34.50	30.03	34.48
σ^2=5	Urban_1	Urban_2	Urban_3	Urban_4	Urban_5	Urban_6	Urban_7	Average
Bicubic	25.56	29.71	24.83	27.37	25.81	26.07	26.18	26.50
GF [56]	27.19	31.07	26.26	29.26	26.55	27.08	27.08	27.78
EDSR [41]	26.19	28.47	25.86	27.53	26.04	26.28	26.71	26.73
DJF [65]	29.91	31.38	28.74	30.62	27.55	28.73	27.66	29.23
Ours	**32.08**	**33.99**	**31.86**	**34.29**	**30.04**	**32.15**	**29.12**	**31.93**
Ours*	32.61	36.30	32.34	35.87	30.47	32.87	29.53	32.86

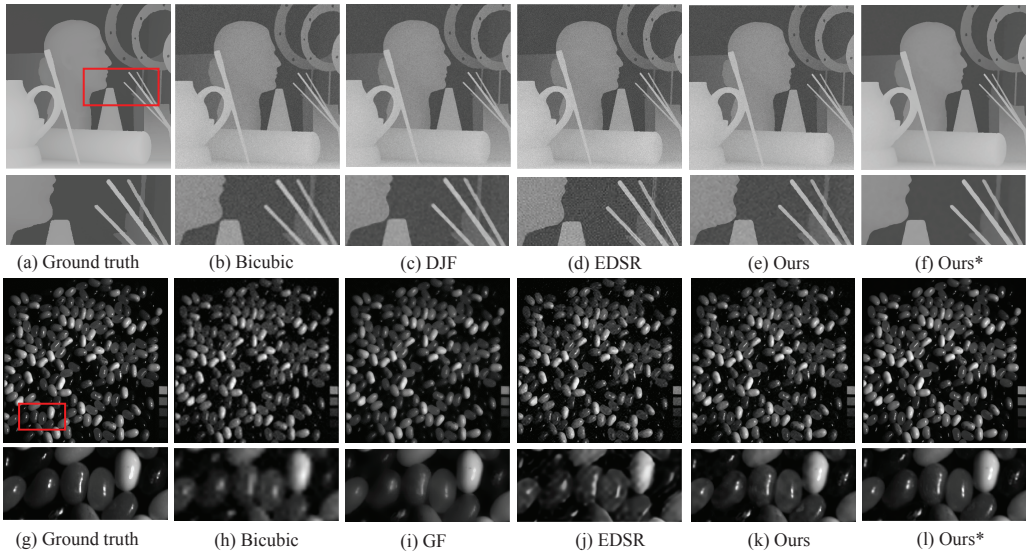


Figure 5.10: Visual comparison of 4× upscaling results in noisy case with $\sigma^2 = 2$. Some regions are enlarged for better comparison. The symbol * indicates that the results are obtained by re-training the models using the noisy datasets.

5.4 Summary

In this chapter, we introduced a joint multi-modal dictionary learning algorithm to perform the MISR task. Based on this algorithm, we further proposed a new deep coupled network for MISR through unfolding the iterative shrinkage and thresholding algorithm (ISTA). We also developed a novel way to initialize the network parameters before training. Compared with the random initialization, our new initialization approach achieves better performance in both training and testing phases. We test the performance of our method with different multi-modal datasets including RGB/depth, RGB/near-infrared and RGB/multi-spectral datasets, for different upscaling factors. The numerical results show that our method outperforms other state-of-the-art methods in both reconstruction accuracy and visual quality.

Chapter 6

Deep CNN for Multi-modal Image Restoration and Fusion

6.1 Introduction

Multi-modal image processing has been attracting increasing interest from the computer vision community, due to a variety of intriguing applications, e.g., image style transfer [45, 101], image fusion [25, 147], RGB guided depth image super-resolution [8, 146], image denoising [148]. Based on the reconstruction target, these applications can be roughly classified into two categories, the multi-modal image restoration (MIR) and multi-modal image fusion (MIF) task. Given an image $\boldsymbol{x}$ and another image $\boldsymbol{y}$, in MIR problem, one aims to recover a better version of $\boldsymbol{x}$ with the guidance of $\boldsymbol{y}$, while the MIF task aims to fuse $\boldsymbol{x}$ and $\boldsymbol{y}$ to a new image $\boldsymbol{z}$ which has both the advantages of $\boldsymbol{x}$ and $\boldsymbol{y}$. Different from uni-modal image processing tasks, e.g., single image super-resolution, multi-modal image processing usually requires proper modelling of the dependencies across different modalities.

In this chapter, we propose a deep convolutional neural network named the Common and Unique information splitting network (CU-Net), by incorporating the convolutional sparse coding modelling into a neural network. To the best of our knowledge,

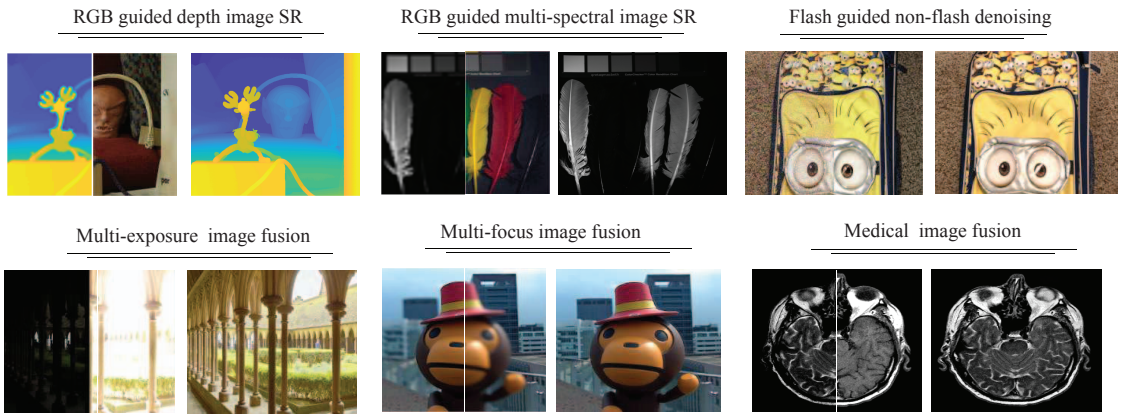


Figure 6.1: Examples of different multi-modal image restoration and fusion tasks. The first row shows the MIR related applications, including RGB guided depth image SR, RGB guided multi-spectral image SR, and flash guided non-flash image denoising. The second row shows the MIF related applications, including multi-exposure image fusion, multi-focus image fusion, and medical image fusion.

this is the first time a universal framework is proposed to solve both the MIR and MIF problems. Our network is able to automatically split the common features from the unique features between the target and guided modality, which is beneficial to both MIR and MIF tasks. Compared with other empirically designed networks, the proposed CU-Net is derived from a new multi-modal convolutional sparse coding (MCSC) model, and thus each part of the network has good interpretability. The MCSC model is developed upon a recent coupled dictionary learning algorithm in [30]. This algorithm is demonstrated to model well the dependencies across modalities, but it is performed at patch level, which means it only models the patch dependencies. In contrast, our MCSC model is defined at image level, and thus we can model the image dependencies across modalities. We test the performance of the proposed method on various MIR and MIF tasks, such as RGB guided depth image super-resolution, flash guided non-flash image denoising, multi-focus and multi-exposure image fusion, as shown in Fig. 6.1. The numerical results validate the effectiveness, flexibility, and universality of our method.

6.2 Multi-modal convolutional sparse coding

In multi-modal image processing, one important issue is to model the dependencies among different modalities. In this section, we first introduce the multi-modal

Notation	Dimension	Description
$\boldsymbol{x}$	$\mathbb{R}^{n\times n}$	the input image
$\boldsymbol{y}$	$\mathbb{R}^{n\times n}$	the guidance image
$\boldsymbol{z}$	$\mathbb{R}^{n\times n}$	the target image
$\boldsymbol{d}_k^c$	$\mathbb{R}^{s\times s}$	the k-th common filter of $\boldsymbol{x}$
$\boldsymbol{d}_k^u$	$\mathbb{R}^{s\times s}$	the k-th unique filter of $\boldsymbol{x}$
$\boldsymbol{h}_k^c$	$\mathbb{R}^{s\times s}$	the k-th common filter of $\boldsymbol{y}$
$\boldsymbol{h}_k^u$	$\mathbb{R}^{s\times s}$	the k-th unique filter of $\boldsymbol{y}$
$\boldsymbol{g}_k^c$	$\mathbb{R}^{s\times s}$	the k-th common filter of $\boldsymbol{z}$
$\boldsymbol{g}_k^u$	$\mathbb{R}^{s\times s}$	the k-th unique filter of $\boldsymbol{z}$ related with $\boldsymbol{x}$
$\boldsymbol{g}_k^v$	$\mathbb{R}^{s\times s}$	the k-th unique filter of $\boldsymbol{z}$ related with $\boldsymbol{y}$
$\boldsymbol{c}_k$	$\mathbb{R}^{n\times n}$	the k-th common feature response
$\boldsymbol{u}_k$	$\mathbb{R}^{n\times n}$	the k-th unique feature response of $\boldsymbol{x}$
$\boldsymbol{v}_k$	$\mathbb{R}^{n\times n}$	the k-th unique feature response of $\boldsymbol{y}$

Table 6.1: Summary of notations

convolutional sparse coding (MCSC) model for the MIR and MIF tasks, and then introduce the optimization formulations, together with the solutions for these two tasks. To make the notations clear, we list the main symbols used in this section in Table 6.1.

6.2.1 The MCSC model

For the MIR task, the problem can be formulated as follows: given a distorted image $\boldsymbol{x} \in \mathbb{R}^{n\times n}$, and a guidance image $\boldsymbol{y} \in \mathbb{R}^{n\times n}$, we aim to reconstruct an image $\boldsymbol{z} \in \mathbb{R}^{n\times n}$ which is a high-quality version of $\boldsymbol{x}$. Here, we assume that the image is square. Since $\boldsymbol{x}$ and $\boldsymbol{y}$ capture the same scene, they should share some common features, but they also have unique features. Take the depth and RGB images for example, discontinuities in the depth image are clearly related to edges in RGB image. However, RGB image contains texture information, which has no relations with depth image. Based on this observation, we model the relationships among different modalities as follows:

$$\boldsymbol{x} = \sum_k \boldsymbol{d}_k^c * \boldsymbol{c}_k + \sum_k \boldsymbol{d}_k^u * \boldsymbol{u}_k, \tag{6.1}$$

$$\boldsymbol{y} = \sum_k \boldsymbol{h}_k^c * \boldsymbol{c}_k + \sum_k \boldsymbol{h}_k^v * \boldsymbol{v}_k, \tag{6.2}$$

$$\boldsymbol{z} = \underbrace{\sum_k \boldsymbol{g}_k^c * \boldsymbol{c}_k}_{\substack{\text{common features} \\ \text{between } \boldsymbol{x} \text{ and } \boldsymbol{y}}} + \underbrace{\sum_k \boldsymbol{g}_k^u * \boldsymbol{u}_k}_{\substack{\text{unique features} \\ \text{of } \boldsymbol{x}}} \tag{6.3}$$

Here, the symbol * denotes the convolutional operation, $\{\boldsymbol{d}_k^c\}_{k=1}^K$, $\{\boldsymbol{h}_k^c\}_{k=1}^K$ and $\{\boldsymbol{g}_k^c\}_{k=1}^K \in \mathbb{R}^{s\times s\times K}$ are the common filters of $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$, respectively; $\{\boldsymbol{d}_k^u\}_{k=1}^K$ and $\{\boldsymbol{h}_k^v\}_{k=1}^K$ are the unique filters of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively; $\{\boldsymbol{g}_k^u\}_{k=1}^K \in \mathbb{R}^{s\times s\times K}$ is the unique filter of $\boldsymbol{z}$ which is related with $\boldsymbol{x}$. The common filters share the same feature responses $\{\boldsymbol{c}_k\}_{k=1}^K \in \mathbb{R}^{n\times n\times K}$, while the unique filters have their own feature responses $\{\boldsymbol{u}_k\}_{k=1}^K$ and $\{\boldsymbol{v}_k\}_{k=1}^K \in \mathbb{R}^{n\times n\times K}$. In summary, $\boldsymbol{x}$ and $\boldsymbol{y}$ share the same common feature responses $\{\boldsymbol{c}_k\}_{k=1}^K$, but also have different unique feature responses $\{\boldsymbol{u}_k\}_{k=1}^K$ and $\{\boldsymbol{v}_k\}_{k=1}^K$ as indicated in Eqs. (6.1) and (6.2). Since the unique information in the guidance image $\boldsymbol{y}$ is not useful for the reconstruction of $\boldsymbol{z}$, the model of $\boldsymbol{z}$ is composed of two parts: the features which are in common between $\boldsymbol{x}$ and $\boldsymbol{y}$, and the unique features of $\boldsymbol{x}$, as in Eq. (6.3). The advantage of this model is that only the useful information is preserved while the information that may interfere with the estimation of $\boldsymbol{z}$ is discarded.

