THE HONG KONG POLYTECHNIC UNIVERSITY

DEPARTMENT OF COMPUTING

# Sparse and Low-rank Models for Image Restoration

By

SHUHANG GU

A Thesis Submitted in Partial Fulfillment of

the Requirements for the Degree of

Doctor of Philosophy

March 2017

Temporary Binding for Examination Purposes

# CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signature)

_____(Name of Student)

## ABSTRACT

Image restoration aims to recover the latent high quality image from its degraded observation. As one of the most classical and fundamental topics in image processing and low-level vision, image restoration has been widely studied in the community, and a variety of approaches have been proposed, including filtering-based approaches, transformation-based approaches and variational approaches. The image sparsity priors have been used, either explicitly or implicitly, in many of these approaches, and have been playing a crucial role to improve the image restoration performance. In particular, the sparse representation based methods have achieved a great success in image restoration in the last decade.

Based on the representation scheme of a signal vector, sparse representation models can be generally categorized into analysis sparse representation (ASR) models and synthesis sparse representation (SSR) models. Low-rank minimization models have also been proposed to exploit the sparsity (i.e., low-rankness) of a matrix of correlated vectors. Different models will have respectively their merits and drawbacks. The ASR based methods regularize the projection coefficients of a signal over an analysis dictionary, and they are able to supply robust priors for image large scale structures. However, the projective coding mechanism of ASR based methods restrict their capacity of benefiting from highly redundant dictionaries, limiting their flexibility in modeling image complex texture structures. The SSR based methods represent a signal as the linear combination of a few atoms in an over-complete dictionary. To model image local structures, most SSR methods partition an image into over-lapped patches to process, which brings the inconsistency issue of over-lapped patches as well as the heavy computation burden. The low-rank methods regularize the number of independent subspaces of a matrix. Since directly minimizing the rank of a matrix is an NP hard problem, many recently developed low-rank methods adopt the nuclear norm (i.e., the

$\ell_1$ norm of singular values) for low-rank approximation. However, the nuclear norm shrinks the singular values equally, ignoring the different importance of different singular values. In this thesis, we address the above-mentioned problems of ASR, SSR and low-rank methods, and develop new sparsity-based models for image restoration.

We first investigate the ASR models for guided image restoration, and present a weighted ASR model learning scheme for RGB image guided depth image restoration. By introducing a guidance weight function, we largely improve the flexibility of ASR models and make it be able to deal with guided image enhancement tasks. Having the objective function of weighted ASR model, we utilize a task-driven training strategy to learn stage-wise dynamic parameters from training data. As a result, the proposed algorithm is able to generate high quality output efficiently. Experiments on guided depth image upsampling and noisy depth image restoration validate the effectiveness of the proposed method.

To address the inconsistency issues in previous patch-based SSR models, we propose a convolutional sparse coding (CSC) scheme for image super-resolution. By working directly on the whole image, the proposed CSC algorithm does not need to partition the image into overlapped patches, and it can exploit the image global correlation to reconstruct more robustly image local structures. State-of-the-art super-resolution results demonstrate the advantage of the proposed CSC method.

Instead of investigating the ASR and SSR models individually, we propose to integrate the two models to exploit their complementary representation mechanisms. In the proposed joint convolutional analysis and synthesis (JCAS) model, a single image is adaptively decomposed into two layers, one is used for SSR and the other for ASR. The intrinsic complementarity of ASR and SSR allows them cooperate to separate the input image into a structure layer and a texture layer. We adaptively train the synthesis dictionary to learn the required texture pattern for specific tasks. The proposed JCAS method shows very competitive performance in single image layer separation tasks, such as texture-cartoon decomposition and rain streak removal.

Besides vector based one dimensional sparse representation models, we also investigate matrix based two dimensional low-rank models for image restoration. Specifically, we extend the nuclear norm minimization (NNM) to weighted nuclear norm minimization (WNNM) by introducing a weight vector to weight different singular values of the data matrix. Although the WNNM model is nonconvex, we prove that the corresponding weighted nuclear norm proximal (WNNP) operator is equivalent to a standard quadratic programming problem with linear constraint. Very importantly, we show that the WNNM problem has closed-form optimal solution when the weights are of non-descending order. With WNNP, several extensions of the WNNM problem, including robust PCA and matrix completion, can be readily derived with the ADMM (alternating direction method of multipliers) paradigm. The proposed WNNM methods achieve state-of-the-art performance in typical low level vision tasks, including image denoising, background subtraction and image inpainting.

In summary, in this thesis we investigate in-depth the sparse representation and low-rank minimization based image restoration methods, and develop several new models for different image restoration applications. Our models not only enrich the understanding of sparsity based statistical image modeling, but also demonstrate state-of-the-art performance in image restoration and other low level vision problems.

**Keywords:** Sparse models, Low-rank models, Image restoration.

PUBLICATIONS

**Journal Papers**

1. **Shuhang Gu**, Qi Xie, Deyu Meng, Wangmeng Zuo, Xiangchu Feng, Lei Zhang, "Weighted Nuclear Norm Minimization and Its Applications to Low Level Vision," International Journal of Computer Vision January 2017, Volume 121, Issue 2, pp 183 - 208.

2. Yuan Xie, **Shuhang Gu**, Yan Liu, Wangmeng Zuo, Wenshen Zhang, Lei Zhang, "Weighted Schatten p-Norm Minimization for Image Denoising and Background Subtraction," In IEEE Trans. on Image Processing, 2016, Volume: 25, Issue: 10, Pages: 4842 - 4857.

3. Wangmeng Zuo, Dongwei Ren, David Zhang, **Shuhang Gu**, Lei Zhang, "Learning Iteration-wise Generalized Shrinkage-Thresholding Operators for Blind Deconvolution," In IEEE Trans. on Image Processing, 2016, Volume: 25, Issue: 4, Pages: 1751 - 1764.

**Conference Papers**

1. **Shuhang Gu**, Deyu Meng, Wangmeng Zuo, Lei Zhang. "Joint Convolutional Analysis and Synthesis Sparse Representation for Single Image Layer Separation." To appear in ICCV 2017.

2. **Shuhang Gu**, Wangmeng Zuo, Shi Guo, Yunjin Chen, Chongyu Chen, Lei Zhang. "Learning Dynamic Guidance for Depth Image Enhancement," In CVPR 2017.

3. **Shuhang Gu**, Wangmeng Zuo, Q. Xie, D. Meng, Xiangchu Feng, Lei Zhang. "Convolutional Sparse Coding for Image Super-resolution," In ICCV 2015.

4. **Shuhang Gu**, Lei Zhang, Wangmeng Zuo and Xiangchu Feng. "Projective dictionary pair learning for pattern classification." In NIPS 2014.

5. **Shuhang Gu**, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng, "Weighted Nuclear Norm Minimization with Application to Image Denoising," In CVPR 2014.

6. Keze Wang, Liang Lin, Wangmeng Zuo, **Shuhang Gu** and Lei Zhang, "Dictionary Pair Classifier Driven Convolutional Neural Networks for Object Detection," In CVPR 2016.

7. Q. Xie, Q. Zhao, D. Meng, Z. Xu, **Shuhang Gu**, W. Zuo and Lei Zhang, "Multispectral Images Denoising by Intrinsic Tensor Sparsity Regularization," In CVPR 2016.

8. Wangmeng Zuo, Dongwei Ren, **Shuhang Gu**, Liang Lin and Lei Zhang, "Discriminative learning of iteration-wise priors for blind deconvolution," In CVPR 2015.

ACKNOWLEDGEMENTS

First and foremost, I want to express my gratitude to my supervisor, Prof. Lei Zhang, for his wonderful guidance, support and generosity. It is my great pleasure to be a student of Prof. Zhang, who is always showing me the right direction, but allowing me to walk the path myself.

I would like to express my gratitude to Prof. Wangmeng Zuo, Prof. Xiangchu Feng and Prof. Deyu Meng in my Ph.D. study. Their valuable suggestions have greatly help me to do good research, and I am very thankful for their very constructive suggestions on my articles. I also want to acknowledge Prof. Michael Elad for offering me the opportunity to visit his lab. I am very lucky to have the chance to learn from him.

I want to express my gratitude to my lab and office mates, for their assistance, for their encouragement and for the wonderful time in The Hong Kong Polytechnic University I have shared with them.

Finally, I want to thank my family for inspiring me to pursue this route. I want to thank them for their endless love, support, and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

# CHAPTER 1
# INTRODUCTION

Due to the recent advances in hardware of imaging systems, digital cameras have become much easier to access. Although the development of hardware has greatly improved the quality of images in the last several decades, image degradation is unavoidable due to the many factors in image acqusition process. Image restoration, which aims to reconstruct a high quality image $\mathbf{x}$ from its degraded observation $\mathbf{y}$, is a classical yet still very active topic in the area of low-level computer vision. Among different approaches to image restoration, sparsity-based models have achieved very competitive results. The goal of this thesis is to investigate new sparse representation based image restoration models and algorithms. In this chapter, we first introduce the image restoration problem in section 1.1, and then review sparsity models for image restoration in section 1.2. Section 1.3 discusses the limitations of previous sparsity-based methods. Section 1.4 summarizes the contributions of this thesis.

## 1.1 Image restoration

Image restoration aims to recover a high quality image from its degraded observation. It is one of the most classical and fundamental problems in image processing and computer vision. On one hand, the ubiquitous use of imaging systems makes image restoration very important to the system performance. On the other hand, the quality of output images plays an crucial role to user experience and the success of the following high level vision tasks such as object detection and recognition.

Typical image restoration problems include denoising [1, 53, 109, 111, 114], deblurring [77, 159], super-resolution [48, 55], inpainting [5, 28] and image layer separation [20, 95, 137]. Although each of these problems has been intensively studied for many years,

current image restoration algorithms still can not fulfill the increasing requirements of high quality images. The main challenge in image restoration lies in that a significant amount of information may be lost during the degradation process, making image restoration a highly ill-posed inverse problem. In order to get a good estimation of the latent image, prior knowledge is required to provide supplementary information. Therefore, how to appropriately model the prior of high quality images is the key issue in image restoration research. In this section, we first provide a brief introduction to some commonly used corruption models, and then briefly review some popular prior modeling methods for image restoration.

### 1.1.1 The image degradation model

Due to the various factors in image acquisition, there are many types of degradations which can deterioriate the quality of images. The degradation model describes the relationship between the degraded image and unknown high quality image, which is very important for the success of restoration tasks. A general image degradation model is:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}, \tag{1.1}$$

where $\boldsymbol{x}$ refers to the unknown high quality image (ground truth), $\boldsymbol{y}$ is the degraded observation, $\boldsymbol{n}$ represents the additive white Gaussian noise (AWGN), and $\boldsymbol{H}$ is a degradation-specific operator. Depending on the operator $\boldsymbol{H}$, typical image restoration tasks includes: denoising, debluring, super-resolution and inpainting, which are introduced as follows.

- Image denoising [1, 53, 109, 111, 114] is one of the most fundamental image restoration task. It aims to reconstruct the clean image $\boldsymbol{x}$ from its noisy observation $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$, i.e., the operator $\boldsymbol{H}$ in (1.1) is an identity matrix. Denoising is not only an important pre-processing step for many vision applications, but also an ideal test bed for evaluating statistical image modeling methods.

- Image debluring [77, 159] tries to recover the latent sharp image $\boldsymbol{x}$ from a blurry observation $\boldsymbol{y} = \boldsymbol{k} \otimes \boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{k}$ is the blur kernel and $\otimes$ is the convolution operation.

According to the availability of the blur kernel $\boldsymbol{k}$, debluring can be divided into blind debluring problems and nonblind debluring problems.

- Image super-resolution [48, 55] aims to reconstruct a high resolution image from a single low resolution image or a sequence of low resolution images. The degradation between high resolution image and low resolution image can be modeled as $\boldsymbol{y} = \boldsymbol{D}(\boldsymbol{k} \otimes \boldsymbol{x}) + \boldsymbol{n}$. The down-sampling operation $\boldsymbol{D}(\cdot)$ is performed on blured image $\boldsymbol{k} \otimes \boldsymbol{x}$. The noise level in super-resolution applications is often assumed to be very small.

- Image inpainting refers to the problem of estimating the damaged or corrupted pixels in an image. The degradation model of inpainting problem can be written as: $\boldsymbol{y} = \boldsymbol{M} \odot \boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{M}$ is a binary mask and $\odot$ is the point-wise multiplication operation between two matrices. The ones and zeros in the given mask matrix $\boldsymbol{M}$ indicate whether the corresponding pixel values in $\boldsymbol{y}$ are available or not.

Besides these classical problems mentioned above, there are many other image restoration tasks such as image layer separation [20, 95, 137], rain streak removal [24, 80, 87], background estimation [32, 78, 94, 150] and depth image restoration [33, 46, 61, 106, 155], etc. We will also investigate some of these restoration tasks in this thesis, and the detailed problem definition will be provided in the corresponding chapters.

### 1.1.2 Natural image prior modeling for image restoration

One can easily see that image restoration is an ill-posed inverse problem. Prior knowledge is often needed to provide extra information for estimating the latent high quality image. Based on how the prior is exploited to generate high quality estimation, previous prior modeling methods for image restoration can be generally divided into two categories: the explicitly modeling methods and the implicitly modeling methods.

**The implicit methods:** This category of methods adopt priors of high quality images implicitly, where the priors are embeded into specific restoration operations. Such an implicitly

modeling strategy was used in most of the image restoration algorithms in the early years [109, 125, 133]. Based on the assumptions of high quality images, different operations have been designed to generate estimations directly from the degraded images. For example, based on the smoothness assumption, filtering-based methods [23, 96, 125, 132] have been widely utilized to remove noise from noisy images. Although image priors are not modeled explicitly, the priors on high quality images are considered in designing the fillters to estimate the clean images. Such implicitly modeling schemes have dominated the area of image restoration for many years. To better model the piece-wise smooth image signal, diffusion methods [109, 133] have been proposed to adaptively smooth image contents. By assuming that the wavelet coefficients of natural image are sparse, shrinkage methods have been developed to denoise images in wavelet domains [27, 39]. Although these hand-crafted methods have limited capacity in producing high-quality restoration results, these studies greatly deepen researchers' understanding on natural image modeling. Many useful conclusions and principles are still applicable to modern image restoration algorithm design.

Recently, attributed to the advances in machine learning, researchers have proposed to learn operations for image restoration. Different methods have been developed to build the complex mapping functions between degraded and high quality image pairs, such as nonlinear regression [52, 69] and neural networks [11, 35]. Since the functions (such as neural networks) learned in these methods are often very complex, the priors embedded in these functions are very hard to analyze. As a result, the functions trained for a specific task are often inapplicable to other restoration tasks. One may need to train different models for different applications or even for the same application with different degradation parameters. Albeit its limited generalization capacity, the highly competitive restoration results obtained by these discriminative learning methods make this category of approaches an active and attractive research topic.

**The explicit methods:** Besides implicitly embedding priors into restoration operations, another category of methods explicitly characterize image priors and adopt the Bayesian method to produce high quality reconstruction results. Having the degradation model $p(\boldsymbol{y}|\boldsymbol{x})$

and specific prior model $p(\boldsymbol{x})$, different estimators can be used to estimate latent image $\boldsymbol{x}$. One popular approach is the *maximum a posterior* (MAP) estimator:

$$\hat{\boldsymbol{x}} = argmax_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y}) = argmax_{\boldsymbol{x}} p(\boldsymbol{y}|\boldsymbol{x}) p(\boldsymbol{x}), \tag{1.2}$$

with which we seek for the most likely estimate of $\boldsymbol{x}$ given the corrupted observation and prior. Compared with other estimators, the MAP estimator often leads to an easier inference algorithm, which makes it the most commonly used estimator for image restoration. However, MAP estimation still has limitations in the case of few measurements [127]. An alternative estimator is the Bayesian least square (BLS) estimator:

$$\hat{\boldsymbol{x}} = E\{\boldsymbol{x}|\boldsymbol{y}\} = \int_{\boldsymbol{x}} \boldsymbol{x} p(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x}. \tag{1.3}$$

BLS marginalizes the posterior probability $p(\boldsymbol{x}|\boldsymbol{y})$ over all possible clean images $\boldsymbol{x}$. Theoretically, it is the optimal estimate in terms of mean square error, and the estimator is also named as minimum mean square error (MMSE) estimator [127].

A wide range of models, such as Independent Component Analysis (ICA) [4], variational models [114], sparse representation models [1] and Markov random field (MRF) [48, 111], have been utilized to characterize priors of natural image. Early studies tend to analyze image signals with analytical mathematical tools and manually designed functional forms to describe natural image prior. Recent methods tend to take advantage of training data and learn parameters to better model high quality image priors. Compared with implicit prior modeling methods, these explicit priors often have a stronger generalization capacity and can be applied to different image restoration applications.

In both the two categories of image restoration approaches, sparsity priors play an extremely important role. Under the Bayesian framework, most state-of-the-arts methods share the similar idea of using sparsity priors. In this thesis, we will investigate some challenging problems appeared in the application of sparsity priors for image restoration tasks.

## 1.2 Sparse representation models for image restoration

Image sparsity priors have been widely used in many computer vision applications. In this section, we provide a brief review of sparse representation based image restoration algorithms. Generally speaking, study on this topic can be divided into analysis-based and synthesis-based methods [41], while the recently developed low-rank models [14] for image restoration can be viewed as a two dimensional extension of conventional sparse models.

### 1.2.1 Analysis sparse representation models for image restoration

The analysis representation approaches represent a signal in terms of its product with a linear operator:

$$\boldsymbol{\alpha}_a = \boldsymbol{P}\boldsymbol{x}, \tag{1.4}$$

where $\boldsymbol{x}$ is the signal vector and $\boldsymbol{\alpha}_a$ is its analysis representation coefficients. Linear operator $\boldsymbol{P}$ is often referred to as the analysis dictionary [113]. Researches on image statistics have shown that the marginal distributions of bandpass filter responses to natural images exhibit clearly non-Gaussianity and heavy tails [47]. Under the MAP framework, most of analysis sparse representation models share a similar form:

$$\hat{\boldsymbol{x}} = argmin_x \Upsilon(\boldsymbol{x}, \boldsymbol{y}) + \Psi(\boldsymbol{P}\boldsymbol{x}), \tag{1.5}$$

where $\Upsilon(\boldsymbol{x}, \boldsymbol{y})$ is the data fidelity term which depends on the degradation model, and $\Psi(\boldsymbol{P}\boldsymbol{x})$ is the regularization term which imposes sparsity prior on the filter responses $\boldsymbol{P}\boldsymbol{x}$.

The analysis dictionary $\boldsymbol{P}$ and the penalty function $\Psi(\cdot)$ play a very important role in the analysis sparse representation model. Early studies utilize signal processing and statistical tools to analytically design dictionaries and penalty functions. One of the most notable analysis-based methods is the Total-Variation (TV) approach [114], which uses a Laplacian distribution to model image gradients, resulting in an $\ell_1$ norm penalty on the gradients of estimated image. In addition to TV and its extensions [17, 19, 20], researchers have also proposed wavelet filters [13, 34, 91] for analysis sparse representation. In these methods, the gradient operator in TV methods is replaced by wavelet filters to model image local

structures. Besides dictionaries, the penalty functions have also been well investigated. Different statistical models have been introduced to model the heavy-tailed distributions of coefficients in natural, which leads to a variety of robust penalty functions, such as $\ell_p$ norm [159], normalized sparsity measure [70], etc.

Although these analytic methods have greatly deepened our understanding on image modeling, they are considered to be over-simplistic to model the complex natural phenomena. With the advance of computing power, machine learning methods have been introduced to learn better priors. From a probabilistic image modeling point of view, some early approaches actually attempt to learn potential functions for predefined filters [156]. Inspired by the pioneer work of field-of-expert (FoE) [111], many different methods have been proposed to learn appropriate filters (analysis dictionary) for predefined potential (penalty) functions [113, 122]. Although these works are introduced from different point of views (for example, MRF [122], co-sparsity [113], etc.), they can all be interpreted from the perspective of analysis representation based regularization. Besides these unsupervised generative learning methods, discriminative learning methods have also been utilized to train priors for specific tasks [25, 57]. By using the image pairs as training data, these discriminative learning methods are able to deliver highly competitive restoration results. However, the learning is often achieved by solving a bi-level optimization problem, which is time-consuming. Recently, Schmidt et al. [116] and Chen et al. [26] proposed to unfold the optimization process of (1.5) and got highly effective and efficient restoration models.

### 1.2.2 Synthesis sparse representation models for image restoration

Different from the analysis representation models, the synthesis sparse representation models represent a signal $\boldsymbol{x}$ as the linear combination of dictionary atoms:

$$\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}_s, \tag{1.6}$$

where $\boldsymbol{\alpha}_s$ is the synthesis coefficient for signal vector $\boldsymbol{x}$, and $\boldsymbol{D}$ is the synthesis dictionary. Such a decomposition model may have different choices of $\boldsymbol{\alpha}_s$, and regularization is required

to provide a well-defined solution. A commonly used criterion is to find a sparse coefficient vector $\boldsymbol{\alpha}_s$, which reconstructs the signal by only a few atoms in $\boldsymbol{D}$. In their seminal work [92], Mallat and Zhang proposed a matching pursuit (MP) algorithm to find an approximated solution of the NP-hard sparse decomposition problem. The orthogonal MP (OMP) [107] method was later proposed to improve the performance of synthesis-based modeling. Besides constraining the non-zero values ($\ell_0$ norm) in the coefficients, researchers have also proposed to utilize its convex envelop, the $\ell_1$ norm, to regularize the synthesis coefficient. Compared with $\ell_0$ norm, the global solution of the convex $\ell_1$ problem can be achieved. One can solve $\ell_1$ norm sparse coding problem with conventional linear programming solvers, or modern methods, such as least angle regression [40] and proximal algorithms [105].

The application of synthesis-based sparse representation for image restoration is quite straight forward under the MAP framework:

$$\hat{\boldsymbol{x}} = argmin_x \Upsilon(\boldsymbol{x}, \boldsymbol{y}) + \Psi(\boldsymbol{\alpha}_s),\ s.t. \boldsymbol{D}\boldsymbol{\alpha}_s = \boldsymbol{x}, \tag{1.7}$$

where $\Upsilon(\boldsymbol{x}, \boldsymbol{y})$ is the fidelity term, and the regularization $\Psi(\cdot)$ on synthesis coefficient $\boldsymbol{\alpha}_s$ provides prior information for estimating the clean image $\boldsymbol{x}$. In early years, the adopted dictionary is often designed under the umbrella of harmonic analysis [112] such as DCT, wavelet and curvelet dictionaries. These dictionaries, however, are far from enough to model the complex structures of natural images, limiting the image restoration performance. To better model the local structures in images, dictionary learning methods have been introduced to improve image restoration performance [1]. Armed with the learned dictionaries, synthesis sparse representation framework has led to state-of-the-art denoising results. Researchers have also made many attempts on designing strong regularization functions $\Psi(\cdot)$ to provide better prior for restoration [16]. For example, Julien et al. [89] and Dong et al. [38] suggested to use the group-sparsity or centralized sparsity to embed the nonlocal self-similarity prior in the sparse representation model, respectively.

Besides modeling prior explicitly under the Bayesian framework, synthesis sparse representation models also play an important role in many implicit prior modeling methods [55, 72, 129, 142]. These methods aim to find a complex mapping function to model

the relationship between corrupted and high quality images. A popular method is to learn coupled dictionaries, which link the coefficients of the two types of data to build a simple relationship. The idea of coupled dictionary learning has led to state-of-the-art algorithms on different applications [55, 72].

### 1.2.3 Low rank models for image restoration

Apart from employing sparsity prior for signal vectors, low-rank models have also been proposed to exploit the sparsity (i.e., low-rankness) of a matrix of correlated vectors. Low rank matrix approximation (LRMA) aims to recover the underlying low rank matrix from its degraded observation. It has achieved a great success in various applications of computer vision [14, 15, 53, 115, 149]. The current development of LRMA can be categorized into two categories: low rank matrix factorization (LRMF) [65, 124] and rank minimization [14, 53, 136]. LRMF factorizes the input matrix $\boldsymbol{Y} \in \Re^{m\times n}$ into two smaller ones $\boldsymbol{A} \in \Re^{m\times k}$ and $\boldsymbol{B} \in \Re^{n\times k}$. Here $k < min(m, n)$ ensures the low-rank property of the reconstructed matrix $\boldsymbol{AB}^T$. Rank minimization methods reconstruct the data matrix through imposing a rank constraint upon the estimated matrix. Since directly minimizing the matrix rank is an NP-hard problem [44], the relaxation methods are more commonly utilized. Candes et al. developed the nuclear norm minimization (NNM) methods [14, 15] and discussed the exact recovery property of NNM. The regularization-based NNM method and its extensions have attracted great research interests, and they have been applied to different applications [62, 84, 149].

The low-rank models have also been successfully applied to image restoration problems [53, 62, 104, 130]. Some early studies [62, 148] take the low-rankness as a global prior, and treat image restoration as an LRMA problem directly. However, such an assumption is too strong and can not reflect the characteristics of natural images. As a result, those global low-rank prior based methods can only perform well on images with special contents, and they will over-smooth details in natural images. Instead of approximating clean images globally, recent methods [53, 130, 136] estimate a group of non-local similar patches. The

combination of nonlocal self-similarity prior and low rank models has proved to be highly effective, leading to state-of-the-art algorithms for different image restoration applications [53].

## 1.3 Limitations of existing sparsity-based models

Although sparsity-based methods have achieved great successes in image restoration applications, they still have some limitations, as we introduced below.

### 1.3.1 Limitations of analysis sparse representation models

Analysis sparse representation (ASR) models characterize the complement subspace of a signal. The zero representation coefficients indicate a orthogonal relationship between the signal and the corresponding bases. Such an ASR prior provides a robust prior for the principal components of the signal, and performs well on approximating the major structures of an image. However, it can not take advantages of a redundant analysis dictionary. As a result, ASR has limited capacity in modeling textures with complex pattern, even when a large scale training dataset with repetitive texture pattern is provided.

To improve the flexibility of ASR, task-driven learning strategy has been proposed to train task-specific analysis priors [25]. Furthermore, some recent works [26, 116] have been proposed to unfold the optimization procedure of the analysis-based models and learn the model parameters from training data. Such models have achieved very good restoration performance on tasks such as image denoising and super-resolution.

However, the discriminative learning based ASR models mentioned above can only deliver limited results on some challenging restoration problems such as non-blind deblurring and guided image restoration. In non-blind debluring, one needs to iteratively estimate the blur kernel as well as the latent clean image, which makes discriminative learning methods hard to learn optimal parameters for the whole estimation procedure. Guided image restoration problem is another challenging task. Although the guidance image is able to

provide additional information about the latent image, there is a much higher expectation on the quality of restored images. Currently, most of the guided image restoration methods use hand-crafted parameters to generate the estimation. It is interesting to investigate if the ASR model is flexible enough to deal with such a challenging problem. In this thesis, we answer this question by proposing a highly effective weighted ASR model. By taking the advantage of ASR in modeling piece-wise image, we utilize the proposed weighted ASR model to deal with the guided depth image enhancement problem. The sate-of-the-art performance on depth image upsampling, hole-filling and denoising validate the effectiveness of the proposed method.

### 1.3.2 Limitations of synthesis sparse representation models

Synthesis sparse representation models (SSR) characterize the union-of-subspaces of signals, which is able to take advantage of highly redundant dictionaries. Given an appropriate dictionary of atoms (learned from training data), an image can be well reconstructed with highly sparse coefficients. However, SSR requires solving a sparse coding problem over an over-complete dictionary, which can be time consuming. To handle the high-dimensional image signal, most of current SSR methods utilize a patch dividing strategy and model prior probability on local patches. The image is first divided into over-lapped patches and each patch is processed independently. It is commonly accepted that more overlapped pixels between neighboring patches will deliver better reconstruction results since each pixel in the output image will be estimated for more times. However, such an overlap-averaging strategy ignores an important constraint in solving the patch estimation problem, i.e., pixels in the overlapped area of adjacent patches should be exactly the same (i.e., consistent). The consistency constraint provides prior information on each single estimation problem. Actually, in the seminal work of example-based restoration method [48], the consistency prior is modeled by an MRF to select HR patches in the external database. Recently, researchers have proposed several elegant aggregation methods [66, 158] to alleviate the inconsistency of overlapped patches, and achieved significant performance improvement in image denois-

ing. Nonetheless, the aggregation issue is still a practical problem which may deteriorate the performance of patch-based methods.

Apart from the aggregation issue, patch-dividing strategy may also generate shifted observations for the same type of local structure. As a result, a large number of atoms are required to characterize the patch samples, bringing great computation burden in the decomposition process. To address this problem, in this thesis, we propose a convolution-based synthesis model to decompose the image globally. The proposed algorithm achieves state-of-the-art performance on the super-resolution task.

Furthermore, instead of using a single type of sparsity model, we propose to integrate the ASR and SSR models for image restoration. We take advantage of the complementary property of the two models, and utilize one ASR and one SSR components to approximate the input image simultaneously. The two models characterize different types of image local structures. The analysis-based priors characterize well the major background information of the image, while the synthesis-based priors characterize the textures well. We apply the proposed model to texture-cartoon decomposition and rain streak removal and achieve state-of-the-art performance.

### 1.3.3 Limitations of low-rank minimization models

As the tightest convex envelop of the rank function, the nuclear norm minimization (NNM) approach has been attracting significant attention due to its rapid development in both theory and implementation. On one hand, Candes et al. [15] proved that from the noisy input, its intrinsic low-rank reconstruction can be exactly achieved with a high probability through solving an NNM problem. On the other hand, Cai et al. [12] proved that the nuclear norm proximal (NNP) problem

$$\hat{\boldsymbol{X}} = \mathbf{prox}_{\lambda\|\cdot\|_*}(\boldsymbol{Y}) = \arg\min_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \lambda\|\boldsymbol{X}\|_*, \tag{1.8}$$

can be easily solved in closed-form by imposing a soft-thresholding operation on the singular values of the observation matrix. By utilizing NNP as the key proximal technique, many

NNM-based models have been proposed in recent years [14, 62, 84].

Albeit its success as aforementioned, NNM still has certain limitations. In traditional NNM, all singular values are treated equally and shrunk with the same threshold. This, however, ignores the prior knowledge we often have on singular values of a practical data matrix. More specifically, larger singular values of an input data matrix quantify the information of its underlying principal directions. For example, the large singular values of a matrix of image similar patches deliver the major edge and texture information. This implies that to recover an image from its corrupted observation, we should shrink less the larger singular values and shrink more the smaller ones. Clearly, traditional NNM model, as well as its corresponding soft-thresholding solvers, are not flexible enough to achieve this goal.

In this thesis, we propose the weighted nuclear norm and study its minimization strategy. The weight vector will enhance the representation capability of the original nuclear norm. Rational weights specified based on the prior knowledge and understanding of the problem will benefit the corresponding weighted nuclear norm minimization (WNNM) model for achieving a better estimation of the latent data from the corrupted input.

### 1.4 Contributions and organization of the thesis

This thesis makes an in-depth study on the three major types of sparsity-based image restoration methods. We correspondingly present several state-of-the-art algorithms for different image restoration tasks.

- To improve the flexibility of analysis-based models, we present a weighted analysis sparse representation model for guided image restoration. The weight generation function is able to introduce information from a guidance image and generate a local structure aware weight map to control the regularization on the estimation. Furthermore, task driven training strategy is utilized to learn stage-wise model parameters from training data, which allows the proposed method to efficiently generate good restoration results for different tasks. The proposed method demonstrates state-of-the-

art results on guided depth image upsampling and hole filling tasks. This part of work can be found in Chapter 2 of this thesis.

- To improve the performance of synthesis-based models, we propose to use convolution-based method, instead of conventional patch-based method, to deal with image restoration problems. Compared with conventional sparse coding methods which process overlapped patches independently, the global decomposition strategy in convolutional sparse coding is more suitable for keeping patch consistency. Our experiments on commonly used test images show that the proposed method achieves very competitive super-resolution results with the state-of-the-art methods not only in PSNR index, but also in visual quality. This part of work can be found in Chapter 3 of this thesis.

- To exploit the advantages of both analysis-based and synthesis-based models, we propose a joint convolutional analysis and synthesis (JCAS) model to deal with the single image layer separation problem. The proposed model can not only achieve competitive decomposition performance on the texture-cartoon decomposition and rain streak removal applications, but also help us better understand the intrinsic characteristics of the two models. This part of work can be found in Chapter 4 of this thesis.

- We investigate the regularization-based LRMA problem, and propose the weighted nuclear norm minimization (WNNM) scheme. We prove that the non-convex weighted nuclear norm proximal (WNNP) problem has a globally optimal solution, and we present the algorithm and discuss its properties. The WNNP can be used to deal with the image denoising problem in couple with the nonlocal self-similarity prior. By extending WNNP to WNNM, we further apply it to matrix completion (WNNM-MC) and robust principal component analysis (WNNM-RPCA), and deliver state-of-the-art image inpainting and backgroud substraction results. This part of work can be found in Chapter 5 of this thesis.

The organization of the thesis is illustrated in Figure 1.1. We first present two implicit prior modeling methods in chapter 2 and chapter 3, which improve previous ASR and

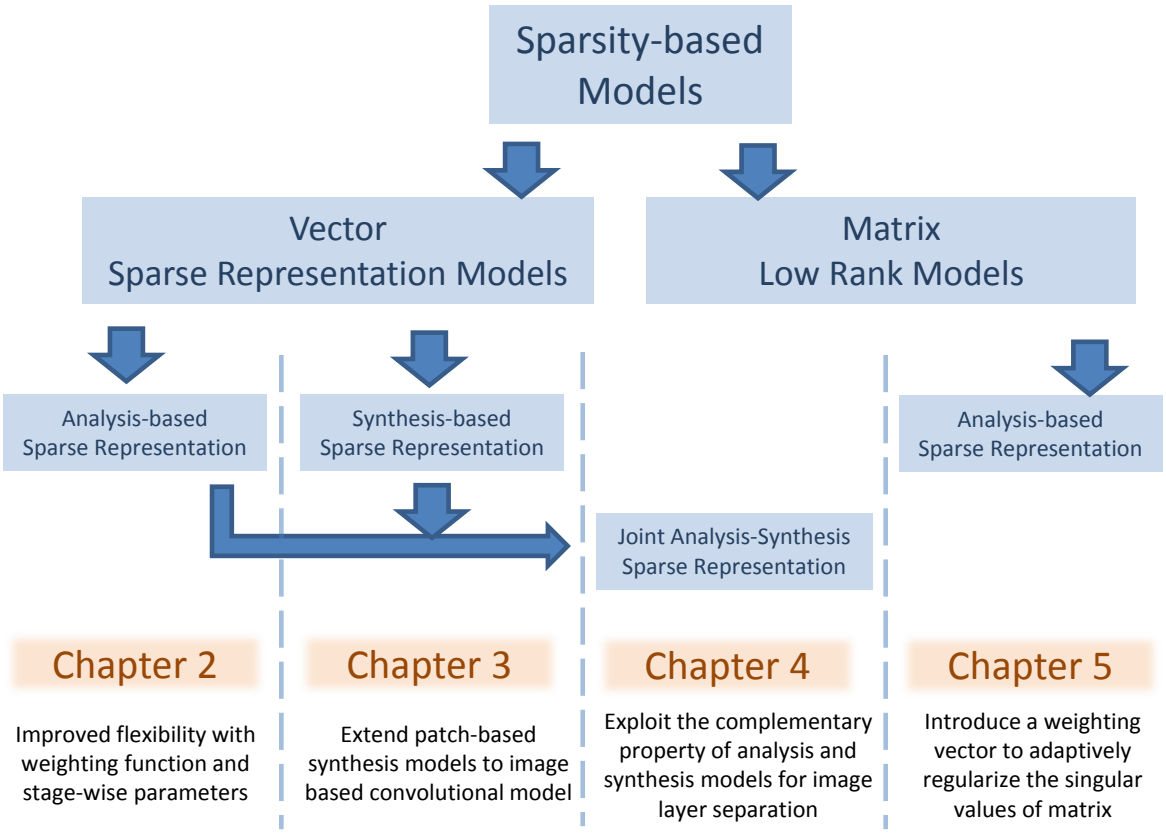



Figure 1.1. Thesis organization.

SSR methods, respectively. We apply them to guided depth image restoration and image super-resolution tasks, and achieve state-of-the-art results. In chapter 4, we take advantage of the complementary property of ASR and SSR models, and utilize the ASR and SSR priors explicitly to regularize different layers. The proposed method achieves state-of-the-art rain streak removal performance without any training data. In chapter 5, we study the low rank model for image restoration. The weighted nuclear norm is proposed to regularize the latent matrix explicitly. By adaptively penalizing different singular values of the data matrix, the proposed method outperforms previous low rank minimization methods in different applications.

# CHAPTER 2

# WEIGHTED ANALYSIS SPARSE REPRESENTATION MODEL LEARNING FOR GUIDED IMAGE RESTORATION

Guided depth image restoration provides a natural and powerful approach to enhance the quality of depth image acquired by consumer-level RGB-D cameras, which has been receiving considerable research interests. One key issue in guided depth image restoration is how to model the dependency between the RGB intensity image and depth images. Most of the existing methods are based on hand-crafted parameters, which are not tailored for depth enhancement and insufficient in characterizing the complex relationship between depth and intensity images. In this chapter, we suggest a task driven model to learn a weighted analysis representation model for guided depth image restoration. The proposed model generalizes the representations of guided weight function and analysis representation model, and learns them in a task driven manner to achieve the goal of guided depth image restoration. In order to tackle the high non-convexity of the proposed model, we unfold the inference process as an iterative algorithm, where stage-wise model parameters are learned from training data. As a consequence, the proposed framework is able to generate high quality results in a few stages. Experimental results on RGB guided depth image upsampling and hole filling validate the effectiveness of the proposed method.

## 2.1 Introduction

### 2.1.1 Depth image restoration and enhancement

High quality depth image plays a fundamental role in many real world applications, such as robotics, human-computer interaction, and augmented reality. Traditional depth sensing is mainly based on stereo or laser measurement, which in general is of high computational complexity or expensive cost. Recently, consumer depth sensing products, e.g., RGB-D cameras and Time of Flight (ToF) range sensors, have became widely available and they offer an economic alternative for dense depth measurements. However, the depth image generated by consumer depth sensors usually is of insufficient quality, e.g., with low resolution, noise or missing values.

Depth image restoration and enhancement have received considerable research interests [33, 46, 61, 106, 155]. One category of methods utilize multiple low quality depth images from the same scene to reconstruct a high quality depth image [61, 155]. The success of such methods, however, relies on high accurate calibration, which may fail when applied to dynamic environment. Another category of methods directly enhance each single depth image [18, 33, 46, 86, 106], guided by a corresponding high quality RGB image. For most consumer level depth sensors, high quality RGB image can be simultaneously acquired with the depth image, making guided depth image enhancement a natural and attractive approach to improve depth image quality.