For the MIF task, we aim to fuse the image $\boldsymbol{x}$ with another image $\boldsymbol{y}$ to obtain a new image $\boldsymbol{z}$ which has both the advantages of $\boldsymbol{x}$ and $\boldsymbol{y}$. Different from the MIR task in which only the common information in $\boldsymbol{y}$ is helpful, here in the MIF task, all the information contained in $\boldsymbol{y}$ might be useful for the reconstruction of $\boldsymbol{z}$. In other words, the fused image $\boldsymbol{z}$ should be composed of three parts: common features shared by $\boldsymbol{x}$ and $\boldsymbol{y}$, unique features of $\boldsymbol{x}$ and unique features of $\boldsymbol{y}$. Thus, in the MIF task, the models for $\boldsymbol{x}$ and $\boldsymbol{y}$ are the same as for the MIR task, but the model for $\boldsymbol{z}$ should be re-formulated as follows:

$$\boldsymbol{z} = \underbrace{\sum_k \boldsymbol{g}_k^c * \boldsymbol{c}_k}_{\substack{\text{common features} \\ \text{between } \boldsymbol{x} \text{ and } \boldsymbol{y}}} + \underbrace{\sum_k \boldsymbol{g}_k^u * \boldsymbol{u}_k}_{\substack{\text{unique features} \\ \text{of } \boldsymbol{x}}} + \underbrace{\sum_k \boldsymbol{g}_k^v * \boldsymbol{v}_k}_{\substack{\text{unique features} \\ \text{of } \boldsymbol{y}}}, \tag{6.4}$$

where $\{\boldsymbol{g}_k^u\}_{k=1}^K$ and $\{\boldsymbol{g}_k^v\}_{k=1}^K \in \mathbb{R}^{s\times s\times K}$ are the unique filters of $\boldsymbol{z}$ related with $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively.

6.2.2 Convolutional sparse coding

In the synthesis phase, given the input images $\boldsymbol{x}$ and $\boldsymbol{y}$, we need to infer the image $\boldsymbol{z}$. To this end, assuming all dictionaries are known, we need to calculate the common and unique feature responses $\{\boldsymbol{c}_k, \boldsymbol{u}_k, \boldsymbol{v}_k\}_{k=1}^K$, through solving the following optimization problem:

$$\begin{aligned}\underset{\{\boldsymbol{c}_k,\boldsymbol{u}_k,\boldsymbol{v}_k\}}{\operatorname{argmin}} &\frac{1}{2}\left\|\boldsymbol{x}-\sum_k(\boldsymbol{d}_k^c*\boldsymbol{c}_k+\boldsymbol{d}_k^u*\boldsymbol{u}_k)\right\|_2^2\\ &+\frac{1}{2}\left\|\boldsymbol{y}-\sum_k(\boldsymbol{h}_k^c*\boldsymbol{c}_k+\boldsymbol{h}_k^v*\boldsymbol{v}_k)\right\|_2^2\\ &+\lambda\sum_k(\|\boldsymbol{c}_k\|_1+\|\boldsymbol{u}_k\|_1+\|\boldsymbol{v}_k\|_1).\end{aligned} \tag{6.5}$$

To solve this problem, our strategy is to alternately update each variable with other variables fixed. This leads to the following three steps:

- Step 1, we fix the common feature response $\boldsymbol{c}_k$ and the unique response $\boldsymbol{v}_k$ of $\boldsymbol{y}$, to update the unique response $\boldsymbol{u}_k$ of $\boldsymbol{x}$.
- Step 2, we fix the common feature response $\boldsymbol{c}_k$ and the unique response $\boldsymbol{u}_k$ of $\boldsymbol{x}$, to update the unique response $\boldsymbol{v}_k$ of $\boldsymbol{y}$.
- Step 3, we fix all the unique responses $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ to update the common response $\boldsymbol{c}_k$.

These three steps should be repeated until convergence. After solving (6.5), the target image $\boldsymbol{z}$ can be reconstructed by either Eq. (6.3) or Eq. (6.4), depending on the task. However, solving (6.5) is very time-consuming, since the solution requires several iterations. In this paper, we aim to solve (6.5) using a deep learning approach, by turning each step above into a deep network module with tunable parameters and adjustable number of layers. The next section introduces in detail how we turn these three steps into a deep network.

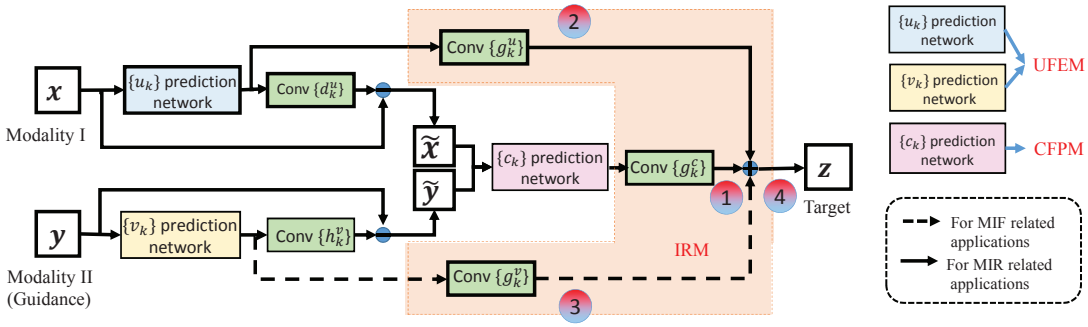


Figure 6.2: Network Architecture of the proposed CU-Net. For the MIR related tasks, the final reconstruction (Point 4) is composed of the common reconstruction (Point 1) and the unique reconstruction (Point 2). For the MIF related tasks, the final reconstruction is composed of the common reconstruction (Point 1) and the two unique reconstructions (Point 2 and Point 3).

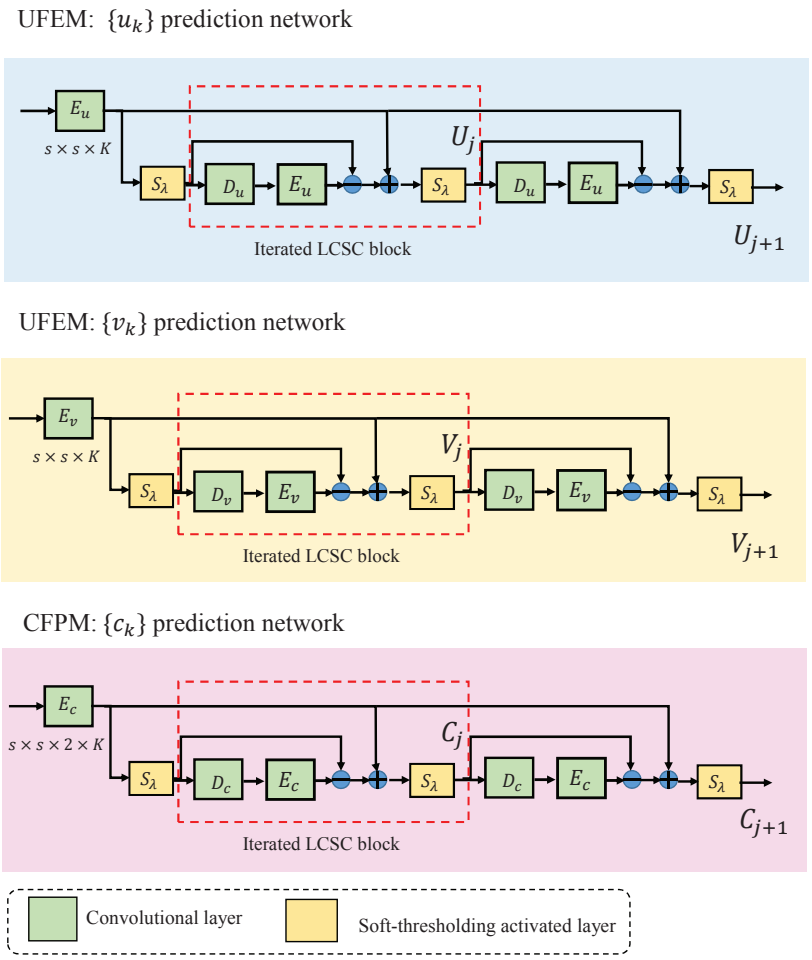


Figure 6.3: The network architectures of UFEM and CFPM.

6.3 Common and unique information splitting network (CU-Net)

The architecture of the proposed CU-Net is shown in Fig. 6.2. We can see that the network is composed of three parts: the unique feature extraction module (UFEM), common feature preservation module (CFPM), and the image reconstruction module (IRM). Next, we will introduce these three modules in detail and explain how we design the architecture of each of them from the MCSC model.

6.3.1 Unique feature extraction module

The UFEM aims to extract the unique features from each source image. Since we have two source images $\boldsymbol{x}$ and $\boldsymbol{y}$, we have two UFEMs, i.e., the $\{\boldsymbol{u}_k\}$ prediction network and the $\{\boldsymbol{v}_k\}$ prediction network, as shown in Fig. 6.2. As discussed in Section 6.2.2, the first step is to fix $\boldsymbol{c}_k$ and $\boldsymbol{v}_k$ to update $\boldsymbol{u}_k$, which turns (6.5) into the following:

$$\underset{\{\boldsymbol{u}_k\}}{\operatorname{argmin}} \frac{1}{2} \left\| \boldsymbol{x} - \sum_k (\boldsymbol{d}_k^c * \boldsymbol{c}_k + \boldsymbol{d}_k^u * \boldsymbol{u}_k) \right\|_2^2 + \lambda \sum_k \|\boldsymbol{u}_k\|_1 . \tag{6.6}$$

Since $\boldsymbol{c}_k$ is fixed, we can further simplify (6.6) by denoting $\hat{\boldsymbol{x}} = \boldsymbol{x} - \sum_k \boldsymbol{d}_k^c * \boldsymbol{c}_k$, which leads to the following expression:

$$\underset{\{\boldsymbol{u}_k\}}{\operatorname{argmin}} \frac{1}{2} \left\| \hat{\boldsymbol{x}} - \sum_k \boldsymbol{d}_k^u * \boldsymbol{u}_k \right\|_2^2 + \lambda \sum_k \|\boldsymbol{u}_k\|_1 . \tag{6.7}$$

Eq. (6.7) is a standard convolutional sparse coding problem, which can be solved using the traditional method in [149]. We instead solve this problem using the learned convolutional sparse coding (LCSC) algorithm proposed in [1], which gives us the solution of $\boldsymbol{u}_k$ as follows:

$$\boldsymbol{U}_{j+1} = S_\lambda(\boldsymbol{U}_j - \boldsymbol{E}_u * \boldsymbol{D}_u * \boldsymbol{U}_j + \boldsymbol{E}_u * \hat{\boldsymbol{x}}), \tag{6.8}$$

where $\boldsymbol{U} \in \mathbb{R}^{n\times n\times K}$ is the stack of feature responses $\{\boldsymbol{u}^k\}_{k=1}^K$, and $\boldsymbol{U}_j$ is the update of $\boldsymbol{U}$ at the j-th iteration, $\boldsymbol{D}_u \in \mathbb{R}^{s\times s\times K}$ and $\boldsymbol{E}_u \in \mathbb{R}^{s\times s\times K}$ are the learnable convolutional layers related to the filters $\{\boldsymbol{d}_k^u\}_{k=1}^K$. For details about how to derive (6.8) by solving (6.7), please refer to the Appendix B. The $S_\lambda(\cdot)$ indicates the soft-thresholding operation with λ as the threshold. Following [1], we can unfold the iterations in (6.8) into a neural network, as shown in the UFEM module (the upper figure) in Fig. 6.3. Theoretically, since each LCSC block in the UFEM module corresponds to one iteration in (6.8), the UFEM module can be extended to any number of LCSC blocks, which makes the network architecture very flexible.

The architecture of $\{\boldsymbol{v}_k\}$ prediction network is obtained in a way similar to that of $\{\boldsymbol{u}_k\}$, and it is derived from the second step in solving Eq. (5.3). When updating $\boldsymbol{v}_k$, we need to fix $\boldsymbol{c}_k$ and $\boldsymbol{u}_k$, so that Eq. (5.3) becomes:

$$\underset{\{\boldsymbol{v}_k\}}{\operatorname{argmin}} \frac{1}{2} \left\| \hat{\boldsymbol{y}} - \sum_k \boldsymbol{h}_k^v * \boldsymbol{v}_k \right\|_2^2 + \lambda \sum_k \|\boldsymbol{v}_k\|_1 , \tag{6.9}$$

where $\hat{\boldsymbol{y}} = \boldsymbol{y} - \sum_k \boldsymbol{h}_k^c * \boldsymbol{c}_k$. Likewise, we can use the LCSC algorithm from [1] to solve (6.9), which gives us the following iterations:

$$\boldsymbol{V}_{j+1} = S_\lambda(\boldsymbol{V}_j - \boldsymbol{E}_v * \boldsymbol{D}_v * \boldsymbol{V}_j + \boldsymbol{E}_v * \hat{\boldsymbol{y}}). \tag{6.10}$$

Here, $\boldsymbol{V} \in \mathbb{R}^{n\times n\times K}$ is the stack of feature responses $\{\boldsymbol{v}^k\}_{k=1}^K$, and $\boldsymbol{V}_j$ is the update of $\boldsymbol{V}$ at the j-th iteration, $\boldsymbol{D}_v \in \mathbb{R}^{s\times s\times K}$ and $\boldsymbol{E}_v \in \mathbb{R}^{s\times s\times K}$ are the learnable convolutional layers related with $\{\boldsymbol{h}_k^v\}_{k=1}^K$. The iterations in Eq. (6.10) can be also unfolded into a neural network, as shown in the middle figure in Fig. 6.3.