The key issue of guided depth image enhancement is how to model and exploit the dependency between intensity and depth images. The first category of methods use the guided filters to transfer the structure (i.e., location and sharpness of the edges) of intensity image to the restored depth map [58]. However, inconsistent structures in the intensity image may be introduced into the depth image. Shen et al. suggested a joint filtering method to enhance and transfer only the consistent structures [118]. The second category of methods include Markov Random Fields (MRF) [33] and variational models (e.g., Total Generalized Variation [46] and non-local means [106]). In these methods, certain forms of objective functions

are adopted for modeling the interdependency. These models usually are intuitive but limited in characterizing the complex relationship between intensity image and depth image. The third category of methods are learning-based models [73, 126], where analysis or synthesis dictionary learning is exploited to model the statistical dependency between intensity and depth images. In these methods, intensity and the associated depth images are simply packed together to learn dictionaries in a group learning manner, which is however insufficient to characterize the complex interdependency between intensity and depth. Moreover, even the dictionary is known, in general the resulting model remains nonconvex and is hard to solve.

### 2.1.2 Motivation

As introduced in the previous section, a number of approaches have been proposed to introduce the guidance information for depth image enhancement. One representative method is to formulate the input image $\boldsymbol{y}$, desired image $\boldsymbol{x}$ and the guidance image $\boldsymbol{g}$ into an optimization model [33, 46, 106]. Several complex functions have been utilized to model the structural dependencies between $\boldsymbol{x}$ and $\boldsymbol{g}$ and have yielded state-of-the-art performance in different applications. Diebel et al. [33] proposed an MRF formulation for depth upsampling, where the guidance image is utilized to control the smoothness of depth map by the prior potential function:

$$\sum\nolimits_i \sum\nolimits_{j \in N(i)} w_{ij}(\boldsymbol{g})(\boldsymbol{x}_i - \boldsymbol{x}_j)^2, \tag{2.1}$$

where $i$ and $j$ are the pixel indexes of an image, and $N(i)$ is the set of neighboring index of $i$. The weight $w_{ij}(\boldsymbol{g})$ is an exponential function of pixel-pair difference $(\boldsymbol{g}_i - \boldsymbol{g}_j)^2$ in the guidance image. Sharing the similar idea that using a guidance image related weighting function to adaptively regularize the smoothness in depth map, different methods have been proposed to deal with the guided depth enhancement problem. Park et al. [106] proposed a non-local mean (NLM) regularizer to control smoothness in depth map, while the weight $w_{ij}(\boldsymbol{g})$ is calculated by considering color, segmentation and edge cues. Recently, Ham et al. [56] proposed a nonconvex method to handle differences in structure between guidance and

input images. Instead of changing the weighting function, the Welsch's function is utilized [56] to regularize depth differences:

$$\sum_i \sum_{j \in N(i)} \phi_\mu(\boldsymbol{g}_i - \boldsymbol{g}_j)(1 - \phi_\nu(\boldsymbol{x}_i - \boldsymbol{x}_j))/\nu, \tag{2.2}$$

where $\phi_\mu(z) = exp(-\mu z^2)$ is the weighting method. Besides these first-order methods which simply model pixel-pair wise differences, there are also works proposed to model high order relationship in depth map with hand-crafted [46] or learned [67] operators.

Actually, the models in Eqns. (2.1) and (2.2) can be viewed as extensions to hand-crafted analysis models, in which a group of inter-pixel difference operators are used as the analysis filters, and weight functions on $\boldsymbol{g}$ are introduced for explicit guidance. Motivated by this observation, we propose a generalized analysis representation model with weight function, and provide a task-driven learning method to learn the weight functions, analysis filters, and penalty functions from training data. With the training data, we can not only find a good way to introduce the guided intensity information, but also good parameters to deal with specific tasks. A flowchart of the proposed method can be found in Fig. 2.1.

## 2.2 Proposed method

In this section, we introduce the proposed model for depth map restoration. We first introduce the detailed formulation of our model. Then, we provide a task driven training method to learn stage-wise model parameters. With the learned stage-wise parameters, guided depth restoration can be achieved in only a few iterations.

### 2.2.1 Weighted analysis sparse representation model

How to take full advantage of the information from the guided image is the key issue in guided image restoration. Many previous optimization based methods [33, 56, 106] promote discontinuity between intensity and depth maps by using guidance related weights to regularize pixel-pair wise differences. To better model the dependency between intensity and

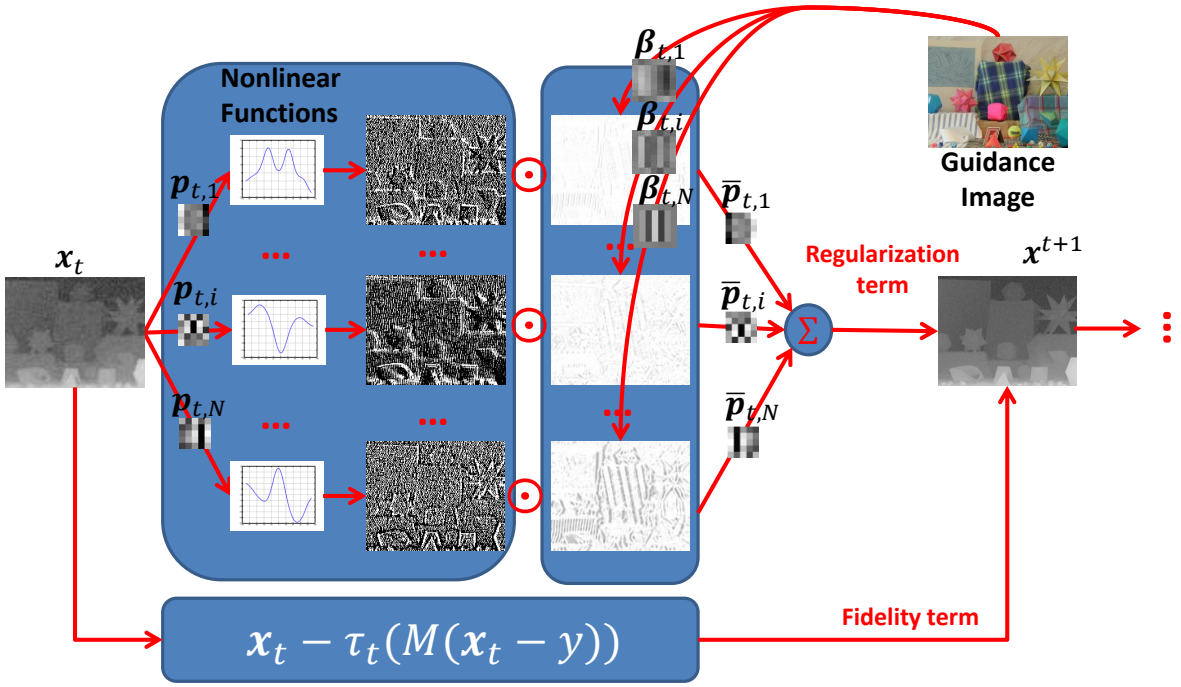



Figure 2.1. Flowchart of the proposed method.

depth images, we propose the following weighted analysis prior model:

$$\mathcal{R}(\boldsymbol{x};\boldsymbol{g}) = \sum\nolimits_i \langle \boldsymbol{w}_i(\boldsymbol{g};\boldsymbol{\beta}_i), \boldsymbol{\rho}_i(\boldsymbol{x};\boldsymbol{p}_i,\boldsymbol{\alpha}_i)\rangle \ , \tag{2.3}$$

where $\langle\cdot,\cdot\rangle$ denotes the standard inner product. The weight $\boldsymbol{w}_i \in \Re^N$ is a column vector associated with each pixel in the intensity image $\boldsymbol{g}$, which is controlled by the parameter $\boldsymbol{\beta}_i$. $\boldsymbol{\rho}_i(\boldsymbol{x};\boldsymbol{p}_i,\boldsymbol{\alpha}_i)$ is also a column vector by point-wisely applying the penalty function $\rho_i$ to the filter response $\boldsymbol{p}_i * \boldsymbol{x}$, i.e.,

$$\boldsymbol{\rho}_i(\boldsymbol{x};\boldsymbol{p}_i,\boldsymbol{\alpha}_i) = (\rho_i((\boldsymbol{p}_i * \boldsymbol{x})_1), \cdots, \rho_i((\boldsymbol{p}_i * \boldsymbol{x})_N))^T \in \Re^N \ ,$$

where $*$ denotes the convolution operator, and the penalty function $\boldsymbol{\rho}_i$ is parameterized by $\boldsymbol{\alpha}_i$. Note that in a standard analysis prior model, the weight $\boldsymbol{w}_i$ is given as a constant. However, in our proposed model, $\boldsymbol{w}_i$ is defined based on the local structure of intensity image, such that $\boldsymbol{w}_i \to 1$ at homogeneous regions, and $\boldsymbol{w}_i \to 0$ at edges. As a consequence, the resulting weighted analysis model will penalize high depth discontinuities at homogeneous regions and allow sharp depth jumps at the corresponding edges.

By plugging the proposed analysis prior model (2.3) into a variational framework, we arrive at the following functional

$$\min_{\boldsymbol{x}} E(\boldsymbol{x}) = \Upsilon(\boldsymbol{x}, \boldsymbol{y}) + \sum\nolimits_i \langle \boldsymbol{w}_i(\boldsymbol{g}; \boldsymbol{\beta}_i), \rho_i(\boldsymbol{P}_i \boldsymbol{x}) \rangle \ , \tag{2.4}$$

where $\boldsymbol{P}_i \in \Re^{N \times N}$ is a highly sparse matrix, implemented as 2D convolution of the image $\boldsymbol{x}$ with the filter kernel $\boldsymbol{p}_i$, i.e., $\boldsymbol{P}_i \boldsymbol{x} \Leftrightarrow \boldsymbol{p}_i * \boldsymbol{x}$.

Most penalty functions used for natural image restoration as well as guided depth map refinement favor small filter responses, and accordingly smooth image edges [33, 46, 106, 111, 114]. While, Chen et al. [26] found that the behavior of penalty functions learned from training data is actually very complex. Though most of the penalty functions tend to shrink the filter response to promote smoothness, there are also some penalty functions which will enlarge filter responses in certain ranges. Such an expansion behavior is helpful to generate sharper image edges. In order to generate high-quality depth map with sharp edges, we follow [26] and investigate penalty functions with flexible shapes. As in both training and test phases (e.g., see (2.10)), the proposed model explicitly involves the first-order derivative of the penalty function $\rho_i$, we alternatively focus on the derivative function $\phi_i = \rho_i'$, which is known as the influence function [26] and can be parameterized as

$$\phi_i(z) = \sum\nolimits_j^M \boldsymbol{\alpha}_{i,j} \texttt{exp}\left(\frac{-(z - \mu_j)^2}{2\gamma_j^2}\right). \tag{2.5}$$

$\phi_i$ is the summation of $M$ Gaussian RFB kernels with center $\mu_j$ and scaling factor $\gamma_j$. This formulation is not only able to generate penalty functions, which shrink the filter response to generate smooth results, but also able to generate functions which could enlarge the filter response in some range to promote sharp edges.

In the proposed model, the weight generating function $\boldsymbol{w}(\boldsymbol{g})$ is introduced to guide the regularization on depth filtering responses based on the structure information in the intensity image. Although the intensity image and the depth map come from the same scene and have certain structure dependencies, the values in the two images have different physical meanings. For example, a black box in front of a white wall or a gray box in front of a black wall may correspond to the same depth map but totally different RGB images. The

weighting function should be able to avoid such interference of structure-unrelated intensity information, while extracting useful structure information to help the depth map locate its edges.

Denote by $\boldsymbol{W}_i$ the reshaped weight map, in our model, we consider the following function to generate the weight map from $\boldsymbol{g}$:

$$\boldsymbol{W}_i(m,n) = \texttt{exp}\left(-\langle \boldsymbol{\beta}_i, \boldsymbol{e}_{m,n}\rangle^2\right) , \tag{2.6}$$

where $\boldsymbol{W}_i(m,n)$ is the weight value in position $(m,n)$, and $\boldsymbol{e}_{m,n} = \frac{\boldsymbol{R}_{m,n}\boldsymbol{g}}{||\boldsymbol{R}_{m,n}\boldsymbol{g}||_2}$ with $\boldsymbol{R}_{m,n}$ an operation to extract local patch in position $(m,n)$ of image $\boldsymbol{g}$. $\boldsymbol{\beta}_i$ is the corresponding linear filter to extract the structural features. The local normalization operation on intensity map is utilized to avoid the effect of different intensity magnitude. The function form $\texttt{exp}\,(-(\cdot)^2)$ in (2.6) makes the weighting function be a step-edge-like function with respect to the filtering response but differentiable. It helps weight map focus on local structure changes in RGB image instead of intensity values.

Having the influence function defined in (2.5) and the weighting function defined in (2.6), we propose a task-driven method to learn the parameters $\{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{p}_i\}_{i=1\cdots N_k}$ from the training data.

### 2.2.2 Training method

Task-driven training strategy aims to learn model parameters to deal with specific tasks [88]. It has been utilized in different tasks and achieved very competitive results. Given $S$ training samples $\{y^{(s)}, x_g^{(s)}\}_{s=1}^S$, where $y^{(s)}$ is the observation and $x_g^{(s)}$ is the corresponding ground truth, task driven training method learns model parameters as sollows [88]:

$$\begin{aligned} \theta^* = \arg\min_{\theta} \sum\nolimits_{s=1}^{S} \ell(x_g^{(s)}, x_\theta^{(s)}(\theta)) \\ s.t. \quad x_\theta^{(s)}(\theta) = f(y^{(s)};\theta)\,, \end{aligned} \tag{2.7}$$

where the generalized function $x_\theta = f(y,\theta)$ represents the process of using the proposed model to generate estimation $x_\theta$, with input data $y$ and model parameters $\theta$. Task specific

loss function $\ell(x_g, x_\theta)$ w.r.t. ground truth $x_g$ and parameter related estimation $x_\theta$ is utilized to learn the best parameter $\theta^*$ for the task.

Our proposed weighted sparse representation model in (2.4) is a highly non-convex minimization problem with many parameters $\{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{p}_i\}_{i=1\cdots N_k}$. Finding the exact solution of the lower-level function $f(y, \theta)$ will make the learning problem very difficult to optimize. Actually, instead of finding the exact solution of the optimization problem in the lower-level problem, researchers have proposed to learn iteration-wise operations to deal with different problems. Gregor et al. [51] have shown that convolutional neural networks can be learned to approximate the coding process of the sparse coding algorithms. Based on the half-quadric splitting method or standard gradient descent method, Schmidt et al. [116] learned a few iteration-wise operations to deal with natural image restoration problems with state-of-the-art performance.

Inspired by these works, we learn stage-wise operations for our objective function (2.4). Solving (2.4) by gradient descent method, we get the following formula in each step:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \left( \nabla_x \Upsilon(\boldsymbol{x}_t, \boldsymbol{y}) + \sum\nolimits_i \boldsymbol{P}_{t,i}^T \text{diag}\left(\boldsymbol{w}_{t,i}\right) \phi_{t,i}(\boldsymbol{P}_{t,i}\boldsymbol{x}_t) \right) , \tag{2.8}$$

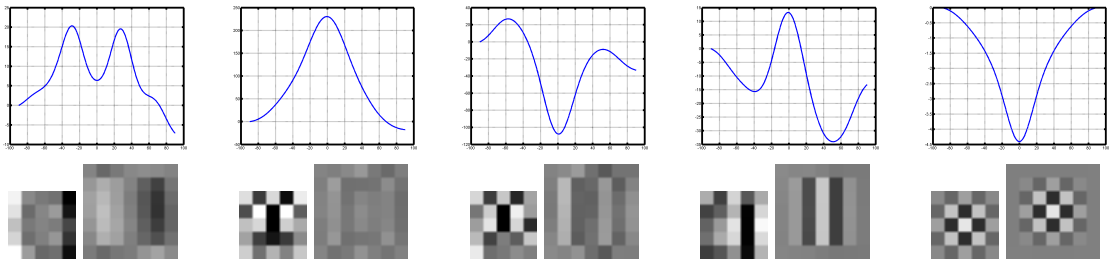
where the subscript $t$ is the stage index and $i$ is the regularization term index. For example, $\boldsymbol{w}_{t,i}$ is the $i$-th weight generation function in the $t$-th stage. $\text{diag}(\boldsymbol{w}_{t,i}) \in \Re^{N\times N}$ is a square matrix with diagonal elements by vector $\boldsymbol{w}_{t,i}$. $\boldsymbol{x}_t$ is the estimation of $\boldsymbol{x}$ in the $t$-th stage.

For both the depth upsampling and hole filling applications considered in this chapter, the fidelity term can be written as follows:

$$\Upsilon(\boldsymbol{x}, \boldsymbol{y}) = \frac{\tau}{2} \|\boldsymbol{M}^{\frac{1}{2}}(\boldsymbol{x} - \boldsymbol{y})\|_2^2, \tag{2.9}$$

where $\boldsymbol{M}$ is a diagonal matrix and $\tau$ is related to the strength of fidelity force. For depth upsampling application, the diagonal elements in $\boldsymbol{M}$ indicate the corresponding points between high resolution estimation $\boldsymbol{x}$ and aligned low resolution input $\boldsymbol{y}$. In the application of hole filling, $\boldsymbol{M}$ is a binary matrix which indicates the differences in observed points.

In our work, we consider the commonly used square error to measure the loss between current estimation and ground truth. Besides, a greedy training strategy is utilized to



(a) (b) (c) (d) (e)

Figure 2.2. Part of learned parameters. In each sub-figure, the upper part is the regressed penalty function by $\boldsymbol{\alpha}_i$; the lower left part is the analysis filters $\boldsymbol{p}_i$ for depth map; and the lower right part is the analysis filters $\boldsymbol{\beta}_i$ for guided intensity image.

learn the parameters in each iteration, which is formulated as

$$\begin{aligned} \{\tau_t, \boldsymbol{\alpha}_{t,i}, \boldsymbol{\beta}_{t,i}, \boldsymbol{p}_{t,i}\} &= \arg\min_{\theta_t} \frac{1}{2} \sum\nolimits_{s=1}^{S} ||\boldsymbol{x}_g^{(s)} - \boldsymbol{x}_{t+1}^{(s)}||_2^2 \\ s.t.\ \boldsymbol{x}_{t+1}^{(s)} &= \boldsymbol{x}_t^{(s)} - \left( \tau_t \boldsymbol{M}(\boldsymbol{x}_t^{(s)} - \boldsymbol{y}^{(s)}) + \sum\nolimits_i \boldsymbol{P}_{t,i}^T \text{diag}\left(\boldsymbol{w}_{t,i}^{(s)}\right) \phi_{t,i}(\boldsymbol{P}_{t,i}\boldsymbol{x}_t^{(s)}) \right). \end{aligned} \tag{2.10}$$

The gradient of the loss function with respect to parameters $\{\tau_t, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{p}_i\}$ can be calculated by the chain rule. Afterwards we use the limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (LBFGS) method [83, 93] to learn parameters in each stage. We experimentally found that we can get very good results after a few stages of process. After training, the inference process of the proposed model is very fast.

### 2.2.3 Discussions

In previous optimization based guided filtering methods, hand-crafted parameters and functions are usually designed based on prior knowledge of the problem. Such hand-crafted design greatly limits the choice of regularizers, and thus is insufficient to model the dependency between intensity and depth images. The proposed weighted analysis sparse representation method provides a more flexible model to characterize the complex relationship between guidance and output images.

To illustrate the complex relationship between depth map and intensity image, we show some learned parameters for noisy depth upsampling with factor 4. The detailed experimental setting can be found in section 2.4. 5 of the 24 groups of learned parameters

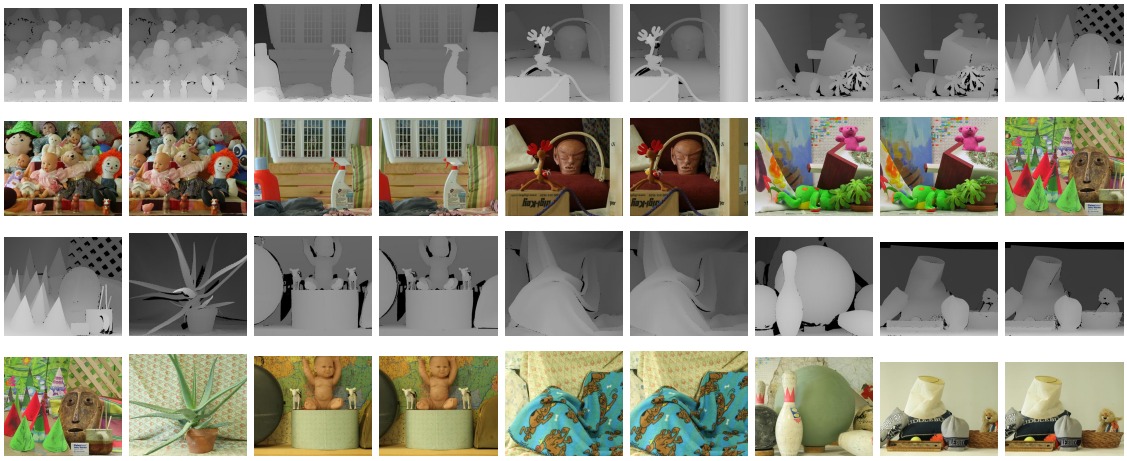


Figure 2.3. The training samples used in the guided depth upsampling experiments.

$\{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{p}_i\}_{i=1\cdots6}$ are shown in Fig. 2.2 (a-e). The filters $\boldsymbol{\beta}_i$ and $\boldsymbol{p}_i$ are reshaped for better visualization, and for $\boldsymbol{\alpha}_i$, we directly show the regressed penalty function. From Fig. 2.2, one can see that the local relationship is modeled by very complex parameters. Different from previous methods which use the same pixel-wise differences to model the co-discontinuities in depth and RGB images, the kernels learned to extract local information from the two images are different. Furthermore, the penalty functions learned by our models show very complex behaviours. Instead of some monotonic shrinkage functions which are in favor of small filter responses to promote smoothness in the estimation, some functions learned by our method show clearly expansion behaviour. Such flexible penalty functions make our model be able to generate high quality depth map with sharp edges in a few steps.

## 2.3 Experiments on depth map upsampling

In this section, we compare the proposed method with other depth upsampling methods. Three commonly used datasets (Middlebury [60], NYU [102] and ToFMark [46]) are utilized to evaluate the depth upsampling performance of the proposed method. Besides the baseline bicubic and bilinear upsampling methods, we compare the proposed methods with a variety of guided upsampling methods. The comparison methods include two filtering based

Table 2.1. Experimental results (RMSE) on the 3 noise-free test images.

| | Art | | | | Books | | | | Moebius | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ×2 | ×4 | ×8 | ×16 | ×2 | ×4 | ×8 | ×16 | ×2 | ×4 | ×8 | ×16 |
| Bicubic | 2.57 | 3.85 | 5.52 | 8.37 | 1.01 | 1.56 | 2.25 | 3.35 | 0.91 | 1.38 | 2.04 | 2.95 |
| Bilinear | 2.83 | 4.15 | 6.00 | 8.93 | 1.12 | 1.67 | 2.39 | 3.53 | 1.02 | 1.50 | 2.20 | 3.18 |
| GF [58] | 2.93 | 3.79 | 4.97 | 7.88 | 1.16 | 1.58 | 2.10 | 3.19 | 1.10 | 1.43 | 1.88 | 2.85 |
| MRF [33] | 3.12 | 3.79 | 5.50 | 8.66 | 1.21 | 1.55 | 2.21 | 3.40 | 1.19 | 1.44 | 2.05 | 3.08 |
| Yang [144] | 4.07 | 4.06 | 4.71 | 8.27 | 1.61 | 1.70 | 1.95 | 3.32 | 1.07 | 1.39 | 1.82 | 2.49 |
| Park [106] | 2.83 | 3.50 | 4.17 | 6.26 | 1.20 | 1.50 | 1.98 | 2.95 | 1.06 | 1.35 | 1.80 | 2.38 |
| TGV [46] | 3.03 | 3.79 | 4.79 | 7.10 | 1.29 | 1.60 | 1.99 | 2.94 | 1.13 | 1.46 | 1.91 | 2.63 |
| SDF [56] | 3.31 | 3.73 | 4.60 | 7.33 | 1.51 | 1.67 | 1.98 | 2.92 | 1.56 | 1.54 | 1.85 | 2.57 |
| DJF [79] | 2.77 | 3.69 | 4.92 | 7.72 | 1.11 | 1.71 | 2.16 | 2.91 | 1.04 | 1.50 | 1.99 | 2.95 |
| Ours | **0.89** | **2.00** | **3.82** | **6.16** | **0.47** | **0.91** | **1.67** | **2.68** | **0.45** | **0.84** | **1.51** | **2.35** |

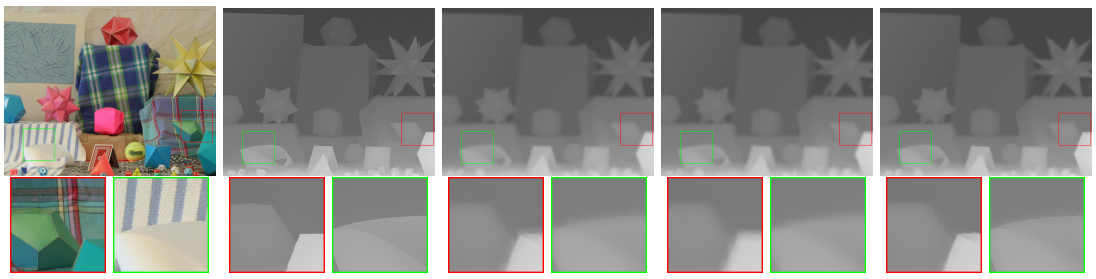
methods [58, 144], and some optimization based methods: MRF based method [33], non-local mean regularized depth upsampling method [106], total generalized variation (TGV) method [46], joint static and dynamic filtering(SDF) method [56]. The root mean square error (RMSE) indexes by recent proposed deep learning based method [79] are also included. Detailed experimental setting will be introduced in the following subsections.

### 2.3.1 Upsampling results on the Middlebury dataset

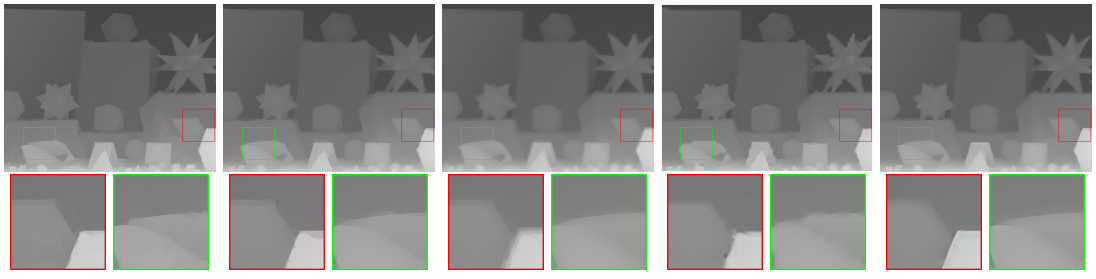
The *Art*, *Books* and *Moebius* images in the Middlebury dataset [60] have been widely utilized to evaluate depth restoration algorithms. Following the experimental setting of [46], we conduct upsampling experiments with both the noise-free and noisy low resolution depth maps on four zooming factors, i.e., 2, 4, 8, 16. For the noise-free experiments, both the training and testing samples are generated by bicubic resizing of the high quality depth maps. While, for the noisy experiments, the testing noisy low-resolution depth maps are obtained from

Table 2.2. Experimental results (RMSE) on the 3 noisy test images.

| | Art | | | | Books | | | | Moebius | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ×2 | ×4 | ×8 | ×16 | ×2 | ×4 | ×8 | ×16 | ×2 | ×4 | ×8 | ×16 |
| Bicubic | 5.32 | 6.07 | 7.27 | 9.59 | 5.00 | 5.15 | 5.45 | 5.97 | 5.34 | 5.51 | 5.68 | 6.11 |
| Bilinear | 4.58 | 5.62 | 7.14 | 9.72 | 3.95 | 4.31 | 4.71 | 5.38 | 4.20 | 4.57 | 4.87 | 5.43 |
| GF [58] | 3.55 | 4.41 | 5.72 | 8.49 | 2.37 | 2.74 | 3.42 | 4.53 | 2.48 | 2.83 | 3.57 | 4.58 |
| MRF [33] | 3.49 | 4.51 | 6.39 | 9.39 | 2.06 | 3.00 | 4.05 | 5.13 | 2.13 | 3.11 | 4.18 | 5.17 |
| Yang [144] | 3.01 | 4.02 | 4.99 | 7.86 | 1.87 | 2.38 | 2.88 | 4.27 | 1.92 | 2.42 | 2.98 | 4.40 |
| Park [106] | 3.76 | 4.56 | 5.93 | 9.32 | 1.95 | 2.61 | 3.31 | 4.85 | 1.96 | 2.51 | 3.22 | 4.48 |
| TGV [46] | 3.19 | 4.06 | 5.08 | 7.61 | 1.52 | 2.21 | 2.47 | 3.54 | 1.47 | 2.03 | 2.58 | **3.56** |
| Chan [18] | 3.44 | 4.46 | 6.12 | 8.68 | 2.09 | 2.77 | 3.78 | 5.45 | 2.08 | 2.76 | 3.87 | 5.57 |
| SDF [56] | 3.36 | 3.86 | 4.93 | 7.85 | 1.59 | 1.92 | 2.60 | 4.16 | 1.64 | 1.85 | 2.67 | 4.21 |
| Ours | **1.82** | **2.94** | **4.40** | **7.05** | **1.13** | **1.61** | **2.33** | **3.49** | **1.29** | **1.72** | **2.57** | 3.79 |



(a) Color image (b) Ground truth (c) GF [58] (d) MRF [33] (e) Yang et al.[144]



(f) Park et al. [106] (g) TGV [46] (h) SDF [56] (i) DJF [79] (j) Ours

Figure 2.4. Depth restoration results by different methods on noise-free data (Moebius).

Table 2.3. Experimental results (RMSE) on the 448 NYU test image.

| | MRF [33] | GF [58] | JBU [110] | TGV [46] | Park [106] | SDF [56] | DJF [79] | Ours |
|---|---|---|---|---|---|---|---|---|
| ×4 | 4.29 | 4.04 | 2.31 | 3.83 | 3.00 | 3.04 | 1.97 | **1.56** |
| ×8 | 7.54 | 7.34 | 4.12 | 6.46 | 5.05 | 5.67 | 3.39 | **2.99** |
| ×16 | 12.32 | 12.23 | 6.98 | 13.49 | 9.73 | 9.97 | 5.63 | **5.24** |

[106], and we prepare training noisy low resolution depth maps by adding white Gaussian noise with standard variation 6 to the clean low resolution depth images.

To prepare training data, we select 18 depth and intensity image pairs in the Middlebury data set [60] and extract 250 72×72 small images as the training dataset. The 18 images are shown in Fig. 2.3. We can see that some images are captured for the same scene with slight viewpoint changes. Since training data set contains limited samples, we further extend it by flipping and rotating the original images. Finally, we get 1000 images of resolution 72×72 in the training data set. Although the extension improves the variation of the training samples, the training data are still not diverse enough because the original 18 training images are limited in colors. In our experiments, instead of using RGB color images, we only use gray level intensity images to guide the restoration.

Besides the model parameters to be learned, there are also some algorithm parameters, e.g., the number of filters and the filter size for the depth map and intensity map, to be fixed. Generally, larger filters are able to model the structural relationship of a larger area. With enough training data, they will lead to better performance. However, utilizing large filters often demands a large number of filters to model local structural prior, which will greatly increase the computational burden in both the training and testing phases. To make a good balance between efficiency and upsampling accuracy, we use 24 5×5 analysis filters $\{\boldsymbol{k}_l\}_{l=1\ldots L}\}$ for depth image. For filters $\{\boldsymbol{\beta}_l\}_{l=1\ldots L}\}$ used to extract information from the intensity image, we set their size as 7×7. For both the noisy and noise-free cases, we use the results by bicubic interpolation as the initialization of $\boldsymbol{x}_0$. Our experimental results show

that the proposed model is able to generate very good upsampling results in a few steps. For the noise free upsampling experiemts, we set the stage numbers for zooming factors 2, 4, 8 and 16 as 4, 5, 6 and 7, respectively. While for the noisy upsampling experiments, the stage numbers are set as 6, 8, 10 and 12. Adding extra stages will further deduce the training loss, but surfer from higher computation burden in both the training and testing phases.

The upsampling experimental results on the 3 noise-free testing images by different methods are shown in Table 2.1. The proposed method consistently shows its advantage over the competing methods. It achieves the best results on all the 3 images with different zooming factors. In Fig. 2.3, we give visual examples of the upsampling results on the *moebius* image with zooming factor 16. We can see that the guided filter method [58] and the MRF method [33] can not generate very sharp edges; while the results by [144][106] and [46] have some artifacts in the edge area. Our method is able to generate high quality depth map with sharper edges and less artifacts.

We further evaluate the proposed method by upsampling experiments on noisy depth maps . The results by different methods are shown in Table 2.2. We do not provide the results by DJF [79] because the sauce codes and results on such setting are not available. The results by [18] are included, which is designed to handle noise in depth super-resolution. The proposed method again achieves the best results.

### 2.3.2 Upsampling results on the NYU dataset

In [79], Li et al. utilized the first 1000 images of NYU dataset [102] as training data, and evaluated their DJF method on the last 448 images of the NYU dataset. In this section, we follow their experimental setting and compare different methods on the 448 images. The results by the other methods are provided by the authors of [79]. To save the training time, we train our model with only the first 100 images from the training data set of [79]. The number and size of the filters are the same as our settings on the Middleburry [60] dataset. The stage number for all the zooming factors 4, 8 and 16 is set as 4.

The experimental result are shown in Table 2.3. Compared with other methods, the

proposed method achieves the best results in terms of RMSE.

### 2.3.3 Upsampling results on real sensor data

Besides synthetic data, we also evaluate the proposed method on real sensor dataset [46]. In which a Time of Flight (ToF) and a CMOS camera are used to obtain low resolution depth maps and intensity images, and the ground truth depth images are generated by a structured light scanner. This dataset is called ToFMark [46].

We utilize the 18 images in Fig. 2.3 as the training images. The noise in the low resolution input of ToFMark dataset is different from previous synthetic data. To generate similar low resolution input for training, we first use a t location-scale distribution to fit the residuals between input and groundtruth data, and then generate additive noise by the distribution parameters. Since the missing values in the depth map are represented as zeros, which may be termed as very sharp edge in the depth map. We use a simple masked joint bilateral filtering [110] method to generate initialization values for the unknown points in the depth map. Although such initialization $\boldsymbol{x}_0$ is still very noisy, our method can still generate very good results in just several stages. In addition, we adopt larger size filters ($7 \times 7$ filters $\boldsymbol{k}_i$ for depth image and $9 \times 9$ filters $\beta_i$ for intensity image) to further improve the performance of the proposed method.

The restoration results are shown in Table 2.4. We compare our method with classical and state-of-the-art methods. Table 2.4 shows that our method produces better results in terms of RMSE. From Fig. 2.5, it is easy to see that our method is capable of generating clean upsampling estimation, while, the results by other methods tend to copy irrelevant textures from the intensity image.

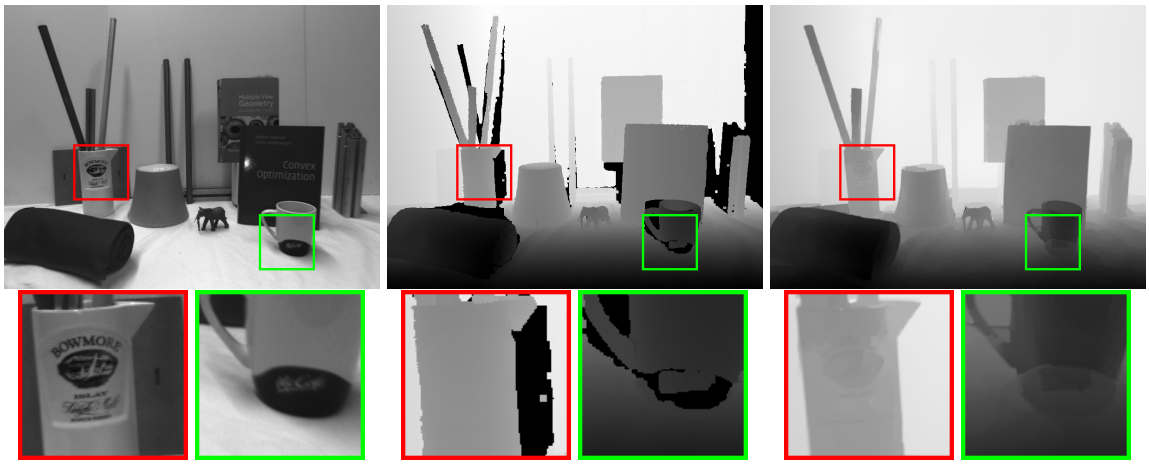
## 2.4 Experiments on incomplete depth map restoration

In this section, we provide some experimental results on other depth map restoration problems. The dataset in [86] is used to test the proposed method, in which there are not only

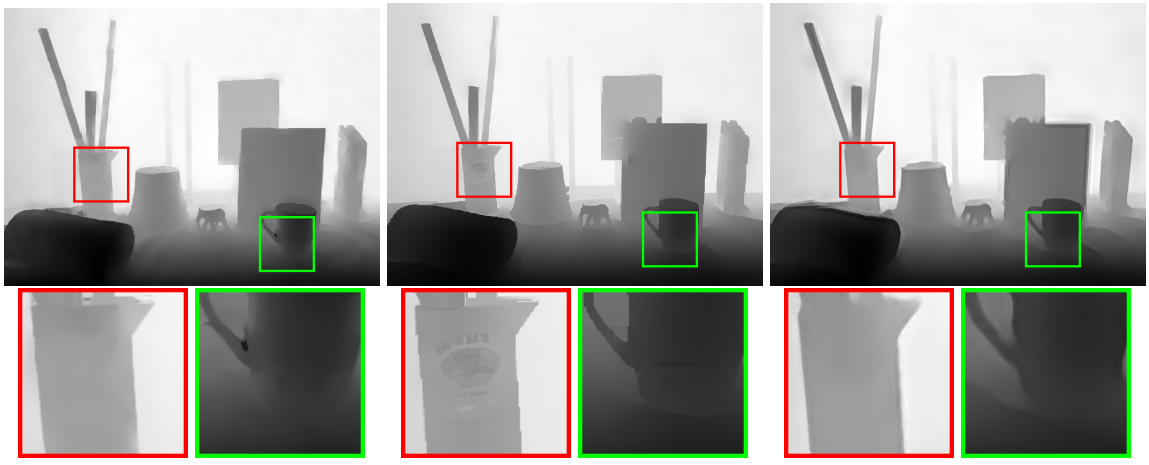
Table 2.4. Experimental results (RMSE) on the 3 test images in [46]

.

| | Nearest Neighbor | Bilinear | He [58] | TGV [46] | Yang [144] | SDF [56] | Ours |
|---|---|---|---|---|---|---|---|
| Books | 18.21 | 17.10 | 15.74 | 12.36 | **12.25** | 12.66 | 12.31 |
| Shark | 21.83 | 20.17 | 18.21 | 15.29 | 14.71 | 14.33 | **14.06** |
| Devil | 19.36 | 18.66 | 27.04 | 14.68 | 13.83 | 10.68 | **9.66** |

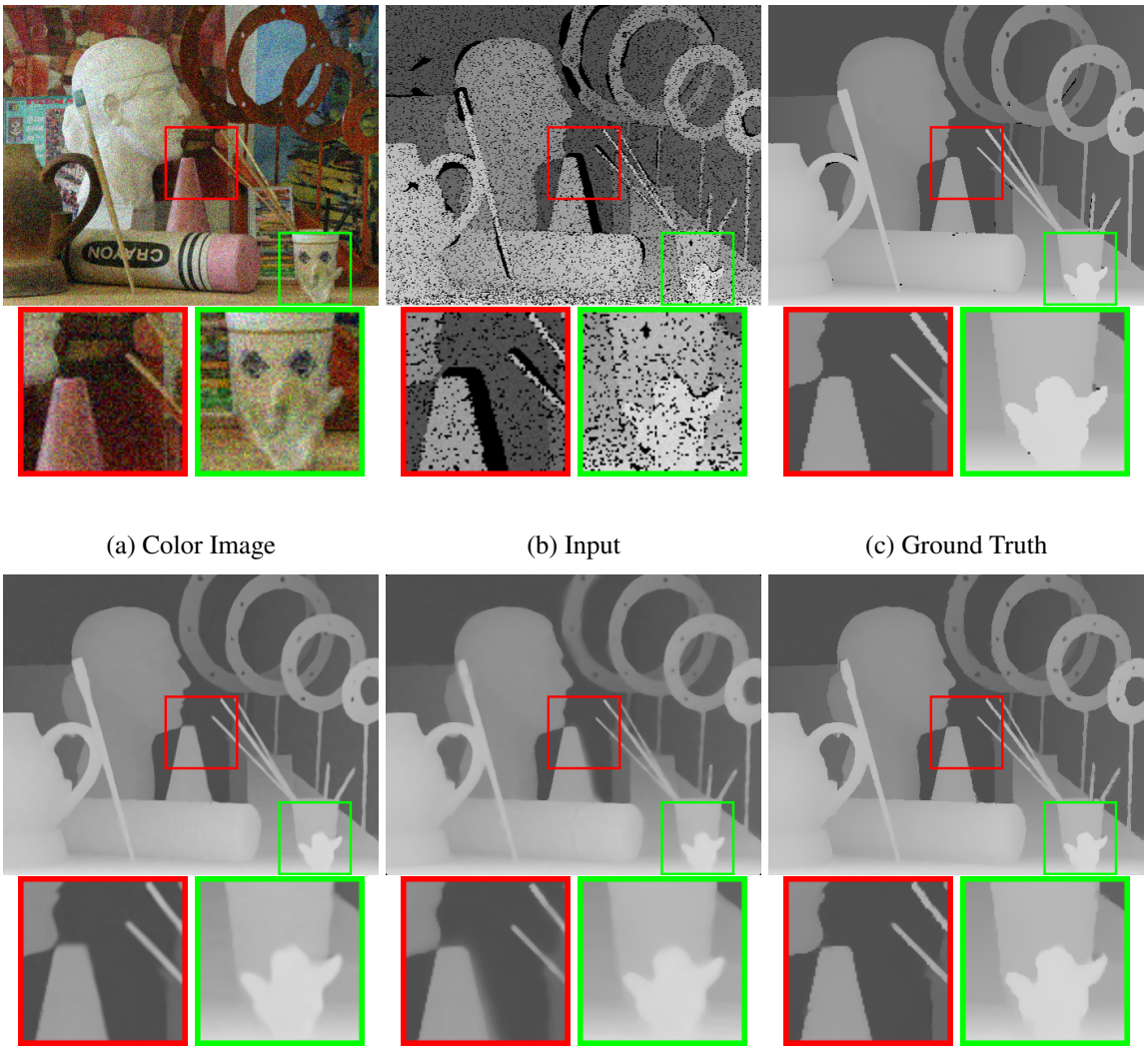


(a) Intensity image (b) Ground true (c) TGV [46]



(d) Yang et al. [144] (e) SDF [56] (f) Ours

Figure 2.5. Depth restoration results by different methods on real data.