6.3.2 Common feature preservation module

Since different modalities capture the same image scene, there exist some consistent features among them. The CFPM aims to preserve these common features. The network architecture of CFPM is derived from the third step in solving Eq. (6.5) which aims to predict the common features $\{c_k\}$. Thus, the CFPM is also called the $\{c_k\}$ prediction network in this paper. When fixing the unique responses u_k and v_k,

Eq. (6.5) can be re-written as follows:

$$\begin{aligned}\underset{\{\boldsymbol{c}_k\}}{\operatorname{argmin}}\frac{1}{2}\left\|\tilde{\boldsymbol{x}}-\sum_k \boldsymbol{d}_k^c * \boldsymbol{c}_k\right\|_2^2+\frac{1}{2}\left\|\tilde{\boldsymbol{y}}-\sum_k \boldsymbol{h}_k^c * \boldsymbol{c}_k\right\|_2^2 \\ +\lambda\sum_k \|\boldsymbol{c}_k\|_1,\end{aligned} \tag{6.11}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \sum_k \boldsymbol{d}_k^u * \boldsymbol{u}_k$ and $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \sum_k \boldsymbol{h}_k^v * \boldsymbol{v}_k$. The first two terms in Eq. (6.11) can be further combined and this yields the following optimization problem:

$$\underset{\{\boldsymbol{c}_k\}}{\operatorname{argmin}}\frac{1}{2}\left\|\boldsymbol{p}-\sum_k \boldsymbol{l}_k^c * \boldsymbol{c}_k\right\|_2^2+\lambda\sum_k \|\boldsymbol{c}_k\|_1, \tag{6.12}$$

where $\boldsymbol{p} \in \mathbb{R}^{n\times n\times 2}$ is the concatenation of $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$, and $\boldsymbol{l}_k^c \in \mathbb{R}^{s\times s\times 2}$ is the concatenation of $\boldsymbol{d}_k^c$ and $\boldsymbol{h}_k^c$. Eq. (6.12) can be then solved using again the LCSC algorithm from [1]. This leads to the following solution:

$$\boldsymbol{C}_{j+1} = S_\lambda(\boldsymbol{C}_j - \boldsymbol{E}_c * \boldsymbol{D}_c * \boldsymbol{C}_j + \boldsymbol{E}_c * \boldsymbol{p}), \tag{6.13}$$

where $\boldsymbol{C} \in \mathbb{R}^{n\times n\times K}$ is the stack of common feature responses $\{\boldsymbol{c}^k\}_{k=1}^K$ and $\boldsymbol{C}_j$ is the update of $\boldsymbol{C}$ at the j-th iteration, $\boldsymbol{D}_c \in \mathbb{R}^{s\times s\times K\times 2}$ and $\boldsymbol{E}_c \in \mathbb{R}^{s\times s\times 2\times K}$ are the learnable convolutional layers related with $\{\boldsymbol{l}_k^c\}_{k=1}^K$. The iterations in Eq. (6.13) can be unfolded and this leads to the CFPM architecture shown in the bottom figure in Fig. 6.3.

6.3.3 Image reconstruction module

After we obtain the feature responses $\{c_k, u_k, v_k\}_{k=1}^K$, the next step is to reconstruct the target image $\boldsymbol{z}$ using either Eq. (6.3) for MIR related tasks or Eq. (6.4) for MIF related tasks. Specifically, for MIR related tasks, the image reconstruction module (IRM) only contains two set of filters $\{\boldsymbol{g}_k^u\}_{k=1}^K$ and $\{\boldsymbol{g}_k^c\}_{k=1}^K$. As shown in Fig. 6.2, the filter $\{\boldsymbol{g}_k^u\}_{k=1}^K$ is directly connected to the output of $\{u_k\}$ prediction network, which corresponds to the term $\sum_k \boldsymbol{g}_k^u * \boldsymbol{u}_k$ in Eq. (6.3). The filter $\{\boldsymbol{g}_k^c\}_{k=1}^K$ is connected to the output of $\{c_k\}$ prediction network, which corresponds to the

term $\sum_k \boldsymbol{g}_k^c * \boldsymbol{c}_k$ in Eq. (6.3). Then, these two terms are added to get $\boldsymbol{z}$. For the MIF related tasks, we use Eq. (6.4) to create the IRM, and it contains three set of filters $\{\boldsymbol{g}_k^u\}_{k=1}^K$, $\{\boldsymbol{g}_k^v\}_{k=1}^K$, and $\{\boldsymbol{g}_k^c\}_{k=1}^K$. The only difference to the MIR related task is the filter $\{\boldsymbol{g}_k^v\}_{k=1}^K$, which is connected to the output of $\{v_k\}$ prediction network, as shown in Fig. 6.2.

6.3.4 Discussion about the CU-Net architecture

It is of interest to note that, although the CU-Net is derived from a theoretical MCSC model, it contains both skip connections and residual blocks. These two elements have both been demonstrated to improve the reconstruction performance of CNN architectures for different tasks. The concept of skip connection was first proposed by Ronneberger *et al.* [150], and later successfully used in the field of image recognition [151], image super-resolution [16, 152] and image denoising [22]. In our network, as shown in Fig. 6.2, the target $\boldsymbol{z}$ is obtained by combining three parts, in which the two parts convolved with $\boldsymbol{g}_k^u$ and $\boldsymbol{g}_k^v$ both use the skip connections. In addition, as we can see in Fig. 6.3, there are two residual lines in our UFEM and CFPM, which are similar to the residual block proposed in [41]. As verified in [41], the residual block can make full use of the hierarchical features and thus significantly improve the image super-resolution performance. In this paper, we also demonstrate the effectiveness of the residual UFEM and CFPM architecture in Section 6.4.5.

6.4 Numerical results

In this section, we verify the effectiveness of our method on various applications, which can be classified into two categories: MIR and MIF related tasks. The MIR related tasks include RGB guided depth image super-resolution, RGB guided multispectral image super-resolution, and flash guided non-flash image denoising. The MIF related tasks include multi-exposure image fusion, multi-focus image fusion and medical image fusion.

6.4.1 Experimental setup

For each task, we train the network using around 150,000 image patches with size 64×64. The Adam optimizer is used to train the network, with a basic learning rate of $1e^{-4}$ and it is decayed by 0.9 every 50 epochs. The number of iterated LCSC blocks in the UFEM and CFPM modules is set to 4. For each convolutional layer, the filter size is 8×8, and the number of filters is 64. We choose the size of mini-batch as 64 and the total number of epochs as 200. Note that for applications which aim to restore an image with multiple channels, e.g., the multi-exposure image fusion, instead of processing each channel independently, we adjust the dimensions of corresponding convolutional layers to accept multi-channel inputs. Specifically, we have $\boldsymbol{D}_u$ and $\boldsymbol{D}_v \in \mathbb{R}^{s\times s\times K\times m}$, $\boldsymbol{E}_u$ and $\boldsymbol{E}_v \in \mathbb{R}^{s\times s\times m\times K}$, $\boldsymbol{D}_c \in \mathbb{R}^{s\times s\times K\times 2m}$ and $\boldsymbol{E}_c \in \mathbb{R}^{s\times s\times 2m\times K}$, where m is the number of channels.

Next, in Section 6.4.2, we show the simulation results of our method on the MIR related tasks, together with the comparisons with other state-of-the-art approaches. In Section 5.2, we show the simulation results on the MIF related tasks. In order to further verify the effectiveness of our deep network, we visualize in Section 5.3 the features generated by different parts of the network, and in Section 5.4 we present comprehensive ablation study results. In Section 5.5, the computational cost is discussed.

6.4.2 MIR related tasks

RGB guided depth image SR

We use the dataset from [136] to train the network, which is composed of 1000 synthetic RGB/Depth image pairs. The testing images are from the Middlebury dataset [154] and the Sintel dataset [155]. The upscaling factor is 4, and we generate the low-resolution depth image by downsampling the high-resolution depth image by the upscaling factor, and then upsample it using the bicubic interpolation. For the high-resolution RGB image, we convert it into the YCbCr format and only the Y channel is used as guidance.

RMSE	Ambush	Cave	Market	Art	Books	Moebius	Reindeer	Average
Bicubic	6.39	6.61	8.83	3.87	1.60	1.32	2.81	4.49
Xie *et al.* [6]	8.79	9.14	12.21	3.79	1.63	1.33	2.77	5.67
Park *et al.* [13]	6.03	7.13	9.45	3.76	1.66	1.42	2.79	4.61
Ferstl *et al.* [14]	5.99	6.40	8.01	3.73	1.65	1.43	2.90	4.30
Lu *et al.* [7]	5.53	6.10	8.31	4.10	2.18	1.56	3.24	4.43
Gu *et al.* [8]	6.04	6.15	8.10	3.52	1.57	1.23	2.66	4.18
SCN [15]	4.29	4.37	5.94	2.59	1.08	0.93	1.94	3.02
EDSR [41]	2.90	3.10	4.64	1.82	0.64	0.68	1.27	2.13
SRFBN [145]	2.96	3.13	4.55	1.83	0.67	0.70	1.30	2.42
RCGD [146]	5.62	5.92	7.62	3.06	1.41	1.19	2.52	3.90
RADAR [19]	3.43	3.57	5.13	2.00	0.68	0.73	1.43	2.42
DGF [17]	16.57	18.09	18.66	10.73	5.25	4.70	6.65	11.52
DJFR [18]	2.95	3.10	4.56	1.96	0.68	0.66	1.15	2.15
CoISTA [153]	2.71	2.94	4.42	1.61	0.59	0.65	1.14	2.01
CU-Net	**2.52**	**2.75**	**4.09**	**1.54**	**0.55**	**0.63**	**1.10**	**1.88**
SSIM	Ambush	Cave	Market	Art	Books	Moebius	Reindeer	Average
Bicubic	0.9685	0.9503	0.9295	0.9687	0.9911	0.9908	0.9852	0.9692
Xie *et al.* [6]	0.9438	0.9221	0.8869	0.9758	0.9917	0.9910	0.9887	0.9571
Park *et al.* [13]	0.9678	0.9379	0.9067	0.9752	0.9912	0.9911	0.9864	0.9638
Ferstl *et al.* [14]	0.9701	0.9563	0.9298	0.9771	0.9915	0.9909	0.9859	0.9717
Lu *et al.* [7]	0.9712	0.9610	0.9266	0.9747	0.9896	0.9896	0.9867	0.9713
Gu *et al.* [8]	0.9766	0.9613	0.9470	0.9779	0.9923	0.9930	0.9883	0.9766
SCN [15]	0.9850	0.9769	0.9664	0.9858	0.9951	0.9949	0.9921	0.9852
EDSR [41]	0.9932	0.9890	0.9823	0.9937	0.9977	0.9970	0.9962	0.9929
SRFBN [145]	0.9931	0.9887	0.9823	0.9936	0.9976	0.9969	0.9959	0.9929
RCGD [146]	0.9818	0.9647	0.9574	0.9834	0.9930	0.9927	0.9894	0.9803
RADAR [19]	0.9919	0.9857	0.9782	0.9922	0.9978	0.9969	0.9956	0.9912
DGF [17]	0.8914	0.7623	0.7877	0.9113	0.9712	0.9630	0.9479	0.8907
DJFR [18]	0.9920	0.9889	0.9814	0.9928	0.9977	0.9970	0.9968	0.9924
CoISTA [153]	0.9939	0.9885	0.9806	0.9948	0.9977	0.9970	0.9969	0.9928
CU-Net	**0.9947**	**0.9895**	**0.9832**	**0.9954**	**0.9980**	**0.9973**	**0.9972**	**0.9936**

Table 6.2: Results on the RGB/depth dataset for 4× upscaling, with the best results in bold and the second best results underlined. Note that RADAR and CoISTA are the methods introduced in Chapters 4 and 5, respectively, in this thesis.

We compare our method with several state-of-the-art methods, which include methods developed for single image super-resolution, e.g., SCN [15], EDSR [41], SRFBN [145], methods for single depth image super-resolution and RGB guided depth image super-resolution, e.g., Xie *et al.* [6], Park *et al.* [13], Ferstl *et al.* [14], Gu *et al.* [8], RCGD [146], RADAR [19], and methods for solving the general MIR problem, e.g., DGF [17], DJFR [18] and CoISTA [153]. The results of these approaches are all obtained by running the software codes provided by the authors.

Table 6.2 presents the comparison results between ours and the other state-of-the-art methods, in terms of root mean square error (RMSE) and SSIM [44]. As can be seen from this table, our CU-Net achieves the best results among all the methods. Fig. 6.4 visualizes the reconstructed depth images using different methods for 4× upscaling. From this figure, we can see that the depth images recovered by our method are quite close to the ground-truth, with clearer and sharper edges.