(a) Color Image (b) Input (c) Ground Truth

(d) Lu et al. [86] (e) Shen et al. [118] (f) Ours

Figure 2.6. Depth restoration results by different methods.

Table 2.5. Experimental results (RMSE) on the 3 test images in [86].

| | Lu et al. [86] | Shen et al. [118] | Ours |
|---|---|---|---|
| Art | 6.77 | 5.65 | **4.96** |
| Books | 2.24 | 2.24 | **1.66** |
| Moebius | 2.18 | 2.27 | **1.76** |

additive Gaussian noise but also some missing values in the depth image. In the following subsections, we first introduce our experimental settings, including the preparation of training data, the setting of initialization and some algorithm parameters. Then, we compare our method with other methods designed for this task, including low rank based method [86] and recently proposed mutual-structure joint filtering method [118].

### 2.4.1 Experimental setting

Lu et al. [86] provided a synthetic data set to evaluate depth restoration methods. 30 depth and RGB image pairs in the Middlebury database [60] are included in the data set. All images are normalized to have the same height 370. Zero mean additive Gaussian noise with stand deviation 25 and 5 are added into the RGB and depth image, respectively. The authors manually set $13\%$ of pixels in depth map as missing values to simulate the depth map acquired from consumer level depth sensors.

To compare the proposed method with other methods, we take the image *Art*, *Books* and *Moebius* as testing images, and use the remaining 27 images as the training images. Since our proposed method does not consider the noise in the RGB image, for fair comparison, we pre-process the RGB image by a state-of-the-art denoising method [53] and use the denoised image to guide the restoration of depth map. Such a method has been utilized in the original paper [86] to compare with other depth restoration methods.

The setting of filter number and size in this noisy depth map restoration experiment is the same as that in the Middleburry upsampling experiment. As our setting in the ToF dataset [46], we also adopt JBF [110] to provide initial values for the missing data. Since less training data are provided in this dataset, we only adopt 4 stages to enhance the input depth image.

### 2.4.2 Experimental results

The restoration results by different methods are shown in Tabel 2.5. The results by [86] and [118] are downloaded from the authors' websites. The proposed method shows significant advantage over the competing methods in terms of RMSE. In Fig. 2.6, we show some visual examples of the restoration results. One can clearly see that our restoration method is able to generate sharp edges when removing noise in the input image.

## 2.5 Conclusions

In this chapter, we improved the flexibility of conventional analysis sparse representation model by introducing a weighting function. The weighting function is able to extract local structural information from the guidance image, and adaptively regularize the analysis coefficients of the latent clean image. Such a formulation provides us a highly flexible framework to model the complex relationship between the guidance and target images. To learn optimal parameters for different applications, we utilized a task-driven training strategy to learn parameters from the training data. The stage-wise dynamic parameters can generate high quality estimations in several stages. We evaluated the proposed model on both the guided depth upsampling and hole-filling applications. Compared with previous algorithms, our method can generate better results with lower RMSE and more pleasant visual quality.

# CHAPTER 3

# CONVOLUTIONAL SPARSE CODING FOR IMAGE SUPER-RESOLUTION

Sparse coding (SC) plays an important role in versatile computer vision applications. Most of the previous SC based image restoration methods partition the image into overlapped patches, and process each patch separately. These methods, however, ignore the consistency of pixels in overlapped patches, which is a strong constraint for image reconstruction. In this chapter, we propose a convolutional sparse coding (CSC) based method to address the consistency issue. To deal with the super-resolution (SR) application, we propose to learn three groups of parameters: (i) a set of filters to decompose the low resolution (LR) image into LR sparse feature maps; (ii) a mapping function to predict the high resolution (HR) feature maps from the LR ones; and (iii) a set of filters to reconstruct the HR images from the predicted HR feature maps via simple convolution operations. By working directly on the whole image, the proposed CSC-SR algorithm does not need to divide the image into overlapped patches, and can exploit the image global correlation to produce more robust reconstruction of image local structures. Experimental results clearly validate the advantages of CSC over patch based SC in SR application.

In this chapter, we first review some recently proposed SR algorithms and discuss the inconsistency issue in these methods. Then, after a brief introduction of the convolutional sparse coding (CSC) method, we present the proposed CSC-SR algorithm. The comparison results with state-of-the-art algorithms show the effectiveness of CSC-SR algorithm. Most of the contents of this chapter have been published in [55].

## 3.1 Introduction

### 3.1.1 Single image super-resolution

The purpose of super-resolution (SR) is to reconstruct a high resolution (HR) image from a single low resolution (LR) image or a sequence of LR images. SR provides a way to overcome the inherent resolution limitations of low-cost imaging sensors, and it also offers a solution to enhance the existing images which are generated by old type imaging equipment. Compared with SR from a sequence of images, single image SR (SISR) is more ill-posed because less information is provided. A key issue of SISR is how to build the relationship between the LR image and the HR image. Since information has been lost in the down-sampling procedure, prior knowledge is needed to provide extra information for estimating the HR image. In the early years of studies, some simple smooth assumptions have been utilized to estimate the missing pixels of the HR image, and different analytical interpolation methods have been proposed to zoom up LR images. However, such kind of simple smooth assumptions are far from enough for reconstructing complex structures in natural images.

The pioneer work in [48] uses Markov random field (MRF) to model the image priors. Inspired by [48], many methods have been developed to model prior knowledge on local structures or patches using natural images [43, 120, 121, 142]. Methods in [43, 120, 121] learn the gradient distribution from high quality natural images to guide the HR estimation in the testing phase. Considering that natural images have complex local structures, instead of modeling the prior on the entire image, most SISR methods utilize the prior knowledge on image patches, which can be further grouped into three categories: example-based, mapping-based, and sparse coding-based methods. For example-based methods, both the external [22, 48, 140] dataset and internal cross-scale relationship [49] can be employed to provide examples of the LR and HR patch pairs. For mapping-based methods, mapping function between the LR and HR images can be directly learned using the LR/HR patch pairs to implicitly incorporate prior knowledge [29, 36, 139]. For sparse coding-based methods, motivated by the progress of sparse coding and dictionary learning, a couple of dictionaries can be trained from the LR and HR image patches, and several approaches have been suggest-

ed to model the relationship between the LR and HR patches in the coding vector domain [59, 129, 142].

### 3.1.2 Motivation

Although patch based methods can greatly reduce the problem size and obtain state-of-the-art performance in SISR [138], previous studies usually process the overlapped patches independently, and the final results are achieved by averaging the overlapped pixels between each patch. It is commonly accepted that more overlapped pixels between neighboring patches will deliver better reconstruction results since each pixel in the output image will be estimated for more times. However, such an "overlap-averaging" mechanism ignores an important constraint in solving the patch estimation problem, i.e., pixels in the overlapped area of adjacent patches should be exactly the same (i.e., consistent). The consistency property is a strong constraint and can provide prior information in dealing with each single estimation problem. Actually, in the seminal work of [48] the consistency prior is modeled by an MRF to select HR patches in the external database. Recently, researchers [66, 158] have proposed several elegant aggregation methods to alleviate the inconsistency of overlapped patches, and achieved significant performance improvement in image denoising. However, for SISR, more evidences and better approaches are still needed to justify the importance of the consistency constraint.

In this chapter, we present a convolutional sparse coding (CSC) based SISR method to demonstrate the effectiveness of consistency constraint and the advantage of global image based CSC over conventional patch based sparse coding. CSC was first proposed by Zeiler et al. [145]. Instead of sparsely representing a vector by the linear combination of dictionary atoms, CSC decomposes the input image into $N$ sparse feature maps by $N$ filters. The convolutional decomposition avoids dividing the whole image into overlapped patches and can naturally utilize the consistency prior in the decomposition procedure. CSC has already been utilized in several works to extract features from images for object recognition [146]. However, compared with the great success of conventional patch based sparse coding in image

reconstruction, no work has been reported that CSC can achieve state-of-the-art performance in image reconstruction.

Previous joint dictionary learning works [59, 129, 142] encode the interpolated LR image and use the corresponding HR dictionary to reconstruct the HR estimation. The LR and HR dictionaries have the same number of components. The interpolation operation before sparse coding greatly increases the computation burden because we need to encode the interpolated image which has the same size of HR image. Furthermore, using the same number of atoms in the LR and HR dictionaries limit the representation capacity of the HR dictionary since HR images are much more complex than the LR images. To address these problems, we use LR and HR filter groups which have different filter numbers and filter sizes to decompose and reconstruct the LR and HR images. A transformation mapping function is introduced to build the relationship between the LR and HR feature maps which are with different sizes in both the spatial and coefficient domain. Such a mechanism not only reduces the computation burden of CSC in the LR image decomposition step, but also improves the representation capacity of HR filters to ensure the performance of our algorithm.

## 3.2 Related works

In this section, we first briefly review the conventional sparse coding method and its application to single image super-resolution (SISR), and then introduce the convolutional sparse coding (CSC) method.

### 3.2.1 Sparse coding for super resolution

Sparse representation encodes a signal vector $\boldsymbol{x}$ as the linear combination of a few atoms in a dictionary $\boldsymbol{D}$, i.e., $\boldsymbol{x} \approx \boldsymbol{D}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is the sparse coding vector. By far, sparse representation has been widely applied in many computer vision applications and achieved state-of-the-art results in various tasks [90, 135, 143]. As for SISR, Yang et al. proposed a sparse coding super resolution (ScSR) method in [142]. In the training phase, given a group of low

resolution (LR) and high resolution (HR) training patch pairs, ScSR aims to jointly learn an HR dictionary $\boldsymbol{D}^h$ and an LR dictionary $\boldsymbol{D}^l$ to reconstruct the HR and LR patches by assuming that each LR/HR patch pair shares the same sparse coding vector. In the testing phase, the input LR image is divided into overlapped patches, and each patch is encoded by the LR dictionary $\boldsymbol{D}^l$ with the sparse coefficient $\boldsymbol{\alpha}$. The corresponding HR patch is reconstructed by $\boldsymbol{D}^h$ and $\boldsymbol{\alpha}$ as $\boldsymbol{D}^h\boldsymbol{\alpha}$. Finally, the HR image can be obtained by aggregating all the estimated HR patches into a whole image.

Inspired by ScSR [142], many sparse coding and dictionary learning based methods have been proposed for SISR. By relaxing the constraint that the LR/HR patch pair has the same coding vector, Wang et al. [129] introduced a transform matrix to allow more complex relationship between the HR and LR coding vectors, and proposed a semi-coupled dictionary learning (SCDL) method for SISR. Subsequently, more complex models have been proposed for better modeling the relationship between the LR and HR spaces with coupled dictionaries. He et al. [59] utilized a non-parametric Bayesian approach to learn dictionaries to build relationship between the LR and HR spaces. Peleg and Elad [108] proposed a statistical model which uses restricted Boltzmann machine (RBM) to model the relationship between the LR and HR coding vectors. Zhu et al. [157] suggested to enhance the flexibility of the HR dictionary by permitting certain deformation in each HR patch.

### 3.2.2 Convolutional sparse coding

Despite its wide applications, sparse coding on an image patch has some drawbacks. First, the scalability of the $\ell_0$ or $\ell_1$ optimization is poor, which limits the application of sparse coding in large scale problems. Second, most of the previous sparse coding based methods partition the whole image into overlapped patches to reduce the burden of modeling and computation. However, the consistency between overlapped patches is ignored and the existing aggregation and averaging strategies can only alleviate this problem.

To take consistency into account, Zeiler et al. [145] proposed a convolutional implementation of sparse coding to sparsely encode the whole image. Instead of decomposing a

signal vector as the multiplication of dictionary matrix and coding vector, the so-called convolutional sparse coding (CSC) model represents an image as the summation of convolutions of the feature maps and the corresponding filters:

$$\min_{\boldsymbol{Z}} \| \boldsymbol{X} - \sum_{i=1}^{N} \boldsymbol{f}_i \otimes \boldsymbol{Z}_i \|_F^2 + \lambda \sum_{i=1}^{N} \| \boldsymbol{Z}_i \|_1, \tag{3.1}$$

where $\boldsymbol{X}$ is an $m \times n$ input image, $\{\boldsymbol{f}_i\}_{i=1,2,\ldots,N}$ is a group of $s \times s$ filters, and $\boldsymbol{Z}_i$ is the feature map corresponding to $\boldsymbol{f}_i$ with size $(m+s-1) \times (n+s-1)$. In the CSC model, we do not need to partition the input image into overlapped patches, and the inconsistency problem of patch based implementation can be avoided.

On the other hand, the convolutional decomposition mechanism also brings some difficulties in optimization. Zeile et al. [145] adopted the continuation method to relax the equality constraints, and employed the conjugate gradient (CG) decent to solve the convolutional least square approximation problem. Bristow et al. [7] proposed a fast CSC algorithm by considering the property of block circulant with circulant block (BCCB) matrix, which solves the problem in the *Fourier* domain. Recently, Wohlberg [134] further improved the algorithm and proposed an efficient alternating direction method of multipliers (ADMM) for CSC.

Despite the study of fast algorithms to solve the CSC problem, little attention has been given on validating the advantages of CSC over conventional patch based sparse coding for image reconstruction. Can CSC benefit image reconstruction? In this chapter, we attempt to answer this question and develop an effective CSC based SISR algorithm.

### 3.3 Convolutional sparse coding for super resolution

In this section, we present our convolutional sparse coding based super-resolution (CSC-SR) method. Like most existing SISR methods, the proposed CSC-SR method also involves a training phase and a testing phase. In the training phase, we learn three groups of parameters: (i) LR filters; (ii) the mapping function between LR and HR feature maps; and (iii) HR filters. In the testing phase, the input LR image is first decomposed into sparse LR feature

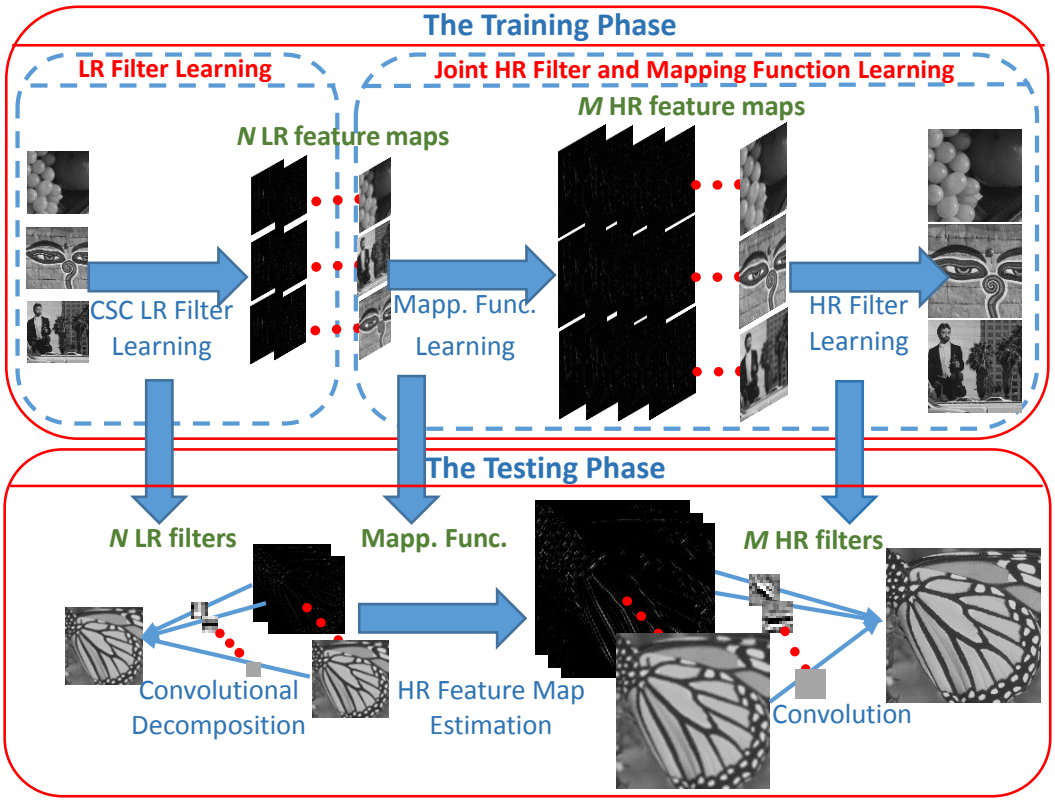



Figure 3.1. Flowchart of the proposed algorithm.

maps by using the learned LR filters. Then, the mapping function is employed to estimate HR feature maps from LR feature maps, and the HR image is reconstructed by simple convolution operation. The flowchart of our algorithm in the training and testing phases is shown in Fig. 3.1.

### 3.3.1 The training phase

In dictionary learning based SISR, a couple of dictionaries together with certain mapping function are generally used to model the relationship between LR and HR images. The LR and HR dictionary learning can be formulated into one objective function, and jointly learned using the training LR/HR patch pairs. However, for the joint dictionary learning methods in [129, 142], because the test HR image is not available, the mechanism of generating coding

vectors in training is different from that in testing, leading to the inconsistency of the coding vectors in the training and testing phases. Several strict joint learning models [88, 141] have been developed to avoid coding inconsistency, but they need to solve a bi-level optimization problem. On the other hand, recent studies [147] have also shown that promising SISR performance can be obtained via separate training of the LR and HR dictionaries. In [147], an LR dictionary is first learned using the LR image dataset, and then an HR dictionary is trained to reconstruct the HR image patches based on the sparse coding vectors of the corresponding LR patches. In this chapter, we extend the method in [147] to CSC, and learn the LR and HR filters separately for SISR.

Suppose that we are given a group of HR images $\{\mathbf{x}_1, \mathbf{x}_k, ..., \mathbf{x}_K\}$ together with the corresponding LR images $\{\mathbf{y}_1, \mathbf{y}_k, ..., \mathbf{y}_K\}$ for training. Because the image index $k$ will not affect the understanding of our algorithm, in the remainder of this chapter, we omit it for the purpose of simplicity.

In our SISR scheme, each LR image is decomposed into one smooth component and one residual component, and we apply different super-resolution prodedures to the two components. We simply apply bi-cubic interpolation to the low frequency LR feature map for the super-resolution of the smooth component, and propose a CSC-SR model for the super-resolution of the residual component.

To extract the smooth component of the LR image $\mathbf{y}$, we first solve the following optimization problem:

$$\min_{\mathbf{Z}} \| \mathbf{y} - \boldsymbol{f}^s \otimes \boldsymbol{Z}_y^s \|_F^2 + \gamma \| \boldsymbol{f}^{dh} \otimes \boldsymbol{Z}_y^s \|_F^2 + \gamma \| \boldsymbol{f}^{dv} \otimes \boldsymbol{Z}_y^s \|_F^2, \tag{3.2}$$

where $\boldsymbol{Z}_y^s$ is the low frequency feature map of LR image $\mathbf{y}$, $\boldsymbol{f}^s$ is a $3 \times 3$ low pass filter and $\boldsymbol{f}^{dh}$ and $\boldsymbol{f}^{dv}$ are the horizontal and vertical differential filters $[1, -1]$ and $[1; -1]$. Based on the special property of BCCB matrix, the closed form solution of (3.2) can be efficiently solved in the *Fourier* domain, and we can decompose the LR image as:

$$\mathbf{y} = \boldsymbol{f}^s \otimes \boldsymbol{Z}_y^s + \boldsymbol{Y},$$

where $\boldsymbol{f}^s \otimes \boldsymbol{Z}_y^s$ denotes the smooth component of the LR image, and $\boldsymbol{Y}$ denotes the residual component which represents the high frequency edge and texture structures in the LR image.

After the decomposition, we learn a group of LR filters to further decompose the residual component $\boldsymbol{Y}$ into $N$ feature maps:

$$\min_{\boldsymbol{Z},\boldsymbol{f}} \|\boldsymbol{Y} - \sum_{i=1}^{N} \boldsymbol{f}_i^l \otimes \boldsymbol{Z}_i^l \|_F^2 + \lambda \sum_{i=1}^{N} \| \boldsymbol{Z}_i^l \|_1, \ s.t. \ \| \boldsymbol{f}_i^l \|_F^2 \leq 1, \tag{3.3}$$

where $\{\boldsymbol{f}_i^l\}_{i=1\sim N}$ are $N$ LR filters, and $\boldsymbol{Z}_i^l$ is the sparse feature map of the $i$th filter. The constraint $\boldsymbol{f}_i^l \ \|_F^2 \leq 1$ is introduced to avoid the trivial solution $\boldsymbol{f} \to \infty, \boldsymbol{Z} \to \boldsymbol{0}$, which has been applied in many previous dictionary learning methods [1, 89].

Similar to other dictionary learning methods, we alternatively optimize the $\boldsymbol{Z}$ and $\boldsymbol{f}$ subproblems. The $\boldsymbol{Z}$ subproblem is a standard CSC problem that can be solved using the algorithm proposed in [134]. For the $\boldsymbol{f}$ subproblem, we need to optimize:

$$\boldsymbol{f}^l = \arg\min_{\boldsymbol{f}} \| \boldsymbol{Y} - \sum_{i=1}^{N} \boldsymbol{f}_i^l \otimes \boldsymbol{Z}_i^l \|_F^2, \ s.t. \| \boldsymbol{f}_i^l \|_F^2 \leq 1. \tag{3.4}$$

The $\boldsymbol{f}$ subproblem can be solved by the ADMM algorithm in the *Fourier* domain [134].

However, when ADMM is employed to solve (3.4), the feature maps of all the training images are required to be loaded in the memory. If the number of training images or the number of LR filters are large, the ADMM algorithm suffers from the problem of high memory demand for solving (3.4). Fortunately, the marriage of the recently developed stochastic average (SA) algorithms and ADMM, i.e., SA-ADMM[152], can be utilized to optimize (3.4). Different from standard ADMM, SA-ADMM adopts the linearization technique which can be deployed to avoid the computation of matrix inversion in our case, and utilizes the SA strategy to avoid the storage of feature maps of all the training images. More details of the optimization procedure can be found in Appendix A.

After the LR filters learning, we further learn the mapping function and the HR filters based on the LR feature maps and the corresponding HR images. Like the LR images, each HR image is decomposed into one smooth component and one residual component. First, bi-cubic interpolation is adopted to enlarge $\boldsymbol{Z}_y^s$, obtaining the low frequency HR feature map

$\boldsymbol{Z}_x^s$. Then, the original HR image can be decomposed as:

$$\mathbf{x} = \boldsymbol{f}^s \otimes \boldsymbol{Z}_x^s + \boldsymbol{X},$$

where $\boldsymbol{f}^s \otimes \boldsymbol{Z}_x^s$ denotes the smooth component, and $\boldsymbol{X}$ denotes the residual component which conveys the high frequency edge and texture structures of HR image $\mathbf{x}$. Given the training set of LR feature maps and HR images, we are able to learn the HR filters and the corresponding feature mapping function.

In most of the previous dictionary learning based SR methods, the LR image is first interpolated to the same size as the HR image, and the sizes of the HR and LR dictionaries are the same. Here we show that a small number of LR filters with small filter size can also achieve satisfactory SISR results while saving the decomposition time in both the training and the testing phases. Thus, we directly perform CSC on the small LR image that is much smaller than the HR image. Furthermore, since the HR image is much more complex than the LR image, we propose to decompose the LR image by a small number of LR filters to reduce the computation burden, while reconstructing the HR image by a larger number of HR filters with more flexible representation capacity.

However, one challenge of the scheme above is that a mapping function needs to be trained to zoom the input LR feature maps to a higher resolution in terms of both spatial size and feature map number. To this end, we propose to train a mapping function between the LR and HR feature maps:

$$\boldsymbol{Z}_j^h(kx, ky) = g(\boldsymbol{Z}_1^l(x, y), \boldsymbol{Z}_2^l(x, y), ..., \boldsymbol{Z}_N^l(x, y); \boldsymbol{W}), \tag{3.5}$$

where $k$ is the zooming factor, $\boldsymbol{Z}_j^h(kx, ky)$ is the coefficient in position $(kx, ky)$ of feature map $\boldsymbol{Z}_j^h$, $\boldsymbol{Z}_i^l(x, y)$ is the coefficient in the corresponding point $(x, y)$ in feature map $\boldsymbol{Z}_i^l$, and $\boldsymbol{W}$ is the parameter of mapping function $g(\bullet)$. For $\boldsymbol{Z}_j^h(x', y')$ with $mod(x', k) \neq 0$ or $mod(y', k) \neq 0$, we simply set $\boldsymbol{Z}_j^h(x', y') = 0$.

The function $g(\bullet)$ should have the ability to generate sparse output from sparse input, and we use a sparse linear transformation matrix to estimate the HR coefficient:

$$\boldsymbol{Z}_j^h(kx, ky) = g(\boldsymbol{Z}_:^l(x, y); \boldsymbol{w}_j) = \boldsymbol{w}_j^T \boldsymbol{z}_:^l(x, y), \;\; s.t. \boldsymbol{w}_j \succeq 0, |\boldsymbol{w}_j|_1 = 1, \tag{3.6}$$

where $\boldsymbol{z}_{:}^{l}(x,y)$ is a vector containing all the coefficients in point $(x,y)$ of the $N$ LR feature maps, and $\boldsymbol{w}_j$ is the transformation vector for the HR feature map $\boldsymbol{Z}_j^h$. We let $\boldsymbol{w}_j \succeq 0$ and $|\boldsymbol{w}_j|_1 = 1$ to ensure the sparsity of $\boldsymbol{W}$. The non-negative simplex constraint used in (3.6) is stronger than some sparsity regularizer (e.g., $\ell_1$ norm). Another thing needs to be noticed is that both the number and size of LR feature maps are enlarged by the mapping function. Compared with the coefficients in the LR feature map, each coefficient in the HR feature map includes the spatial information from a larger local area. The spatial size of HR filters should also be set larger to reconstruct the HR image.

After choosing the form of mapping function, our joint HR filter and mapping function learning model is formulated as:

$$\begin{aligned} \{\boldsymbol{f}^h, \boldsymbol{W}\} = \min_{\boldsymbol{f},\boldsymbol{W}} \| \boldsymbol{X} - \sum\nolimits_{j=1}^{M} \boldsymbol{f}_j^h \otimes g(\boldsymbol{Z}_{:}^{l}; \boldsymbol{w}_j) \|_F^2, \\ s.t. \ \| \boldsymbol{f}_j^h \|_F^2 \leq e; \quad \boldsymbol{w}_j \succeq 0, |\boldsymbol{w}_j|_1 = 1, \end{aligned} \tag{3.7}$$

where $e$ is a scalar to constrain the energy of HR filters, and the specific number of $e$ for different zooming factor will be introduced in the parameter setting section of this chapter. Since the size of HR filter is different from the size of LR filter, the energy constraint should also be different. We optimize the objective function by alternatively updating the filter $\boldsymbol{f}^h$ and the mapping function parameter $\boldsymbol{W}$. For fixed $\boldsymbol{W}$, the filter updating subproblem defined in (3.4) can be solved by the SA-ADMM algorithm. For fixed $\boldsymbol{f}$, the subproblem on $\boldsymbol{W}$ is more complex, and we need to solve the following optimization problem:

$$\{\boldsymbol{W}\} = \arg\min_{\boldsymbol{W}} \| \boldsymbol{X} - \sum_{j=1}^{M} \boldsymbol{f}_j^h \otimes g(\boldsymbol{Z}_{:}^{l}; \boldsymbol{w}_j) \|_F^2, \ \ s.t. \ \ \boldsymbol{w}_j \succeq 0, |\boldsymbol{w}_j|_1 = 1. \tag{3.8}$$

We also solve (3.8) by the SA-ADMM algorithm. Please refer to the Appendix A for the details of the optimization procedure.

With the optimization algorithms for solving the $\boldsymbol{f}$ and $\boldsymbol{W}$ subproblems, we summarize the training algorithm for our CSC-SR method in Algorithm 3.1.

**Algorithm 3.1** The Training Algorithm for Convolutional Sparse Coding based Super Resolution (CSC-SR)

**Input:** Training image pairs $\{\mathbf{x}, \mathbf{y}\}$, LR&HR filter number $N$ and $M$, LR&HR filter sizes $s^l$ and $s^h$, regularization parameter $\gamma$ and $\lambda$;

1: Solve (3.2) to decompose LR image, get high frequency component $\boldsymbol{Y}$ of LR image;
2: Solve the CSC filter learning problem on $\boldsymbol{Y}$, get $\boldsymbol{Z}^l$ and $\boldsymbol{f}^l$;
3: Extract low frequency component of the bi-cubic interpolated LR image, get the texture structure of HR image $\boldsymbol{X}$;
4: Learn the HR filters and the mapping function;

**Output:** LR filters $\boldsymbol{f}^l$, HR filters $\boldsymbol{f}^h$ and mapping function $\boldsymbol{W}$

### 3.3.2 The testing phase

After training, we have the LR filters $\{\boldsymbol{f}^l\}$, HR filters $\{\boldsymbol{f}^h\}$ and the mapping function $g(\bullet; \boldsymbol{W})$. Given a testing LR image $\mathbf{y}$, we extract its texture structure and decompose it by the LR filters to get LR sparse feature maps $\{\boldsymbol{Z}^l\}$. Then, the HR feature map can be estimated by the function $\{\boldsymbol{Z}^h\} = g(\boldsymbol{Z}^l; \boldsymbol{W})$. Finally, the high frequency texture structure in the HR output image is obtained by the summation of the convolutions of the HR feature maps and the corresponding HR filters:

$$\widehat{\boldsymbol{X}} = \sum\nolimits_{j=1}^{M} \boldsymbol{f}_j^h \otimes \boldsymbol{Z}_j^h. \tag{3.9}$$

We can then combine $\widehat{\boldsymbol{X}}$ with the smooth component to generate the final HR estimation. To achieve better SR performance, the back projection operation which is widely used in other sparse coding based SR methods [59, 142, 157] can also be utilized to improve the final HR estimation.

## 3.4 Experimental results

In this section, we first provide a brief convergence analysis of the proposed training algorithm. Then, we present an experiment to illustrate the advantages of the convolutional decomposition mechanism over the patch based method. After a discussion of parameter setting, we compare our algorithm with representative SR methods.

The experimental setting is the same as [142]. LR training and testing images are generated by resizing the HR groundtruth image by bi-cubic interpolation. Using the same

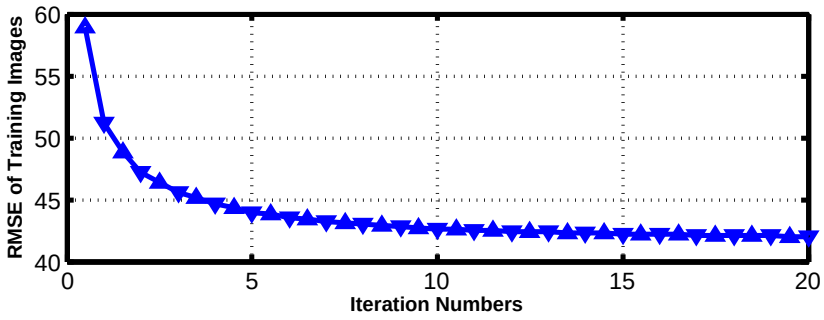



Figure 3.2. The convergence curve in the joint HR filter and mapping function training.

91 training images provided by Yang et al. [142], we randomly crop 1,000 60×60 smaller images from these images to train our model. Then, the corresponding LR training images with zooming factors 2 and 3 are of size $30 \times 30$ and $20 \times 20$, respectively. To avoid boundary effects of *Fourier* domain implementation, 8 pixels are padded on the image boundary.

### 3.4.1 Convergence analysis

In our CSC-SR training algorithm, apart from training filters to decompose the LR images, model (3.7) is also proposed to jointly train the HR filters and mapping function. The objective function in (3.7) is a bi-convex optimization problem [50]. For fixed $\boldsymbol{W}$, the problem is convex to $\boldsymbol{f}$, and for fixed $\boldsymbol{f}$, the function is convex to $\boldsymbol{W}$. We alternatively optimize the $\boldsymbol{f}$ and the $\boldsymbol{W}$ sub-problems, which is actually an alternate convex search (ACS) algorithm [50]. Since our objective function has a general lower bound 0, if we can get the optimal solutions of updating $\boldsymbol{W}$ and $\boldsymbol{f}$, the joint HR filter and mapping function training is guaranteed to converge in terms of function energy.

It is empirically found that the optimization of joint HR filter and mapping function training converges rapidly. Fig. 3.4.1 shows the convergence curve of our algorithm in an experiment with 200 training images. Because the energy of objective function is proportional to the number of pixels in training images, in Fig. 3.4.1 the energy of objective function is normalized by the pixel number of training images. The symbol "△" represents the root

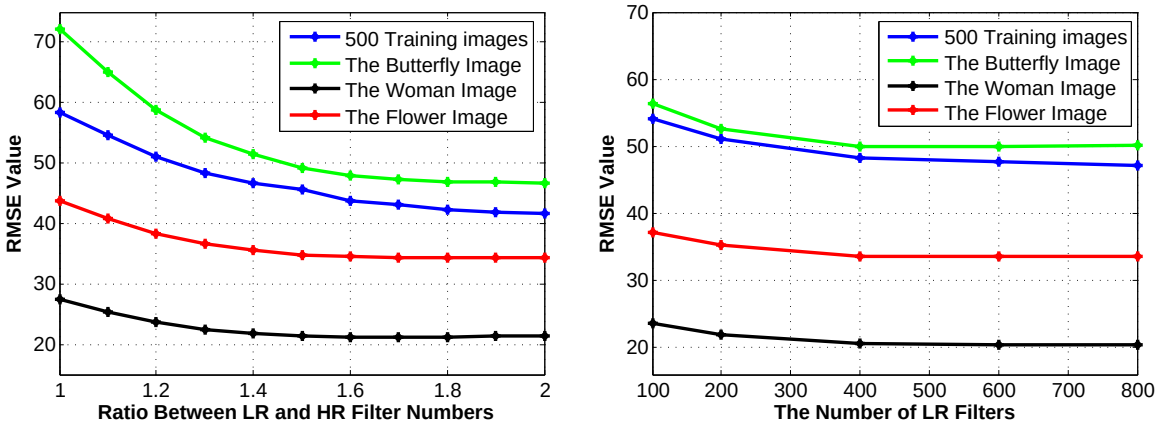
mean square error (RMSE) between the training images and their HR estimates after updating filters $\boldsymbol{f}$ and the symbol "$\bigtriangledown$" shows the RMSE after updating the mapping function $g(\bullet; \boldsymbol{W})$. In most of our experiments, our algorithm will converge in 10 iterations.

### 3.4.2 CSC *vs.* sparse coding for SR

Table 3.1. SR results (PSNR, dB) by patch based sparse coding method ScSR [142] and the proposed convolutional based sparse coding method (without mapping function learning).

| | Zooming Factor 2 | | | |
|---|---|---|---|---|
| | $\mathbf{ScSR}_{256}$ | $\mathbf{ScSR}_{512}$ | $\mathbf{CSC}_{256}$ | $\mathbf{CSC}_{512}$ |
| Butterfly | 30.43 | 31.10 | 30.97 | 31.56 |
| Bird | 40.02 | 40.44 | 40.20 | 40.51 |
| Comic | 27.75 | 27.98 | 27.90 | 28.10 |
| Woman | 34.48 | 34.89 | 34.62 | 34.99 |
| Foreman | 36.18 | 36.49 | 36.46 | 36.56 |

To validate our argument that global decomposition by convolution is more appropriate for SR, we compare the convolutional based CSC and a representative patch based sparse coding (SC) method. The ScSR [142] method is a typical patch based SC method for SR. It trains a pair of HR and LR dictionaries on the training set, and uses the sparse coding coefficients of the LR image to reconstruct the HR image by the HR dictionary. To have a fair comparison between CSC and SC methods, we omit the mapping function introduced in our method, and train a pair of LR filters and HR filters to reconstruct the LR and HR images with the same representation feature map. In the testing phase, we decompose the interpolated LR image and use exactly the same feature map to reconstruct the HR estimation. The SR resluts (PSNR) by different methods (with dictionary size 256 and 512) on 5 images are shown in Table 3.4.2. The results on other images are similar. We see that CSC-SR is always




(a) (b)

Figure 3.3. (a) The RMSE values with different HR/LR filter number ratio on the training dataset and 3 testing images. (b) The RMSE values with different LR filter number on the training dataset and 3 testing images.

better than ScSR with the same number of dictionary atoms.