RGB guided multi-spectral image SR

For the task of RGB guided Multi-spectral image SR, both the training and testing images are from the the Columbia multi-spectral database [156]. We randomly select 7 images for testing and use the remaining images for training. To verify the effectiveness of our method, we compare it with the following approaches: JBF [55], GF [56], JFSM [63], SDF [64], EDSR [41], SRFBN [145], MMSR [72], DGF [17], DJFR [18] and CoISTA [153]. Table 6.3 shows the comparison results in terms of PSNR and SSIM for 4× upscaling. As we can see, our method can recover the HR multi-spectral images with high accuracy. Specifically, the average PSNR value of our method is 1.55 dB higher than the second best method DJFR [18]. Fig. 6.5 shows examples of the upsampling results using different methods.

Flash guided non-flash image denoising

We use the new flash/non-flash image dataset provided by [157] for both training and testing. We randomly select 400 flash/non-flash image pairs from the dataset [157] for training. For testing, to make the results more convincing, we select 9 images

PSNR	Chart toy	Egyptian	Feathers	Glass tiles	Jelly beans	Oil painting	Paints	Average
Bicubic	28.94	36.57	30.80	26.65	27.81	31.67	29.29	30.25
JBF [55]	32.56	38.73	33.60	27.52	30.29	32.77	31.94	32.49
GF [56]	34.09	40.24	33.60	29.46	30.90	35.03	31.73	33.58
SDF [64]	31.87	39.43	33.45	28.22	30.32	32.86	31.96	32.59
JFSM [63]	32.98	40.39	33.89	28.98	31.18	35.91	32.76	33.73
EDSR [41]	33.45	40.03	35.55	29.75	32.81	32.69	37.28	34.51
SRFBN [145]	33.43	40.04	35.53	29.53	32.97	32.68	36.06	34.32
MMSR [72]	37.55	45.21	39.68	34.52	38.87	37.50	39.31	38.95
DGF [17]	34.19	37.81	31.22	29.93	28.94	36.10	31.72	32.85
DJFR [18]	37.86	45.69	40.13	34.97	39.16	37.76	39.36	39.28
CoISTA [153]	36.58	45.91	39.62	33.99	38.92	37.26	38.40	38.67
CU-Net	**39.65**	**47.35**	**42.47**	**35.93**	**40.69**	**38.84**	**40.86**	**40.83**
SSIM	Chart toy	Egyptian	Feathers	Glass tiles	Jelly beans	Oil painting	Paints	Average
Bicubic	0.9424	0.9786	0.9562	0.9242	0.9302	0.8943	0.9493	0.9393
JBF [55]	0.9653	0.9735	0.9637	0.9341	0.9498	0.8962	0.9699	0.9504
GF [56]	0.9788	0.9796	0.9748	0.9593	0.9658	0.9441	0.9702	0.9675
SDF [64]	0.9694	0.9795	0.9650	0.9374	0.9433	0.9126	0.9655	0.9532
JFSM [63]	0.9295	0.9705	0.9425	0.9397	0.9451	0.9560	0.9430	0.9466
EDSR [41]	0.9836	0.9829	0.9875	0.9706	0.9838	0.9178	0.9914	0.9739
SRFBN [145]	0.9838	0.9822	0.9873	0.9676	0.9845	0.9182	0.9907	0.9735
MMSR [72]	0.9934	0.9919	0.9933	0.9912	0.9883	0.9797	0.9943	0.9903
DGF [17]	0.9559	0.9620	0.9336	0.9339	0.9459	0.9649	0.9680	0.9520
DJFR [18]	0.9935	0.9922	0.9939	0.9915	0.9885	0.9805	0.9944	0.9906
CoISTA [153]	0.9914	**0.9961**	0.9937	0.9907	0.9956	0.9690	0.9949	0.9902
CU-Net	**0.9960**	0.9930	**0.9965**	**0.9944**	**0.9972**	**0.9811**	**0.9971**	**0.9936**

Table 6.3: Results on the RGB/MS dataset for 4× upscaling, with the best results in bold and the second best results underlined. Note that CoISTA is the method introduced in Chapter 5 in this thesis.

σ^2=25	Minion	Towel	Elmo	Book	Tampax	Typewriter	Plant	Flower	Aloe	Average
CBM3D [21]	33.63	37.25	36.03	36.17	35.94	34.37	33.82	35.62	31.86	34.97
DnCNN [22]	34.13	37.58	36.65	35.61	36.55	34.87	34.42	36.26	32.70	35.42
DJFR [18]	31.59	36.86	34.69	34.16	34.45	32.81	31.90	34.69	30.78	33.55
MuGIF [20]	30.49	35.42	33.75	32.62	33.46	31.51	30.95	33.24	29.49	32.33
CU-Net	**34.24**	**37.99**	**36.82**	**36.86**	**36.97**	**35.07**	**34.42**	**36.40**	**32.83**	**35.73**
σ^2=50	Minion	Towel	Elmo	Book	Tampax	Typewriter	Plant	Flower	Aloe	Average
CBM3D [21]	29.94	34.36	32.50	33.15	32.37	31.30	30.17	32.33	27.98	31.57
DnCNN [22]	30.32	34.65	33.24	32.37	32.95	31.87	30.83	33.04	28.49	31.97
DJFR [18]	28.55	33.27	31.79	31.74	31.58	29.90	28.84	31.30	28.05	30.56
MuGIF [20]	26.93	32.02	30.94	29.72	30.54	28.57	27.45	29.93	28.97	29.45
CU-Net	**31.08**	**35.91**	**34.20**	**34.11**	**34.22**	**32.40**	**31.16**	**33.60**	**29.17**	**32.87**
σ^2=75	Minion	Towel	Elmo	Book	Tampax	Typewriter	Plant	Flower	Aloe	Average
CBM3D [21]	27.85	32.34	30.75	31.40	30.31	29.56	28.20	30.40	25.83	29.63
DnCNN [22]	28.33	32.75	31.16	31.83	30.74	29.97	28.46	30.83	26.25	30.04
DJFR [18]	26.65	31.77	30.63	29.54	29.86	28.85	27.54	29.25	25.56	28.85
MuGIF [20]	24.89	30.30	29.82	28.40	28.54	26.46	25.82	27.84	24.49	27.40
CU-Net	**29.12**	**34.25**	**32.55**	**32.93**	**32.19**	**30.55**	**29.17**	**31.73**	**27.03**	**31.06**

Table 6.4: Results on the Flash/non-flash dataset for image denoising, with the best results in bold and the second best results underlined.

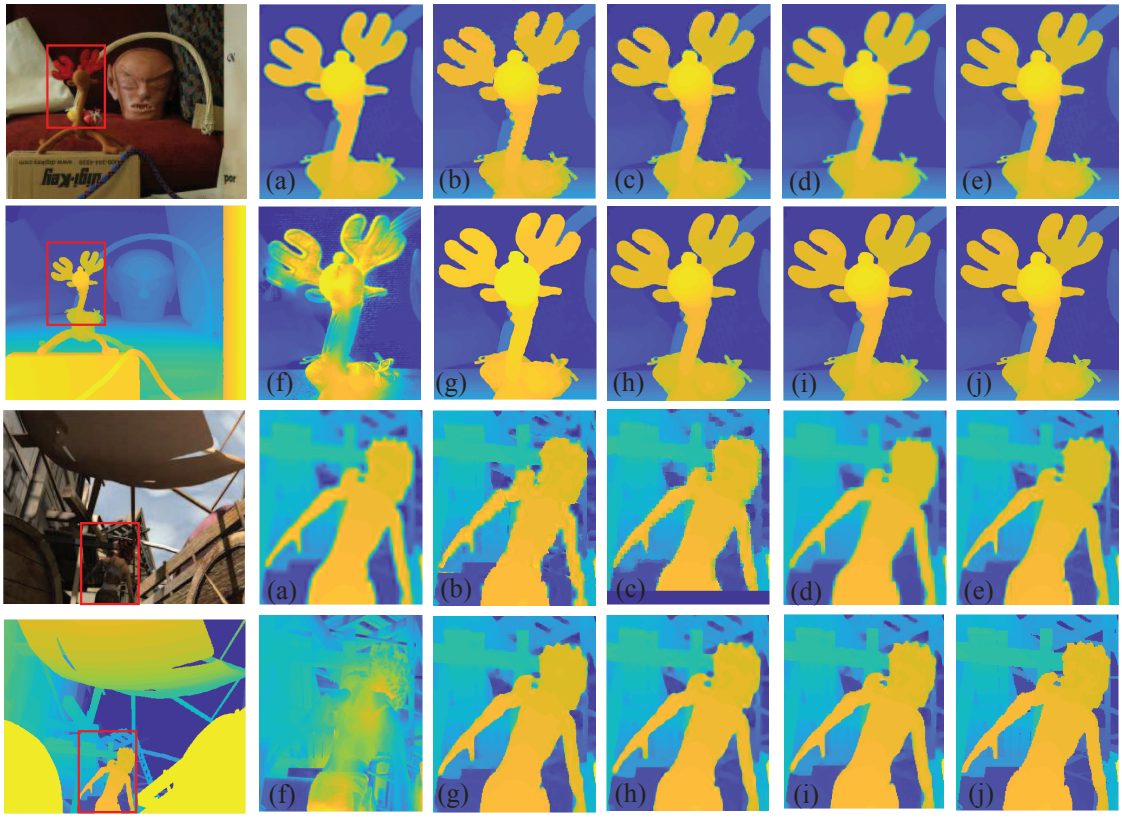


Figure 6.4: Visual comparisons of RGB guided depth SR results for 4× upscaling: (a) Bicubic, (b) Lu *et al.* [7], (c) Xie *et al.* [6], (d) Gu *et al.* [8], (e) SCN [15], (f) DGF [17], (g) DJFR [18], (h) RADAR [19], (i) Our CU-Net, (j) Ground truth.

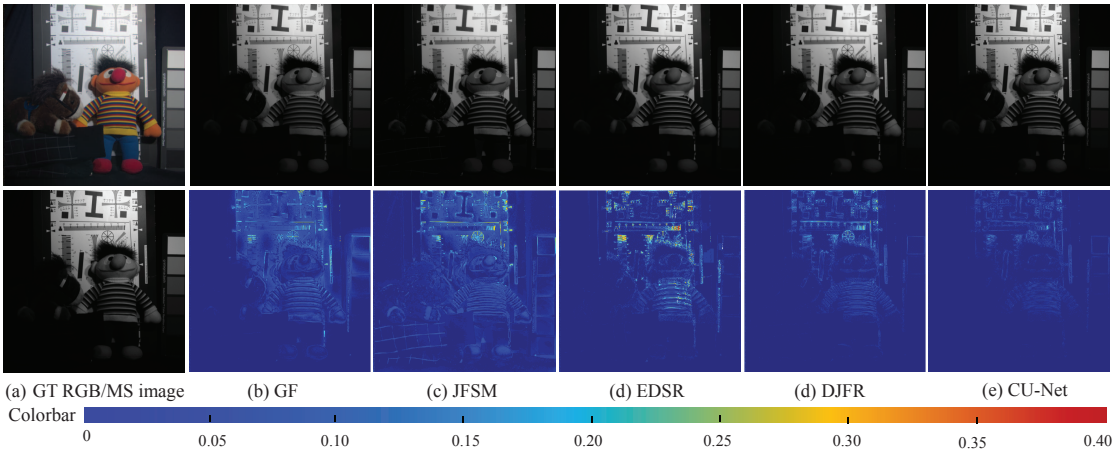


Figure 6.5: Visual comparisons of RGB guided multi-spectral image SR results for 4× upscaling. The first row shows the super-resolved images using different methods, and the second row shows the error maps between the images in the first row and the ground-truth.

from three different categories in [157], i.e., the toy, plant and object. Note that the testing images are different from the training images. The noisy non-flash images are obtained by adding white Gaussian noise to the clean images, with three different noise levels, i.e., $\sigma^2 = 25$, 50, and 75.

To verify the effectiveness of our method, we compare it with the other four state-of-the-art methods, including CBM3D [21] and DnCNN [22] which are specifically aimed at the image denoising task, DJFR [18] and MuGIF [20] which are general MIR methods. Table 6.4 shows the comparison results in terms of PSNR, in which the first four images are from the toy category, the middle four images are from the object category and the last four images are from the plant category. As we can see from this table, our method outperforms other methods for different noise levels. This is further verified in Fig. 6.6 in which we visualize the denoised non-flash images using different methods with $\sigma^2 = 75$. In the case of $\sigma^2 = 75$, the non-flash image is severely affected by noise, with most of the details removed by the heavy noise, and thus the guidance of the flash image becomes very important. As we can see in Fig. 6.6, our method is able to remove the noise effectively, and at the same time recover the fine details and sharp edges, while others either have blurred edges [21, 22] or unclear details [20]. Take the image book for example, we can clearly recover the texts which is hard to read in the noisy image, while the other methods struggle to do so. This confirms that our method is able to make full use of the information in the guidance image, even in the case that most information in the noisy non-flash image has been lost.

6.4.3 MIF related tasks

Multi-exposure image fusion

For the multi-exposure image fusion task, we aim to combine one under-exposed image with an over-exposed one to obtain a photo-realistic natural image. We use the datasets from [158] for training and testing. The paper [158] provides the images with seven exposure levels, and we choose the first and sixth levels as the

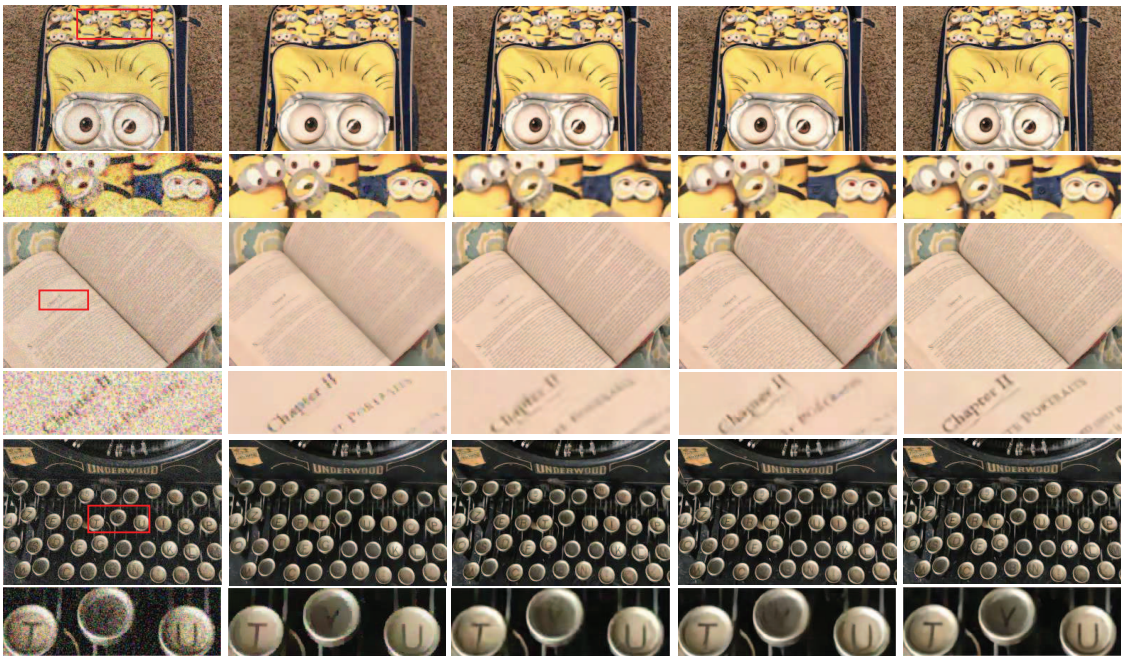


Figure 6.6: Visual comparisons of flash guided non-flash image denoising results with $\sigma^2 = 75$: (a) Noisy non-flash input image, (b) MuGIF [20], (c) CBM3D [21], (d) DnCNN [22], (e) Our CU-Net.

under-exposed and over-exposed images, respectively. Some examples of the images are shown in Fig. 6.7. As we can see from this figure, the under-exposed image is extremely dark, and the over-exposed image is extremely bright, so that many details are lost in both of them. Our method is able to detect the useful information in these two images and re-assemble them to a new image which contains all the relevant information and looks visually pleasing. We compare our method with the other two state-of-the-art approaches on multi-exposure image fusion task: the SPD-MEF [23] and MEF-OPT [24]. As we can see in Fig. 6.7, the SPD-MEF method suffers severe non-uniform brightness artifacts and contrast loss, and thus it loses many important details. For example, in the third picture of the first row, there should be a person standing between the two pillars, but the SPD-MEF method fails to discover it. The MEF-OPT method performs better than SPD-MEF, but the brightness is still not uniform across the whole images. In addition, the combined image has halo effects around the edges (the fourth figure in the third row). In contrast, the images generated by our method are with consistent and uniform brightness across the

whole image, and do not have contrast loss or halo effects.

Multi-focus image fusion

Due to the finite depth-of-field of a camera, it is difficult to capture an image with all the objects in focus. The multi-focus image fusion task aims to fuse two or more images focused on different depth planes to obtain an all-in-focus image. In this paper, we fuse one near-focus image with one far-focus image for an all-in-focus image. Examples of those images are shown in Fig. 6.8. The training dataset is from the General 100 dataset [2]. The near-focus and far-focus image pairs are generated by blurring the randomly chosen foreground and background of an image using Gaussian blurring. The testing images are from the Lytro multi-focus image dataset [159]. We compare our method with the following three state-of-the-art approaches: CSR [25], Deepfuse network [26] and Densefuse network [27]. The visual comparison results are shown in Fig. 6.8. As we can see from this figure, the combined images generated by our method are more visually pleasant than the other approaches, with every part in focus, and with clear and sharp boundary edges. In contrast, the comparison methods, CSR [25], Deepfuse network [26] and Densefuse network [27], all lead to different levels of blurring artefacts across the boundary areas, as shown in the close-ups of the toy *dog* and the *fence*.

Medical image fusion

Medical image fusion plays an important role in clinical disease diagnosis. Since different medical images usually capture different features of the organs or tissues, a good combined image can deliver more useful information, which helps the effective diagnosis of some diseases. In this paper, we choose to fuse two different magnetic resonance (MR) images, i.e., the T1 weighted MR image and T2 weighted MR image. The testing dataset is from the Whole Brain Atlas [160] from Harvard medical school. For deep learning based medical image processing, one main problem is the lack of large training dataset. Here, we use the model trained by the multi-focus image fusion task and then fine tune it using a small medical dataset which is

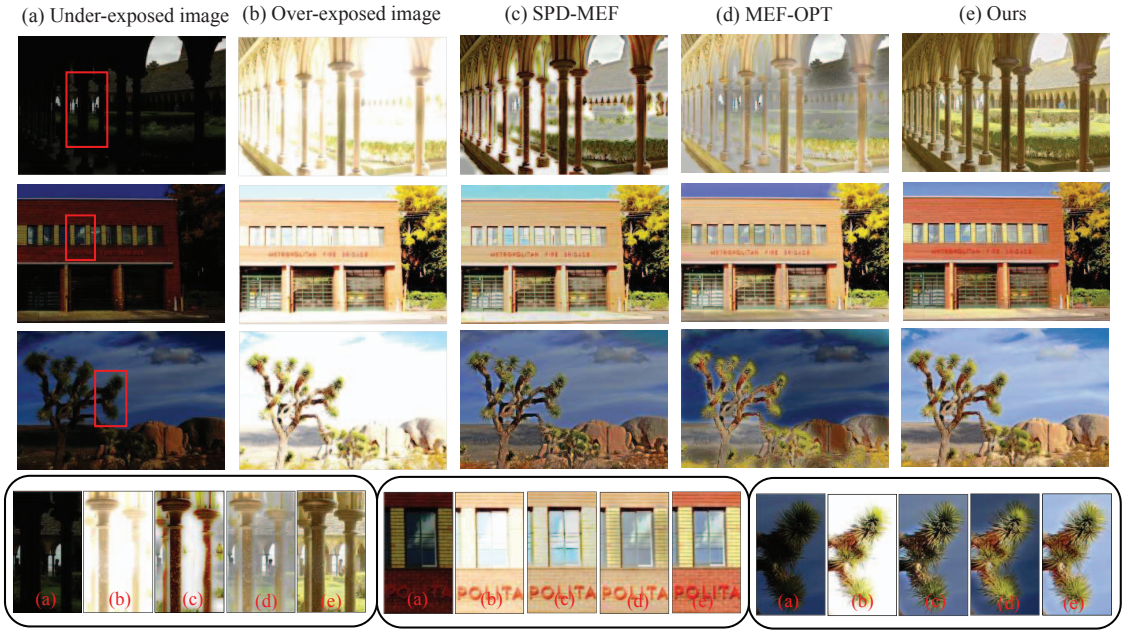


Figure 6.7: Visual comparisons of multi-exposure image fusion results: (a) input under-exposed image, (b) input over-exposed image, (c) SPD-MEF [23], (d) MEF-OPT [24], (e) our CU-Net. The last row shows the enlarged regions in the corresponding images. Better view in electronic version.

composed of 30 image pairs from the Whole Brain Atlas [160]. Fig. 6.9 shows the fused images using different methods, including CSR [25], Deepfuse network [26], Densefuse network [27] and our CU-Net. As we can see, the method CSR [25] and Deepfuse network [26] both lose the image energy and have low contrast. Our method performs well in preserving the structures in T2-weighted MR image, but we lose the brightness in T1-weighted MR image. In contrast, the method Densefuse [27] performs good in preserving the brightness in T1-weighted MR image, but loses some details in T2-weighted MR image.

6.4.4 Common and unique reconstruction visualization

As shown in Fig. 6.2, the final reconstructed image is composed of one common reconstruction and one or two unique reconstructions (one for MIR related tasks and two for MIF related tasks). To better verify the effectiveness of our method, we will visualize the common and unique components which contribute to the fi-

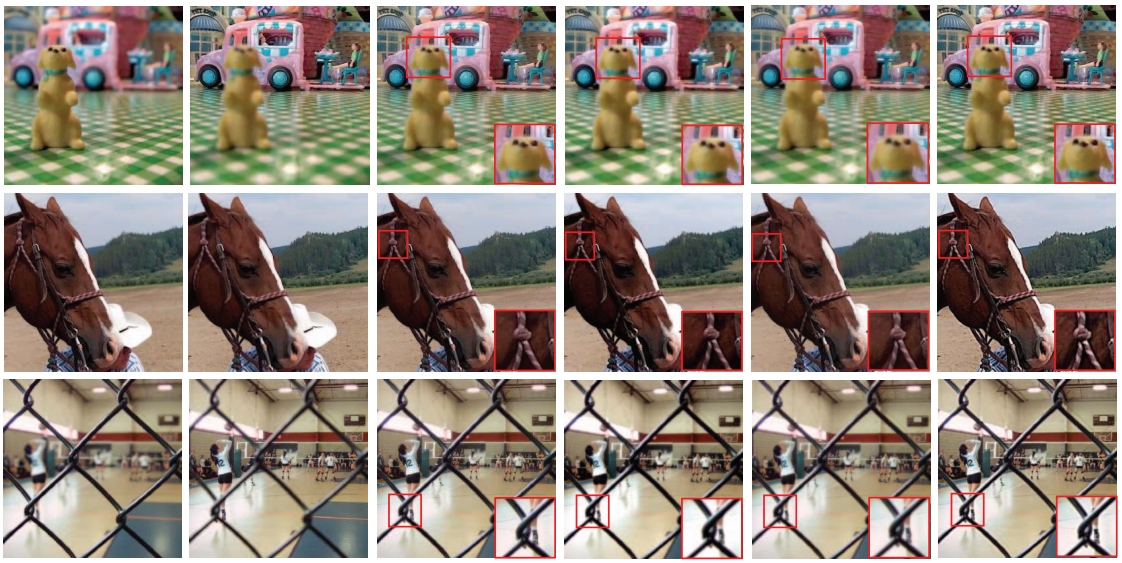
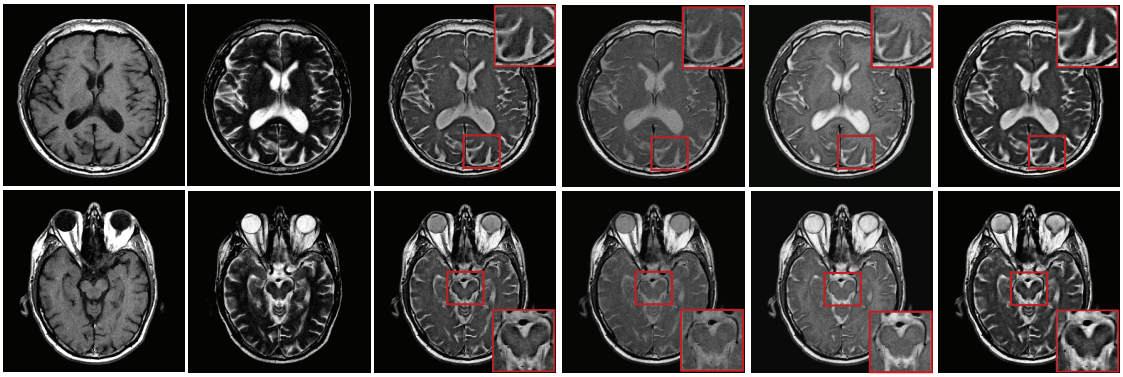


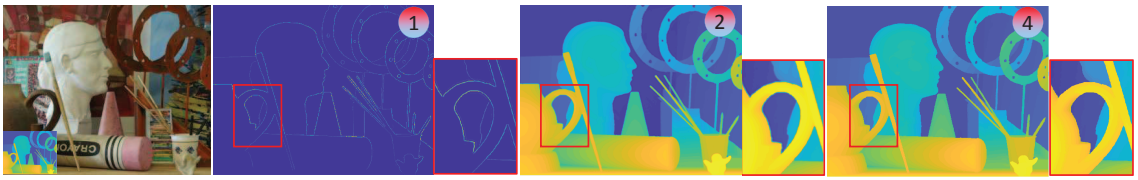
Figure 6.8: Visual comparisons of multi-focus image fusion results. (a) The near-focus image, (b) The far-focus image, (c) CSR [25], (d) Deepfuse network [26], (e) Densefuse network [27], (f) our CU-Net. Better view in electronic version.

nal reconstruction, i.e., the Point 1, 2, 3, and 4 in Fig. 6.2. For the MIR tasks, we take the RGB guided depth SR as example. Fig. 6.10 shows the common and unique reconstructed images for the RGB guided depth SR task. As we can see, the common reconstruction only contains the high-resolution edge information, which is consistent with the fact that the only common element between the RGB and depth images is the edge details. The depth information is entirely provided by the unique reconstruction, which is reasonable because the RGB image does not contain any depth information. However, in the unique reconstruction, depth discontinuities are still blurred. In the final reconstructed image, the blurring effect is successfully eliminated due to the addition of the common reconstruction which has sharp high-resolution edges. For the MIF related tasks, Fig. 6.11 visualizes the three components of the final fused image for the multi-exposure fusion task. We can clearly see from this figure that the common reconstruction contains the common parts shared between the under-exposed and over-exposed images, while the two unique reconstructions preserve the unique features in each image, e.g., see the yellow and red marks in the images. By adding these three components together, we can obtain the final fused image, which contains both the common feature between



(a) Source image 1 (b) Source image 2 (c) CSR (d) DeepFuse (e) DenseFuse (d) Ours

Figure 6.9: Visual comparisons of medical image fusion results. (a) T1-weighted MR image, (b) T2-weighted MR image, (c) CSR [25], (d) Deepfuse network [26], (e) Densefuse network [27], (f) our CU-Net. Better view in electronic version.