### 3.4.3 Parameters setting

A key parameter in all the dictionary based image reconstruction methods is the number of dictionary atoms. With a large number of dictionary atoms, we are able to capture the image sparsity property better, but suffer from heavier space and time complexity. Here, we validate the effectiveness of using different filter numbers and choose an appropriate ratio between the LR filter number $N$ and HR filter number $M$. We train different models on 500 images. The number of LR filters is fixed to 200 and the ratio between HR and LR filters is set from 1 to 2 with step length 0.1. The RMSE values on training images and 3 testing images are shown in Fig.3.4.3 (a). Compared with ratio 1, using more HR filters can provide better HR estimation. In all of our following experiments, we set the ratio between HR filter number and LR filter number as 3/2 to make a balance between SR performance and algorithm complexity.

Besides the ratio between LR and HR filter numbers, another important parameter in our algorithm is the number of LR filter number. We test a wide range of LR filter numbers

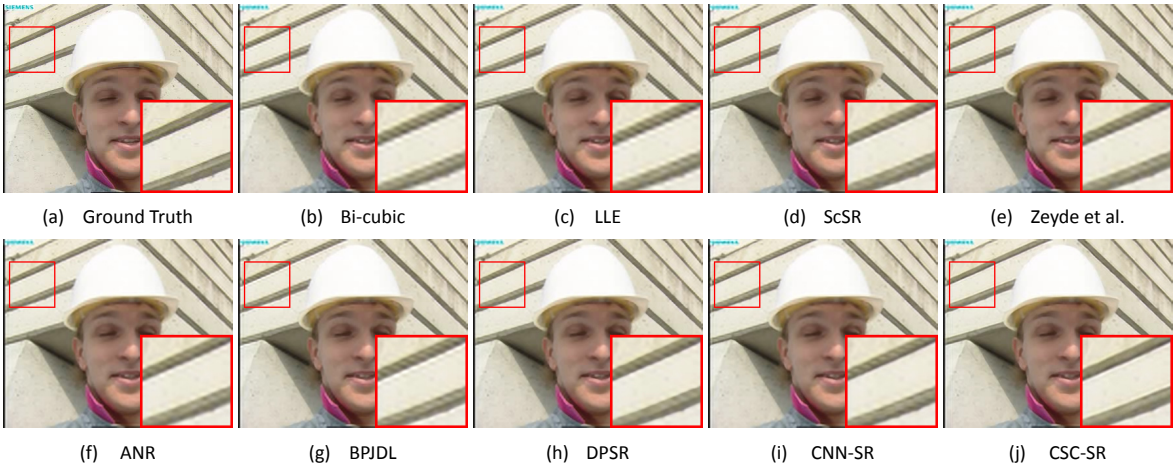



Figure 3.4. Super resolution results on image *Foreman* by different algorithms (zooming factor 3).

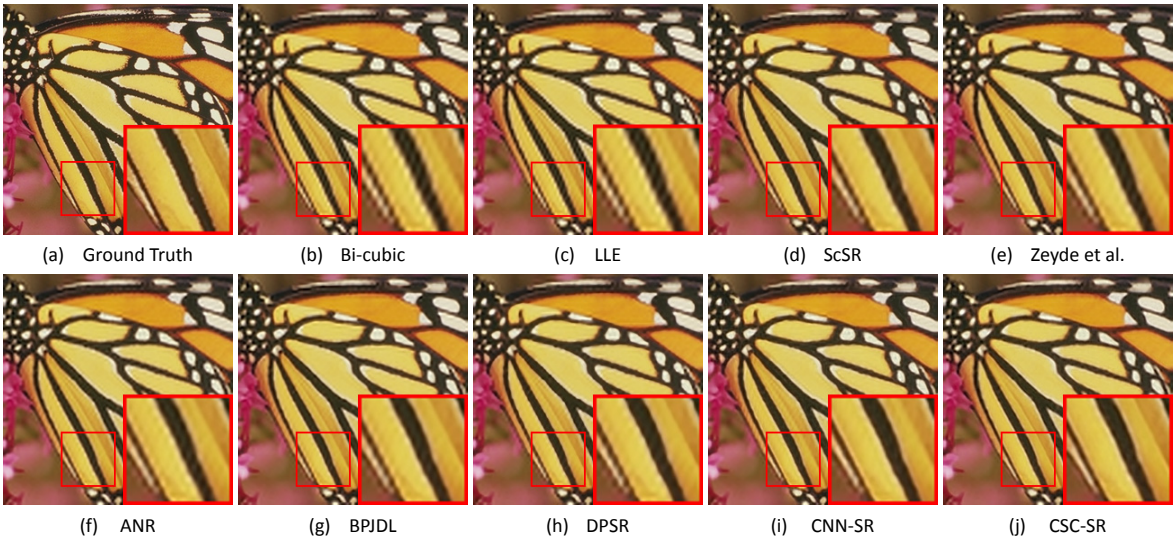



Figure 3.5. Super resolution results on image *Butterfly* by different algorithms (zooming factor 3).


Figure 3.6. Super resolution results on image *Lena* by different algorithms (zooming factor 3).

with 500 training images, and the SR results with different LR filter numbers are shown in Fig.3.4.3 (b). Generally, a larger LR filter number leads to better SR results. To achieve the best performance, we train 800 LR filters in the following experiments.

Other parameters include the size of LR and HR filters, regularization parameters $\gamma$ and $\lambda$, and the HR filter energy constraint parameter $e$. In all our following experiments, we set the size of LR filter as 5 and set the size of HR filter as $5\times$*Zooming Factor*. The regularization parameters $\gamma$ and $\lambda$ are set as 30 and 0.02, and the energy constraint parameter $e$ is set as 4 and 9 for zooming factors 2 and 3, respectively.

### 3.4.4 Comparison with state-of-the-arts

In this section, we compare the proposed CSC-SR methods with several state-of-the-art SR methods. The comparison methods include ScSR [142], LLE [22], the Zeyde′s method [147], anchored neighborhood regression method (ANR) [123], the Beta process joint dictionary learning method (BPJDL) [59], deformable patch super resolution method (DPSR) [157] and the recently proposed convolutional neural network based method CNN-SR [36]. All methods follow the experimental setting of [142], in which the LR images are resized from

Table 3.2. Super resolution results (PSNR, dB) by different methods.

| | **Zooming Factor=2** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **LLE** | **ScSR** | **Zeyde** | **ANR** | **BPJDL** | **DPSR** | **CNN** | **CSC** |
| Butterfly | 28.99 | 31.33 | 30.91 | 30.65 | 31.43 | 31.28 | **32.20** | 31.97 |
| Face | 35.57 | 35.69 | 35.69 | 35.70 | **35.75** | 35.64 | 35.60 | 35.68 |
| Bird | 38.93 | 40.53 | 40.25 | 40.23 | 40.98 | 39.77 | 40.63 | **41.40** |
| Comic | 27.34 | 28.02 | 27.89 | 27.92 | 28.24 | 27.98 | 28.28 | **28.43** |
| Woman | 33.71 | 34.95 | 34.74 | 34.70 | 35.23 | 34.64 | 34.93 | **35.27** |
| Foreman | 35.43 | 36.89 | 36.27 | 36.44 | 36.49 | **36.84** | 36.19 | 36.59 |
| Coast. | 30.30 | 20.60 | 30.61 | 30.56 | **30.63** | 30.55 | 30.49 | **30.63** |
| Flowers | 31.72 | 32.73 | 32.52 | 32.43 | 32.91 | 32.49 | 33.04 | **33.14** |
| Zebra | 32.54 | 33.46 | 33.58 | 33.36 | 33.62 | 33.22 | 33.30 | **33.74** |
| Lena | 35.95 | 36.46 | 36.39 | 36.42 | 36.58 | 36.27 | 36.48 | **36.63** |
| Bridge | 27.43 | 27.67 | 27.70 | 27.62 | 27.77 | 27.58 | 27.70 | **27.83** |
| Baby | 38.31 | 38.41 | 38.46 | 38.55 | **38.54** | 38.29 | 38.41 | 38.43 |
| Peppers | 35.82 | 36.72 | 36.60 | 36.38 | 36.71 | 36.55 | 36.75 | **36.88** |
| Man | 30.14 | 30.70 | 30.60 | 30.57 | 30.80 | 27.56 | 30.82 | **30.96** |
| Barbara | 28.59 | 28.70 | **28.75** | 28.62 | 28.68 | 28.60 | 28.59 | 28.73 |
| **AVE.** | 32.719 | 32.964 | 33.397 | 33.343 | 33.624 | 33.156 | 33.553 | **33.754** |

Table 3.3. Super resolution results (PSNR, dB) by different methods.

| | **Zooming Factor=3** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **LLE** | **ScSR** | **Zeyde** | **ANR** | **BPJDL** | **DPSR** | **CNN** | **CSC** |
| Butterfly | 24.93 | 26.27 | 26.07 | 25.96 | 26.37 | 26.95 | **27.54** | 27.06 |
| Face | 33.41 | 33.53 | 33.63 | 33.67 | 33.46 | 33.62 | 33.58 | **33.74** |
| Bird | 33.78 | 34.33 | 34.67 | 34.63 | 34.44 | 34.73 | 34.83 | **35.46** |
| Comic | 23.81 | 24.09 | 24.11 | 24.13 | 24.14 | 24.24 | **24.41** | 24.39 |
| Woman | 29.61 | 30.32 | 30.50 | 30.39 | 30.45 | 30.72 | 30.88 | **31.11** |
| Foreman | 31.74 | 32.55 | 32.46 | 32.60 | 31.96 | 33.04 | 32.24 | **33.09** |
| Coast. | 26.98 | 26.97 | 27.21 | 27.10 | 27.01 | 27.16 | 27.18 | **27.23** |
| Flowers | 28.10 | 28.56 | 28.62 | 28.60 | 28.65 | 28.83 | **29.01** | 29.00 |
| Zebra | 28.02 | 28.61 | 28.82 | 28.67 | 28.75 | 28.98 | 28.91 | **29.23** |
| Lena | 32.62 | 33.06 | 33.17 | 33.17 | 33.14 | 33.26 | 33.41 | **33.53** |
| Bridge | 24.91 | 25.02 | 25.10 | 25.04 | 24.98 | 25.06 | 25.05 | **25.14** |
| Baby | 34.85 | 34.96 | 35.24 | 35.20 | 35.15 | **35.25** | 35.01 | 35.18 |
| Peppers | 32.91 | 33.46 | 33.78 | 33.58 | 33.64 | 33.94 | 34.06 | **34.12** |
| Man | 26.31 | 26.66 | 26.70 | 26.68 | 26.73 | 24.83 | 26.87 | **26.97** |
| Barbara | 26.78 | 26.72 | **26.86** | 26.75 | 26.83 | 26.82 | 26.66 | 26.71 |
| **AVE.** | 29.249 | 29.674 | 29.796 | 29.744 | 29.713 | 29.829 | 29.976 | **30.131** |

ground truth HR images by bi-cubic interpolation. We download the source codes from the author′s websites, and use the recommended parameters by the authors.

We perform the SR comparison on 15 widely used test images. The PSNR values by the competing methods are shown in Table 3.3. CSC-SR achieves better results than patch-based joint dictionary learning methods on most of testing images. Compared with the state-of-the-art CNN-SR method, the proposed CSC-SR methods also achieves higher PSNR index on many testing images. Overall, CSC-SR improves the average PSNR value of CNN-SR with 0.2 dB and 0.15 dB for zooming factors 2 and 3, respectively.

Let′s then compare the visual quality of the SR results. In Figures 3.4.3, 3.4.3 and 3.4.3, we show the SR results of images *Foreman*, *Butterfly* and *Lena* by the competing algorithms. As highlighted in the small window, the SR results by other competing algorithms have obvious ringing artifacts in strong edge area, while the edges reconstructed by the CSC-SR method are more natural. In summary, the results generated by the proposed CSC-SR method have more textures and less artifacts, producing visually more pleasant SR outputs. More examples of SR results can be found in the supplementary file.

## 3.5 Conclusion

To address the inconsistency issue in previous patch-based synthesis models, we proposed a convolutional sparse coding based super resolution (CSC-SR) method. CSC directly decomposes the whole image by filtering, which naturally takes the consistency of pixels in overlapped patches into consideration. We introduced a mapping function between the L-R and HR sparse coding feature maps for SR. Different from previous patch based sparse coding methods, the convolutional decomposition mechanism of CSC can keep the spatial information of input signal in the feature maps, and exploit the consistency of neighboring patches for better image reconstruction. Compared with other state-of-the-art SR methods, our algorithm achieves not only very competitive PSNR index, but also more pleasant visual quality of image texture and edge structures.

# CHAPTER 4

# JOINT CONVOLUTIONAL ANALYSIS AND SYNTHESIS SPARSE REPRESENTATION FOR SINGLE IMAGE LAYER SEPARATION

In the previous two chapters, we studied the analysis and synthesis models, respectively. Generally speaking, the two models have compensate capabilites on image modeling. The synthesis sparse representation (SSR) models represent a signal as the linear combination of a small number of atoms chosen out of a dictionary, while the analysis sparse representation (ASR) model imposes sparsity on the responses of the signal to analysis operators. Such complementary representation mechanisms of SSR and ASR make them be advantageous in characterizing large-scale structures (e.g., edges) and fine-scale textures, respectively. In this chapter, to exploit their complementary representation mechanisms, we integrate the two models and propose a joint convolutional analysis and synthesis (JCAS) sparse representation model for image decomposition. The convolutional implementation is adopted to more effectively exploit the image global information. In the proposed JCAS model, a single image is adaptively decomposed into two layers, one is used for SSR and the other for ASR. In addition, the synthesis dictionary is adaptively learned from the given image to capture the texture pattern for specific tasks. The developed JCAS model exhibits very encouraging performance in many single image layer separation tasks.

## 4.1 Introduction

### 4.1.1 Image layer decomposition

Image layer separation aims to decompose the input image as the summation of two or more components for certain tasks. Based on different requirements of these tasks, a variety of decomposition models have been suggested [74, 76, 80, 95] by assuming different priors for the decomposition results. In this section, we provide a brief review on the tasks of rain streak removal and texture cartoon decomposition, which are two typical image layer decomposition tasks.

Rain streak removal aims to separate a rainy image into a rain-free background layer and a rain streak layer. The key issue of this task is to appropriately characterize the two layers. By assuming that rain streaks only appear in the high-frequency part of the image, some models [24, 80, 87] were designed by first decomposing the image into low-frequency and high-frequency parts in a classical manner, and then separating the rain streak layer from background details in the high-frequency part. However, these methods often over-smooth image details and generate blurry background estimation. Recent works have been proposed to directly extract the rain streak layer from the input image, e.g., the discriminative sparse coding method [87] and the Gaussian mixture models (GMM) [80]. These methods adopt the same type of models to characterize the background part as well as the rain streak part, and external data are used to train dictionaries or GMM models for the two layers. Different from the existing models for this task, the proposed JCAS model takes benefit from the complementary capabilities of ASR and SSR, and yields an effective rain streak removal method without any extra training data.

Another important layer separating application is texture-cartoon decomposition. Given an input image $\boldsymbol{Y}$, texture-cartoon decomposition aims to separate it into a cartoon (piecewise smooth) layer and a texture layer. Following the influential work in [95], most of current methods design specific priors for the two layers. One commonly used formulation is shown as follows:

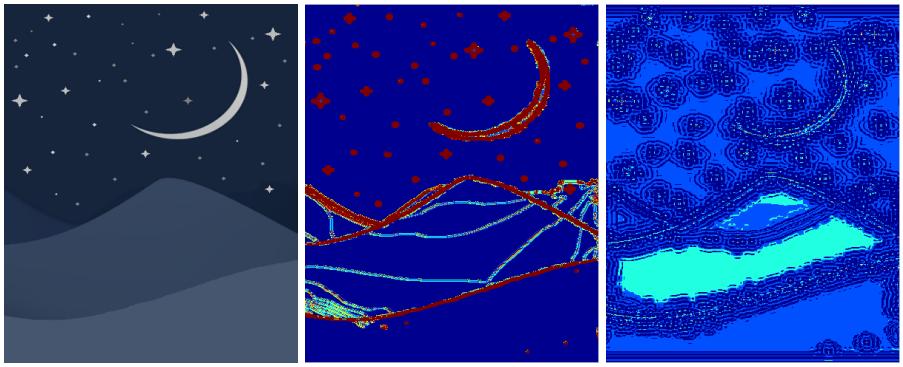
$$min_{\boldsymbol{U},\boldsymbol{V}}\|\boldsymbol{Y}-\boldsymbol{U}-\boldsymbol{V}\|_F^2+\lambda P_1(\boldsymbol{U})+\gamma P_2(\boldsymbol{V}), \tag{4.1}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ represent the cartoon and texture components of the original image $\boldsymbol{Y}$, respectively. The first term $\|\boldsymbol{Y}-\boldsymbol{U}-\boldsymbol{V}\|_F^2$ maintains the fidelity between the original image and the estimated layers. The regularization functions $P_1(*)$ and $P_2(*)$ encode the different characteristics of layers $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. The positive constants $\lambda$ and $\gamma$ are regularization parameters. Different functional forms of $P_1(*)$ and $P_2(*)$ have been investigated for the texture-cartoon decomposition task. $P_1(*)$ imposed on the cartoon layer $\boldsymbol{U}$ is often set as the analysis-based TV regularizer or its extensions [9]. How to properly set the forms of the regularizers on $\boldsymbol{V}$, however, still lacks consensus. Some works design this regularizer on the original image [20] while other on its certain transformations [3]. Some methods do not use any regularizers for texture representation [137].

### 4.1.2 Motivation

As introduced in chapter 2, the analysis-based sparse representation (ASR) methods represent a signal by modeling the distribution of its projection responses over basis (analysis dictionary). While, the synthesis-based sparse representation (SSR) models characterize a signal by regularizing the sparseness of synthesis coefficients, i.e., the signal should be able to be reconstructed by only a few atoms of the dictionary. It is easy to see that though both models aim to exploit the image sparsity, they emphasize different aspects of image characteristics and have intrinsic differences.

ASR characterizes the complement subspace of a signal. The signal is projected onto all the bases in the analysis dictionary, and each analysis dictionary atom contributes to modeling the signal subspace. Such a mechanism provides a robust prior for the principal component of the signal, and performs well on approximating the major structure of the image [41]. However, it adopts zero coefficients to indicate the orthogonality between the signal and the corresponding basis, and thus cannot take benefit from increased redundancy of the analysis dictionary. As a result, ASR has limited capacity in modeling textures with complex patterns, even when textures appear repetitively across the entire image. Comparatively, SSR characterizes the union of signal subspaces by selecting several dictionary atoms



(a) (b) (c)

Figure 4.1. Sparsity maps by analysis and synthesis models. (a) Input image. (b) Number of nonzeros in analysis sparse representation map. (c) Number of nonzeros in synthesis sparse representation map. Dark blue indicates coefficients with less nonzeros and red indicates coefficients with more than five nonzeros.

to reconstruct the signal, and accordingly is able to take benefit from a highly redundant dictionary [41]. Given an appropriate dictionary (learned from training data), repetitive signals with specific patterns can be reconstructed with highly sparse coefficients. However, due to the synthesis reconstruction mechanism, SSR is not as effective as ASR in characterizing image large structures. In Fig. 4.1, we utilize an ASR and an SSR models to approximate the same input image, and show the number of non-zeros in their respective coefficient maps. As clearly depicted in Fig. 4.1, the analysis coefficients obtained by ASR can be very sparse in smooth area, while the coefficients are denser in texture areas. The right image in Fig. 4.1 shows the sparsity of synthesis coefficients in each area. We train the synthesis dictionary on the input image, and adopt convolutional sparse coding to approximate the input image. One can see that SSR needs more nonzero coefficients to approximate the smooth area.

In this chapter, in order to take the advantage of both ASR and SSR, we propose a joint convolutional analysis and synthesis model (JCAS) for image layer decomposition. More specifically, we propose to use ASR and SSR to approximate the two components $\boldsymbol{U}$ and $\boldsymbol{V}$ of image $\boldsymbol{Y}$, respectively. Although sparsity prior has been widely utilized in previous image decomposition methods [20, 87, 95, 137], most of these methods use the same type of sparsity model (with different parameters) to characterize different layers. To the best of our knowledge, this is the first work to utilize the complementary property of ASR and SSR for

image decomposition. Furthermore, we adopt a convolutional implementation for the SSR part. Such a convolutional sparse coding approach avoids patch-dividing and enables us to utilize only several atoms to model the complex (but highly repetitive) textures in an image.

## 4.2 Our method

### 4.2.1 JCAS model

Single image layer separation is an ill-posed problem, and thus priors of the desired solution are required to provide supplementary information. In the proposed JCAS model, a synthesis-based prior model and an analysis-based prior model are utilized to regularize the two layers, respectively.

Using the MAP estimator, the separation results can be achieved by solving the following objective function:

$$min_{\boldsymbol{U},\boldsymbol{Z}} \|\boldsymbol{Y}-\boldsymbol{U}-\sum\nolimits_{j}^{N}\boldsymbol{f}_{S,j}\otimes\boldsymbol{Z}_j\|_F^2+\lambda\sum_{i}^{M}\|\boldsymbol{f}_{A,i}\otimes\boldsymbol{U}\|_1+\gamma\sum_{j}^{N}\|\boldsymbol{Z}_j\|_1, \tag{4.2}$$

where $\|*\|_1$ is the $\ell_1$ norm, and $\lambda$ and $\gamma$ are regularization parameters imposed on the analysis and synthesis prior terms, respectively. Here we model the layer $\boldsymbol{V}$ as $\boldsymbol{V}=\sum_j^N \boldsymbol{f}_{S,j}\otimes\boldsymbol{Z}_j$, where $\boldsymbol{f}_{S,j}$ is the $j$-th atom of convolutional synthesis dictionary, $\boldsymbol{Z}_j$ is its corresponding coefficient map, and "$\otimes$" denotes the convolution operation. Note that we use the convolutional sparse coding for SSR to avoid partitioning the image into patches.

The analysis prior terms $\{\|\boldsymbol{f}_{A,i}\otimes\boldsymbol{U}\|_1\}_{i=1,\ldots,M}$ are introduced to characterize the $\boldsymbol{U}$ component by regularizing the sparseness of its filter responses over analysis filters. As discussed in the previous sections, ASR is capable of better modeling the major structure of an image. Thus, the $\boldsymbol{U}$ layer is corresponding to the cartoon and background layers in the texture-cartoon decomposition and rain streak removal applications, respectively. For layer $\boldsymbol{V}=\sum_j^N \boldsymbol{f}_{S,j}\otimes\boldsymbol{Z}_j$, we regularize its synthesis coefficients $\{\boldsymbol{Z}\}_{j=1,\ldots,N}$ over convolutional synthesis dictionary $\{\boldsymbol{f}_{S,j}\}_{j=1,\ldots,N}$. Compared with ASR, SSR is a highly effective model to reconstruct complex but repetitive textures. Thus, approximating the texture and rain streak

components with the synthesis layer $\boldsymbol{V}$ will lead to a lower energy of the objective function.

### 4.2.2 Choice of dictionaries

In (4.2), an analysis dictionary $\{\boldsymbol{f}_{A,i}\}_{i=1,\ldots,M}$ and a synthesis dictionary $\{\boldsymbol{f}_{S,j}\}_{j=1,\ldots,N}$ are adopted to assign priors for the two image layers, respectively. In this section, we present details on how to properly choose the two dictionaries.

The specification of dictionary in a sparse coding model plays an important role in deducing an appropriate sparse representation of the input signal [112]. The early studies on sparse representation often utilize mathematical tools to analyze the signal data, and directly design and fix a class of functions as the dictionary for data representation in a hand-craft manner. During the last decade, in order to get a finer adaption to specific instances of the data, dictionary learning methods have been investigated from different points of view [112]. Compared with hand-crafted dictionaries, the dictionary learned from data is often capable of delivering better results due to its adaptability to the targeted scenario. However, for those applications (such as texture-cartoon decomposition) where training data are hard to collect to train the desired dictionary, hand-crafted dictionary is still more preferred due to its simplicity and efficiency. In this chapter, we utilize different strategies to set the two dictionaries for ASR and SSR, based on their different characteristics.

ASR utilizes the analysis dictionary to model the rareness of a signal, and each dictionary atom will be compared with the signal (by the inner product). Although this limits the employment of a highly redundant dictionary to provide flexible prior, it makes ASR a very robust model in capturing major structures of an image. Even with an extremely simple analysis dictionary (e.g. the gradient operators), some algorithms can achieve very competitive results in different applications [114]. Thus, in this chapter, we directly adopt the simple gradient operators (1st order and 2nd order) as our analysis dictionary for fast decomposition.

Different from the ASR model, the SSR method selects dictionary atoms to reconstruct the given signal. Having an over-complete dictionary, SSR is able to reconstruct the

input signal with very sparse coefficients. However, hand-crafted dictionary is hard to reconstruct the complex image structures using only a few atoms, and thus dictionary learning method is required to learn synthesis dictionary from training data [1]. In this chapter, we learn a convolutional synthesis dictionary of the texture layer from the input image itself. Such a strategy not only avoids the requirement of external training data with candidate texture types, but also makes JCAS be able to represent the texture layer with only several atoms. The detailed synthesis dictionary learning method will be introduced in the following optimization section.

### 4.2.3 Optimization

As introduced in the previous sections, our method learns the synthesis dictionary during the decomposition process. Thus, for the objective function in (4.2), the synthesis dictionary $\{\boldsymbol{f}_{S,j}\}_{j=1,\dots,N}$ is a variable to be optimized. We rewrite the convolution in a matrix multiplication form, and add some constraints to ensure the boundness of the synthesis filters. The new objective function for our JCAS model has the following form:

$$\begin{aligned} min_{\boldsymbol{u},\boldsymbol{f}_S,\boldsymbol{z}} \|\boldsymbol{y}-\boldsymbol{u}-\sum_j \boldsymbol{F}_{S,j}\boldsymbol{z}_j\|_2^2+\lambda\sum_i\|\boldsymbol{F}_{A,i}\boldsymbol{u}\|_1+\gamma\sum_j\|\boldsymbol{z}_j\|_1, \\ s.t.\ \ \|\boldsymbol{f}_{S,j}\|_F^2 \leq 1, \end{aligned} \tag{4.3}$$

where $\boldsymbol{y}$, $\boldsymbol{u}$ and $\boldsymbol{z}_j$ are the vectorization of image $\boldsymbol{Y}$, background or cartoon layer $\boldsymbol{U}$ and feature map $\boldsymbol{Z}_j$, respectively. $\boldsymbol{F}_{A,i}$ and $\boldsymbol{F}_{S,j}$ are the corresponding block circulant with circulant block (BCCB) matrices of filters $\boldsymbol{f}_{A,i}$ and $\boldsymbol{f}_{S,j}$, respectively. We update the three variables alternatively, and the details of the optimization of each sub-problem are described as follows.

**Updating $\boldsymbol{u}$** To solve the subproblem with respect to $\boldsymbol{u}$, we fix $\{\boldsymbol{f}_{S,j}\}_{j=1\dots N}$ and $\{\boldsymbol{z}_j\}_{j=1\dots N}$ and solve the following optimization problem:

$$min_{\boldsymbol{u}}\|\boldsymbol{y}-\boldsymbol{u}-\sum_j \boldsymbol{F}_{S,j}\boldsymbol{z}_j\|_2^2+\lambda\sum_i\|\boldsymbol{F}_{A,i}\boldsymbol{u}\|_1. \tag{4.4}$$

By introducing a group of auxiliary variables $\{\boldsymbol{s}_i = \boldsymbol{F}_{A,i}\boldsymbol{u}\}_{i=1,\dots,M}$, we solve (4.4) by the

ADMM algorithm:

$$\begin{cases} \boldsymbol{u}^{k+1} = (\frac{\mu_k}{2}\sum_i \boldsymbol{F}_{A,i}^T\boldsymbol{F}_{A,i} + \boldsymbol{I})^{-1}(\boldsymbol{y} - \sum_j \boldsymbol{F}_{S,j}\boldsymbol{z}_j + \frac{\mu_k}{2}\sum_i \boldsymbol{F}_{A,i}^T\boldsymbol{s}_i + \frac{1}{\mu_k}\sum_i \boldsymbol{F}_{A,i}\boldsymbol{L}_i); \\ \boldsymbol{s}_i^{k+1} = \mathcal{S}_{\frac{\lambda}{\mu_k}}(\boldsymbol{F}_{A,i}\boldsymbol{u}^{k+1} + \frac{1}{\mu_k}\boldsymbol{L}_i); \\ \boldsymbol{L}_i^{k+1} = \boldsymbol{L}_i^k + \mu_k(\boldsymbol{F}_{A,i}\boldsymbol{u}^{k+1} - \boldsymbol{s}_i); \\ if\ \mu_k < \mu_{max}, \mu_{k+1} = \mu_k * \rho; \end{cases} \tag{4.5}$$

where $\boldsymbol{L}_i$ is the Lagrange variable for $\boldsymbol{s}_i$, and $\mu_{max}$ and $\rho$ are the parameters in the algorithm. $\mathcal{S}_{\frac{\lambda}{\mu_k}}(*)$ denotes the soft-thresholding operator with parameter $\frac{\lambda}{\mu_k}$, which is the solution for the $\ell_1$-norm proximation problem. Thanks to the property of BCCB matrix, the closed-form solution in the $\boldsymbol{u}$-step in (4.5) can be efficiently obtained in the FFT domain.

**Updating $\boldsymbol{z}$** Fixing $\boldsymbol{u}$ and the synthesis dictionary $\boldsymbol{f}_S$, we solve the following sub-problem to obtain $\boldsymbol{z}$:

$$min_{\boldsymbol{z}} \|\boldsymbol{y} - \boldsymbol{u} - \sum_j \boldsymbol{F}_{S,j}\boldsymbol{z}_j\|_2^2 + \gamma \sum_j \|\boldsymbol{z}_j\|_1. \tag{4.6}$$

The optimization problem in (4.6) is a standard convolutional sparse coding problem. We utilize the algorithm in [134] to solve it, which adopts the ADMM scheme and exploits the FFT to improve computation efficiency.

**Updating $\boldsymbol{f}_S$** With the fixed $\boldsymbol{u}$ and coefficients $\boldsymbol{z}$, we need to update the synthesis dictionary. Let $vec(\boldsymbol{f}_{S,j} \otimes \boldsymbol{Z}_j) = \boldsymbol{F}_{S,j}\boldsymbol{z}_j = \mathcal{Z}\mathbf{f}_S$, where $\mathbf{f}_S$ is the vectorization of all the filters $\{\boldsymbol{f}_{S,j}\}_{j=1,\dots,N}$, $\mathcal{Z} = [\mathcal{Z}_1, \dots, \mathcal{Z}_j, \dots, \mathcal{Z}_N]$, and $\mathcal{Z}_j$ is generated by collecting the patches in $\boldsymbol{Z}_j$. The objective function can be re-written as the following equivalent form:

$$min_{\mathbf{f}_S} \|\boldsymbol{y} - \boldsymbol{u} - \mathcal{Z}\mathbf{f}_S\|_2^2, \quad s.t.\ \|\boldsymbol{f}_{S,j}\|_F^2 \leq 1, \tag{4.7}$$

We utilize a proximal gradient descent method to solve (4.7):

$$\begin{cases} \mathbf{f}_S^{t+0.5} = \mathbf{f}_S^t - \tau\mathcal{Z}^T(\boldsymbol{y} - \boldsymbol{u} - \mathcal{Z}\mathbf{f}_S^t); \\ \mathbf{f}_S^{t+1} = \mathrm{Prox}_{\|\cdot\|\leq 1}(\mathbf{f}_S^{t+0.5}). \end{cases} \tag{4.8}$$

In (4.8), $\tau$ is the step length of the gradient descent step, and $\mathrm{Prox}_{\|\cdot\|\leq 1}(*)$ is the $\ell_2$-ball proximal operator, which makes each filter satisfy the constraint $\|\boldsymbol{f}_{S,j}\|_F^2 \leq 1$:

$$\mathrm{Prox}_{\|\cdot\|\leq 1}(\boldsymbol{x}) = \begin{cases} \boldsymbol{x} & \text{if } \|\boldsymbol{x}\|_2 \leq 1; \\ \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2} & \text{if } \|\boldsymbol{x}\|_2 > 1. \end{cases} \tag{4.9}$$

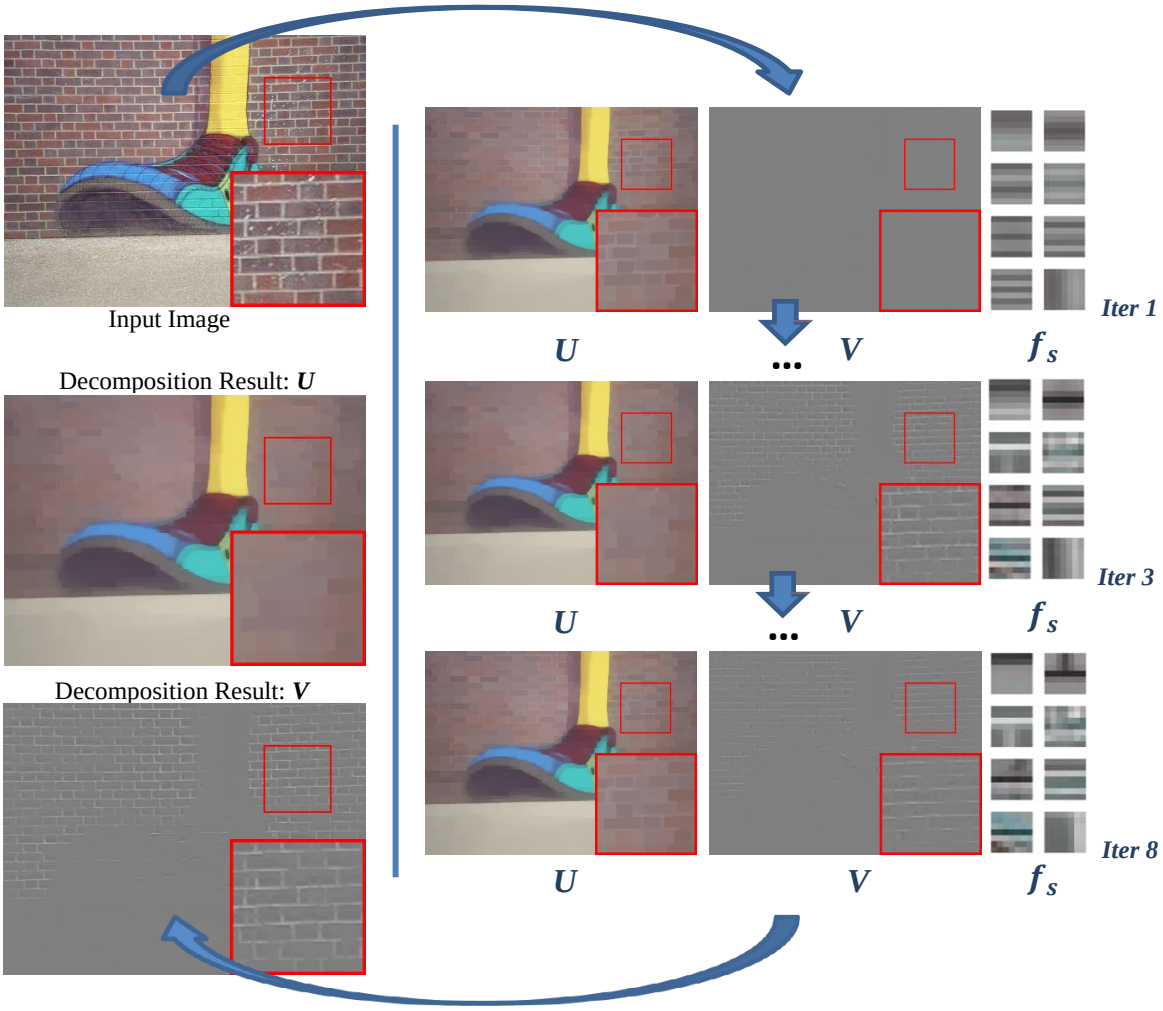



Figure 4.2. Some intermediate results of JCAS for the texture-cartoon decomposition. The synthesis dictionary is able to gradually capture the pattern of textures and remove texture from input image.

**Algorithm 4.1** JCAS algorithm for image decomposition

**Input:** Input image $\boldsymbol{Y}$, analysis filters $\{\boldsymbol{f}_{A,i}\}_{i=1,\ldots,M}$, regularization parameters $\lambda$, $\gamma$

1: **for** $k$=1:$K$ **do**
2: Solve $\boldsymbol{u}^k$ by (4.5)
3: if $k == 1$, initialize $\{\boldsymbol{f}_{S,j}\}_{j=1,\ldots,N}$ as the PCA bases of the patches in $(\boldsymbol{y} - \boldsymbol{u}^1)$
4: Solve $\boldsymbol{z}_j^k$ by (4.6)
5: Update synthesis filters $\{\boldsymbol{f}_{S,j}\}_{j=1,\ldots,N}$ by (4.8)
6: **end for**

**Output:** Decomposition results $\boldsymbol{U}$ and $\boldsymbol{V}$

The whole procedures of our method are summarized in Algorithm 4.1. Since all the three sub-problems involved in our algorithm are convex, each step will not increase the energy of the objective function (4.3). For our lower bounded objective function (4.3), the optimization process is guaranteed to converge in terms of energy. We experimentally found that the energy of loss function reduces rapidly. For all the experiments in this chapter, we set the maximum number of iterations as 15.

### 4.2.4 Discussions

The proposed JCAS model is a non-convex model. Given the input image and the analysis dictionary, we need to estimate not only the image layers but also the synthesis dictionary. For such a non-convex optimization problem, the initialization and the optimization order of the variables play an important role in our algorithm. In Algorithm 4.1, we initialize $\{\boldsymbol{z}_{S,j}^0\}_{j=1,\ldots,N}$ as all zero matrix and solve the $\boldsymbol{u}$ sub-problem first. The estimation $\boldsymbol{u}^1$ provide us a coarse evaluation of the background layer, and the residual image $\boldsymbol{y} - \boldsymbol{u}^1$ contains background details as well as repetitive textures. Then, we extract patches in the residual image $\boldsymbol{y} - \boldsymbol{u}^1$ and utilize the PCA dictionary to initialize the synthesis dictionary $\{\boldsymbol{f}_{S,j}^1\}_{j=1,\ldots,N}$. Having the synthesis dictionary, we are able to get an estimation of the texture layer $\sum_j^N \boldsymbol{F}_{S,j}^1 \boldsymbol{z}_j^1$ by solving the convolutional sparse coding problem. Due to the sparsity regularization and the constraint on the number of synthesis dictionary atoms, the synthesis approximation $\sum_j^N \boldsymbol{F}_{S,j} \boldsymbol{z}_j^1$ of the residual image tends to focus on the texture pattern while ignoring the details from background image. As a result, the details removed in the previous iteration are still in the residual image $\boldsymbol{y} - \boldsymbol{u} - \sum_j^N \boldsymbol{F}_{S,j} \boldsymbol{z}_j^1$. Such a fact helps us gradually extract the

texture layer without over-smoothing the background layer.