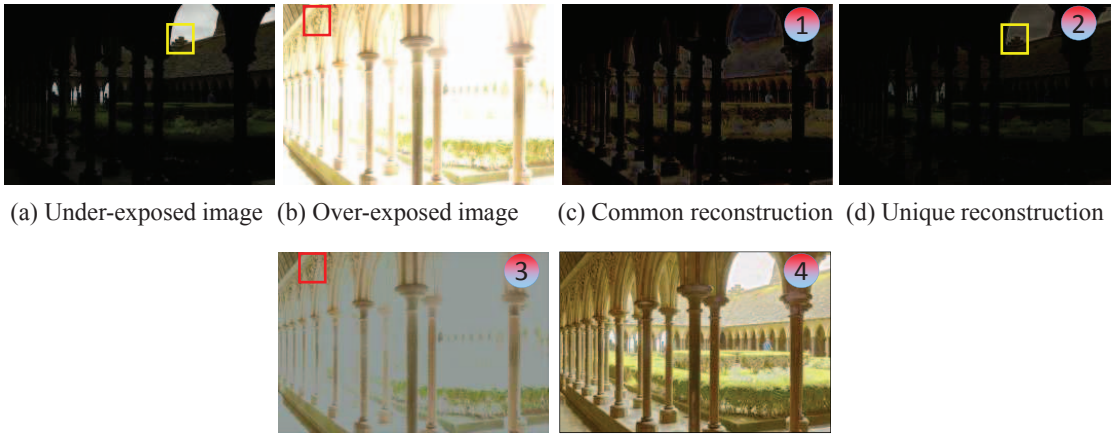
(a) RGB/depth images (b) Common reconstruction part (c) Unique reconstruction part (d) Final reconstruction

Figure 6.10: Visualization of the reconstructed image of different parts in the CU-Net for the RGB guided depth SR task. (b) is the common reconstruction which corresponds to Point 1 in Fig. 6.2, (c) is the unique reconstruction, which corresponds to Point 2 in Fig. 6.2, and (d) is the final reconstruction, which is the sum of (b) and (c), corresponding to the Point 4 in Fig. 6.2. Better view in electronic version.

the two extremely exposed images and the unique feature of each single image.

6.4.5 Ablation study

Residual architecture. As shown in Fig. 6.3, the UFEM and CFPM modules have residual structures. In the first ablation study, we explore the importance of this residual architecture on the reconstruction performance. To do this, we firstly remove the two residual arrows in the modules, which makes UFEM and CFPM pure feed-forward CNN modules. We then re-train the network using the same training strategy. Table 6.5 shows the results on the RGB guided depth image SR task with and without the residual structures, respectively. As we can see, the residual architecture indeed helps increase the reconstruction accuracy.



(a) Under-exposed image (b) Over-exposed image (c) Common reconstruction (d) Unique reconstruction

(e) Unique reconstruction (f) Final reconstruction

Figure 6.11: Visualization of the reconstructed image of different parts in the CU-Net for the multi-exposure image fusion task. (c) is the common reconstruction which corresponds to Point 1 in Fig. 6.2, (d) and (e) are the unique reconstructions, which correspond to Point 2 and 3 in Fig. 6.2, respectively, and (f) is the final reconstruction, which is the sum of (c), (d) and (e). Better view in electronic version.

Filter size. We further analyse the effects of filter size $s \times s$ on the reconstruction performance. Table 6.6 shows the results on the RGB guided depth SR when we gradually increase the filter size s from 3 to 9. We can see that the reconstruction performance is improved with larger s. However, when the filter is too large, e.g., $s = 9$, the reconstruction accuracy decreases. The possible reason for this phenomenon is that a very large filter support may overlook local details, and this leads to less accurate reconstructions. In this paper, we choose to use a filter size of 8×8.

Network depth. The network depth also plays an important role in improving the reconstruction performance. Here, we use the number of LCSC blocks to indicate the network depth. Table 6.7 presents the results with different numbers of LCSC blocks. We can see that the reconstruction accuracy improves with the depth of the network, however, the depth of the network also increases the model size and training complexity. In this paper, we use 4 LCSC blocks in each module, which represents a good trade-off between reconstruction accuracy and training complexity.

Architecture	RMSE	PSNR (dB)	SSIM
Residual	1.88	44.65	0.9936
Non-residual	2.03	43.22	0.9898

Table 6.5: Importance of the residual module architecture.

Filter size	$s = 3$	$s = 4$	$s = 5$	$s = 6$	$s = 7$	$s = 8$	$s = 9$
RMSE	2.13	2.05	1.96	1.93	1.90	1.88	1.92
PSNR/ dB	43.45	43.94	44.24	44.39	44.51	44.65	44.44
SSIM	0.9915	0.9921	0.9931	0.9934	0.9935	0.9936	0.9934

Table 6.6: Effects of the filter size.

Network depth	1	2	3	4	5	6
RMSE	2.07	1.98	1.94	1.88	1.84	1.81
PSNR/ dB	43.88	44.18	44.35	44.65	44.90	45.12
SSIM	0.9920	0.9930	0.9933	0.9936	0.9938	0.9940
Model size /KB	66	99	132	165	198	231

Table 6.7: Effects of the network depth in each module.

Task	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Image resolution	1320×1080	512×512	900×600	900×600	520×520	256×256
Running time	0.85s	0.24s	0.38s	0.42s	0.27s	0.09s

Table 6.8: The average running time (in seconds) of our method on various tasks.

6.4.6 Running speed

For the real-time applications, the computational complexity of a method is an important factor to be considered. Our method is a deep learning based method using a feed-forward network architecture, which has the advantage of fast running speed. In Table 6.8, we present the running time of our method for the six applications presented in this paper. The task 1 to task 6 indicate the RGB guided depth image SR, RGB guided MS image SR, flash guided non-flash image denoising, multi-exposure image fusion, multi-focus image fusion, and medical image fusion, respectively. The time is recorded by implementing the experiments on a PC with an GEFORCE GTX 1080 Ti GPU.

6.5 Summary

In this chapter, we addressed the general multi-modal image restoration and image fusion problems by proposing a novel and flexible CNN architecture, named Common and Unique information splitting network (CU-Net). Different from other approaches, our network architecture is derived from a new proposed multi-modal convolutional sparse coding (MCSC) model, which makes each part of our network interpretable. To verify the effectiveness of our CU-Net, we conduct exhaustive experiments on six different multi-modal image restoration and fusion tasks. We also make a comprehensive ablation study to explore the contribution of each component in the network and the effects of some critical network hyper-parameters, such as filter size and network depth. The experimental results show that our method outperforms other state-of-the-art methods in all the tasks considered, with a small model size and fast running speed.

Chapter 7

Conclusion

7.1 Summary of Thesis Achievements

In this thesis, we considered the problems of inferring a high-quality image from either a single low-quality image or with the guidance of another image from a different modality. Firstly, we presented a wavelet domain style transfer algorithm to achieve a good trade-off between the perceptual and objective image quality in single image super-resolution. Next, we considered the problem of inferring a high-resolution depth image from its corresponding low-resolution depth image with the guidance of a high-resolution RGB image. Then, towards the general multi-modal image super-resolution problems, we proposed a joint multi-modal dictionary learning algorithm to model the dependencies across modalities and then establish a deep neural network based on this modelling. Finally, we moved to solve the more general multi-modal image restoration and image fusion problems, by firstly developing a multi-modal convolutional sparse coding model and then by constructing a common and unique information splitting network.

In Chapter 3, we presented a wavelet domain style transfer (WDST) algorithm to achieve a good perception-distortion trade-off in single image super-resolution (SISR). One challenging problem for SISR is that there exists a fundamental trade-

off between the perceptual and objective image quality, which leads to the following problem: the methods which aim to achieve high objective quality are usually with low perceptual quality, and the methods which aim to achieve high perceptual quality are often with low objective quality. In order to achieve both high perceptual and high objective image quality, we need to split the elements affecting the perceptual quality from those affecting the objective quality. One of the main contributions in our method is that we find the wavelet transform to be effective in splitting those elements. Specifically, after applying the wavelet transform on an image, the low-frequency sub-band contributes more to the objective image quality, while the high-frequency sub-bands contribute more to the perceptual quality. Based on this observation, we can focus on improving the objective quality of the low-frequency sub-band, and enhancing the perceptual quality of the high-frequency sub-bands. Since there are various existing SISR methods, we verified the effectiveness of our observation by combining two super-resolved images generated by two existing SISR methods, i.e., one image is with high objective quality and the other is with high perceptual quality. After applying the stationary wavelet transform (SWT), we further improve the objective quality by applying a VDSR network to the low-frequency sub-band. For the high-frequency sub-bands, we employed the WDST algorithm to improve their perceptual quality. These new sub-bands are then re-combined into an image using inverse SWT.

In Chapter 4, we presented a RGB guided depth image super-resolution method named RADAR through combining the finite rate of innovation (FRI) theory with a triple dictionary learning algorithm. We find that the rows and columns of a depth image can be modelled as piecewise polynomials. This good characteristic allows us to upscale the depth image using the FRI theory. We developed two FRI-based depth upscaling frameworks to cope with both the clean and noisy depth images. In order to further improve the quality of depth images, we use the corresponding RGB image as side information, and a triple dictionary learning algorithm is used to model the relationships between the depth and RGB images. Since these two

images share some common features but also have unique features, we represent each modality with one common and one unique dictionary with the constraint that the common dictionary should share the same sparse representations. Furthermore, to speed up the super-resolution process, we introduced a new projection-based rapid upscaling (PRU) algorithm which classify the training samples into various clusters and then pre-calculate the projections from the joint low-resolution depth and RGB pairs to the high-resolution depth. The combined use of a model-based algorithm (FRI theory) and a data-driven algorithm (PRU algorithm) makes our method resilient to different blurring kernels and achieves at the same time high reconstruction accuracy.

In Chapter 5, we presented a deep coupled ISTA network to solve the general multi-modal image super-resolution (MISR) problem. We firstly proposed a joint multi-modal dictionary learning (JMDL) model to correlate different modalities. This model jointly learns three dictionaries for the LR and HR versions of one modality and the HR version of the guided modality, respectively. In addition, two transform matrices are learned to correlate the three dictionaries in the sparse representation space. Based on the JMDL model, we designed a new deep coupled neural network for solving the MISR problem, by unfolding the iterative shrinkage and thresholding algorithm (ISTA). Since the network initialization plays an important role in the network training and testing performance, we developed a new layer-wise optimization algorithm (LOA) to initialize the network parameters. Compared to other widely used initialization algorithms, the LOA is demonstrated to optimize the training loss function better and to improve reconstruction accuracy. We tested the performance of the proposed method on various multi-modal datasets, including RGB/depth, RGB/multi-spectral (MS), and RGB/near-infrared (NIR) images, which validates the effectiveness and universality of our method

In Chapter 6, we presented a deep convolutional neural network to solve the general multi-modal image restoration (MIR) and multi-modal image fusion (MIF) problems. Different from other methods based on deep learning, our network architecture

is designed by drawing inspirations from a new proposed multi-modal convolutional sparse coding (MCSC) model. The MCSC model has two variations to solve the general multi-modal image restoration (MIR) and multi-modal image fusion (MIF) problems, respectively. In this model, we represent each modality with two convolutional dictionaries: one for common feature representation and the other for unique feature representation. Based on the MCSC model, we design a deep convolutional neural network, named Common and Unique splitting network (CU-Net).The CU-Net is composed of three modules, i.e., the unique feature extraction module (UFEM), common feature preservation module (CFPM), and image reconstruction module (IRM). The architecture of each module is derived from the corresponding part in the MCSC model, which consists of several learned convolutional sparse coding (LCSC) blocks. The CU-Net was demonstrated to automatically split the common features from the unique features between the target and guided modality, which is beneficial to both MIR and MIF tasks. We tested the performance of the proposed method on various MIR and MIF tasks, such as RGB guided depth image super-resolution, flash guided non-flash image denoising, multi-focus and multiexposure image fusion. The numerical results validated the effectiveness, flexibility, and universality of our method.