In Figure 4.2, we provide some intermediate results of JCAS for texture-cartoon decomposition. In the first iteration, the weak analysis-prior (with the simple gradient operators as the analysis dictionary) provides a coarse estimation of the background. To avoid over-smoothing of the background, a small regularization parameter $\lambda$ is adopted, and there are still a large amount of textures in $\boldsymbol{u}^1$. Furthermore, with the PCA initialized dictionary, the synthesis component $\boldsymbol{v}^1$ is not able to provide a good approximation to the texture. In the following iterations, the synthesis dictionary gradually captures the texture patterns, and $\sum_j^N \boldsymbol{F}_{S,j}\boldsymbol{z}_j$ extracts the texture layer from the residual $\boldsymbol{y} - \boldsymbol{u}$. Since the compact synthesis component only focuses on the textures, the image details in the first iteration of background estimation are still in the residual image. The following iterations will not lose details but gradually remove textures. As a result, the proposed method is able to remove the repetitive textures (e.g., the brickwork joint) while keeping the illuminance of background layer (e.g., bricks with different color) unchanged. In the next section, we provide more experimental results on the texture-cartoon decomposition and rain streak removal applications to demonstrate the superiority of the proposed JCAS model.

## 4.3 Experimental results

In this section, we evaluate the proposed JCAS model on the rain streak removal and texture-cartoon decomposition tasks.

### 4.3.1 Experimental results on rain streak removal

Rain streak removal aims to decompose a rainy image to a rain-free background and a rain streak layer. Due to the complex appearance of rain streaks as well as outdoor background in images, rain streak removal is a challenging image layer decomposition problem. In the last several years, many models [24, 64, 68, 80, 87] have been proposed to deal with the rain streak removal problem. To validate the effectiveness of the proposed JCAS model, we

compare JCAS with state-of-the-art rain streak removal algorithms in this section.

As introduced in the previous sections, to better capture the texture pattern, we proposed to learn a synthesis dictionary from the input image during the decomposition process. Attributed to the convolutional representation behavior in our model, we are able to use 4 convolutional dictionary atoms of size $7\times7$ to reconstruct the rain streak layer. Some priors of rain images are further utilized to improve the removal performance. Specifically, we take benefit from the directional prior as well as the non-negative prior of rain streaks. The vertical orientation prior has been utilized in [68]. With this prior, we only adopt the horizonal gradient filters $[-1, 1]$ and $[-1, 0, 1]$ as the analysis dictionary for rain removal application. While, we also know the non-negativeness of the rain layer, and we incorporate it by adding positive constraints for both the synthesis coefficients and dictionary in (4.3). This prior will not introduce further computation burden in the optimization process. By simply changing the proximal steps in the $\boldsymbol{z}$ and $\boldsymbol{f}_S$ subproblems to their non-negative version, we can get the non-negative estimation.

We compare the proposed method with different rain streak removal algorithms, which include the frequency domain decomposition method [64], the low-rank appearance model (LRA) [24], the discriminative sparse coding (DSC) method [87] and the layer-prior method (LP) [80]. The code of the LRA algorithm [24] is written by ourselves, while the codes of other competing methods are provided by authors of these methods. To validate the effectiveness of joint sparse representation, we also provide the rain removal results by a single ASR prior as the baseline.

To quantitatively compare the proposed JCAS algorithm with other methods, we perform rain streak removal experiments on 14 synthetic rainy images. The first two images are from [64] and the remaining 12 images are provided by [80]. The parameters for each algorithm are the same on all the 14 images. Specifically, we set the parameters $\lambda$ and $\gamma$ in our JCAS model as 0.005 and 0.02, and the same parameter for the analysis term $\lambda = 0.005$ is utilized for the baseline method ASR. For the other competing methods, we have carefully tuned their parameters for their best performance on the dataset.

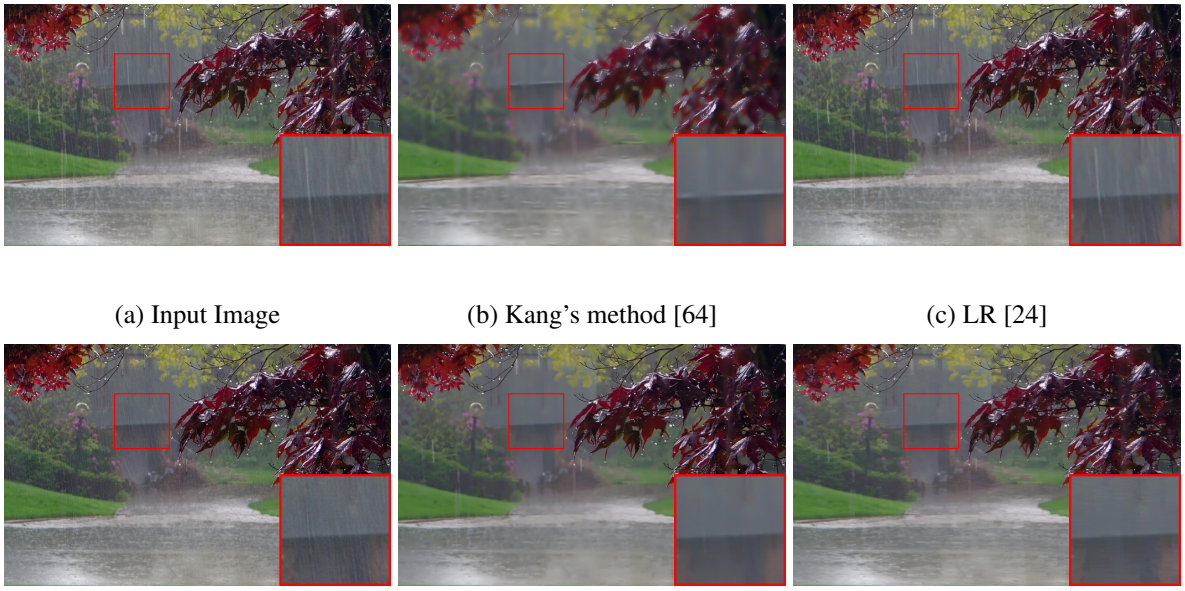
Table 4.1. Experimental results (SSIM) of all competing methods on 14 images.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | Ave. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ASR | 0.53 | 0.65 | 0.81 | 0.88 | 0.79 | 0.94 | 0.91 | 0.93 | 0.94 | 0.82 | 0.89 | 0.83 | 0.85 | 0.82 | 0.83 |
| Kang's [64] | 0.54 | 0.73 | 0.68 | 0.75 | 0.81 | 0.72 | 0.57 | 0.71 | 0.79 | 0.74 | 0.70 | 0.70 | 0.58 | 0.73 | 0.70 |
| LRA [24] | 0.57 | 0.75 | 0.79 | 0.84 | 0.76 | 0.82 | 0.86 | 0.88 | 0.91 | 0.80 | 0.86 | 0.79 | 0.82 | 0.78 | 0.80 |
| DSC [87] | 0.52 | 0.59 | 0.79 | 0.85 | 0.72 | 0.94 | 0.89 | 0.92 | 0.93 | 0.78 | 0.8867 | 0.77 | 0.84 | 0.77 | 0.76 |
| LP [80] | 0.56 | **0.77** | 0.86 | 0.91 | **0.92** | 0.92 | 0.87 | **0.94** | 0.94 | 0.88 | 0.90 | 0.86 | 0.84 | 0.91 | 0.87 |
| JCAS | **0.58** | 0.76 | **0.88** | **0.94** | 0.88 | **0.95** | **0.91** | 0.94 | **0.96** | **0.91** | **0.94** | **0.90** | **0.90** | **0.92** | **0.88** |

(a) Input Image (b) Groundtruth (c) ASR (d) Kang's method [64]

(e) LR [24] (f) DSC [87] (g) LP [80] (h) JCAS

Figure 4.3. Rain streak removal results on a synthetic image by different methods.



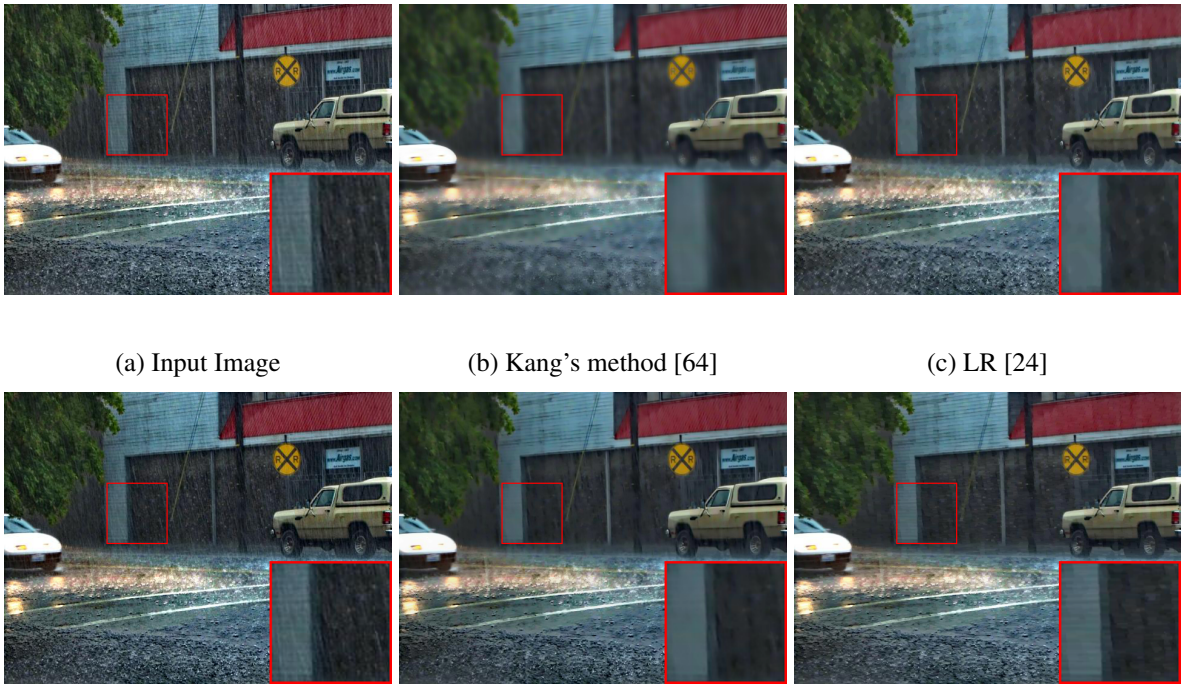
(a) Input Image (b) Kang's method [64] (c) LR [24]

(d) DSC [87] (e) LP [80] (f) JCAS

Figure 4.4. Visual comparison of different rain streak removal algorithms on a real rainy images.

We follow the experimental setting in [80] and compare different results with their structure similarity (SSIM) indexes [131] between the ground truth image. The SSIM indexes by different methods on the testing images are provided in Table 4.1, where the best results are highlighted in bold. The proposed JCAS algorithm achieves the best results on 11 out of 14 testing images, and the second best on the others. Furthermore, the much higher SSIM index of JCAS over ASR validates the effectiveness of our idea of joint ASR and SSR approximation. By extracting repetitive textures from the input image, the synthesis model helps the analysis model to better characterize the latent background. An example of rain removal on a synthetic image is provided in Figure 4.3. Kang's method produces an over-smoothed estimation which loses many details in the background. The results by other competing methods preserve most of the details in the background but also keep some streak residuals. Compared with these methods, the proposed JCAS provides a cleaner background estimation which has less rain streak residuals.

In Fig. 4.4 and Fig. 4.5, we show the results on two real rainy images. The highlight windows clearly show the advantages of the proposed algorithm. It is easy to see that JCAS removes more rain streaks and keeps details better in the background layer.



(a) Input Image (b) Kang's method [64] (c) LR [24]

(d) DSC [87] (e) LP [80] (f) JCAS

Figure 4.5. Visual comparison of different rain streak removal algorithms on a real rainy images.

### 4.3.2 Experimental results on texture-cartoon decomposition

Texture-cartoon decomposition aims to separate the input image into a cartoon part $\boldsymbol{U}$, which consists of only the image contrasted shape, and a textural part $\boldsymbol{V}$, which consists of the oscillating patterns [9]. A successful texture-cartoon decomposition can lead to improvements for subsequent image processing operations. Furthermore, as a classical image layer decomposition problem, texture-cartoon decomposition also provides researchers a good testing bed for image prior modeling algorithms.

In this part, we test the proposed model on the texture-cartoon decomposition application. Since the textures in texture-cartoon decomposition problem are more complex than rain streaks, we adopt 8 filters in the synthesis dictionary. The comparison methods include relative total variation (RTV) [137], fast cartoon+texture filtering (FCTF) [9] and recently proposed static and dynamic guidance filtering method (SDF) [56]. The codes of the comparison methods are provided by the authors of these methods. We have carefully tuned their parameters on each of the testing images for fair comparison.

Since there is no groundtruth for cartoon-texture decomposition, it is hard to compare the different methods quantitatively. Therefore, in Fig. 4.6, Fig. 4.7 and Fig. 4.8, we provide some visual examples of the cartoon estimations by different methods. In the *Floor* image, RTV [137] and FCTF [9] algorithms tend to blur the edges, and thus lose some details in the estimation. SDF [56] is able to generate cartoon estimation with sharp edges. However, it tends to ignore the details in low-contrast areas, e.g., the white areas on the bird wings have been removed in its result. Compared with other methods, the proposed JCAS model contains more fine detailed structures on the bird wings and the branch areas. A more illustrative example is provided in 4.7, where RTV [137] and SDF [56] methods fail to remove the white grids in dark area and produce blurry background estimations in the low-contrast areas. FCTF [9] and JCAS successfully remove the high-contrast textures, however, the illuminance of the extremity in the pink island area has been changed in the result of FCTF [9]. In Fig 4.8, we provide another visual example, compared with other methods, the proposed JCAS method is able to generate better cartoon estimation with sharper edges.

## 4.4 Conclusion

In this chapter, we integrated the ASR and SSR models to deal with the single image layer separation problem. An analysis-component and a synthesis-component were utilized to approximate the input image jointly in the developed model. The complementary property of the ASR and SSR models makes the two components compensate for each other well in modeling different types of image structures. As a result, the proposed JCAS model is able to finely extract textures in an input image without over-smoothing the background layer. Our experimental results on texture-cartoon decomposition and rain streak removal validate the effectiveness of the proposed model. The proposed JCAS model is expected to inspire more future investigations on the behaviors of analysis-based and synthesis-based prior modeling methods.

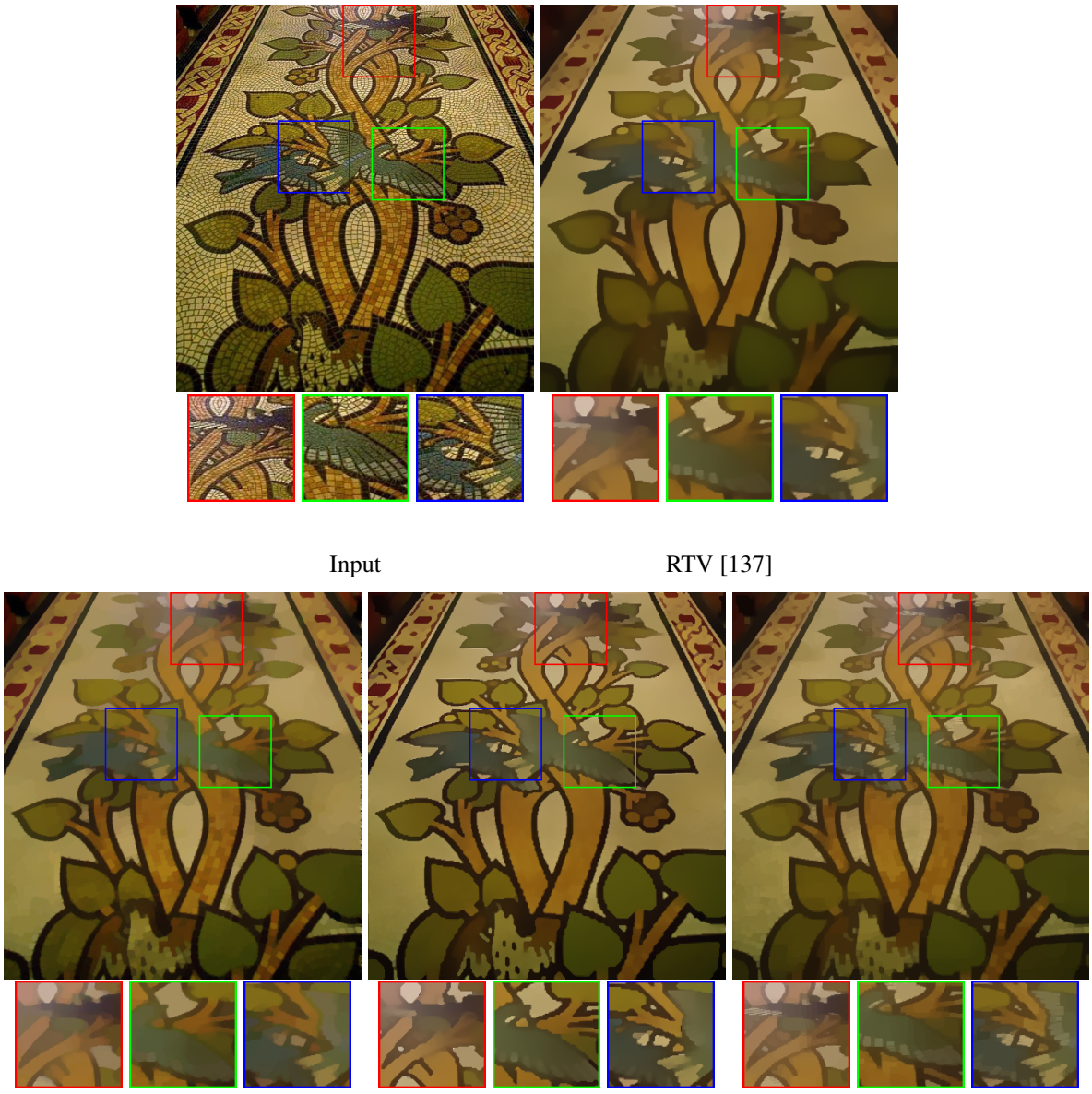



Figure 4.6. The texture removal results by different methods on the *Floor* image.

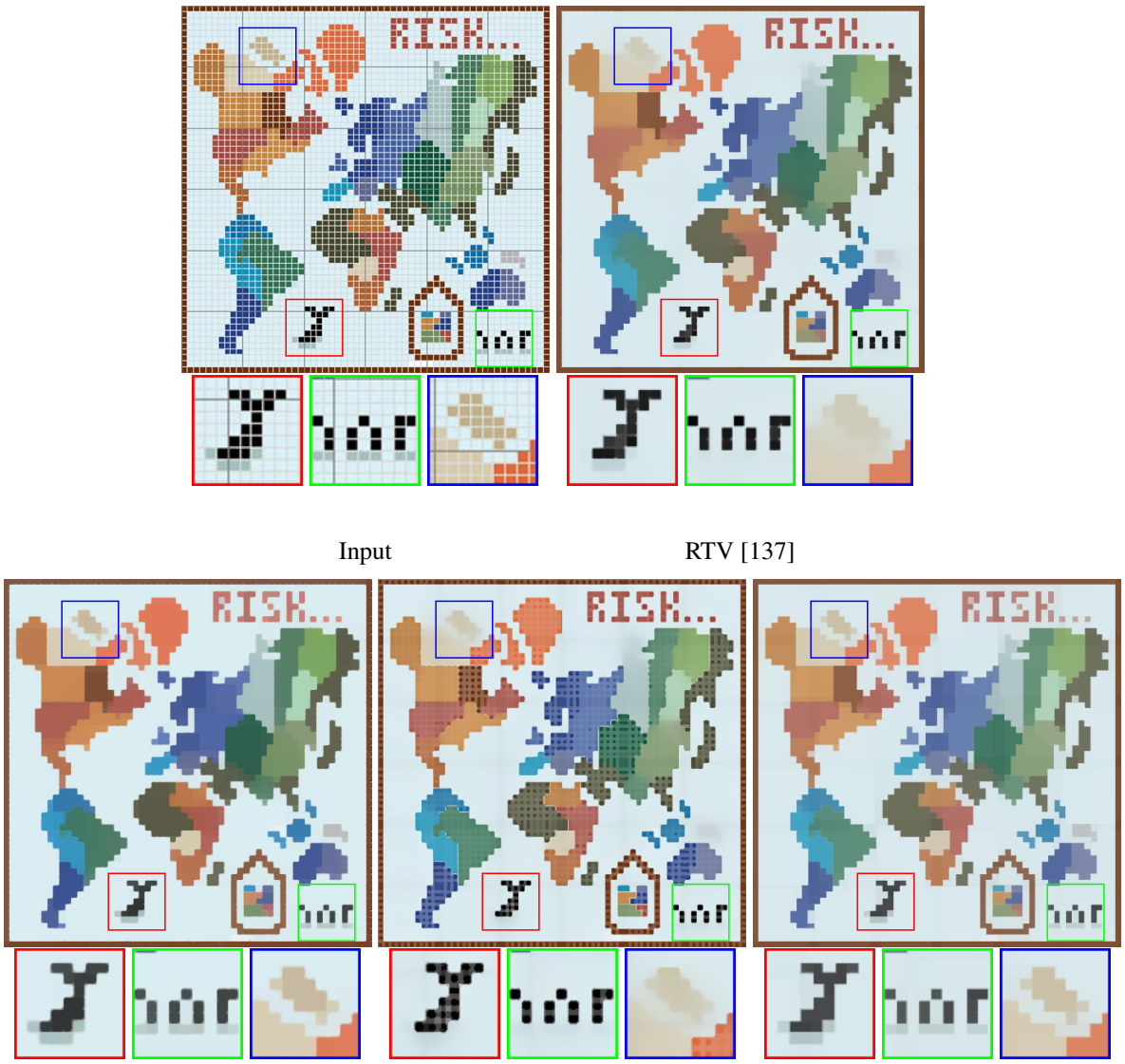



Figure 4.7. The texture removal results by different methods on the *Map* image.

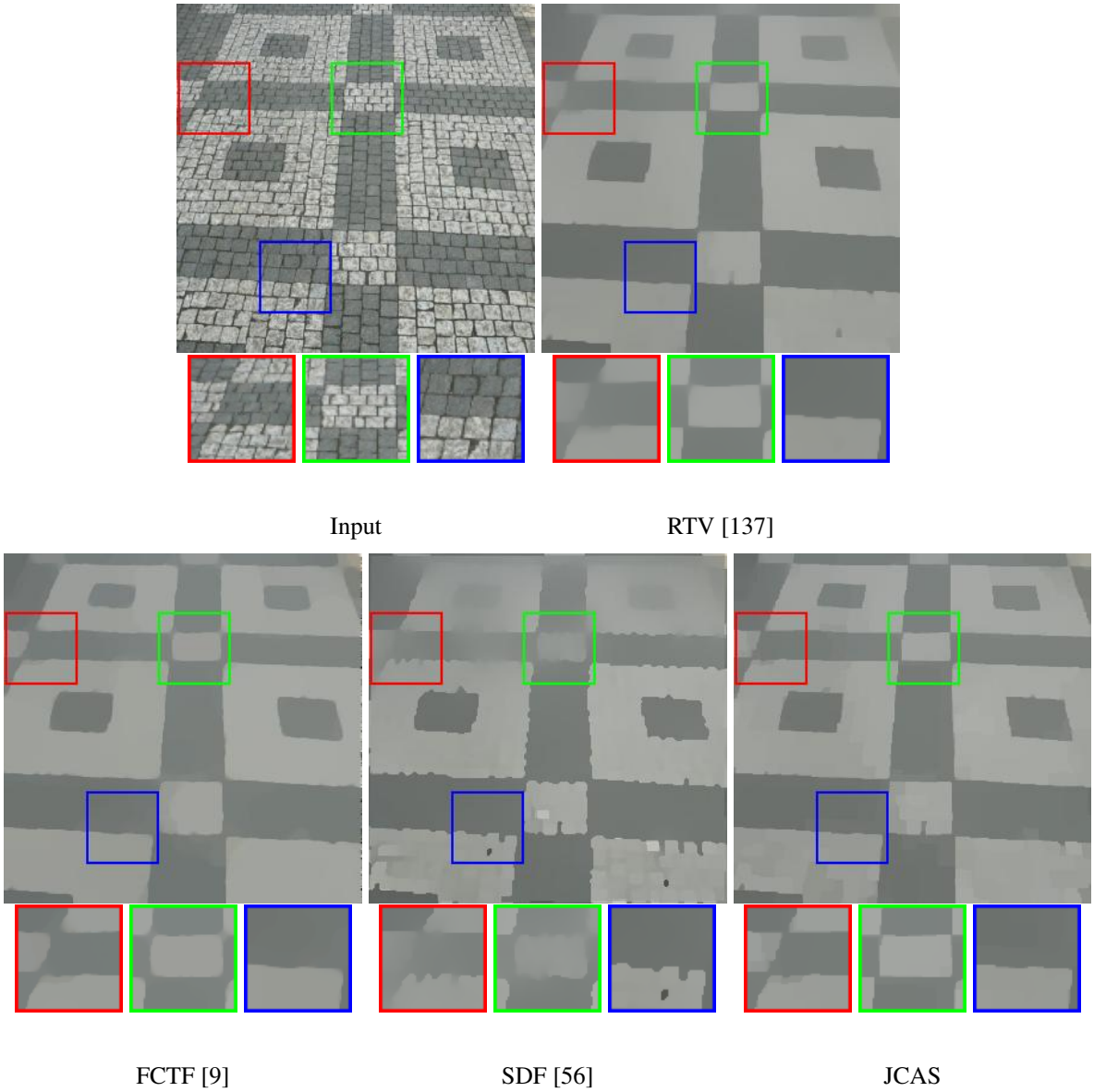



Figure 4.8. The texture removal results by different methods on the *Ground* image.

# CHAPTER 5

# WEIGHTED NUCLEAR NORM MINIMIZATION AND ITS APPLICATIONS TO LOW LEVEL VISION

In the previous chapters, we studied sparse representation models for one dimensional signal vectors. When we have a group of related vectors, simply encoding each vector individually will not exploit the correlation among those vectors. By stacking these vectors into a matrix, many methods have been adopted to model the low-rankness of the data matrix, which can be viewed as a special sparsity prior that characterize the sparsity of matrix independent subspaces. One representative work is the nuclear norm minimization (NNM) approach, which minimizes the convex envelop of the rank function. NNM regularizes each singular value equally, composing an easily calculated convex norm. However, this restricts its capability and flexibility in dealing with many practical problems, where the singular values have clear physical meanings and should be treated differently. In this chapter we propose the weighted nuclear norm minimization (WNNM) problem, which adaptively assigns weights on different singular values. As the key step of solving general WNNM models, the theoretical properties of the weighted nuclear norm proximal (WNNP) operator are investigated. Albeit nonconvex, we prove that WNNP is equivalent to a standard quadratic programming problem with linear constrains, which facilitates solving the original problem with off-the-shelf convex optimization solvers. In particular, when the weights are sorted in a non-descending order, its optimal solution can be easily obtained in closed-form. With WNNP, the solving strategies for multiple extensions of WNNM, including robust PCA and matrix completion, can be readily constructed under the alternating direction method of multipliers paradigm. Furthermore, inspired by the reweighted sparse coding scheme, we present an automatic

weight setting method, which greatly facilitates the practical implementation of WNNM. The proposed WNNM methods achieve state-of-the-art performance in typical low level vision tasks, including image denoising, background subtraction and image inpainting. Most of the contents in this chapter have been published in [54] and [53].

## 5.1 Introduction

### 5.1.1 Low rank matrix approximation

Low rank matrix approximation (LRMA), which aims to recover the underlying low rank matrix from its degraded observation, has a wide range of applications in computer vision and machine learning. For instance, human facial images can be modeled as reflections of a Lambertian object and approximated by a low dimensional linear subspace; this low rank nature leads to a proper reconstruction of a face model from occluded/corrupted face images [65, 84, 151]. In recommendation systems, the LRMA approach has achieved outstanding performance on the celebrated Netflix competition, whose low-rank insight is based on the fact that the customers′ choices are mostly affected by only a few common factors [115]. In background subtraction, the video clip captured by a surveillance camera is generally taken under a static scenario in background and with relatively small moving objects in foreground over a period, naturally resulting in its low-rank property; this has inspired various effective techniques on background modeling and foreground object detection in recent years [14, 101]. In image processing, it has also been shown that the matrix formed by non-local similar patches in a natural image is of low rank; such a prior knowledge benefits the image restoration tasks [130]. Owing to the rapid development of convex and non-convex optimization techniques in past decades, there have been a flurry of studies in LRMA, and many important models and algorithms have been reported [10, 12, 14, 15, 42, 45, 65, 82, 119].

The current development of LRMA can be categorized into two categories: the low rank matrix factorization (LRMF) approaches and the rank minimization approaches. Given a matrix $\boldsymbol{Y} \in \Re^{m\times n}$, LRMF aims to factorize it into two smaller ones, $\boldsymbol{A} \in \Re^{m\times k}$ and

$\boldsymbol{B} \in \Re^{n\times k}$, such that their product $\boldsymbol{AB}^T$ can reconstruct $\boldsymbol{Y}$ under certain fidelity loss functions. Here $k < min(m, n)$ ensures the low-rank property of the reconstructed matrix $\boldsymbol{AB}^T$. A variety of LRMF methods have been proposed, including the classical singular value decomposition (SVD) under $\ell_2$-norm loss, robust LRMF methods under $\ell_1$-norm loss, and many probabilistic methods [10, 42, 65, 71, 98, 119].

Rank minimization methods represent another main branch along this line of research. These methods reconstruct the data matrix through imposing an additional rank constraint upon the estimated matrix. Since direct rank minimization is NP hard and is difficult to solve, the problem is generally relaxed by substitutively minimizing the nuclear norm of the estimated matrix, which is a convex relaxation of minimizing the matrix rank [44]. This methodology is called as nuclear norm minimization (NNM). The nuclear norm of a matrix $\boldsymbol{X}$, denoted by $\|\boldsymbol{X}\|_*$, is defined as the sum of its singular values, i.e., $\|\boldsymbol{X}\|_* = \sum_i \sigma_i(\boldsymbol{X})$, where $\sigma_i(\boldsymbol{X})$ denotes the *i*-th singular value of $\boldsymbol{X}$. The NNM approach has been attracting significant attention due to its rapid development in both theory and implementation. On one hand, it has been proved in [14] that from the noisy input, its intrinsic low-rank reconstruction can be exactly achieved with a high probability through solving an NNM problem. On the other hand, it has been proved in [12] that the nuclear norm proximal (NNP) problem

$$\hat{\boldsymbol{X}} = \mathbf{prox}_{\lambda\|\cdot\|_*}(\boldsymbol{Y}) = \arg\min_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \lambda\|\boldsymbol{X}\|_* \tag{5.1}$$

can be easily solved in closed-form by imposing a soft-thresholding operation on the singular values of the observation matrix:

$$\hat{\boldsymbol{X}} = \boldsymbol{U}\mathcal{S}_{\frac{\lambda}{2}}(\boldsymbol{\Sigma})\boldsymbol{V}^T, \tag{5.2}$$

where $\boldsymbol{Y} = \boldsymbol{U\Sigma V}^T$ is the SVD of $\boldsymbol{Y}$ and $\mathcal{S}_{\frac{\lambda}{2}}(\boldsymbol{\Sigma})$ is the soft-thresholding function on diagonal matrix $\boldsymbol{\Sigma}$ with parameter $\frac{\lambda}{2}$. For each diagonal element $\boldsymbol{\Sigma}_{ii}$ in $\boldsymbol{\Sigma}$, there is

$$\mathcal{S}_{\frac{\lambda}{2}}(\boldsymbol{\Sigma})_{ii} = \max\left(\boldsymbol{\Sigma}_{ii} - \frac{\lambda}{2}, 0\right). \tag{5.3}$$

By utilizing NNP as the key proximal technique [100], many NNM-based models have been proposed in recent years [12, 63, 81, 82].

### 5.1.2 Motivation

Although NNM is successful, it does have limitations. In traditional NNM, all singular values are treated equally and shrunk with the same threshold $\frac{\lambda}{2}$ as defined in (5.3). This, however, ignores the prior knowledge we often have on singular values of a practical data matrix. More specifically, larger singular values of an input data matrix quantify the information of its underlying principal directions. For example, the large singular values of a matrix of image similar patches deliver the major edge and texture information. This implies that to recover an image from its corrupted observation, we should shrink less the larger singular values while shrinking more the smaller ones. Clearly, traditional NNM model, as well as its corresponding soft-thresholding solvers, are not flexible enough to handle such issues.

To improve the flexibility of NNM, we propose the weighted nuclear norm and study its minimization in this chapter. The weighted nuclear norm of a matrix $\boldsymbol{X}$ is defined as

$$\|\boldsymbol{X}\|_{\boldsymbol{w},*} = \sum_i w_i \sigma_i(\boldsymbol{X}), \tag{5.4}$$

where $\boldsymbol{w} = [w_1, \ldots, w_n]^T$ and $w_i \geq 0$ is a non-negative weight assigned to $\sigma_i(\boldsymbol{X})$. The weight vector will enhance the representation capability of the original nuclear norm. Rational weights specified based on the prior knowledge and understanding of the problem will benefit the corresponding weighted nuclear norm minimization (WNNM) model for achieving a better estimation of the latent data from the corrupted input. The difficulty of solving the WNNM model, however, lies in that it is non-convex in general cases, and the sub-gradient method [12] used for achieving the closed-form solution of an NNP problem is no longer applicable. In this chapter, we investigate in detail how to properly and efficiently solve such a non-convex WNNM problem.

As the NNP operator to the NNM problem, the following weighted nulcear norm

proximal (WNNP)[1] operator determines the general solving regime of the WNNM problem:

$$\hat{\boldsymbol{X}} = \mathbf{prox}_{\|\cdot\|_{\boldsymbol{w},*}}(\boldsymbol{Y}) = \arg\min_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \|\boldsymbol{X}\|_{\boldsymbol{w},*}. \tag{5.5}$$

We prove that the WNNP problem can be equivalently transformed to a quadratic programming (QP) problem with linear constraints. This allows us to easily reach the global optimum of the original problem by using off-the-shelf convex optimization solvers. We further show that when the weights are non-descending, the global optimum of WNNP can be easily achieved in closed-form, i.e., by a so-called weighted soft-thresholding operator. Such an efficient solver makes it possible to utilize weighted nuclear norm in more complex applications. Particularly, we propose the WNNM-based robust principle component analysis (WNNM-RPCA) model and WNNM-based matrix completion (WNNM-MC) model, and solve these models by virtue of the WNNP solver. Furthermore, inspired by the previous developments of reweighted sparse coding, we present a rational scheme to automatically set the weights for the given data.

To validate the effectiveness of the proposed WNNM models, we test them on several typical low level vision tasks. Specifically, we first test the performance of the proposed WNNP model on image denoising. By utilizing the nonlocal self-similarity prior of images [8], the WNNP model achieves superior performance to state-of-the-art denoising algorithms. We perform the WNNM-RPCA model on the application of background subtraction. Both the quantitative results and visual examples demonstrate the superiority of the proposed model beyond previous low-rank learning methods. We further apply WNNM-MC to the image inpainting task, and it also shows competitive results with state-of-the-art methods.

[1] A general proximal operator is defined on a convex problem to guarantee an accurate projection. Although the problem here is nonconvex, we can strictly prove that it is equivalent to a convex quadratic programing problem in Section 5.2. We thus also call it a proximal operator throughout the chapter for convenience.

## 5.2 Weighted Nuclear Norm Minimization for Low Rank Modeling

In this section, we first introduce the general solving scheme for the WNNP problem (5.5), and then introduce its WNNM-RPCA and WNNM-MC extensions. Note that these WNNM models are more difficult to optimize than conventional NNM ones due to the non-convexity of the involved weighted nuclear norm. Furthermore, the sub-gradient method proposed in [12] to solve NNP is not applicable to WNNP. We thus construct new solving regime for this problem. Obviously, NNM is a special case of WNNM when all the weights $w_i$ are set the same. Our solution thus covers that of the traditional NNP.

### 5.2.1 Weighted nuclear norm proximal for WNNM

To analyze the proximal operation of weighted nuclear norm, we first introduce the following Lemma, which comes from the classical von Neumanns trace inequality [97]:

**Lemma 5.2.1.** *(von Neumanns trace inequality [97]) For any $m \times n$ matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $tr\left(\boldsymbol{A}^T\boldsymbol{B}\right) \leq \sum_i \sigma_i(\boldsymbol{A})\sigma_i(\boldsymbol{B})$, where $\sigma_1(\boldsymbol{A}) \geq \sigma_2(\boldsymbol{A}) \geq \cdots \geq 0$ and $\sigma_1(\boldsymbol{B}) \geq \sigma_2(\boldsymbol{B}) \geq \cdots \geq 0$ are the descending singular values of $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. The case of equality occurs if and only if it is possible to find unitaries $\boldsymbol{U}$ and $\boldsymbol{V}$ that simultaneously singular value decompose $\boldsymbol{A}$ and $\boldsymbol{B}$ in the sense that*

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}_A\boldsymbol{V}^T, and\ \boldsymbol{B} = \boldsymbol{U}\boldsymbol{\Sigma}_B\boldsymbol{V}^T,$$

*where $\boldsymbol{\Sigma}_A$ and $\boldsymbol{\Sigma}_B$ denote the ordered eigenvalue matrices with singular value $\sigma(\boldsymbol{A})$ and $\sigma(\boldsymbol{B})$ along the diagonal with the same order, respectively.*

Based on the result of Lemma 5.2.1, we can deduce the following important theorem.

**Theorem 5.2.1.** *Given $\boldsymbol{Y} \in \Re^{m\times n}$, without loss of generality, we assume that $m \geq n$, and let $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ be the SVD of $\boldsymbol{Y}$, where $\boldsymbol{\Sigma} = \begin{pmatrix} diag(\sigma_1, \sigma_2, ..., \sigma_n) \\ \boldsymbol{0} \end{pmatrix} \in \Re^{m\times n}$.*

*The global optimum of the WNNP problem in* (5) *can be expressed as* $\hat{\boldsymbol{X}} = \boldsymbol{U}\hat{\boldsymbol{D}}\boldsymbol{V}^T$*, where* $\boldsymbol{D} = \begin{pmatrix} diag(d_1, d_2, ..., d_n) \\ \boldsymbol{0} \end{pmatrix}$ *is a diagonal non-negative matrix and* $(d_1, d_1, ..., d_n)$ *is the solution to the following convex optimization problem:*

$$\begin{aligned} &\min_{d_1,d_2,...,d_n} \sum_{i=1}^{n} (\sigma_i - d_i)^2 + w_i d_i, \\ &s.t. \quad d_1 \geq d_2 \geq ... \geq d_n \geq 0. \end{aligned} \tag{5.6}$$

The proof of Theorem 5.2.1 can be found in the Appendix B.3. Theorem 5.2.1 shows that the WNNP problem can be transformed into a new optimization problem (5.6). It is interesting that (5.6) is a quadratic optimization problem with linear constraints, and its global optimum can be easily calculated by off-the-shelf convex optimization solvers. This means that for the non-convex WNNP problem, we can always get its global solution through (5.6). In the following corollary, we further show that the global solution of (5.6) can be achieved in closed-form when the weights are sorted in a non-descending order.

**Corollary 5.2.1.** *If* $\sigma_1 \geq \ldots \geq \sigma_n \geq 0$ *and the weights satisfy* $0 \leq w_1 \leq \ldots \leq w_n$*, then the global optimum of* (5.6) *is* $\hat{d}_i = max\left(\sigma - \frac{w_i}{2}, 0\right)$.

The proof of Corollary 5.2.1 is given in Appendix B.4. The conclusion in Corollary 5.2.1 is very useful. The singular values of a matrix are sorted in a non-ascending order, and the larger singular values usually correspond to the subspaces of more important components of the data matrix. We thus always expect to shrink less the larger singular values to keep the major and faithful information of the latent data. In this sense, Corollary 5.2.1 guarantees that we have a closed-form optimal solution for the WNNP problem by the weighted singular value soft-thresholding operation:

$$\mathbf{prox}_{\lambda\|\cdot\|_{\boldsymbol{w},*}}(\boldsymbol{Y}) = \boldsymbol{U}\mathcal{S}_{\frac{\boldsymbol{w}}{2}}(\boldsymbol{\Sigma})\boldsymbol{V}^{\mathbf{T}},$$

where $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the SVD of $\boldsymbol{Y}$, and $\mathcal{S}_{\frac{\boldsymbol{w}}{2}}(\boldsymbol{\Sigma})$ is the generalized soft-thresholding operator with weight vector $\boldsymbol{w}$

$$\mathcal{S}_{\frac{\boldsymbol{w}}{2}}(\boldsymbol{\Sigma})_{ii} = \max\left(\boldsymbol{\Sigma}_{ii} - \frac{w_i}{2}, 0\right).$$

Note that when all the weights $w_i$ are set the same, the above WNNP solver exactly degenerates to the NNP solver for the traditional NNM problem.

A recent work by Lu et al. [85] has proved a similar conclusion to our Corollary 5.2.1. As Lu et al. analyzed the generalized singular value regularization model with different penalty functions for the singular values, the condition in their paper is the monotonicity property of the proximal operator which is determined by the penalty function. While our work attains the WNNP solver in general weighting cases rather than only in the case of nonascendingly ordered weights. Interested readers may refer to the proof in Appendix B and [85] for details.

**Algorithm 5.1** WNNM-RPCA

**Input:** Observation data $\boldsymbol{Y}$, weight vector $\boldsymbol{w}$

Initialize $\mu_0 > 0$, $\rho > 1$, $\theta > 0$, $k = 0$, $\boldsymbol{X}_0 = \boldsymbol{Y}$, $\boldsymbol{L}_0 = \boldsymbol{0}$;
**do**
  $\boldsymbol{E}_{k+1} = \arg\min_{\boldsymbol{E}} \|\boldsymbol{E}\|_1 + \frac{\mu_k}{2}\|\boldsymbol{Y} + \mu_k^{-1}\boldsymbol{L}_k - \boldsymbol{X}_k - \boldsymbol{E}\|_F^2$;
  $\boldsymbol{X}_{k+1} = \arg\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{w,*} + \frac{\mu_k}{2}\|\boldsymbol{Y} + \mu_k^{-1}\boldsymbol{L}_k - \boldsymbol{E}_{k+1} - \boldsymbol{X}\|_F^2$;
  $\boldsymbol{L}_{k+1} = \boldsymbol{L}_k + \mu_k(\boldsymbol{Y} - \boldsymbol{X}_{k+1} - \boldsymbol{E}_{k+1})$;
  Update $\mu_{k+1} = \rho * \mu_k$;
  $k = k + 1$;
**while** $\|\boldsymbol{Y} - \boldsymbol{X}_{k+1} - \boldsymbol{E}_{k+1}\|_F / \|\boldsymbol{Y}\|_F > \theta$

**Output:** Matrix $\boldsymbol{X} = \boldsymbol{X}_{k+1}$ and $\boldsymbol{E} = \boldsymbol{E}_{k+1}$;

### 5.2.2 WNNM for robust PCA

In last section, we analyzed the optimal solution to the WNNP operator for the WNNM problem. Based on our definition of WNNP in (5.5), $\mathbf{prox}_{\|\cdot\|_{w,*}}(\boldsymbol{Y})$ is the low rank approximation to the observation matrix $\boldsymbol{Y}$ under the $F$-norm data fidelity term. However, in real applications, the observation data may be corrupted by outliers or sparse noise with large magnitude. In such cases, the large magnitude noise, even with small amount, tends to greatly affect the $F$-norm data fidelity and lead to a biased low rank estimation. The recently proposed NNM-based RPCA (NNM-RPCA) method [14] alleviates this problem by optimizing the following problem:

$$\min_{\boldsymbol{E},\boldsymbol{X}} \|\boldsymbol{E}\|_1 + \|\boldsymbol{X}\|_*, \quad s.t. \ \ \boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}. \tag{5.7}$$

Using the $\ell_1$-norm to model the error $\boldsymbol{E}$, model (5.7) guarantees a more robust matrix approximation in the presence of outliers/sparse noise. In particular, it is proved in [14] that if the low rank matrix $\boldsymbol{X}$ and the sparse component $\boldsymbol{E}$ satisfy certain conditions, the NNM-RPCA model (5.7) can exactly recover $\boldsymbol{X}$ with a probability close to 1.

In this section, we propose to use the weighted nuclear norm to reformulate (5.7), leading to the following WNNM-based RPCA (WNNM-RPCA) model:

$$\min_{\boldsymbol{E},\boldsymbol{X}} \|\boldsymbol{E}\|_1 + \|\boldsymbol{X}\|_{w,*}, \ \ s.t. \ \ \boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}. \tag{5.8}$$

As in NNM-RPCA, we also employ the alternating direction method of multipliers (ADMM) to solve the WNNM-RPCA problem. Its augmented Lagrange function is

$$\Gamma(\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{L}, \mu) = \|\boldsymbol{E}\|_1 + \|\boldsymbol{X}\|_{w,*} + \langle \boldsymbol{L}, \boldsymbol{Y} - \boldsymbol{X} - \boldsymbol{E} \rangle + \frac{\mu}{2}\|\boldsymbol{Y} - \boldsymbol{X} - \boldsymbol{E}\|_F^2, \tag{5.9}$$

where $\boldsymbol{L}$ is the Lagrange multiplier and $\mu$ is a positive scaler. The optimization procedure is described in Algorithm 5.1. Note that the convergence of this ADMM algorithm is more difficult to analyze than the convex NNM-RPCA model due to the non-convexity of the WNNM-RPCA model. We give the following weak convergence result to facilitate the construction of a rational termination condition for Algorithm 5.1.

**Theorem 5.2.2.** *If the weights are sorted in a non-descending order, the sequences* $\{\boldsymbol{E}_k\}$ *and* $\{\boldsymbol{X}_k\}$ *generated by Algorithm 5.1 satisfy:*

$$(1) \lim_{k\to\infty} \|\boldsymbol{X}_{k+1} - \boldsymbol{X}_k\|_F = 0,$$

$$(2) \lim_{k\to\infty} \|\boldsymbol{E}_{k+1} - \boldsymbol{E}_k\|_F = 0.$$

$$(3) \lim_{k\to\infty} \|\boldsymbol{Y} - \boldsymbol{E}_{k+1} - \boldsymbol{X}_{k+1}\|_F = 0.$$

The proof of Theorem 5.2.2 can be found in Appendix B.4.1. Please note that the proof of Theorem 5.2.2 relies on the unboundedness of the parameter $\mu_k$. In most of previous ADMM based methods, an upper bound of $\mu_k$ is needed to ensure the optimal solution for convex objective functions [82]. However, since our WNNM-RPCA model is non-convex

for general weight conditions, we use an unbounded $\mu_k$ to guarantee the convergence of Algorithm 5.1. If $\mu_k$ increases too fast, the iteration may stop quickly and we might not get a good solution. Thus in both Algorithm 5.1 and the following Algorithm 5.2, a small $\rho$ is adopted to prevent $\mu_k$ from increasing too fast. Please refer to the experiments in sections 5.4 and 5.5 for more details.

**Algorithm 5.2** WNNM-MC

**Input:** Observation data $\boldsymbol{Y}$, indicator matrix $\Omega$, weight vector $\boldsymbol{w}$,

1: Initialize $\mu_0 > 0$, $\rho > 1$, $\theta > 0$, $k = 0$, $k = 0$, $\boldsymbol{X}_0 = \boldsymbol{Y}$, $\boldsymbol{L}_0 = \boldsymbol{0}$;
2: **do**
3: $\quad \boldsymbol{E}_{k+1} = \arg\min_{\boldsymbol{E}} \|\boldsymbol{Y} + \mu_k^{-1}\boldsymbol{L}_k - \boldsymbol{X}_k - \boldsymbol{E}\|_F^2$ $s.t. \|\mathcal{P}_\Omega(\boldsymbol{E})\|_F^2 = 0$;
4: $\quad \boldsymbol{X}_{k+1} = \arg\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{w,*} + \frac{\mu_k}{2}\|\boldsymbol{Y} + \mu_k^{-1}\boldsymbol{L}_k - \boldsymbol{E}_k - \boldsymbol{X}\|_F^2$;
5: $\quad \boldsymbol{L}_{k+1} = \boldsymbol{L}_k + \mu_k(\boldsymbol{Y} - \boldsymbol{X}_{k+1} - \boldsymbol{E}_{k+1})$;
6: $\quad$ Update $\mu_{k+1} = \rho * \mu_k$;
7: $\quad k = k + 1$;
8: **while** $\|\boldsymbol{Y} - \boldsymbol{X}_{k+1} - \boldsymbol{E}_{k+1}\|_F / \|\boldsymbol{Y}\|_F > \theta$

**Output:** $\boldsymbol{X} = \boldsymbol{X}_{k+1}$;

### 5.2.3 WNNM for matrix completion

In section 5.2.2, we introduced the WNNM-RPCA model and provided the ADMM algorithm to solve it. In this section, we further use WNNM to deal with the matrix completion problem, and propose the following WNNM-based matrix completion (WNNM-MC) model:

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{\boldsymbol{w},*} \quad s.t. \quad \mathcal{P}_\Omega(\boldsymbol{X}) = \mathcal{P}_\Omega(\boldsymbol{Y}), \tag{5.10}$$

where $\Omega$ is a binary support indicator matrix of the same size as $\boldsymbol{Y}$, and zeros in $\Omega$ indicate the missing entries in the observation matrix. $\mathcal{P}_\Omega(\boldsymbol{Y}) = \Omega \odot \boldsymbol{Y}$ is the element-wise matrix multiplication (Hardamard product) between the support matrix $\Omega$ and the variable $\boldsymbol{Y}$. The constraint implies that the estimated matrix $\boldsymbol{X}$ agrees with $\boldsymbol{Y}$ in the observed entries.

By introducing a variable $\boldsymbol{E}$, we reformulate (5.10) as

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{w,*}, \quad s.t. \ \boldsymbol{X} + \boldsymbol{E} = \boldsymbol{Y}, \ \ \mathcal{P}_\Omega(\boldsymbol{E}) = 0. \tag{5.11}$$

The ADMM algorithm for solving WNNM-MC can then be constructed in Algorithm 5.2. For non-descending weights, both the subproblems in steps 3 and 4 of Algorithm 5.2 have

closed-form optimal solutions. However, as the weighted nuclear norm is not convex, it is difficult to accurately analyze the convergence of the algorithm. Like in Theorem 5.2.2, in the following Theorem 5.2.3, we also present a weak convergence result.

**Theorem 5.2.3.** *If the weights are sorted in a non-descending order, the sequence* $\{\boldsymbol{X}_k\}$ *generated by Algorithm 5.2 satisfies*

$$(1) \lim_{k\to\infty} \|\boldsymbol{X}_{k+1} - \boldsymbol{X}_k\|_F = 0,$$

$$(2) \lim_{k\to\infty} \|\boldsymbol{Y} - \boldsymbol{E}_{k+1} - \boldsymbol{X}_{k+1}\|_F = 0.$$

The proof of Theorem 5.2.2 is similar to Theorem 5.2.3, and thus we omit it here.

### 5.2.4 The setting of weighting vector

In previous sections, we proposed to utilize the WNNM model to solve different problems. By introducing the weight vector, the WNNM model improves the flexibility of the original NNM model. However, the weight vector itself also brings more parameters in the model. Appropriate setting of the weights plays a crucial role in the success of the proposed WNNM model.

In [16], an effective reweighting mechanism is proposed to enhance the sparsity of sparse coding solutions by adaptively tuning weights through the following formula:

$$w_i^{\ell+1} = \frac{C}{|x_i^{\ell}| + \varepsilon}, \tag{5.12}$$

where $x_i^{\ell}$ is the $i$-th sparse coding coefficient in the $\ell$-th iteration and $w_i^{\ell+1}$ is its corresponding regularization parameter in the $(\ell+1)$-th iteration, $\varepsilon$ is a small positive number to avoid dividing by zero and $C$ is a compromising constant. Such a reweighting procedure has proved to be capable of getting a good resemble of the $\ell_0$ norm and the model achieves superior performance in compressive sensing.

Inspired by the success of reweighted sparse coding, we can adopt a similar reweighting strategy in WNNM by replacing $x_i^{\ell}$ in (5.12) with the singular value $\sigma_i(\boldsymbol{X}_{\ell})$ in the $\ell^{th}$

iteration. Because (5.12) is monotonically decreasing with the singular values, the non-descending order of weights with respect to singular values will be kept throughout the reweighting process. Very interestingly, the following remark indicates that we can directly get the closed-form solution of WNNP operator with such a reweighting strategy.

**Remark 5.2.1.** *Let* $\mathbf{Y} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ *be the SVD of* $\mathbf{Y}$*, where* $\boldsymbol{\Sigma} = \begin{pmatrix} diag(\sigma_1(\mathbf{Y}), \sigma_2(\mathbf{Y}), ..., \sigma_n(\mathbf{Y})) \\ \boldsymbol{0}, \end{pmatrix}$, *and* $\sigma_i(\mathbf{Y})$ *denotes the* $i$*-th singular value of* $\mathbf{Y}$*. If the regularization parameter* $C$ *is positive and the positive value* $\varepsilon$ *is small enough to make the inequality* $\varepsilon < min\left(\sqrt{C}, \frac{C}{\sigma_1(\mathbf{Y})}\right)$ *hold, by using the reweighting formula* $w_i^\ell = \frac{C}{\sigma_i(\mathrm{X}_\ell)+\varepsilon}$ *with initial estimation* $\mathbf{X}_0 = \mathbf{Y}$*, the reweighted WNNP problem* $\{\min_{\mathbf{X}} \|\mathbf{Y}-\mathbf{X}\|_F^2 + \|\mathbf{X}\|_{\mathbf{w},*}\}$ *has the closed-form solution:* $\mathbf{X}^* = \mathbf{U}\widetilde{\boldsymbol{\Sigma}}\mathbf{V}^T$*, where*

$$\widetilde{\boldsymbol{\Sigma}} = \begin{pmatrix} diag(\sigma_1(\mathbf{X}^*), \sigma_2(\mathbf{X}^*), ..., \sigma_n(\mathbf{X}^*)) \\ \boldsymbol{0}, \end{pmatrix},$$

*and*

$$\sigma_i(\mathbf{X}^*) = \begin{cases} 0 & \text{if } c_2 < 0 \\ \frac{c_1+\sqrt{c_2}}{2} & \text{if } c_2 \geq 0 \end{cases} \tag{5.13}$$

*where*

$$c_1 = \sigma_i(\mathbf{Y}) - \varepsilon, \;\; c_2 = (\sigma_i(\mathbf{Y}) + \varepsilon)^2 - 4C.$$

The proof of Remark 5.2.1 can be found in Appendix B.5. Remark 5.2.1 shows that although a reweighting strategy $w_i^\ell = \frac{C}{\sigma_i(X_\ell)+\varepsilon}$ is used, we do not need to iteratively perform the thresholding and weight calculation operations. Based on the relationship between the singular value of observation matrix *X* and the regularization parameter $C$, the convergence of the sigular value of estimated matrix after the reweighting process can be directly obtained. In each iteration of both the WNNM-RPCA and WNNM-MC algorithms, such a reweighting strategy is performed on the WNNP subproblem (step 4 in Algorithms 5.1 and 5.2) to adjust

weights based on current $\boldsymbol{X}_k$. Thanks to Remark 1, utilizing reweighting strategy in step 4 of Algorithm 5.1 and Algorithm 5.2 will increase little the computation burden. We are able to use an operation to directly shrink the original singular value $\sigma_i(Y)$ to 0 or $\frac{c_1+\sqrt{c_2}}{2}$, just like the soft-thresholding operation in the NNM method.

In implementation, we initialize $\boldsymbol{X}_0$ as the observation $\boldsymbol{Y}$. The above weight setting strategy greatly facilitates the WNNM calculations. Note that there remains one parameter $C$ in the WNNM implementation. In all of our experiments, we set it by experience for certain tasks. Please see the following sections for details.

### 5.3 Image Denoising by WNNM

In this section, we validate the proposed WNNM model in application of image denoising. Image denoising is one of the fundamental problems in low level vision, and is an ideal test bed to investigate and evaluate the statistical image modeling techniques and optimization methods. Image denoising aims to reconstruct the original image $\boldsymbol{x}$ from its noisy observation $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{n}$ is generally assumed to be additive white Gaussian noise (AWGN) with zero mean and variance $\sigma_n^2$.

The seminal work of nonlocal means [8] triggers the study of nonlocal self-similarity (NSS) based methods for image denoising. NSS refers to the fact that there are many repeated local patterns across a natural image, and those nonlocal similar patches to a given patch offer helpful remedy for its better reconstruction. The NSS-based image denoising algorithms such as BM3D [30], LSSC [89], NCSR [38] and SAIST [37] have achieved state-of-the-art denoising results. In this section, we utilize the NSS prior to develop the following WNNM-based denoising algorithm.

#### 5.3.1 Denoising algorithm

For a local patch $\boldsymbol{y}_j$ in image $\boldsymbol{y}$, we can search for its nonlocal similar patches across a relatively large area around it by methods such as block matching [30]. By stacking those

**Algorithm 5.3** Image Denoising by WNNM

**Input:** Noisy image $\boldsymbol{y}$

Initialize $\hat{\boldsymbol{x}}^{(0)} = \boldsymbol{y}, \boldsymbol{y}^{(0)} = \boldsymbol{y}$
**for** $k$=1:$K$ **do**
    Iterative regularization $\boldsymbol{y}^{(k)} = \hat{\boldsymbol{x}}^{(k-1)} + \delta(\boldsymbol{y} - \hat{\boldsymbol{y}}^{(k-1)})$
    **for** each patch $\boldsymbol{y}_j$ in $\boldsymbol{y}^{(k)}$ **do**
        Find similar patch group $\boldsymbol{Y}_j$
        Apply the WNNP operator to $\boldsymbol{Y}_j$ to estimate $\boldsymbol{X}_j$
    **end for**
    Aggregate $\boldsymbol{X}_j$ to form the clean image $\hat{\boldsymbol{x}}^{(k)}$
**end for**

**Output:** Denoised image $\hat{\boldsymbol{x}}^{(K)}$

nonlocal similar patches into a matrix, denoted by $\boldsymbol{Y}_j$, we have $\boldsymbol{Y}_j = \boldsymbol{X}_j + \boldsymbol{N}_j$, where $\boldsymbol{X}_j$ and $\boldsymbol{N}_j$ are the original clean patch matrix and the corresponding corruption matrix, respectively. Intuitively, $\boldsymbol{X}_j$ should be a low rank matrix, and the LRMA methods can be used to estimate $\boldsymbol{X}_j$ from $\boldsymbol{Y}_j$. By aggregating all the estimated patches $\boldsymbol{Y}_j$, the whole image can be reconstructed. Indeed, the NNM method has been adopted in [62] for video denoising, and we apply the proposed WNNP operator to each $\boldsymbol{Y}_j$ to estimate $\boldsymbol{X}_j$ for image denoising. By using the noise variance $\sigma_n^2$ to normalize the $F$-norm data fidelity term $\|\boldsymbol{Y}_j - \boldsymbol{X}_j\|_F^2$ , we have the following energy function:

$$\hat{\boldsymbol{X}}_j = \arg\min_{\boldsymbol{X}_j} \tfrac{1}{\sigma_n^2}\|\boldsymbol{Y}_j - \boldsymbol{X}_j\|_F^2 + \|\boldsymbol{X}_j\|_{\boldsymbol{w},*}. \tag{5.14}$$

Throughout our experiments, we set the parameter $C$ as $\sqrt{2n}$ by experience, where $n$ is the number of similar patches.

By applying the above procedures to each patch and aggregating all patches together, the image $\boldsymbol{x}$ can be reconstructed. In practice, we can run several iterations of this reconstruction process across all image patches to enhance the denoising outputs. The whole denoising algorithm is summarized in Algorithm 5.3.

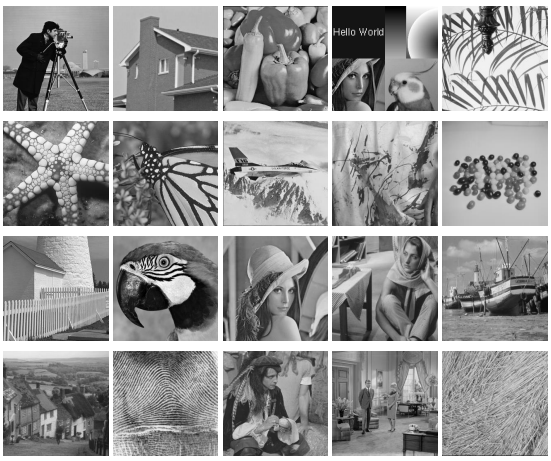



Figure 5.1. The 20 test images used in image denoising experiments.

### 5.3.2 Experimental setting

We compare the proposed WNNM-based image denoising algorithm with several state-of-the-art denoising methods, including BM3D[2] [30], EPLL[3] [158], LSSC[4] [89], NCSR[5] [38] and SAIST[6] [37]. The baseline NNM algorithm is also compared. All the competing methods exploit the image nonlocal redundancies.

There are several other parameters ($\delta$, the iteration number $K$ and the patch size) in the proposed algorithm. For all noise levels, the iterative regularization parameter $\delta$ is fixed to 0.1. The iteration number $K$ and the patch size are set based on noise level. For higher noise level, we choose bigger patches and run more times the iteration. By experience, we set patch sizes to $6 \times 6$, $7 \times 7$, $8 \times 8$ and $9 \times 9$ for $\sigma_n \leq 20$, $20 < \sigma_n \leq 40$, $40 < \sigma_n \leq 60$ and $60 < \sigma_n$, respectively. The iteration number $K$ is set to 8, 12, 14, and 14, and the number of selected non-local similar patches is set to 70, 90, 120 and 140, respectively, on these noise levels.

[2]http://www.cs.tut.fi/ foi/GCF-BM3D/BM3D.zip

[3]http://people.csail.mit.edu/danielzoran/noiseestimation.zip

[4]http://lear.inrialpes.fr/people/mairal/software.php

[5]http://www4.comp.polyu.edu.hk/∼cslzhang/code/NCSR.rar

[6]http://www.csee.wvu.edu/ xinl/demo/saist.html

For NNM, we use the same parameters as WNNM except for the uniform weight $\sqrt{n}\sigma_n$. The source codes of the comparison methods are obtained directly from the original authors, and we use the default parameters. Our code can be downloaded from http://www4.comp.polyu.edu.hk/ ∼cslzhang/code/WNNM_code.zip.

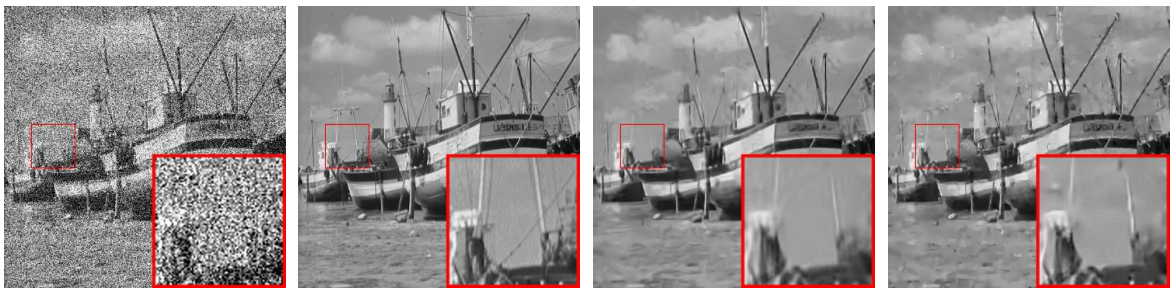
Table 5.1. The average PSNR (dB) values by competing methods on the 20 test images. The best results are highlighted in bold.

| | $\sigma_n$=10 | $\sigma_n$=20 | $\sigma_n$=30 | $\sigma_n$=40 | $\sigma_n$=50 | $\sigma_n$=75 | $\sigma_n$=100 |
|---|---|---|---|---|---|---|---|
| **NNM** | 33.462 | 30.040 | 27.753 | 26.422 | 25.048 | 22.204 | 21.570 |
| **BM3D** | 34.326 | 30.840 | 28.905 | 27.360 | 26.406 | 24.560 | 23.247 |
| **EPLL** | 34.008 | 30.470 | 28.463 | 27.060 | 25.965 | 24.020 | 22.706 |
| **LSSC** | 34.507 | 30.950 | 28.877 | 27.500 | 26.436 | 24.450 | 23.061 |
| **NCSR** | 34.456 | 30.890 | 28.849 | 27.390 | 26.326 | 24.340 | 22.996 |
| **SAIST** | 34.555 | 30.970 | 28.980 | 27.590 | 26.521 | 24.620 | 23.296 |
| **WNNM** | **34.772** | **31.200** | **29.214** | **27.780** | **26.752** | **24.860** | **23.555** |

### 5.3.3 Experimental results on 20 test images

We evaluate the competing methods on 20 widely used test images, whose thumbnails are shown in Fig. 5.1. The first 12 images are of size $256 \times 256$, and the other 8 images are of size $512 \times 512$. AWGN with zero mean and variance $\sigma_n^2$ are added to those test images to generate the noisy observations. We evaluate the proposed algorithm on a wide range of noise levels. The average denoising results on different noise levels by all competing methods can be found in Table 5.1.

From Table 5.1, we can see that the proposed WNNM method achieves the highest PSNR in almost all cases. It achieves 1.3dB-2dB improvement over the NNM method on average and outperforms the benchmark BM3D method by 0.3dB-0.45dB consistently on all the four noise levels. Such an improvement is notable since few methods can surpass BM3D



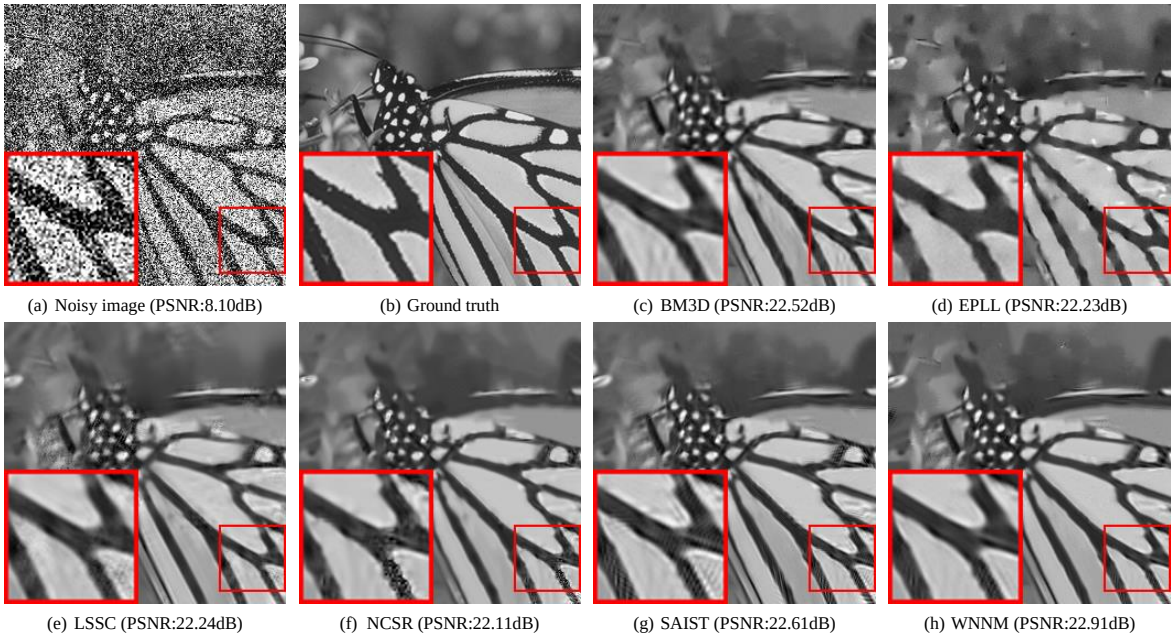
(a) Noisy image (PSNR:14.16dB) (b) Ground truth (c) BM3D (PSNR:26.78dB) (d) EPLL (PSNR:26.65dB)

(e) LSSC (PSNR:26.77dB) (f) NCSR (PSNR:26.66dB) (g) SAIST (PSNR:26.63dB) (h) WNNM (PSNR:26.98dB)

Figure 5.2. Denoising results on image *Boats* by competing methods (noise level $\sigma_n = 50$). The demarcated area is enlarged in the right bottom corner for better visualization. The figure is better seen by zooming on a computer screen.

more than 0.3dB on average [75].

In Fig. 5.2 and Fig. 5.3, we compare the visual quality of the denoised images by the competing algorithms. Fig. 5.2 demonstrates that the proposed WNNM reconstructs more image details from the noisy observation. Compared with WNNM, the LSSC, NCSR and SAIST methods over-smooth more textures in the sands area of image *Boats*, and the BM3D and EPLL methods generate more artifacts. More interestingly, as can be seen in the demarcated window, the proposed WNNM is capable of well reconstructing the tiny masts of the boat, while the masts are almost unrecognizable in the reconstructed images by other methods. Fig. 5.3 shows an example with strong noise. It is easy to see that WNNM generates less artifacts and preserves better the image edge structures compared with other competing methods. In summary, WNNM shows strong denoising capability, producing more pleasant denoising outputs in visualization and higher PSNR indices in quantity.



(a) Noisy image (PSNR:8.10dB) (b) Ground truth (c) BM3D (PSNR:22.52dB) (d) EPLL (PSNR:22.23dB)

(e) LSSC (PSNR:22.24dB) (f) NCSR (PSNR:22.11dB) (g) SAIST (PSNR:22.61dB) (h) WNNM (PSNR:22.91dB)

Figure 5.3. Denoising results on image *Monarch* by competing methods (noise level $\sigma_n = 100$). The demarcated area is enlarged in the left bottom corner for better visualization. The figure is better seen by zooming on a computer screen.

### 5.4 WNNM-RPCA for background subtraction

SVD/PCA aims to find the principal (affine) directions along which the data variance can be maximized. It has been widely used in the area of data modeling, compression, and visualization. In the conventional PCA model, the error is measured by the $\ell_2$-norm fidelity, which is optimal to suppress additive Gaussian noise. However, there are occasions that outliers or sparse noise are corrupted in data, which may disable SVD/PCA in estimating the ground truth subspace. To address this problem, multiple RPCA models have been proposed to robustify PCA, and have been employed in different applications such as structure from motion, ranking and collaborative filtering, face reconstruction and background subtraction [154].

Recently, the NNM-RPCA model has been proposed [14], which can be efficiently solved by ADMM, and can guarantee the exact reconstruction of the original data under certain conditions. Here, we propose WNNM-RPCA to further enhance the flexibility of NNM-RPCA. In the following, we first design synthetic simulations to comprehensively compare

the performance between WNNM-RPCA and NNM-RPCA, and then show the superiority of the proposed method in background subtraction by comparing with typical low-rank learning methods designed for this task.

### 5.4.1 Experimental results on synthetic data

Table 5.2. Relative error of low rank matrix recovery results by NNM-RPCA and WNNM-RPCA, with $p_e$ fixed as 0.05, and $p_r$ varying from 0.05 to 0.45 with step length 0.05.

| $Rank(\boldsymbol{X})$ | 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 | 180 |
|---|---|---|---|---|---|---|---|---|---|
| NNM-RPCA | 2.41e-8 | 3.91e-8 | 5.32e-8 | 7.91e-8 | 2.90e-4 | 1.72e-2 | 6.49e-2 | 0.13 | 0.21 |
| WNNM-RPCA | 1.79e-8 | 3.49e-8 | 5.83e-8 | 6.53e-8 | 9.28e-8 | 1.30e-7 | 1.68e-7 | 2.02e-7 | 2.43e-7 |

Table 5.3. Relative error of low rank matrix recovery results by NNM-RPCA and WNNM-RPCA, with $p_e$ fixed as 0.1, and $p_r$ varying from 0.05 to 0.45 with step length 0.05.

| $Rank(\boldsymbol{X})$ | 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 | 180 |
|---|---|---|---|---|---|---|---|---|---|
| NNM-RPCA | 2.26e-8 | 4.58e-8 | 7.44e-8 | 2.50e-4 | 2.31e-2 | 6.16e-2 | 9.96e-2 | 0.15 | 0.22 |
| WNNM-RPCA | 2.34e-8 | 3.71e-8 | 6.03e-8 | 8.87e-8 | 1.37e-7 | 1.82e-7 | 2.24e-7 | 4.80e-3 | 2.41e-2 |

To quantitatively evaluate the performance of the proposed WNNM-RPCA model, we generate synthetic low rank matrix recovering simulations for testing. The ground truth low rank data matrix $\boldsymbol{X} \in \Re^{m \times m}$ is obtained by the multiplication of two low rank matrices: $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{B}^T$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are both of size $m \times r$. Here $r = p_r \times m$ constrains the upper bound

Table 5.4. Relative error of low rank matrix recovery results by NNM-RPCA and WNNM-RPCA, with $p_e$ fixed as 0.2, and $p_r$ varying from 0.05 to 0.45 with step length 0.05.

| $Rank(\boldsymbol{X})$ | 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 | 180 |
|---|---|---|---|---|---|---|---|---|---|
| NNM-RPCA | 4.22e-8 | 6.84e-8 | 8.89e-3 | 5.80e-2 | 9.29e-2 | 0.12 | 0.14 | 0.18 | 0.24 |
| WNNM-RPCA | 3.68e-8 | 6.09e-8 | 1.18e-7 | 1.72e-7 | 3.76e-4 | 2.94e-2 | 5.42E-2 | 6.82e-2 | 7.53e-2 |

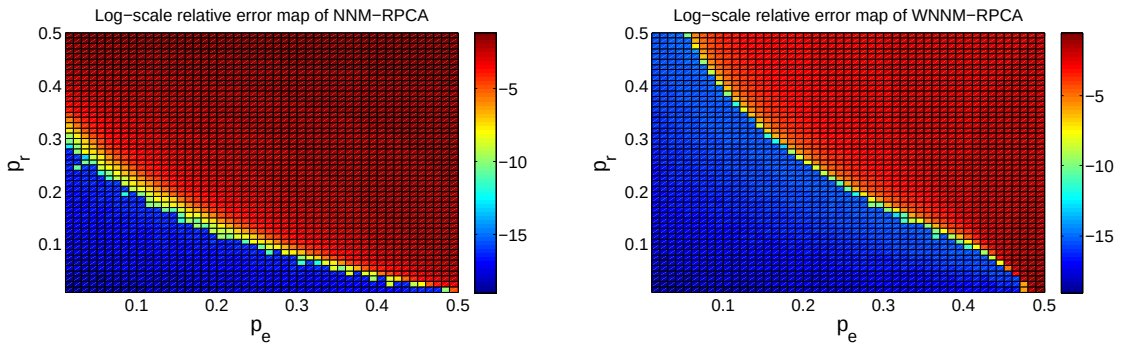



Figure 5.4. The log-scale relative error $log\frac{\|\hat{\boldsymbol{X}}-\boldsymbol{X}\|_F^2}{\|\boldsymbol{X}\|_F^2}$ of NNM-RPCA and WNNM-RPCA with different rank and outlier rate settings $\{p_r, p_e\}$.

of $Rank(\boldsymbol{X})$. In all experiments, each element of $\boldsymbol{A}$ and $\boldsymbol{B}$ is generated from a Gaussian distribution $\mathcal{N}(0, 1)$. The ground truth matrix $\boldsymbol{X}$ is corrupted by sparse noise $\boldsymbol{E}$ which has $p_e \times m^2$ non-zero entries. The non-zero entries in $\boldsymbol{E}$ are located in random positions and the value of each non-zero element is generated from a uniform distribution between [-5, 5]. We set $m = 400$, and let both $p_r$ and $p_e$ vary from $0.01$ to $0.5$ with step length $0.01$. For each parameter setting $\{p_r, p_e\}$, we generate the synthetic low-rank matrix 10 times and the final results are measured by the average of these 10 runs.

For the NNM-RPCA model, there is an important parameter $\lambda$. We set it as $1/\sqrt{m}$ following the suggested setting of [14]. For our WNNM-RPCA model, the parameter $C$ is empirically set as the square root of matrix size, i.e., $C = \sqrt{m \times m} = m$, in all experiments. The parameters in ADMM for both methods are set as $\rho = 1.05$. Typical experimental results are listed in Tables 5.2-5.4 for easy comparison.

It is easy to see that when the rank of matrix is low or the number of corrupted entries is small, both NNM-RPCA and WNNM-RPCA models are able to deliver accurate estimation of the ground truth matrix. However, with the rank of matrix or the number of corrupted entries getting larger, NNM-RPCA fails to deliver an accurate estimation of the ground truth matrix, yet the error of the results by WNNM-RPCA is much smaller than NNM-RPCA in these cases. In Fig. 5.4, we show the log-scale relative error map of the recovered matrices by NNM-RPCA and WNNM-RPCA with different settings of $\{p_r, p_e\}$. It is clear that the success area of WNNM-RPCA is much larger than NNM-RPCA, which means that WNNM-RPCA has much better low-rank matrix reconstruction capability in the presence of outliers/sparse noise.

### 5.4.2 Experimental results on background subtraction

As an important application in video surveillance, background sbutraction refers to the problem of separating the moving objects in foreground and the stable scene in background. The matrix $\boldsymbol{Y}$ obtained by stacking the video frames as columns forms a low-rank matrix with stationary background corrupted by sparse moving objects in the foreground. Thus, RPCA

model is appropriate to deal with this problem. We compare WNNM-RPCA with NNM-RPCA and several representative low-rank learning models, including the classic iteratively reweighted least squares (IRLS) based RPCA model[7] [32], and the $\ell_2$-norm and $\ell_1$-norm based matrix factorization models: singular value decomposition (SVD), Bayesian robust matrix factorization (BRMF)[8] [128] and RegL1ALM[9] [151]. The results of a recently proposed Mixture of Gaussian (MoG) model[10] [94] [150] are also included. These comparison methods range over state-of-the-art $\ell_2$ norm, $\ell_1$ norm and probabilistic subspace learning methods, including both categories of rank minimization and LRMF based approaches. We downloaded the codes of these algorithms from the corresponding authors' websites and keep their initialization and stopping criteria unchanged. The code of the proposed WNNM-RPCA model can be downloaded at http://www4.comp.polyu.edu.hk/∼cslzhang/code/WNNM_RPCA_code.zip.