7.2 Future Work

7.2.1 Fast wavelet domains style transfer

Like the conventional style transfer work [3], we need many iterations to solve the optimization problem related to the style transfer. This is time-consuming, i.e., around 60 seconds for each sub-band. Inspired by the real-time artistic style transfer work [45], our future aim is to train a feed-forward network to predict the fused sub-band, so that the computational complexity can be significantly decreased. There are also some recent works which aim to get a closed-form solution for style transfer [161, 162] using whitening and coloring transforms, and this may also be a future

research avenue.

7.2.2 Perceptual based multi-modal image restoration

In this thesis, all our multi-modal image restoration methods aim to minimize the mean squared error (MSE) which is related with the objective quality of the restored image. As pointed out by many papers, the MSE is not always consistent with the human visual system (HVS). Recently, we can see many image restoration methods aiming to optimize the perceptual image quality, e.g., image super-resolution, image enhancement, image inpainting. However, there is still not much work on improving the perceptual quality in the multi-modal image restoration area. The generative adversarial network (GAN) has been widely used in single image super-resolution to improve the perceptual quality of super-resolved images [3, 48]. Thus, our future work will be to employ a GAN to enhance the perceptual quality of restored images for multi-modal image restoration.

7.2.3 Extension from image to video processing

In this thesis, either the single image super-resolution or the multi-modal image restoration tasks focused on image processing. One possible future research direction is to apply the algorithms to video sequences. Compared to images, processing the video sequences is more challenging due to the temporal constraint between adjacent frames. We can see many successful extensions from image to video processing, such as video super-resolution, video inpainting. The main difference is that the network architecture should be adjusted to guarantee the temporal consistency among adjacent frames. In addition, there is redundant information among adjacent frames which can be used to improve the reconstruction performance.

7.2.4 Joint multi-modal image compression

We have explored many interesting properties for multi-modal images in this thesis. One of them is that the images from different modalities share some common features, and we have designed a network to effectively extract these common features among different modalities. Therefore, when we compress this kinds of multi-modal images, instead of compressing each of them individually, we can jointly compress them by dealing with the common elements first and then the residual unique elements. In this way, bits can be saved since we only need to compress the common part once.

Appendix A

Chapter 4

A.1 FRI reconstruction on 1-D signal

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

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

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 $\varphi(t)$ [163]. Moreover, it is possible to show that there exist coefficients $c_{m,n}$ such that:

$$\sum_{n \in \mathbb{Z}} c_{m,n} \varphi(t-n) \simeq e^{jw_0 mt}. \tag{A.2}$$

Under some mild assumptions, the approximation in Eq. (A.2) is exact, otherwise the error is small [29]. 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 [118]. 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. (A.2)

$$s_m = \sum_n c_{m,n} y_n, \quad m \in \{0, 1, 2, ...L\}. \tag{A.3}$$

Taking Eq. (A.1) in to Eq. (A.3), we obtain

$$\begin{aligned} s_m &\overset{(a)}{=} \langle x(t), \sum_n c_{m,n} \varphi(t-n) \rangle \\ &\overset{(b)}{=} \int_{-\infty}^{+\infty} \sum_{k=0}^{K-1} a_k \delta(t - t_k) e^{jw_0 mt} dt \\ &\overset{(c)}{=} \sum_{k=0}^{K-1} a_k u_k^m, \quad m \in \{0, 1, 2, ...L\}, \end{aligned} \tag{A.4}$$

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. (A.2), and (c) by setting $u_k = e^{jw_0 t_k}$.

We now have the values of s_m from Eq. (A.3) and we hope to figure out the values of a_k and u_k in Eq. (A.4). To this end, following the Prony's method [164], 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,

$$H(z) = \sum_{m=0}^{K} h_m z^{-m} = \prod_{k=0}^{K-1} (1 - u_k z^{-1}). \tag{A.5}$$

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,

$$\begin{aligned} h_m * s_m &= \sum_{l=0}^{K} h_l s_{m-l} \\ &= \sum_{l=0}^{K} h_l \sum_{k=0}^{K-1} a_k u_k^{m-l} \\ &= \underbrace{\sum_{l=0}^{K} h_l u_k^{-l}}_{H(u_k)=0} \sum_{k=0}^{K-1} a_k u_k^m = 0 \end{aligned} \tag{A.6}$$

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 $h_0 = 1$,

$$\begin{bmatrix} s_{K-1} & s_{K-2} & \dots & s_0 \\ s_K & s_{K-1} & \dots & s_1 \\ \vdots & \vdots & \ddots & \vdots \\ s_{2K-2} & s_{2K-3} & \dots & s_{K-1} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_K \end{bmatrix} = - \begin{bmatrix} s_K \\ s_{K+1} \\ \vdots \\ s_{2K-1} \end{bmatrix}. \tag{A.7}$$

Eq. (A.7) 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. (A.4) 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,

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ u_0 & u_1 & \dots & u_{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_0^{K-1} & u_1^{K-1} & \dots & u_{K-1}^{K-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{K-1} \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{K-1} \end{bmatrix}. \tag{A.8}$$

Eq. (A.8) 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.

Appendix B

Chapter 6

B.1 Learned Convolutional Sparse Coding [1]

The typical convolutional sparse coding problem with ℓ_1 constraint is formulated as follows:

$$\underset{\{\boldsymbol{u}_k\}}{\text{Argmin}} \frac{1}{2} \left\| \boldsymbol{x} - \sum_k \boldsymbol{d}_k * \boldsymbol{u}_k \right\|_2^2 + \lambda \sum_k \|\boldsymbol{u}_k\|_1 , \tag{B.1}$$

where $\boldsymbol{x} \in \mathbb{R}^{n \times n}$ is the input image, $\{\boldsymbol{d}^k\}_{k=1}^K \in \mathbb{R}^{s \times s}$ is a set of known filters, and $\{\boldsymbol{u}^k\}_{k=1}^K \in \mathbb{R}^{n \times n}$ is the filter response to be computed. Since the convolutional operation is linear, we can construct a Toeplitz matrix $\boldsymbol{D}_k \in \mathbb{R}^{n^2 \times n^2}$ to make $\boldsymbol{d}_k * \boldsymbol{u}_k = \boldsymbol{D}_k \boldsymbol{u}'_k$ where $\boldsymbol{u}'_k \in \mathbb{R}^{n^2}$ is the vectorized $\boldsymbol{u}_k$. Then, we can turn (B.1) to the following:

$$\underset{\{\boldsymbol{u}'_k\}}{\text{Argmin}} \frac{1}{2} \left\| \boldsymbol{x}' - \sum_k \boldsymbol{D}_k \boldsymbol{u}'_k \right\|_2^2 + \lambda \sum_k \|\boldsymbol{u}'_k\|_1 , \tag{B.2}$$

where $\boldsymbol{x}' \in \mathbb{R}^{n^2}$ is the vectorized $\boldsymbol{x}$. By concatenating all the $\boldsymbol{D}_k$, we can have $\boldsymbol{D} \in \mathbb{R}^{n^2 \times Kn^2} = \{\boldsymbol{D_1 D_2} \cdots \boldsymbol{D_K}\}$. By putting all the $\boldsymbol{u}'_k$ in a column, we can have $\boldsymbol{u} \in \mathbb{R}^{Kn^2} = \{\boldsymbol{u}'_1; \boldsymbol{u}'_2; \cdots \boldsymbol{u}'_K\}$. Then, Eq. (B.2) can be turned to a traditional sparse coding problem:

$$\underset{\boldsymbol{u}}{\text{Argmin}} \frac{1}{2} \|\boldsymbol{x}' - \boldsymbol{D}\boldsymbol{u}\|_2^2 + \lambda \|\boldsymbol{u}\|_1 . \tag{B.3}$$

The Eq. (B.3) can be solved using the iterated shrinkage and thresholding algorithm (ISTA) [85], which gives us the following iterative solution:

$$\boldsymbol{u}_{j+1} = S_\lambda(\boldsymbol{u}_j + \boldsymbol{D}^T(\boldsymbol{x}' - \boldsymbol{D}\boldsymbol{u}_j)). \tag{B.4}$$

By replacing the matrix multiplication in Eq. (B.4) with convolutional operations, we can have the following:

$$\boldsymbol{U}_{j+1} = S_\lambda(\boldsymbol{U}_j - \boldsymbol{E} * \boldsymbol{F} * \boldsymbol{U}_j + \boldsymbol{E} * \boldsymbol{x}), \tag{B.5}$$

where $\boldsymbol{E} \in \mathbb{R}^{s\times s\times K}$ and $\boldsymbol{F} \in \mathbb{R}^{s\times s\times K}$ are the two sets of filters which lead to the Toeplitz matrices $\boldsymbol{D^T}$ and $\boldsymbol{D}$, respectively, and $\boldsymbol{U} \in \mathbb{R}^{n\times n\times K}$ is the stack of $\{\boldsymbol{u}^k\}_{k=1}^K$ to be learned in Eq. (B.1). In the learned convolutional sparse coding (LCSC), the $\boldsymbol{E}$ and $\boldsymbol{F}$ are learnable as convolutional layers in a deep network.

Bibliography

[1] H. Sreter and R. Giryes, "Learned convolutional sparse coding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2191–2195, IEEE, 2018.

[2] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European conference on computer vision*, pp. 391–407, Springer, 2016.

[3] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4681–4690, 2017.

[4] S. Martull, M. Peris, and K. Fukui, "Realistic CG stereo image dataset with ground truth disparity maps," in *ICPR workshop*, vol. 111, pp. 117–118, 2012.

[5] 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*, pp. 993–1000, 2013.

[6] 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.

[7] J. Lu and D. Forsyth, "Sparse depth super resolution," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2245–2253, 2015.

[8] S. Gu, W. Zuo, S. Guo, Y. Chen, C. Chen, and L. Zhang, "Learning dynamic guidance for depth image enhancement," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3769–3778, 2017.

[9] 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*, pp. 360–376, Springer, 2016.

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

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

[12] 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.

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

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

[15] D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, and T. S. Huang, "Robust single image super-resolution via deep networks with sparse prior," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3194–3207, 2016.

[16] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1646–1654, 2016.

[17] H. Wu, S. Zheng, J. Zhang, and K. Huang, "Fast end-to-end trainable guided filter," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1838–1847, 2018.

[18] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Joint image filtering with deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1909–1923, 2019.

[19] X. Deng, P. Song, M. R. Rodrigues, and P. L. Dragotti, "Radar: Robust algorithm for depth image super resolution based on fri theory and multimodal dictionary learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[20] X. Guo, Y. Li, J. Ma, and H. Ling, "Mutually guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[21] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image restoration by sparse 3d transform-domain collaborative filtering," in *Image Processing: Algorithms and Systems VI*, vol. 6812, p. 681207, International Society for Optics and Photonics, 2008.

[22] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[23] K. Ma, H. Li, H. Yong, Z. Wang, D. Meng, and L. Zhang, "Robust multi-exposure image fusion: a structural patch decomposition approach," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2519–2532, 2017.

[24] K. Ma, Z. Duanmu, H. Yeganeh, and Z. Wang, "Multi-exposure image fusion by optimizing a structural similarity index," *IEEE Transactions on Computational Imaging*, vol. 4, no. 1, pp. 60–72, 2018.

[25] Y. Liu, X. Chen, R. K. Ward, and Z. J. Wang, "Image fusion with convolutional sparse representation," *IEEE Signal Processing Letters*, vol. 23, no. 12, pp. 1882–1886, 2016.

[26] K. R. Prabhakar, V. S. Srikar, and R. V. Babu, "Deepfuse: A deep unsupervised approach for exposure fusion with extreme exposure image pairs.," in *International Conference on Computer Vision (ICCV)*, pp. 4724–4732, 2017.

[27] H. Li and X.-J. Wu, "Densefuse: A fusion approach to infrared and visible images," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2614–2623, 2018.

[28] 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.

[29] 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.

[30] P. Song, X. Deng, J. F. Mota, N. Deligiannis, P.-L. Dragotti, and M. Rodrigues, "Multimodal image super-resolution via joint sparse representations induced by coupled dictionaries," *IEEE Transactions on Computational Imaging*, 2019.

[31] 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.

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

[33] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3467–3478, 2012.

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

[35] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, and L. Zhang, "Convolutional sparse coding for image super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1823–1831, 2015.

[36] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[37] 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*, pp. 184–199, Springer, 2014.

[38] J. Kim, J. Kwon Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1637–1645, 2016.

[39] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint*, 2016.

[40] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proceedings of the IEEE international conference on computer vision*, pp. 4539–4547, 2017.

[41] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Conference on Computer Vision and Pattern Recognition workshops (CVPRW)*, pp. 136–144, 2017.

[42] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1664–1673, 2018.

[43] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 286–301, 2018.

[44] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[45] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, pp. 694–711, Springer, 2016.

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[48] M. S. Sajjadi, B. Schölkopf, and M. Hirsch, "Enhancenet: Single image super-resolution through automated texture synthesis," in *International Conference on Computer Vision (ICCV)*, pp. 4501–4510, IEEE, 2017.

[49] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *European Conference on Computer Vision (ECCV)*, pp. 63–79, Springer, 2018.

[50] R. Mechrez, I. Talmi, F. Shama, and L. Zelnik-Manor, "Maintaining natural image statistics with the contextual loss," in *Asian Conference on Computer Vision*, pp. 427–443, Springer, 2018.

[51] W. Zhang, Y. Liu, C. Dong, and Y. Qiao, "Ranksrgan: Generative adversarial networks with ranker for image super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3096–3105, 2019.