Four benchmark video sequences provided by Li et al. [78] are adopted in our experiments, including two outdoor scenes (*Fountain* and *Watersurface*) and two indoor scenes (*Curtain* and *Airport*). In each sequence, 20 frames of ground truth foreground regions were provided by Li et al. for quantitative comparison. For all the comparison methods, parameters are fixed on the four sequences. We follow the experimental setting in [150] and constrain the maximum rank to 6 for all the factorization based methods. The regularization parameter $\lambda$ for the $\ell_1$-norm sparse term in the NNM-RPCA model is set to $\frac{1}{2\sqrt{max(m,n)}}$, since we empirically found that it can perform better than the recommended parameter $\frac{1}{\sqrt{max(m,n)}}$ in the original paper [14] in this series of experiments. For the proposed WNNM-RPCA model, we set $C = \sqrt{2max(m^3, n^3)}$ in all experiments.

To quantitatively compare the performance of competing methods, we use $S(A, B) = \frac{A \cap B}{A \cup B}$ to measure the similarity between the estimated foreground regions and the ground truth ones. To generate the binary foreground map, we applied the Markov random field (MRF)

[7] hhttp://www.cs.cmu.edu/ ftorre/codedata.html

[8] http://winsty.net/brmf.html

[9] https://sites.google.com/site/yinqiangzheng/

[10] http://www.cs.cmu.edu/ deyum/Publications.htm

Table 5.5. Quantitative performance ($S$) comparison of background subtraction results obtained by different methods.

| Method | Watersurface | Fountain | Airport | Curtain |
|---|---|---|---|---|
| **SVD** | 0.0995 | 0.2840 | 0.4022 | 0.1615 |
| **IRLS** | 0.4917 | 0.4894 | 0.4128 | 0.3524 |
| **BRMF** | 0.5786 | 0.5840 | 0.4694 | 0.5998 |
| **RegL1ALM** | 0.1346 | 0.4248 | 0.4420 | 0.2983 |
| **MoG** | 0.2782 | 0.4342 | 0.4921 | 0.3332 |
| **NNM-RPCA** | 0.7703 | 0.5859 | 0.3782 | 0.3191 |
| **WNNM-RPCA** | **0.7884** | **0.6043** | **0.5144** | **0.7863** |

model to label the absolute value of the estimated sparse error. The MRF labeling problem was solved by the multi-label optimization tool box [6]. The quantitative results of $S$ by different methods are shown in Tabel 5.5. One can see that on all the four utilized sequences, the proposed WNNM-RPCA model outperforms all other competing methods.

The visual results of representative frames in the *Watersurface* and *Curtain* sequences are shown in Fig. 5.5 and Fig. 5.6. From these figures, we can see that WNNM-RPCA method is able to deliver clear background estimation even under prominently embedded foreground moving objects. This on the other hand facilitates a more accurate foreground estimation. Comparatively, in the results estimated by the other methods, there are some ghost shadows in the background, leading to relatively less complete foreground detection results.

## 5.5 WNNM-MC for Image Inpainting

Matrix completion refers to the problem of recovering a matrix from only partial observation of its entries. It is a well known ill-posed problem which needs prior of the ground truth matrix as supplementary information for reconstruction. Fortunately, in many practical instance, the matrix to be recovered has a low-rank structure. Such prior knowledge has been utilized in many low-rank matrix completion (LRMC) methods, such as ranking and collaborative filtering [115] and image inpainting [148]. Matrix completion can be solved by both the matrix factorization or the rank minimization approaches. As the exact recovery property of the NNM-based methods has been proved in [15], this methodology has received great research interest, and many algorithms have been proposed to solve the NNM-MC problem [12, 82]. In the following, we provide experimental results on synthetic data and image inpainting to show the superiority of the proposed WNNM-MC model to the traditional NNM-MC technology.

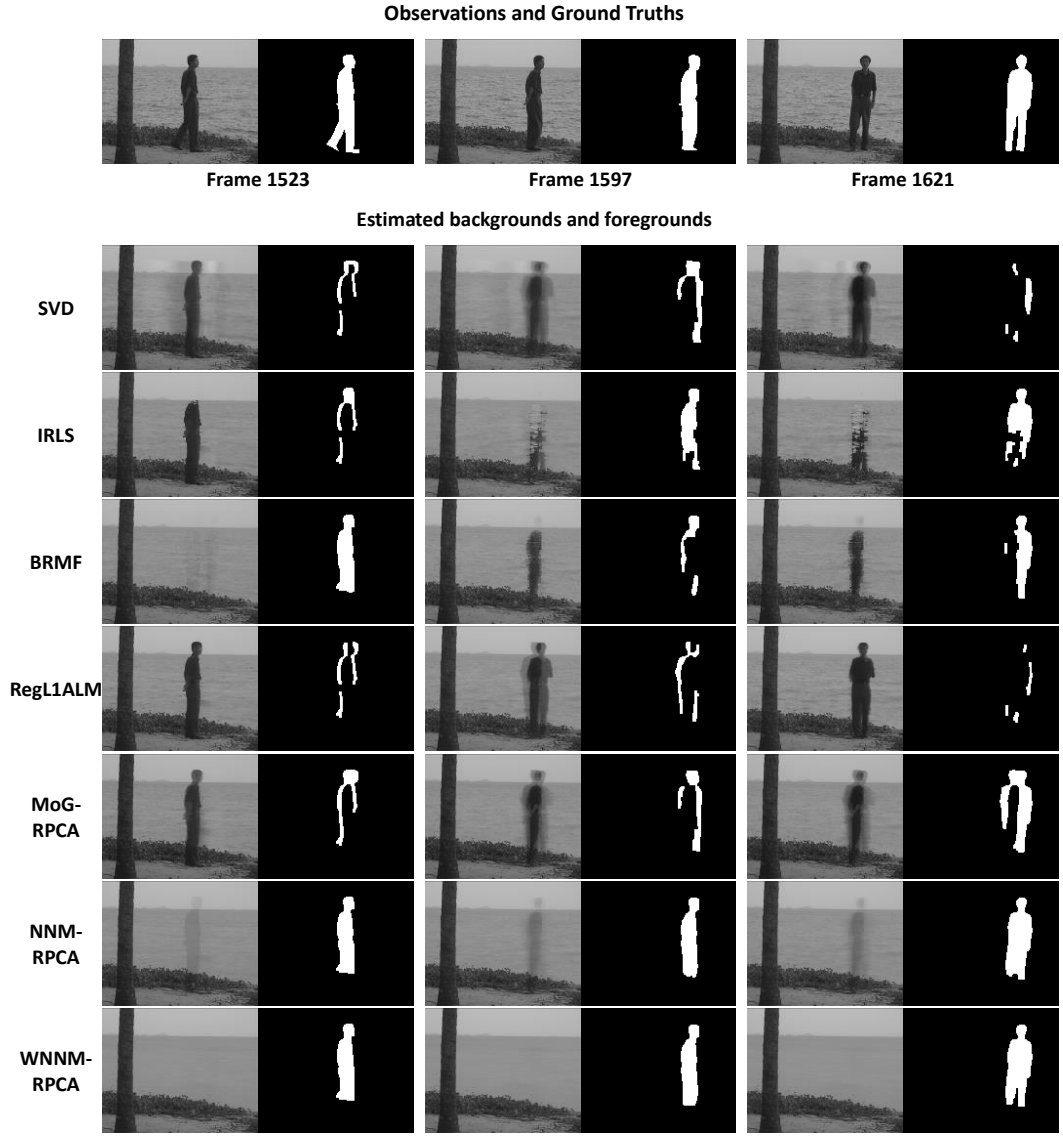



Figure 5.5. Performance comparison in visualization of competing methods on the *Water-surface* sequence. First row: the original frames and annotated ground truth foregrounds. Second row to the last row: estimated backgrounds and foregrounds by SVD, IRLS, BRMF, RegL1ALM, MoGRPCA, NNM-RPCA and WNNM-RPCA, respectively.

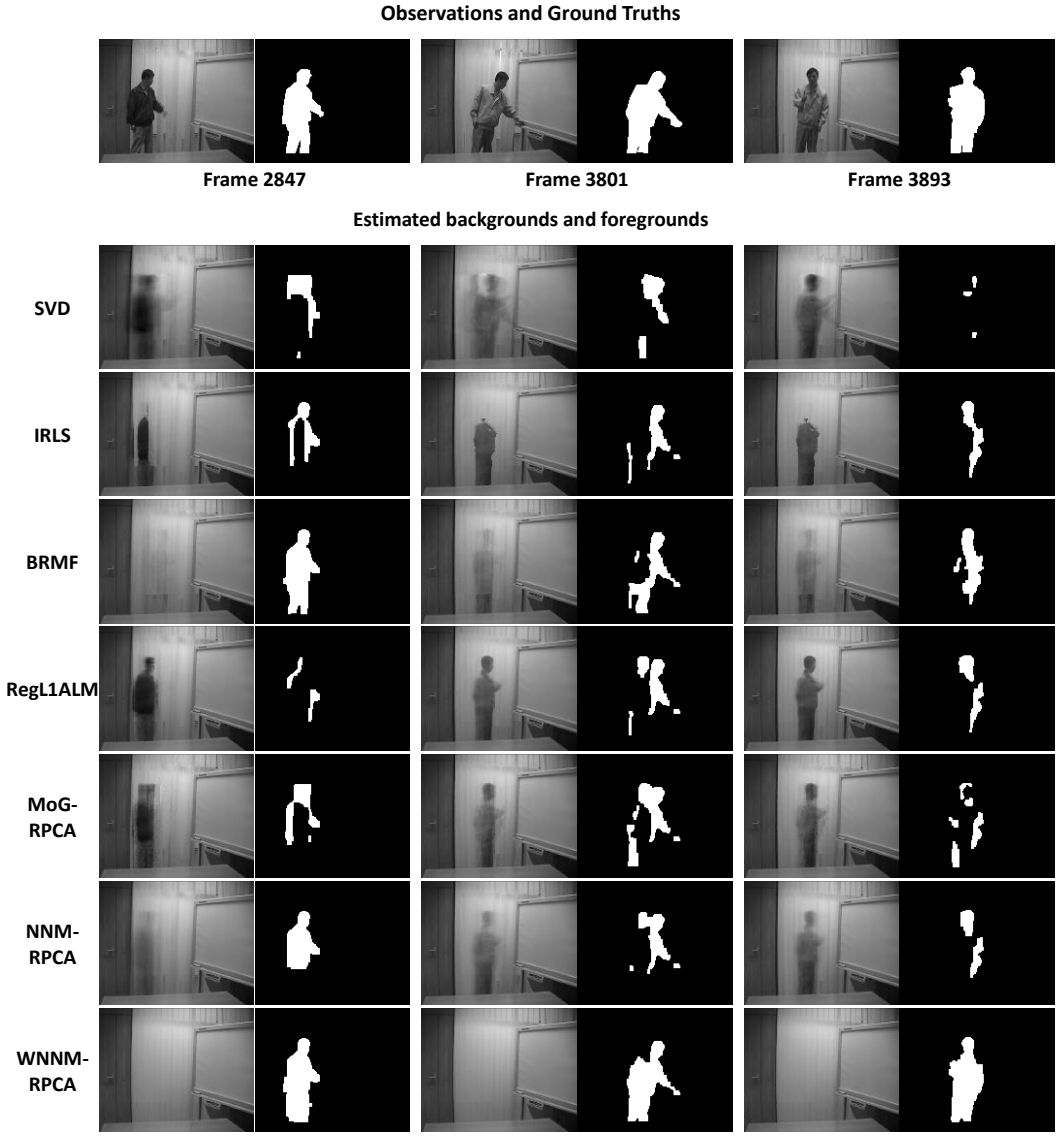



Figure 5.6. Performance comparison in visualization of competing methods on the *Curtain* sequence. First row: the original frames and annotated ground truth foregrounds. Second row to the last row: estimated backgrounds and foregrounds by SVD, IRLS, BRMF, RegL1ALM, MoGRPCA, NNM-RPCA and WNNM-RPCA, respectively.

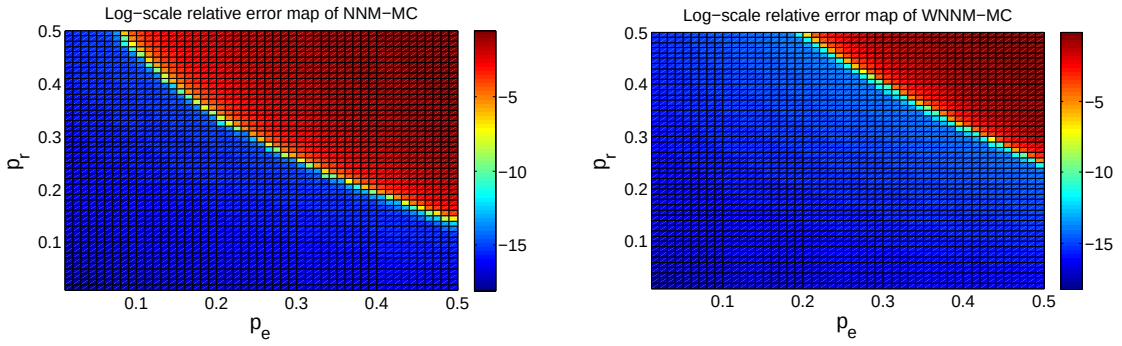



Figure 5.7. The log-scale relative error $log\frac{\|\hat{\boldsymbol{X}}-\boldsymbol{X}\|_F^2}{\|\boldsymbol{X}\|_F^2}$ of NNM-MC and WNNM-MC with different rank and outlier rate settings $\{p_r, p_e\}$.

Table 5.6. Relative error of low rank matrix recovery results by NNM-MC and WNNM-MC, with $p_e$ fixed as 0.1, and $p_r$ varying from 0.05 to 0.45 with step length 0.05.

| $Rank(\boldsymbol{X})$ | 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 | 180 |
|---|---|---|---|---|---|---|---|---|---|
| NNM-RPCA | 5.51e-8 | 7.25e-8 | 9.51e-8 | 1.12e-7 | 1.43e-7 | 1.76e-7 | 2.10e-7 | 2.58e-7 | 9.97e-5 |
| WNNM-RPCA | 4.40e-8 | 7.72e-8 | 9.44e-8 | 1.18e-7 | 1.41e-7 | 1.77e-7 | 2.09e-7 | 2.53e-7 | 3.25e-7 |

Table 5.7. Relative error of low rank matrix recovery results by NNM-MC and WNNM-MC, with $p_e$ fixed as 0.2, and $p_r$ varying from 0.05 to 0.45 with step length 0.05.

| $Rank(\boldsymbol{X})$ | 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 | 180 |
|---|---|---|---|---|---|---|---|---|---|
| NNM-RPCA | 7.33e-8 | 1.03e-7 | 1.22e-7 | 1.65e-7 | 2.01e-7 | 2.76e-7 | 1.91e-2 | 8.52e-2 | 0.14 |
| WNNM-RPCA | 6.35e-8 | 9.21e-8 | 1.30e-7 | 1.60e-7 | 1.94e-7 | 2.48e-7 | 3.32e-7 | 4.66e-7 | 7.21e-7 |

Table 5.8. Relative error of low rank matrix recovery results by NNM-MC and WNNM-MC, with $p_e$ fixed as 0.3, and $p_r$ varying from 0.05 to 0.45 with step length 0.05.

| $Rank(\boldsymbol{X})$ | 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 | 180 |
|---|---|---|---|---|---|---|---|---|---|
| NNM-RPCA | 9.20e-8 | 1.21e-7 | 1.61e-7 | 2.06e-7 | 0.53e-5 | 8.94e-2 | 0.18 | 0.25 | 0.30 |
| WNNM-RPCA | 9.31e-8 | 1.21e-7 | 1.60e-7 | 2.13e-7 | 2.81e-7 | 4.00e-7 | 6.15e-7 | 1.71e-2 | 0.22 |

### 5.5.1 Experimental results on synthetic data

We first compare NNM-MC with WNNM-MC using synthetic low-rank matrices. Similar to our experimental setting in the RPCA problem, we generate the ground truth low-rank matrix by a multiplication of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of size $m \times r$. Here $r = p_r \times m$ constrains the upper bound of $Rank(\boldsymbol{X})$. All of their elements are generated from the Gaussian distribution $\mathcal{N}(0, 1)$. In the observed matrix $\boldsymbol{Y}$, $p_e \times m^2$ entries in the ground truth matrix $\boldsymbol{X}$ are missing. We set $m = 400$, and let $p_r$ and $p_e$ vary from $0.01$ to $0.5$ with step length $0.01$. For each parameter setting of $\{p_r, p_e\}$, 10 groups of synthetic data are generated for testing and the performance of each method is assessed by the average of the 10 runs on these groups.

In all the experiments we fix parameters $\lambda = 1/\sqrt{m}$ and $C = m$ in NNM-MC and WNNM-MC, respectively. The parameter $\rho$ in the ALM algorithm is set to 1.05 for both methods. Typical experimental results are listed in Tables 5.6-5.8.

It can be easily observed that when the rank of latent ground truth matrix $\boldsymbol{X}$ is relatively low, both NNM-RPCA and WNNM-RPCA can successfully recover it with high accuracy. The advantage of WNNM-MC over NNM-MC is reflected when dealing with more challenging cases. Table 5.7 shows that when $20\%$ of entries in the matrix are missing, NNM-MC will not have good recovery accuracy once the rank is higher than 120, while WNNM-MC can still have very high accuracy. Similar observations can be made in Table 5.8.

The log-scale relative error map with different settings of $\{p_r, p_e\}$ are shown in Fig. 5.7. From this figure, it is clear to see that WNNM-MC has a much larger success area than NNM-MC.

### 5.5.2 Experimental results on image inpainting

We then test the proposed WNNM-MC model on image inpainting. In some previous works, the whole image is assumed to be a low rank matrix and matrix completion is directly performed on the image to get the inpainting result. However, a natural image is only approximately low rank, and the small singular values in the long tail distribution include many details. Simply using the low rank prior on the whole image may fail to recover the missing pixels or lose too much detailed information in the image. As in the image denoising experiments, we utilize the NSS prior and perform WNNM-MC on each group of non-local similar patches for this task. We initialize the inpainting by the field of experts (FOE) [111] method to search the non-local patches, and then for each patch group we perform WNNM-MC to get an updated low-rank reconstruction. After the first round estimation of the missing values in all patch groups, all reconstructed patches are aggregated to get the recovered image. We then perform a new stage of similar patch searching based on the first round estimation, and iteratively implement the similar process to converge to a final inpainting output.

The first 12 test images[11] with size $256\times256$ in Fig. 5.1 are used to evaluate WNNM-MC. Random masks with 25%, 50% and 75% missing pixels and a text mask are used to test the inpainting performance, respectively. We compare WNNM-MC with NNM-MC and several representative and state-of-the-art inpainting methods, including the TV method [21], FOE method[12] [111], variational nonlocal example-based (VNL) method[13] [2] and the beta process dictionary learning (BPDL) method[14] [153]. The setting of the TV inpainting method

[11] The color versions of images #3, #5, #6, #7, #9, #11 are used in this MC experiment.

[12] http://www.gris.informatik.tu-darmstadt.de/ sroth/research/foe

[13] http://gpi.upf.edu/static/vnli/interp/interp.html

[14] http://people.ee.duke.edu/ mz1/Softwares

Table 5.9. Inpainting results (PSNR, dB) by different methods.

| | Random mask with 25% missing entries | | | | | | Random mask with 50% missing entries | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **TV** | **FOE** | **VNL** | **BPDL** | **NNM** | **WNNM** | **TV** | **FOE** | **VNL** | **BPDL** | **NNM** | **WNNM** |
| C.Man | 32.20 | 30.23 | 26.98 | 33.39 | 34.12 | **35.21** | 27.41 | 27.42 | 25.71 | 28.59 | 29.42 | **30.58** |
| House | 39.37 | 41.90 | 33.69 | 42.03 | 42.90 | **44.59** | 34.25 | 36.84 | 32.35 | 37.63 | 37.45 | **38.83** |
| Peppers | 37.44 | 38.46 | 31.00 | 39.66 | 39.65 | **41.53** | 32.16 | 34.51 | 28.80 | 34.80 | 34.02 | **35.85** |
| Montage | 32.28 | 28.00 | 28.45 | 35.86 | 37.48 | **39.63** | 26.47 | 24.53 | 26.41 | 29.68 | 29.91 | **31.02** |
| Leaves | 32.10 | 30.52 | 27.57 | 36.77 | 36.27 | **38.95** | 26.07 | 27.22 | 25.32 | 30.35 | 29.87 | **32.32** |
| StarFish | 34.20 | 35.34 | 27.44 | 36.94 | 36.79 | **38.93** | 29.05 | 31.18 | 26.13 | 31.84 | 31.36 | **33.03** |
| Monarch | 34.27 | 32.92 | 28.55 | 36.74 | 36.51 | **38.14** | 28.84 | 29.16 | 26.84 | 31.17 | 31.02 | **32.75** |
| Airplane | 30.80 | 30.34 | 26.05 | 33.19 | 32.94 | **33.76** | 26.61 | 28.11 | 24.62 | 29.00 | 28.72 | **29.30** |
| Paint | 34.75 | 35.87 | 28.87 | 37.27 | 36.84 | **38.79** | 29.29 | 31.45 | 27.22 | 32.62 | 31.52 | **33.02** |
| J.Bean | 41.56 | 44.57 | 34.97 | 45.15 | 44.49 | **48.04** | 35.61 | 38.88 | 32.46 | 40.07 | 37.47 | **40.93** |
| Fence | 30.24 | 31.96 | 28.98 | 35.85 | 36.55 | **37.91** | 25.06 | 29.97 | 27.43 | 31.47 | 31.77 | **32.85** |
| Parrot | 32.88 | 30.69 | 27.50 | 33.37 | 33.93 | **35.09** | 27.77 | 28.24 | 25.80 | 28.73 | 29.20 | **30.52** |
| **AVE.** | 34.340 | 34.234 | 29.170 | 37.185 | 37.372 | **39.215** | 29.049 | 30.625 | 27.425 | 32.162 | 31.812 | **33.416** |
| | Random mask with 75% missing entries | | | | | | Text mask | | | | | |
| C-Man | 23.65 | 23.35 | 23.65 | 24.17 | 23.55 | **25.69** | 28.49 | 27.27 | 26.75 | 26.24 | 29.74 | **31.68** |
| House | 29.59 | 31.63 | 31.07 | 31.61 | 28.62 | **34.12** | 34.75 | 37.41 | 34.46 | 32.01 | 37.69 | **39.60** |
| Peppers | 27.16 | 28.86 | 26.72 | 28.97 | 27.22 | **30.04** | 33.90 | 34.82 | 31.72 | 29.57 | 34.55 | **37.04** |
| Montage | 22.47 | 21.92 | 23.62 | 24.89 | 23.35 | **25.65** | 27.20 | 26.06 | 27.54 | 25.62 | 29.77 | **31.47** |
| Leaves | 20.15 | 20.69 | 22.30 | 23.11 | 22.08 | **25.01** | 25.48 | 24.21 | 26.39 | 23.07 | 27.32 | **30.69** |
| Starfish | 24.44 | 26.43 | 24.36 | 26.57 | 25.45 | **27.11** | 29.39 | 30.97 | 27.53 | 27.24 | 31.66 | **33.44** |
| Mornar. | 23.97 | 24.10 | 24.77 | 26.08 | 24.84 | **27.45** | 28.35 | 27.51 | 27.56 | 26.10 | 29.89 | **32.85** |
| Plane | 23.16 | 24.20 | 22.03 | 24.87 | 24.05 | **25.47** | 28.70 | 28.79 | 25.95 | 26.61 | 29.66 | **30.57** |
| Paint | 24.05 | 25.81 | 24.73 | 26.71 | 25.02 | **26.93** | 29.54 | 30.50 | 28.47 | 31.86 | 31.32 | **32.96** |
| J.Bean | 29.95 | 32.89 | 29.48 | 32.46 | 28.00 | **33.40** | 34.50 | 37.12 | 32.59 | 32.62 | 35.71 | **38.91** |
| Fence | 21.03 | 23.56 | 25.49 | 26.12 | 25.65 | **28.38** | 25.44 | 27.55 | 29.55 | 27.01 | 32.44 | **34.62** |
| Parrot | 22.85 | 23.33 | 23.54 | 24.53 | 24.02 | **25.61** | 28.07 | 27.54 | 26.58 | 25.83 | 29.54 | **30.61** |
| **AVE.** | 24.372 | 25.564 | 25.146 | 26.674 | 25.155 | **27.905** | 29.484 | 29.979 | 28.757 | 27.814 | 31.606 | **33.702** |

follows the implementation of [31][15], and the codes for other comparison methods are provided by the original authors. The source code of the proposed WNNM-MC model can be downloaded at http://www4.comp.polyu.edu.hk/∼cslzhang/code/WNNM_MC_code.zip.

The PSNR results by different methods are shown in Table 5.9. It is easy to see that WNNM-MC achieves much better results than the other methods. Visual examples on a random mask and a text mask are shown in Figs. 5.8 and 5.9, respectively. More visual examples can be found in the supplementary file. From the enlarged demarcated windows, we can see that inpainting methods based on image local prior (e.g., TV, FOE and BPDL) are able to recover the image smooth areas, while they have difficulties in extracting the details in edge and texture areas. The VNL, NNM and WNNM methods utilized the rational NSS prior, and thus the results are more visually plausible. However, in some challenging cases when the percentage of missing entries is high, it can be observed that VNL and NNM more or less generate artifacts across the recovered image. As a comparison, the proposed WNNM-MC model has much better visual quality of the inpainting results.

## 5.6 Discussions

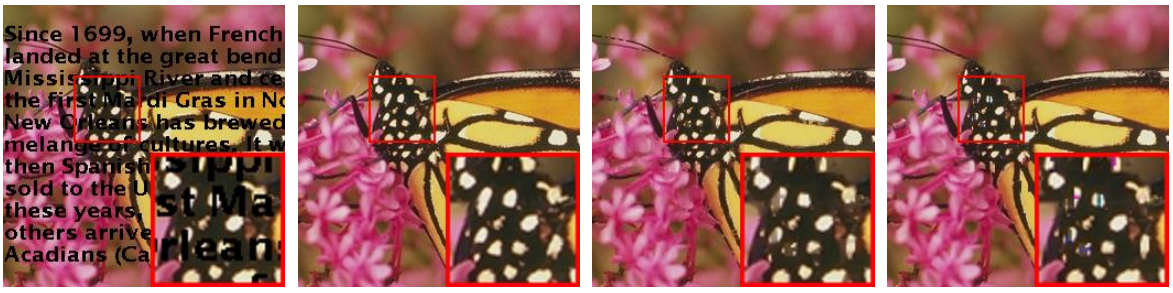
To improve the flexibility of the original nuclear norm, we proposed the weighted nuclear norm and studied its minimization strategy in this chapter. Based on the observed data and the specific application, different weight setting strategies can be utilized to achieve better performance. Inspired by [16], we utilized the reweighting strategy $w_i^\ell = \frac{C}{|\sigma_i(X_\ell)|+\varepsilon}$ to approximate the $\ell_0$ norm on the singular values of the data matrix. Other than this setting, there also exist other weight setting strategies for certain types of data and applications. For instance, the truncated nuclear norm regularization (TNNR) [148] and the partial sum minimization (PSM) [104] were proposed to regularize only the smallest $N - r$ singular values. Actually, they can be viewed as a special case of WNNM, where weight vector $[w_{1\cdots r} = 0, w_{r+1\cdots N} = \lambda]$ is used to approximate the rank function of matrix. The Schatten-$p$

[15] http://www.imm.dtu.dk/ pcha/mxTV/

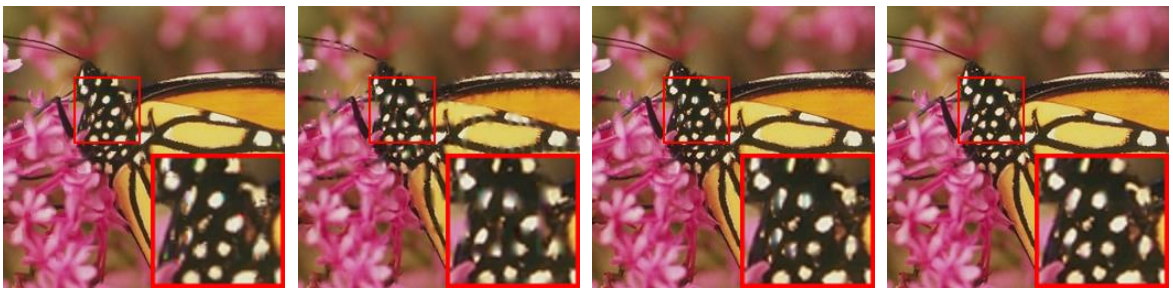
(a) Input image (b) Ground truth (c) TV (PSNR: 24.44 dB) (d) FOE (PSNR: 26.43 dB)

(e) VNL (PSNR: 24.36 dB) (f) BPDL (PSNR: 26.57 dB) (g) NNM (PSNR: 25.45 dB) (h) WNNM (PSNR: 27.11 dB)

Figure 5.8. Inpainting results on image *Starfish* by different methods (Random mask with 75% missing values).



(a) Input image (b) Ground truth (c) TV (PSNR: 28.35 dB) (d) FOE (PSNR: 27.51 dB)



(e) VNL (PSNR: 27.56 dB) (f) BPDL (PSNR: 26.10 dB) (g) NNM (PSNR: 29.89 dB) (h) WNNM (PSNR: 32.85 dB)

Figure 5.9. Inpainting results on image *Monarch* by different methods (Text mask).

Table 5.10. The average PSNR (dB) values of denoising results by competing methods on the 20 test images. The best results are highlighted in bold.

| | $\sigma_n$=10 | $\sigma_n$=20 | $\sigma_n$=30 | $\sigma_n$=40 | $\sigma_n$=50 | $\sigma_n$=75 | $\sigma_n$=100 |
|---|---|---|---|---|---|---|---|
| **NNM** | 33.462 | 30.040 | 27.753 | 26.422 | 25.048 | 22.204 | 21.570 |
| **TNNM** | 34.125 | 30.558 | 28.627 | 27.250 | 26.256 | 24.284 | 23.109 |
| **SPNM** | 34.739 | 31.048 | 29.075 | 27.488 | 26.582 | 24.757 | 23.174 |
| **WNNM** | **34.772** | **31.200** | **29.214** | **27.780** | **26.752** | **24.860** | **23.555** |

norm minimization (SPNM) methods [103] [99] can also be understood as special cases of WNNM since the $\ell_p$ norm proximal problem can be solved by the iteratively thresholding method [117]. The same strategy as in our work can be used to set weights for each singular values.

In Tables 5.10, 5.11 and 5.12, we compare the proposed WNNM with the TNNR and SPNM methods on image denosing, background subtraction and image inpainting, respectively. The results obtained by the NNM method are also shown in the tables as baseline comparison. The experimental settings on the three applications are exactly the same as the experiments in Sections 5.3, 5.4 and 5.5. For the TNNR method, there are two parameters $r$ and $\lambda$, which represent the truncation position and the regularization parameter for the remaining $N - r$ singular values. For the SPNM model, we need to set the $p$ value for the $\ell_p$ norm and the regularization parameter $\gamma$. We tried our best to adjust these parameters for the two models for different applications. The remaining parameters are the same as the WNNM model on each task.

From the tables, we can find that regularizing more flexibly the singular values is beneficial for low rank models. Besides, WNNM outperforms TNNR and SPNM in the testing applications, which validates the superiority of the proposed method along this line of research. Nonetheless, it is worth to note that there may exist other weighting mechanisms

Table 5.11. The average PSNR (dB) values of inpainting results by competing methods on the 12 test images. The best results are highlighted in bold.

| | 25% missing entries | 50% missing entries | 75% missing entries | Text Mask |
|---|---|---|---|---|
| **NNM** | 37.372 | 31.812 | 25.155 | 31.606 |
| **TNNM** | 37.789 | 32.378 | 27.201 | 32.738 |
| **SPNM** | 39.203 | 33.196 | 27.800 | 33.668 |
| **WNNM** | **39.215** | **33.416** | **27.905** | **33.702** |

Table 5.12. Quantitative performance ($S$) comparison of background subtraction results obtained by different methods.

| Method | Watersurface | Fountain | Airport | Curtain |
|---|---|---|---|---|
| **NNM-RPCA** | 0.7703 | 0.5859 | 0.3782 | 0.3191 |
| **TNNM-RPCA** | 0.7772 | 0.5926 | 0.3829 | 0.3310 |
| **SPNM-RPCA** | **0.7906** | 0.6033 | 0.3714 | 0.3233 |
| **WNNM-RPCA** | 0.7884 | **0.6043** | **0.5144** | **0.7863** |

which might achieve better performance on certain computer vision tasks. Actually, since we have proved in Theorem 5.2.1 that the WNNP problem is equivalent to an element wise proximal problem with order constrains, a wide range of weighting and reweighting strategies can be used to regularize the singular values of a data matrix. One important research topic of our future work is thus to investigate more sophisticated weight setting mechanisms for different types of data and applications.

## 5.7 Conclusion

In this chapter, we extended the standard nuclear norm minimization scheme to the weighted nuclear norm minimization (WNNM) problem. We first presented the solving strategy for the weighted nuclear norm proximal (WNNP) operator under $\ell_2$-norm fidelity loss to facilitate the solving of different WNNM paradigms. We then extended the WNNM model to robust PCA (RPCA) and matrix completion (MC) tasks and constructed efficient algorithms to solve them based on the derived WNNP operator. Inspired by previous results on reweighted sparse coding, we further designed a rational scheme for automatic weight setting, which offers closed form solutions of the WNNP operator and eases the utilization of WNNM in different applications.

We validated the effectiveness of the proposed WNNM models on multiple low level vision tasks. For baseline WNNM, we applied it to image denoising, and achieved state-of-the-art performance in both quantitative PSNR measures and visual qualities. For WNNM-RPCA, we applied it to background subtraction, and validated its superiority among up-to-date low-rank learning methods. For WNNM-MC, we applied it to image inpainting, and demonstrated its superior performance to state-of-the-art inpainting technologies.

# CHAPTER 6

# CONCLUSION AND FUTURE WORKS

The image sparsity prior plays an important role in many image restoration algorithms. In this thesis, we studied the analysis-based and synthesis-based sparse representation methods, as well as the low-rank minimization methods, and developed new sparsity models for different image restoration applications.

To improve the flexibility of conventional analysis sparse representation (ASR) models, in chapter 2 we proposed a weighted ASR learning model for guided image restoration. A weighting function was learned to introduce structural information of the guidance image into the target image. We unfolded the sparse optimization into several stages and exploited the task-driven training strategy to learn state-wise parameters. Our experimental results on the guided depth image enhancement clearly demonstrated the advantages of the proposed algorithm.

In chapter 3, we addressed the inconsistency issue in the commonly used patch-based synthesis sparse representation (SSR) models, and proposed a convolutional sparse coding (CSC) algorithm for super-resolution (SR). With CSC, we decomposed an input low resolution image without patch dividing, and learned a mapping function between the sparse coding feature maps of low-resolution and high-resolution images for SR. The proposed CSC-SR algorithm preserves the consistency of neighboring patches and delivers highly competitive SR results with pleasant visual quality.

To integrate the advantages of ASR and SSR models in describing image major structures and repetitive textures, in chapter 4 we proposed a joint convolutional analysis and synthesis (JCAS) model for single image layer separation. The superior layer separation results

over previous methods on applications of rain streak removal and texture-cartoon decomposition demonstrated the effectiveness of the proposed JCAS model.

Finally, in chapter 5 we extended the conventional matrix nuclear norm minimization to the weighted nuclear norm minimization (WNNM). We showed that although the WNNM model is non-convex, it still has a globally optimal solution. Particularly, when the weights are assigned in a non-descending order, the optimal solution has a closed-form. We then extended the WNNM model to robust PCA (RPCA) and matrix completion (MC) tasks. The proposed WNNM models demonstrated state-of-the-art performance on multiple low level vision tasks, including image denoising, background subtraction and image inpainting.

The proposed algorithms in this thesis not only advance much the performance of sparsity-based image restoration methods, but also deepen the understanding of sparsity-based statistical modeling of natural images. In our future work, we will investigate the following problems.

First, our proposed weighted ASR model un-folds the sparse optimization problem and adopts the task-driven training strategy to learn state-wise parameters. It can be viewed as a special convolutional neural network (CNN) structure. It is an interesting problem to investigate other structures for guided image restoration tasks and study their relationship with CNN.

Second, in the proposed CSC-SR algorithm the LR and HR filters are trained separately for low-resolution image decomposition and high-resolution image reconstruction. How to train the two groups of filters jointly in an end-to-end manner is an important direction to further improve our CSC-SR algorithm.

Third, the proposed JCAS algorithm has achieved state-of-the-art performance on rain streak removal and texture-cartoon decomposition. We will adopt it to other image enhancement applications, such as high dynamic range imaging and image detail enhancement, in which layer separation is an important pre-processing step.

At last, the proposed WNNM model has achieved state-of-the-art performance on

several low level vision problems with the same weight setting. It is interesting to investigate whether there is a better weight setting strategy, or how to more effectively set the weights for a given task.

# Appendix

APPENDIX A

## A.1 A Brief Introduction to SA-ADMM

Here we briefly introduce the algorithm of SA-ADMM. For more details of the algorithm and the convergence analysis, please refer to the original paper [152].

The ADMM algorithm is proposed to solve the following problem:

$$\min_{x,y} \phi(x) + \psi(y) \quad s.t. Ax + By = c. \tag{1}$$

In many real applications, $\phi(x)$ in (1) can be written as

$$\phi(x) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(x) + \Omega(x), \tag{2}$$

where $\ell_i$ is the contribution from the $i$-th sample, and $n$ is the number of samples. For such a problem, the original ADMM algorithm suffers from a heavy computation burden in the step of updating $x$ when $n$ is a large number.

If the function $\ell_i(x)$ in (2) is $L$-smooth[1], SA-ADMM algorithm can be used to solve the problem (1). More specifically, in our case $\Omega = 0$. For problem (1) with $\phi(x) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(x)$, the SA-ADMM algorithm updates the variable $x$, $y$ and the Lagrangian variable $\alpha$ alternatively:

$$\begin{aligned}
&x_{t+1} \leftarrow argmin_x \frac{1}{n}\sum_{i=1}^{n} \ell_i(x_{\tau_i(t)}) + \bigtriangledown \ell_i(x_{\tau_i(t)})^T (x - x_{\tau_i(t)}) + \frac{L}{2}\|x - x_{\tau_i(t)}\|^2 + \frac{\rho}{2}\|Ax + By_t - c - \alpha_t\|^2, \\
&y_{t+1} \leftarrow argmin_y \ \psi(y) + \frac{\rho}{2}\|Ax_{t+1} + By - c + \alpha_t\|^2, \\
&\alpha_{t+1} \leftarrow \ \alpha_t + Ax_{t+1} + By_{t+1} - c,
\end{aligned} \tag{3}$$

[1]Let $\|\cdot\|$ be the Euclidean norm. For a differentiable function $f$, we use $\bigtriangledown f$ to denote its gradient. A function $f$ is $L$-smooth if $\|\bigtriangledown f(x) - \bigtriangledown f(y)\| \leq L\|x - y\|$.

where $L$ is the scalar in the $L$-smooth definition, and $\tau_i(t)$ is defined as

$$\tau_i(t) = \begin{cases} t & i = k(t) \\ \tau_i(t-1) & otherwise \end{cases}. \tag{4}$$

The updating strategy of $y$ and $\alpha$ is the same as the standard ADMM algorithm. For the $x$ subproblem, by letting its derivative to zero, we have:

$$x_{t+1} \leftarrow (\rho A^T A + LI)^{-1}[L\overline{x}_t - \rho A^T(By_t - c + \alpha_t) - \overline{\bigtriangledown \ell_t}], \tag{5}$$

where $\overline{x}_t = \frac{1}{n}\sum_{i=1}^{n} x_{\tau_i(t)}$, and $\overline{\bigtriangledown \ell}_t = \frac{1}{n}\sum_{i=1}^{n} \bigtriangledown \ell_i(x_{\tau_i(t)})$.