[52] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6228–6237, 2018.

[53] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, "Learning a no-reference quality metric for single-image super-resolution," *Computer Vision and Image Understanding*, vol. 158, pp. 1–16, 2017.

[54] J.-H. Choi, J.-H. Kim, M. Cheon, and J.-S. Lee, "Deep learning-based image super-resolution considering quantitative and perceptual quality," *arXiv preprint arXiv:1809.04789*, 2018.

[55] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Transactions on Graphics (ToG)*, vol. 26, no. 3, p. 96, 2007.

[56] K. He, J. Sun, and X. Tang, "Guided image filtering," in *European Conference on Computer Vision*, pp. 1–14, Springer, 2010.

[57] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond," in *International Conference on Computer Vision (ICCV)*, pp. 49–56, 2013.

[58] F. Kou, W. Chen, C. Wen, and Z. Li, "Gradient domain guided image filtering," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4528–4539, 2015.

[59] R. J. Jevnisek and S. Avidan, "Co-occurrence filter," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3184–3192, 2017.

[60] T. Brox, O. Kleinschmidt, and D. Cremers, "Efficient nonlocal means for denoising of textural patterns," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1083–1092, 2008.

[61] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *European conference on computer vision*, pp. 815–830, Springer, 2014.

[62] X. Shen, C. Zhou, L. Xu, and J. Jia, "Mutual-structure for joint filtering," in *International Conference on Computer Vision (ICCV)*, pp. 3406–3414, 2015.

[63] X. Shen, Q. Yan, L. Xu, L. Ma, and J. Jia, "Multispectral joint image restoration via optimizing a scale map," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2518–2530, 2015.

[64] B. Ham, M. Cho, and J. Ponce, "Robust guided image filtering using nonconvex potentials," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 192–207, 2018.

[65] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep joint image filtering," in *European Conference on Computer Vision*, pp. 154–169, Springer, 2016.

[66] B. Kim, J. Ponce, and B. Ham, "Deformable kernel networks for joint image filtering," 2018.

[67] J. Pan, J. Dong, J. S. Ren, L. Lin, J. Tang, and M.-H. Yang, "Spatially variant linear representation models for joint filtering," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1711, 2019.

[68] I. Marivani, E. Tsiligianni, B. Cornelis, and N. Deligiannis, "Learned multimodal convolutional sparse coding for guided image super-resolution," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2891–2895, IEEE, 2019.

[69] C. Guo, C. Li, J. Guo, R. Cong, H. Fu, and P. Han, "Hierarchical features driven residual learning for depth map super-resolution," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2545–2557, 2018.

[70] X. Song, Y. Dai, and X. Qin, "Deeply supervised depth map super-resolution as novel view synthesis," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[71] Y. Wen, B. Sheng, P. Li, W. Lin, and D. D. Feng, "Deep color guided coarse-to-fine convolutional network cascade for depth image super-resolution," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 994–1006, 2019.

[72] F. Lahoud, R. Zhou, and S. Susstrunk, "Multi-modal spectral image super-resolution," in *European Conference on Computer Vision (ECCV)*, 2018.

[73] Z. Shi, C. Chen, Z. Xiong, D. Liu, Z.-J. Zha, and F. Wu, "Deep residual attention network for spectral image super-resolution," in *European Conference on Computer Vision (ECCV)*, pp. 0–0, 2018.

[74] S. Lohit, D. Liu, H. Mansour, and P. T. Boufounos, "Unrolled projected gradient descent for multi-spectral image fusion," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7725–7729, IEEE, 2019.

[75] S. Li, X. Kang, L. Fang, J. Hu, and H. Yin, "Pixel-level image fusion: A survey of the state of the art," *Information Fusion*, vol. 33, pp. 100–112, 2017.

[76] G. Pajares and J. M. De La Cruz, "A wavelet-based image fusion tutorial," *Pattern recognition*, vol. 37, no. 9, pp. 1855–1872, 2004.

[77] L. Cao, L. Jin, H. Tao, G. Li, Z. Zhuang, and Y. Zhang, "Multi-focus image fusion based on spatial frequency in discrete cosine transform domain," *IEEE Signal Processing Letters*, vol. 22, no. 2, pp. 220–224, 2014.

[78] Q. Zhang and B.-l. Guo, "Multifocus image fusion using the nonsubsampled contourlet transform," *Signal processing*, vol. 89, no. 7, pp. 1334–1346, 2009.

[79] B. Yang and S. Li, "Pixel-level image fusion with simultaneous orthogonal matching pursuit," *Information fusion*, vol. 13, no. 1, pp. 10–19, 2012.

[80] Q. Wei, J. Bioucas-Dias, N. Dobigeon, and J.-Y. Tourneret, "Hyperspectral and multispectral image fusion based on a sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, pp. 3658–3668, 2015.

[81] J. Hu and S. Li, "The multiscale directional bilateral filter and its application to multisensor image fusion," *Information Fusion*, vol. 13, no. 3, pp. 196–206, 2012.

[82] J. H. Jang, Y. Bae, and J. B. Ra, "Contrast-enhanced fusion of multisensor images using subband-decomposed multiscale retinex," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3479–3490, 2012.

[83] Y. Liu, X. Chen, H. Peng, and Z. Wang, "Multi-focus image fusion with a deep convolutional neural network," *Information Fusion*, vol. 36, pp. 191–207, 2017.

[84] J. Ma, W. Yu, P. Liang, C. Li, and J. Jiang, "Fusiongan: A generative adversarial network for infrared and visible image fusion," *Information Fusion*, vol. 48, pp. 11–26, 2019.

[85] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on pure and applied mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[86] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.

[87] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[88] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 399–406, Omnipress, 2010.

[89] P. Sprechmann, A. Bronstein, and G. Sapiro, "Learning efficient structured sparse models," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 219–226, Omnipress, 2012.

[90] Z. Wang, Q. Ling, and T. Huang, "Learning deep l0 encoders," in *AAAI Conference on Artificial Intelligence*, pp. 2194–2200, 2016.

[91] Z. Joey, D. Kai, D. Jiawei, P. Xi, Y. Hao, J. P. Sinno, T. Ivor, L. Y, Q. Zheng, and S. M. G. Rick, "SC2Net: Sparse lstms for sparse coding," in *AAAI Conference on Artificial Intelligence*, AAAI, 2018.

[92] J.-T. Chien and C.-H. Lee, "Deep unfolding for topic models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 318–331, 2018.

[93] M. Borgerding, P. Schniter, and S. Rangan, "Amp-inspired deep networks for sparse linear inverse problems," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4293–4308, 2017.

[94] J. Sun, H. Li, Z. Xu, *et al.*, "Deep admm-net for compressive sensing mri," in *Advances in Neural Information Processing Systems*, pp. 10–18, 2016.

[95] C. Metzler, A. Mousavi, and R. Baraniuk, "Learned d-amp: Principled neural network based compressive image recovery," in *Advances in Neural Information Processing Systems*, pp. 1770–1781, 2017.

[96] C. Bertocchi, E. Chouzenoux, M.-C. Corbineau, J.-C. Pesquet, and M. Prato, "Deep unfolding of a proximal interior point method for image restoration," *arXiv preprint arXiv:1812.04276*, 2018.

[97] J. Sun and Z. Xu, "Color image denoising via discriminatively learned iterative shrinkage," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4148–4159, 2015.

[98] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, 2017.

[99] B. Jawerth and W. Sweldens, "An overview of wavelet based multiresolution analyses," *SIAM review*, vol. 36, no. 3, pp. 377–412, 1994.

[100] J.-L. Starck, J. Fadili, and F. Murtagh, "The undecimated wavelet decomposition and its reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 297–309, 2007.

[101] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423, 2016.

[102] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

[103] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 674–693, 1989.

[104] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 114–125, 2017.

[105] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor em-

bedding," in *British machine vision conference (BMVC)*, pp. 1–10, IEEE, 2012.

[106] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *International Conference on Computer Vision (ICCV)*, vol. 2, pp. 416–423, IEEE, 2001.

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

[108] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, "The 2018 PIRM challenge on perceptual image super-resolution," in *European Conference on Computer Vision (ECCV)*, pp. 334–355, Springer, 2018.

[109] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883, 2016.

[110] P. N. Michelini, D. Zhu, and H. Liu, "Multi–scale recursive and perception–distortion controllable image super–resolution," in *European Conference on Computer Vision (ECCV)*, pp. 3–19, Springer, 2018.

[111] T. Vu, T. M. Luu, and C. D. Yoo, "Perception-enhanced image super-resolution via relativistic generative adversarial networks," in *European Conference on Computer Vision*, pp. 98–113, Springer, 2018.

[112] X. Deng, "Enhancing image quality via style transfer for single image super-resolution," *IEEE Signal Processing Letters*, vol. 25, no. 4, pp. 571–575, 2018.

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

[114] 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.

[115] 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.

[116] 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*, pp. 1–7, 2015.

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

[118] 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.

[119] S. Mallat, *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

[120] 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.

[121] 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.

[122] 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.

[123] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[124] 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.

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

[126] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *the IEEE Computer Vision and Pattern Recognition*, vol. 1, pp. I–I, IEEE, 2003.

[127] 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)* (A. Fitzgibbon et al. (Eds.), ed.), Part IV, LNCS 7577, pp. 611–625, Springer-Verlag, Oct. 2012.

[128] 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.

[129] 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*, pp. 71–84, Springer, 2012.

[130] M.-Y. Liu, O. Tuzel, and Y. Taguchi, "Joint geodesic upsampling of depth images," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 169–176, IEEE, 2013.

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

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

[133] 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.

[134] S. Bahrampour, N. M. Nasrabadi, A. Ray, and W. K. Jenkins, "Multimodal task-driven dictionary learning for image classification," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 24–38, 2016.

[135] P. Song, X. Deng, J. F. Mota, N. Deligiannis, P. L. Dragotti, and M. R. Rodrigues, "Multimodal image super-resolution via joint sparse representations induced by coupled dictionaries," *arXiv preprint arXiv:1709.08680*, 2017.

[136] G. Riegler, D. Ferstl, M. Rüther, and H. Bischof, "A deep primal-dual network for guided depth super-resolution," *arXiv preprint arXiv:1607.08569*, 2016.

[137] S. Wang, L. Zhang, Y. Liang, and Q. Pan, "Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2216–2223, IEEE, 2012.

[138] Y. Zhuang, Y. Wang, F. Wu, Y. Zhang, and W. Lu, "Supervised coupled dictionary learning with group structures for multi-modal retrieval.," in *AAAI Conference on Artificial Intelligence*, pp. 1070–1076, 2013.

[139] H. Zou, "The adaptive LASSO and its oracle properties," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[140] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Metaface learning for sparse representation based face recognition," in *International Conference on Image Processing (ICIP)*, pp. 1601–1604, IEEE, 2010.

[141] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[142] R. Rubinstein and M. Elad, "Dictionary learning for analysis-synthesis thresholding," *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 5962–5972, 2014.

[143] X. Song, Y. Dai, and X. Qin, "Deeply supervised depth map super-resolution as novel view synthesis," in *IEEE Transactions on Circuits and Systems for Video Technology*, 2019, Early access.

[144] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

[145] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback network for image super-resolution," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3867–3876, 2019.

[146] W. Liu, X. Chen, J. Yang, and Q. Wu, "Robust color guided depth map restoration," *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 315–327, 2017.

[147] H. Li and X.-J. Wu, "Densefuse: A fusion approach to infrared and visible images," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2614–2623, 2019.

[148] Q. Yan, X. Shen, L. Xu, S. Zhuo, X. Zhang, L. Shen, and J. Jia, "Cross-field joint image restoration via scale map," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1537–1544, 2013.

[149] B. Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 301–315, 2015.

[150] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical*

image computing and computer-assisted intervention, pp. 234–241, Springer, 2015.

[151] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[152] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 624–632, 2017.

[153] X. Deng and P. L. Dragotti, "Deep coupled ISTA network for multi-modal image super-resolution," *IEEE Transactions on Image Processing*, 2019, to appear.

[154] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, IEEE, 2007.

[155] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conference on Computer Vision (ECCV)*, pp. 611–625, Springer, 2012.

[156] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, "Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2241–2253, 2010.

[157] Y. Aksoy, C. Kim, P. Kellnhofer, S. Paris, M. Elgharib, M. Pollefeys, and W. Matusik, "A dataset of flash and ambient illumination pairs from the crowd," in *European Conference on Computer Vision (ECCV)*, pp. 634–649, 2018.

[158] J. Cai, S. Gu, and L. Zhang, "Learning a deep single image contrast enhancer from multi-exposure images," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 2049–2062, 2018.

[159] M. Nejati, S. Samavi, and S. Shirani, "Multi-focus image fusion using dictionary-based sparse representation," *Information Fusion*, vol. 25, pp. 72–84, 2015.

[160] D. Summers, "Harvard whole brain atlas: www. med. harvard. edu/aanlib/home. html," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 74, no. 3, pp. 288–288, 2003.

[161] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in *Advances in neural information processing systems*, pp. 386–396, 2017.

[162] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz, "A closed-form solution to photorealistic image stylization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 453–468, 2018.

[163] 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.

[164] 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*, pp. 24–76, 1975.