Denote by $(x^*, y^*)$ the optimal solution of (1) , Zhong et al. have proved that

$$\begin{aligned} &\mathbb{E}[\Phi(\overline{x}_T, \overline{y}_T) - \Phi(x^*, y^*) + \gamma\|A\overline{x}_T + B\overline{y}_T - c\|] \\ \leq &\frac{1}{2T}\{\|x^* - x_0\|_{H_x}^2 + nL\|x^* - x_0\|^2 + \|y^* - y_0\|_{H_y}^2 + 2\rho(\frac{\gamma^2}{\rho^2} + \|\alpha_0\|^2)\}, \end{aligned} \tag{6}$$

where $\|x\|_H = x^T H x$ for a positive semi-defined matrix $H$, $H_x = L_A I - \rho A^T A$ and $H_y = \rho B^T B$. Eq. (6) shows that the $\{\overline{x}_T, \overline{y}_T\}$ generated by SA-ADMM will converge with speed $\mathcal{O}(\frac{1}{T})$.

### A.2 Filter training by SA-ADMM

The filter training problem in our CSC model aims to optimize the following problem:

$$\boldsymbol{f} = \arg\min_{\boldsymbol{f}} \sum_{k=1}^{K} \| \boldsymbol{Y}_k - \sum\nolimits_{i=1}^{N} \boldsymbol{f}_i \otimes \boldsymbol{Z}_{k,i} \|_F^2, \ s.t. \|\boldsymbol{f}_i\|_F^2 \leq e. \tag{7}$$

Note that here we do not omit the index $k$ for the training image. $\boldsymbol{Y}_k$ is the $k$th training image and $\boldsymbol{Z}_{k,i}$ is the feature map produced by the $i$th filter $\boldsymbol{f}_i$ on $\boldsymbol{Y}_k$. Based on the properties of convolution and Kronecker product, we have the following equation:

$$vec(\boldsymbol{f} \otimes \boldsymbol{Z}) = \boldsymbol{F} * vec(\boldsymbol{Z}) = (\boldsymbol{I} \odot vec(\boldsymbol{Z}))vec(\boldsymbol{F}^T) = \mathbb{Z}^T * vec(f), \tag{8}$$

where $\odot$ and $vec(\bullet)$ denote the Kronecker product and the vectorization operation, respectively. $\boldsymbol{I}$ is the Identity matrix and $\boldsymbol{F}$ is the BCCB matrix corresponding to filter $\boldsymbol{f}$. $\mathbb{Z} = Image2Patch(\boldsymbol{Z})$ is the output of an *Image2Patch* operation on matrix $\boldsymbol{Z}$ with the size of filter $\boldsymbol{f}$, e.g., extracting all the patches from $\boldsymbol{Z}$ with the same size of $\boldsymbol{f}$.

Based on the above equations, the filter learning problem can be transformed to

$$\boldsymbol{f} = \arg\min_{\boldsymbol{f}} \sum_{k=1}^{K} \| \ vec(\boldsymbol{Y}_k) - [\mathbb{Z}_{k,1}^T, \mathbb{Z}_{k,2}^T, \ldots, \mathbb{Z}_{k,N}^T] * [vec(\boldsymbol{f}_1)^T, vec(\boldsymbol{f}_2)^T, \ldots, vec(\boldsymbol{f}_N)^T]^T \ \|_F^2, \quad s.t. \|\boldsymbol{f}_i\|_F^2 \leq e. \tag{9}$$

For the purpose of simplicity, we denote $[\mathbb{Z}_{k,1}^T, \mathbb{Z}_{k,2}^T, \ldots, \mathbb{Z}_{k,N}^T]$ by $\mathbb{Z}$ and $[vec(\boldsymbol{f}_1)^T, vec(\boldsymbol{f}_2)^T, \ldots, vec(\boldsymbol{f}_N)^T]^T$ by $\boldsymbol{f}$, the filter training problem with a large number of training images can be written as:

$$\boldsymbol{f} = \arg\min_{\boldsymbol{f}} \sum_{k} \|\boldsymbol{y}_k - \mathbb{Z}_k * \boldsymbol{f}\|^2 \quad s.t. \|\boldsymbol{f}_i\|^2 \leq e. \tag{10}$$

By introducing an augmented variable $\boldsymbol{s} = \boldsymbol{f}$, we can solve (10) by the SA-ADMM algorithm in (3):

$$\begin{aligned} \boldsymbol{f}_{t+1} &= [L\bar{\boldsymbol{f}}_t - \rho(\boldsymbol{d}_t - \boldsymbol{s}_t) - \frac{1}{K}\sum_{k=1}^{K} \mathbb{Z}_k^T(\mathbb{Z}_k \boldsymbol{f}_{\tau_j(t)} - \boldsymbol{Y}_k)]/(\rho + L) \\ \boldsymbol{s}_{t+1} &= argmin_{\boldsymbol{s}} \ \frac{\rho}{2}\|\boldsymbol{f}_{t+1} + \boldsymbol{d}_t - \boldsymbol{s}\|^2, \ \ s.t. \|\boldsymbol{s}_i\|^2 \leq e \\ \boldsymbol{d}_{t+1} &= \boldsymbol{d}_t + \boldsymbol{f}_{t+1} - \boldsymbol{s}_{t+1} \end{aligned} \tag{11}$$

For our square loss function in Eq. (7), a general scalar $L$ which satisfies the $L$-smooth condition is the upper bound on the eigenvalues of $\mathbb{Z}^T\mathbb{Z}$. The $\boldsymbol{s}$ problem is a proximal problem with $\ell_2$-norm ball constraint, which has a closed-form solution. Please note that, here $(\rho + L)$ is a scalar, and the updating of $\boldsymbol{f}$ does not need any matrix inverse calculation.

### A.2.1 Mapping Function Learning by SA-ADMM

The mapping function learning problem in our CSC model aims to optimize the following problem:

$$\{\boldsymbol{W}\} = \arg\min_{\boldsymbol{W}} \sum_{k=1}^{K} \| \boldsymbol{X}_k - \sum_{j=1}^{M} \boldsymbol{f}_j^h \otimes g(\boldsymbol{Z}_{k,:}^l; \boldsymbol{w}_j) \|_F^2, \quad s.t. \ \boldsymbol{w}_j \succeq 0, |\boldsymbol{w}_j|_1 = 1. \tag{12}$$

Denote by $\tilde{\boldsymbol{Z}}_i^l$ the upsampling of LR feature map

$$\tilde{\boldsymbol{Z}}_{k,i}^l(x', y') = \begin{cases} \boldsymbol{Z}_{k,i}^l(x, y) & if \ mod(x', factor) = 0 \ and \ mod(y', factor) = 0 \\ 0 & otherwise \end{cases}, \tag{13}$$

then we have

$$[vec(\boldsymbol{Z}_{k,1}^h), vec(\boldsymbol{Z}_{k,2}^h), \ldots, vec(\boldsymbol{Z}_{k,M}^h)] = [vec(\tilde{\boldsymbol{Z}}_{k,1}^l), vec(\tilde{\boldsymbol{Z}}_{k,2}^l), \ldots, vec(\tilde{\boldsymbol{Z}}_{k,N}^l)] * \boldsymbol{W}, \quad (14)$$

where $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_M]$ is the linear mapping function matrix, and $\boldsymbol{w}_j = [w_{j,1}, w_{j,2}, \ldots, w_{j,N}]^T$ is the linear transform vector used to predict the $j$th HR feature map. Utilizing the BCCB matrix corresponding to the HR filters, the original problem (12) can be rewritten as

$$\{\boldsymbol{W}\} = \sum_{k=1}^{K} \arg\min_{\boldsymbol{W}} \|vec(\boldsymbol{X}) - [\boldsymbol{F}_1^h, \ldots, \boldsymbol{F}_M^h] * \begin{bmatrix} [vec(\tilde{\boldsymbol{Z}}_{k,1}^l), \ldots, vec(\tilde{\boldsymbol{Z}}_{k,N}^l)] & & \\ & \cdots & \\ & & [vec(\tilde{\boldsymbol{Z}}_{k,1}^l), \ldots, vec(\tilde{\boldsymbol{Z}}_{k,N}^l)] \end{bmatrix} * vec(\boldsymbol{W})\|_F^2$$
$$s.t. \ \ \boldsymbol{w}_j \succeq 0, |\boldsymbol{w}_j|_1 = 1.$$

Let

$$\boldsymbol{A} = \left\{ \boldsymbol{F}_1^h * [vec(\tilde{\boldsymbol{Z}}_1^l), vec(\tilde{\boldsymbol{Z}}_2^l), \ldots, vec(\tilde{\boldsymbol{Z}}_N^l)], \ldots, \boldsymbol{F}_M^h * [vec(\tilde{\boldsymbol{Z}}_1^l), vec(\tilde{\boldsymbol{Z}}_2^l), \ldots, vec(\tilde{\boldsymbol{Z}}_N^l)] \right\}, \quad (15)$$

and then the mapping function training problem has the form

$$\{\boldsymbol{W}\} = \sum_{k=1}^{K} \arg\min_{\boldsymbol{W}} \|vec(\boldsymbol{X}) - \boldsymbol{A} * vec(\boldsymbol{W})\|_F^2 \quad s.t. \ \ \boldsymbol{w}_j \succeq 0, |\boldsymbol{w}_j|_1 = 1. \quad (16)$$

We solve (16) by the SA-ADMM algorithm

$$\begin{aligned} &vec(\boldsymbol{W})_{t+1} = [Lvec(\bar{\boldsymbol{W}})_t - \rho(\boldsymbol{T}_t - \boldsymbol{S}_t) - \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{A}_k^T(\boldsymbol{A}_k vec(\boldsymbol{W}_{\tau_j(t)}) - \boldsymbol{X}_k)]/(\rho + L) \\ &\boldsymbol{S}_{t+1} = argmin_{\boldsymbol{S}} \frac{\rho}{2}\|\boldsymbol{W}_{t+1} + \boldsymbol{T}_t - \boldsymbol{S}\|^2, \quad s.t. \ \ \boldsymbol{s}_j \succeq 0, \sum \boldsymbol{s}_j = 1 \\ &\boldsymbol{T}_{t+1} = \boldsymbol{T}_t + \boldsymbol{W}_{t+1} - \boldsymbol{S}_{t+1} \end{aligned} \quad (17)$$

Different from the $\ell_2$-norm proximal problem in (11) which has a closed-form solution, the second optimization problem in (17) is a proximal problem with nonnegative simplex constraint. Although it does not has a closed-form solution, we have the following **Remark** to show that each column of $\boldsymbol{S}$ can be solved very efficiently.

**Remark A.2.1.** *Let* $\mathbf{e} = (1, 1, \ldots, 1)^T$*; problem*

$$\min_{\mathbf{a} \in \Re^n} \|\mathbf{a} - \mathbf{b}\|_F^2 \quad s.t. \mathbf{e}^T\mathbf{a} = 1, -\mathbf{a} \leq 0, \quad (18)$$

*has a globally optimal solution*

$$\mathbf{a}^* = \mathbf{b} - \frac{\sum_{i=1}^{r} \mathbf{b}_{\tau_i} - 1}{r} + [\frac{\sum_{i=1}^{r} \mathbf{b}_{\tau_i} - 1}{r}\mathbf{e} - \mathbf{b}]_+, \tag{19}$$

*where* $\{\tau_1, \tau_2, \ldots, \tau_n\}$ *is an index sequence which satisfies* $b_{\tau_1} \geq b_{\tau_2} \geq \cdots b_{\tau_n}$. $r$ *is an integer which satisfies* $b_{\tau_r} > \frac{\sum_{i=1}^{r} \mathbf{b}_{\tau_i} - 1}{r}$ *and* $b_{\tau_{r+1}} \leq \frac{\sum_{i=1}^{r} \mathbf{b}_{\tau_i} - 1}{r}$.

*Proof.* The Lagrange function of (18)

$$\mathfrak{L}(\boldsymbol{a}, \lambda, \boldsymbol{v}) = \|\boldsymbol{a} - \boldsymbol{b}\|_F^2 + \lambda(\boldsymbol{e}^T\boldsymbol{a} - 1) - \boldsymbol{v}^T\boldsymbol{a} \quad s.t. \boldsymbol{v} \geq 0 \tag{20}$$

is a convex function. Let the partial derivative of $\mathfrak{L}$ w.r.t. $\boldsymbol{a}$ equal to zero, we have the optimal solution of $\boldsymbol{a}$:

$$\boldsymbol{a}^* = \boldsymbol{a} - \frac{\lambda}{2}\boldsymbol{e} + \frac{1}{2}\boldsymbol{v}. \tag{21}$$

Substitute (21) into (20), we have the dual function of (18)

$$\begin{aligned} g(\lambda, \boldsymbol{v}) &= \| - \frac{\lambda}{2}\boldsymbol{e} + \frac{1}{2}\boldsymbol{v}\|_F^2 + \lambda(\boldsymbol{e}^T(\boldsymbol{b} - \frac{\lambda}{2}\boldsymbol{e} + \frac{1}{2}\boldsymbol{v}) - 1) - \boldsymbol{e}^T(\boldsymbol{e} - -\frac{\lambda}{2}\boldsymbol{e} + \frac{1}{2}\boldsymbol{v}) \\ &= \frac{n}{4}\lambda^2 + \frac{1}{4}\|\boldsymbol{v}\|_F^2 - \frac{1}{2}\lambda\boldsymbol{e}^T\boldsymbol{v} + \lambda\boldsymbol{e}^T\boldsymbol{b} - \frac{n}{2}\lambda^2 + \frac{1}{2}\lambda\boldsymbol{e}^T\boldsymbol{v} - \lambda - \boldsymbol{v}^T\boldsymbol{b} + \frac{\lambda}{2}\boldsymbol{v}^T\boldsymbol{e} - \frac{1}{2}\|\boldsymbol{v}\|_F^2 \\ &= -\frac{n}{4}\lambda^2 - \frac{1}{4}\|\boldsymbol{v}\|_F^2 + \frac{1}{2}\lambda\boldsymbol{e}^T\boldsymbol{v} + \lambda\boldsymbol{e}^T\boldsymbol{b} - \lambda - \boldsymbol{v}^T\boldsymbol{b} \\ &= -\frac{1}{4}\|\lambda\boldsymbol{e} - \boldsymbol{v}\|_F^2 + \boldsymbol{b}^T(\lambda\boldsymbol{e} - \boldsymbol{v}) - \lambda. \end{aligned} \tag{22}$$

Thus, the dual problem

$$\begin{aligned} max_{\lambda,\boldsymbol{v}} g(\lambda, \boldsymbol{v}) &= max_{\lambda,\boldsymbol{v}} - \frac{1}{4}\|\lambda\boldsymbol{e} - \boldsymbol{v}\|_F^2 + \boldsymbol{b}^T(\lambda\boldsymbol{e} - \boldsymbol{v}) - \lambda \\ &= \boldsymbol{b}^T\boldsymbol{b} + max_{\lambda,\boldsymbol{v}} - \frac{1}{4}\|\lambda\boldsymbol{e} - \boldsymbol{v} - 2\boldsymbol{b}\|_F^2 - \lambda \end{aligned} \tag{23}$$

is a concave function, and the optimal solution can be achieved by letting the partial derivative equal to zero. The optimal solution of $\boldsymbol{v}$ is

$$\boldsymbol{v}^* = [\lambda\boldsymbol{e} - 2\boldsymbol{b}]_+ = max(\lambda\boldsymbol{e} - 2\boldsymbol{b}, 0). \tag{24}$$

We then substitute (24) into (23), and calculate its derivative w.r.t. $\lambda$:

$$\frac{\partial\{-\frac{1}{4}\|\lambda\boldsymbol{e}-2\boldsymbol{b}-[\lambda\boldsymbol{e}-2\boldsymbol{b}]_+\|_F^2-\lambda\}}{\partial\lambda}=-\frac{1}{2}\mathfrak{I}(\lambda\boldsymbol{e}<2\boldsymbol{b})^T(\lambda\boldsymbol{e}-2\boldsymbol{b})-1, \tag{25}$$

in which

$$(\mathfrak{I}(\boldsymbol{x}))_i=\begin{cases}1 & x_i\ is\ true\\ & ,i=1,2,\ldots,n.\\ 0 & x_i\ is\ false\end{cases} \tag{26}$$

Let $-\frac{1}{2}\mathfrak{I}(\lambda\boldsymbol{e}<2\boldsymbol{b})^T(\lambda\boldsymbol{e}-2\boldsymbol{b})-1$ be zero, we have

$$\lambda=\frac{2(\sum_{i=1}^r\boldsymbol{b}_{\tau_i}-1)}{r}, \tag{27}$$

where $\{\tau_1,\tau_2,\ldots,\tau_n\}$ is an index sequence which satisfies $b_{\tau_1}\geq b_{\tau_2}\geq\cdots b_{\tau_n}$, and $r$ is an integer which satisfies $b_{\tau_r}>\frac{\sum_{i=1}^r\boldsymbol{b}_{\tau_i}-1}{r}$ and $b_{\tau_{r+1}}\leq\frac{\sum_{i=1}^r\boldsymbol{b}_{\tau_i}-1}{r}$.

Based on (21), (24) and (27), the optimal solution for problem (18) is

$$\boldsymbol{a}^*=\boldsymbol{b}-\frac{\sum_{i=1}^r\boldsymbol{b}_{\tau_i}-1}{r}+[\frac{\sum_{i=1}^r\boldsymbol{b}_{\tau_i}-1}{r}\boldsymbol{e}-\boldsymbol{b}]_+.$$

□

APPENDIX B

### B.3 Proof of Theorem 1

*Proof.* For any $\boldsymbol{X}, \boldsymbol{Y} \in \Re^{m\times n}(m > n)$ , denote by $\bar{\boldsymbol{U}}\boldsymbol{D}\bar{\boldsymbol{V}}^T$ and $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ the singular value decomposition of matrix $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively, where $\boldsymbol{\Sigma} = \begin{pmatrix} diag(\sigma_1, \sigma_2, ..., \sigma_n) \\ \mathbf{0} \end{pmatrix} \in \Re^{m\times n}$, and $\boldsymbol{D} = \begin{pmatrix} diag(d_1, d_2, ..., d_n) \\ \mathbf{0} \end{pmatrix}$ are the diagonal singular value matrices. Based on the property of Frobenius norm, the following derivations hold:

$$\begin{aligned} &\|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \|\boldsymbol{X}\|_{w,*} \\ =&Tr\left(\boldsymbol{Y}^T\boldsymbol{Y}\right) - 2Tr\left(\boldsymbol{Y}^T\boldsymbol{X}\right) + Tr\left(\boldsymbol{X}^T\boldsymbol{X}\right) + \sum_i^n w_i d_i \\ =&\sum_i^n \sigma_i^2 - 2Tr\left(\boldsymbol{Y}^T\boldsymbol{X}\right) + \sum_i^n d_i^2 + \sum_i^n w_i d_i. \end{aligned}$$

Based on the von Neumann trace inequality in Lemma 1, we know that $Tr\left(\boldsymbol{Y}^T\boldsymbol{X}\right)$ achieves its upper bound $\sum_i^n \sigma_i d_i$ if $\boldsymbol{U} = \bar{\boldsymbol{U}}$ and $\boldsymbol{V} = \bar{\boldsymbol{V}}$. Then, we have

$$\begin{aligned} &\min_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \|\boldsymbol{X}\|_{w,*} \\ \Leftrightarrow &\min_{\boldsymbol{D}} \sum_i^n \sigma_i^2 - 2\sum_i^n \sigma_i d_i + \sum_i^n d_i^2 + \sum_i^n w_i d_i \\ &s.t.\ d_1 \geq d_2 \geq ... \geq d_n \geq 0 \\ \Leftrightarrow &\min_{\boldsymbol{D}} \sum_i (d_i - \sigma_i)^2 + w_i d_i \\ &s.t.\ d_1 \geq d_2 \geq ... \geq d_n \geq 0. \end{aligned}$$

From the above derivation, we can see that the optimal solution of the WNNP problem in (5.5) is

$$\boldsymbol{X}^* = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T,$$

where $\boldsymbol{D}$ is the optimum of the constrained quadratic optimization problem in (5.6).

End of proof. □

### B.4 Proof of Corollary 1

*Proof.* Without considering the constraint, the optimization problem (5.6) degenerates to the following unconstrained formula:

$$\begin{aligned} &\min_{d_i \geq 0} (d_i - \sigma_i)^2 + w_i d_i \\ \Leftrightarrow &\min_{d_i \geq 0} \left( d_i - (\sigma_i - \frac{w_i}{2}) \right)^2 . \end{aligned}$$

It is not difficult to derive its global optimum as:

$$\bar{d_i} = max \left( \sigma_i - \frac{w_i}{2}, 0 \right), \ i = 1, 2, ..., n. \tag{28}$$

Since we have $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$ and the weight vector has a non-descending order $w_1 \leq w_2 \leq ... \leq w_n$, it is easy to see that $\bar{d_1} \geq \bar{d_2} \geq ... \geq \bar{d_n}$. Thus, $\bar{d}_{i=1,2,...,n}$ satisfy the constraint of (5.6), and the solution in (28) is then the globally optimal solution of the original constrained problem in (5.6).

End of proof. □

### B.4.1 Proof of Theorem 2

*Proof.* Denote by $\boldsymbol{U}_k\boldsymbol{\Lambda}_k\boldsymbol{V}_k^T$ the SVD of the matrix $\{\boldsymbol{Y}+\mu_k^{-1}\boldsymbol{L}_k-\boldsymbol{E}_{k+1}\}$ in the $(k+1)$-th iteration, where $\boldsymbol{\Lambda}_k=\{diag(\sigma_k^1,\sigma_k^2,...,\sigma_k^n)\}$ is the diagonal singular value matrix. Based on the conclusion of Corollary 1, we have

$$\boldsymbol{X}_{k+1}=\boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^T, \tag{29}$$

where $\boldsymbol{\Sigma}_k=\mathcal{S}_{\mathbf{w}/\mu_k}(\boldsymbol{\Lambda}_k)$ is the singular value matrix after weighted shrinkage. Based on the Lagrange multiplier updating method in step 5 of Algorithm 5.1, we have

$$\begin{aligned}
\|\boldsymbol{L}_{k+1}\|_F &= \|\boldsymbol{L}_k+\mu_k(\boldsymbol{Y}-\boldsymbol{X}_{k+1}-\boldsymbol{E}_{k+1})\|_F \\
&= \mu_k\|\mu_k^{-1}\boldsymbol{L}_k+\boldsymbol{Y}-\boldsymbol{X}_{k+1}-\boldsymbol{E}_{k+1}\|_F \\
&= \mu_k\|\boldsymbol{U}_k\boldsymbol{\Lambda}_k\boldsymbol{V}_k^T-\boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^T\|_F \\
&= \mu_k\|\boldsymbol{\Lambda}_k-\boldsymbol{\Sigma}_k\|_F \\
&= \mu_k\|\boldsymbol{\Lambda}_k-\mathcal{S}_{\mathbf{w}/\mu_k}(\boldsymbol{\Lambda}_k)\|_F \\
&\leq \mu_k\sqrt{\sum_i\left(\frac{w_i}{\mu_k}\right)^2} \\
&= \sqrt{\sum_i w_i^2}.
\end{aligned} \tag{30}$$

Thus, $\{\boldsymbol{L}_k\}$ is bounded.

To analyze the boundedness of $\Gamma(\boldsymbol{X}^{k+1},\boldsymbol{E}^{k+1},\boldsymbol{L}^k,\mu^k)$, first we can see the following inequality holds because in step 3 and step 4 we have achieved the globally optimal solutions of the $\boldsymbol{X}$ and $\boldsymbol{E}$ subproblems:

$$\Gamma(\boldsymbol{X}_{k+1},\boldsymbol{E}_{k+1},\boldsymbol{L}_k,\mu_k)\leq\Gamma(\boldsymbol{X}_k,\boldsymbol{E}_k,\boldsymbol{L}_k,\mu_k).$$

Then, based on the way we update $\boldsymbol{L}$:

$$\boldsymbol{L}_{k+1}=\boldsymbol{L}_k+\mu_k(\boldsymbol{Y}-\boldsymbol{X}_{k+1}-\boldsymbol{E}_{k+1}),$$

there is

$$\begin{aligned}
&\Gamma(X_k, E_k, L_k, \mu_k) \\
=&\Gamma(X_k, E_k, L_{k-1}, \mu_{k-1}) + \frac{\mu_k - \mu_{k-1}}{2} \|Y - X_k - E_k\|_F^2 \\
&+ \langle L_k - L_{k-1}, Y - X_k - E_k \rangle \\
=&\Gamma(X_k, E_k, L_{k-1}, \mu_{k-1}) + \frac{\mu_k - \mu_{k-1}}{2} \left\|\mu_{k-1}^{-1}(L_k - L_{k-1})\right\|_F^2 \\
&+ \left\langle L_k - L_{k-1}, \mu_{k-1}^{-1}(L_k - L_{k-1}) \right\rangle \\
=&\Gamma(X_k, E_k, L_{k-1}, \mu_{k-1}) + \frac{\mu_k + \mu_{k-1}}{2\mu_{k-1}^2} \|L_k - L_{k-1}\|_F^2 .
\end{aligned}$$

Denote by $\Theta$ the upper bound of $\|\boldsymbol{L}_k - \boldsymbol{L}_{k-1}\|_F^2$ for all $\{k = 1, \ldots, \infty\}$. We have

$$\begin{aligned}
\Gamma(\boldsymbol{X}_{k+1}, \boldsymbol{E}_{k+1}, \boldsymbol{L}_k, \mu_k) \leq &\Gamma(\boldsymbol{X}_1, \boldsymbol{E}_1, \boldsymbol{L}_0, \mu_0) \\
&+ \Theta \sum_{k=1}^{\infty} \frac{\mu_k + \mu_{k-1}}{2\mu_{k-1}^2}.
\end{aligned}$$

Since the penalty parameter $\{\mu_k\}$ satisfies $\sum_{k=1}^{\infty} \mu_k^{-2}\mu_{k+1} < +\infty$, we have

$$\sum_{k=1}^{\infty} \frac{\mu_k + \mu_{k-1}}{2\mu_{k-1}^2} \leq \sum_{k=1}^{\infty} \mu_{k-1}^{-2}\mu_k < +\infty.$$

Thus, we know that $\Gamma(\boldsymbol{X}^{k+1}, \boldsymbol{E}^{k+1}, \boldsymbol{L}^k, \mu^k)$ is also upper bounded.

The boundedness of $\{\boldsymbol{X}^k\}$ and $\{\boldsymbol{E}^k\}$ can be easily deduced as follows:

$$\begin{aligned}
&\|\boldsymbol{E}_k\|_1 + \|\boldsymbol{X}_k\|_{w,*} \\
=&\Gamma(\boldsymbol{X}_k, \boldsymbol{E}_k, \boldsymbol{L}_{k-1}, \mu_{k-1}) + \frac{\mu_{k-1}}{2}\Big(\frac{1}{\mu_{k-1}^2}\|\boldsymbol{L}_{k-1}\|_F^2 \\
&\qquad - \|\boldsymbol{Y} - \boldsymbol{X}_k - \boldsymbol{E}_k + \frac{1}{\mu_{k-1}}\boldsymbol{L}_{k-1}\|_F^2\Big) \\
=&\Gamma(\boldsymbol{X}_k, \boldsymbol{E}_k, \boldsymbol{L}_{k-1}, \mu_{k-1}) - \frac{1}{2\mu_{k-1}}(\|\boldsymbol{L}_k\|_F^2 - \|\boldsymbol{L}_{k-1}\|_F^2).
\end{aligned}$$

Thus, $\{\boldsymbol{X}_k\}$, $\{\boldsymbol{E}_k\}$ and $\{\boldsymbol{L}_k\}$ generated by the proposed algorithm are all bounded.

There exists at least one accumulation point for $\{\boldsymbol{X}_k, \boldsymbol{E}_k, \boldsymbol{L}_k\}$. Specifically, we have

$$\lim_{k\to\infty} \|\boldsymbol{Y} - \boldsymbol{X}_{k+1} - \boldsymbol{E}_{k+1}\|_F = \lim_{k\to\infty} \frac{1}{\mu_k}\|\boldsymbol{L}_{k+1} - \boldsymbol{L}_k\|_F = 0,$$

and the accumulation point is a feasible solution to the objective function.

We then prove that the change of the variables in adjacent iterations tends to be zero. For the $\boldsymbol{E}$ subproblem in step 3, we have

$$\begin{aligned}
&\lim_{k\to\infty} \|\boldsymbol{E}_{k+1} - \boldsymbol{E}_k\|_F \\
=& \lim_{k\to\infty} \|\mathcal{S}_{\frac{1}{\mu_k}}\left(\boldsymbol{Y} + \mu_k^{-1}\boldsymbol{L}_k - \boldsymbol{X}_k\right) - \left(\boldsymbol{Y} + \mu_k^{-1}\boldsymbol{L}_k - \boldsymbol{X}_k\right) \\
&\qquad\qquad - 2\mu_k^{-1}\boldsymbol{L}_k - \mu_{k-1}^{-1}\boldsymbol{L}_{k-1}\|_F \\
\leq& \lim_{k\to\infty} \frac{mn}{\mu_k} + \|2\mu_k^{-1}\boldsymbol{L}_k + \mu_{k-1}^{-1}\boldsymbol{L}_{k-1}\|_F = 0,
\end{aligned}$$

in which $\mathcal{S}_{\frac{1}{\mu_k}}(\cdot)$ is the soft-thresholding operation with parameter $\frac{1}{\mu_k}$, and $m$ and $n$ are the size of matrix $\boldsymbol{Y}$.

To prove $\lim_{k\to\infty} \|\boldsymbol{X}_{k+1} - \boldsymbol{X}_k\|_F = 0$, we recall the updating strategy in Algorithm 1 which makes the following inequalities hold:

$$\boldsymbol{X}_k = \boldsymbol{U}_{k-1}\mathcal{S}_{\mathbf{w}/\mu_{k-1}}(\boldsymbol{\Lambda}_{k-1})\boldsymbol{V}_{k-1}^T,$$

$$\boldsymbol{X}_{k+1} = \boldsymbol{Y} + \mu_k^{-1}\boldsymbol{L}_k - \boldsymbol{E}_{k+1} - \mu_k^{-1}\boldsymbol{L}_{k+1},$$

where $\boldsymbol{U}_{k-1}\boldsymbol{\Lambda}_{k-1}\boldsymbol{V}_{k-1}^T$ is the SVD of the matrix $\{\boldsymbol{Y} + \mu_{k-1}^{-1}\boldsymbol{L}_{k-1} - \boldsymbol{E}_k\}$ in the $k$-th iteration.

We then have

$$
\begin{aligned}
&\lim_{k\to\infty} \|\boldsymbol{X}_{k+1} - \boldsymbol{X}_k\|_F \\
=&\lim_{k\to\infty} \|(\boldsymbol{Y} + \mu_k^{-1}\boldsymbol{L}_k - \boldsymbol{E}_{k+1} - \mu_k^{-1}\boldsymbol{L}_{k+1}) - \boldsymbol{X}_k\|_F \\
=&\lim_{k\to\infty} \|(\boldsymbol{Y} + \mu_k^{-1}\boldsymbol{L}_k - \boldsymbol{E}_{k+1} - \mu_k^{-1}\boldsymbol{L}_{k+1}) - \boldsymbol{X}_k \\
&\qquad + (\boldsymbol{E}_k + \mu_{k-1}^{-1}\boldsymbol{L}_{k-1}) - (\boldsymbol{E}_k + \mu_{k-1}^{-1}\boldsymbol{L}_{k-1})\|_F \\
\leq&\lim_{k\to\infty} \|\boldsymbol{Y}+\mu_{k-1}^{-1}\boldsymbol{L}_{k-1}-\boldsymbol{E}_k-\boldsymbol{X}_k\|_F+\|\boldsymbol{E}_k-\boldsymbol{E}_{k+1}+\mu_k^{-1}\boldsymbol{L}_k \\
&\qquad -\mu_k^{-1}\boldsymbol{L}_{k+1} - \mu_{k-1}^{-1}\boldsymbol{L}_{k-1}\|_F \\
\leq&\lim_{k\to\infty} \|\boldsymbol{\Lambda}_{k-1} - \mathcal{S}_{\mathbf{w}/\mu_{k-1}}(\boldsymbol{\Lambda}_{k-1})\|_F + \|\boldsymbol{E}_k - \boldsymbol{E}_{k+1}\|_F \\
&\qquad + \|\mu_k^{-1}\boldsymbol{L}_k - \mu_k^{-1}\boldsymbol{L}_{k+1} - \mu_{k-1}^{-1}\boldsymbol{L}_{k-1}\|_F \\
=&\ 0.
\end{aligned}
$$

End of proof. □

### B.5 Proof of Remark 1

*Proof.* Based on the conclusion of Theorem 5.2.1, the WNNM problem can be equivalently transformed to a constrained singular value optimization problem. Furthermore, when utilizing the reweighting strategy $w_i^{\ell+1} = \frac{C}{\sigma_i^{\ell}(\boldsymbol{X})+\varepsilon}$, the singular values of $\boldsymbol{X}$ are consistently sorted in a non-ascending order. The weight vector thus follows the non-descending order. It is then easy to deduce that the sorted orders of the sequences $\{\sigma_i(\boldsymbol{Y}), \sigma_i(\boldsymbol{X}_\ell), w_i^{\ell}; i = 1, 2, \cdots, n\}$ keep unchanged during the iteration. Thus, the optimization for each singular value $\sigma_i(\boldsymbol{X})$ can be analyzed independently. For the purpose of simplicity, in the following development we omit the subscript $i$ and denote by $y$ a singular value of matrix $\boldsymbol{Y}$, and denote by $x$ and $w$ the corresponding singular value of $\boldsymbol{X}$ and its weight.

For the weighting strategy $w^\ell = \frac{C}{x^{\ell-1}+\varepsilon}$, we have

$$x^\ell = max\left(y - \frac{C}{x^{\ell-1}+\varepsilon}, 0\right).$$

Since we initialize $x^0$ as the singular value of matrix $\boldsymbol{X}_0 = \boldsymbol{Y}$, and each $x^\ell$ is a result of soft-thresholding operation on positive value $y = \sigma_i(\boldsymbol{Y})$, $\{x^\ell\}$ is a non-negative sequence. The convergence value $\lim_{\ell\to\infty} x^\ell$ for different conditions are analyzed as follows.

(1) $c_2 < 0$

From the definition of $c_1$ and $c_2$, we have $(y+\varepsilon)^2 - 4C < 0$. In such case, the quadratic system $x^2 + (\varepsilon - y)x + C - y\varepsilon = 0$ does not have a real solution and function $f(x) = x^2 + (\varepsilon - y)x + C - y\varepsilon$ gets its positive minimum value $C - y\varepsilon - \frac{(y-\varepsilon)^2}{4}$ at $x = \frac{y-\varepsilon}{2}$.

$\forall \tilde{x} \geq 0$, the following inequalities hold

$$\begin{aligned} f(\tilde{x}) &\geq f\left(\frac{y-\varepsilon}{2}\right) \\ \tilde{x}^2 + (\varepsilon - y)\tilde{x} &\geq -\frac{(y-\varepsilon)^2}{4} \\ \tilde{x} - \frac{C - y\varepsilon - \frac{(y-\varepsilon)^2}{4}}{\tilde{x}+\varepsilon} &\geq y - \frac{C}{\tilde{x}+\varepsilon}. \end{aligned}$$

The sequence $x^{\ell+1} = max\left(y - \frac{C}{x^\ell+\varepsilon}, 0\right)$ with initialization $x^0 = y$ is a monotonically decreasing sequence for any $x^\ell \geq 0$. We have $x^\ell < y$, and

$$x^\ell - \left(y - \frac{C}{x^\ell+\varepsilon}\right) > \frac{C - y\varepsilon - \frac{(y-\varepsilon)^2}{4}}{y+\varepsilon}.$$

If $x^\ell \leq \frac{C-y\varepsilon}{y}$, we have $y - \frac{C}{x^\ell+\varepsilon} \leq 0$ and $x^{\ell+1} = max\left(y - \frac{C}{x^\ell+\varepsilon}, 0\right) = 0$. If $x^\ell > \frac{C-y\varepsilon}{y}$, $\exists N \in \mathbb{N}$ makes $x^{\ell+N} < x^\ell - N \cdot \frac{C-y\varepsilon-\frac{(y-\varepsilon)^2}{4}}{y+\varepsilon}$ less than $\frac{C-y\varepsilon}{y}$. The sequence $\{x^\ell\}$ will shrink to 0 monotonically.

(2) $c_2 \geq 0$

In such case, we can know that $y > 0$, because if $y = 0$, we will have $c_2 = (y+\varepsilon)^2 -$

$4C = \varepsilon^2 - 4C < 0$. For positive $C$ and sufficiently small value $\varepsilon$, we can know that $c_1$ is also non-negative:

$$c_2 = (y+\varepsilon)^2 - 4C \geq 0$$

$$(y+\varepsilon)^2 \geq 4C$$

$$y - \varepsilon \geq 2(\sqrt{C} - \varepsilon)$$

$$c_1 = y - \varepsilon \geq 0.$$

Having $c_2 \geq 0$, $c_1 \geq 0$, we have

$$\bar{x}_2 = \frac{y - \varepsilon + \sqrt{(y-\varepsilon)^2 - 4(C - \varepsilon y)}}{2} > 0.$$

For any $x > \bar{x}_2 > 0$, the following inequalities hold:

$$f(x) = x^2 + (\varepsilon - y)x + C - y\varepsilon > 0$$

$$\left[x - \left(y - \frac{C}{x+\varepsilon}\right)\right](x+\varepsilon) > 0$$

$$x > y - \frac{C}{x+\varepsilon}.$$

Furthermore, we have

$$x > y - \frac{C}{x+\varepsilon} > y - \frac{C}{\bar{x}_2 + \varepsilon} = \bar{x}_2.$$

Thus, for $x^0 = y > \bar{x}_2$, we always have $x^\ell > x^{\ell+1} > \bar{x}_2$, the sequence is monotonically decreasing and has lower bound $\bar{x}_2$. The sequence will converge to $\bar{x}_2$, as we prove below.

We propose a proof by contradiction. If $x^\ell$ converges to $\hat{x} \neq \bar{x}_2$, then we have $\hat{x} > \bar{x}_2$ and $f(\hat{x}) > 0$. By the definition of convergence, we can obtain that $\forall \epsilon > 0$, $\exists N \in \mathbb{N}$ s.t. $\forall \ell \geq N$, the following inequality must be satisfied

$$|x^\ell - \hat{x}| < \epsilon. \tag{31}$$

We can also have the following inequalies

$$\begin{aligned} f(x^N) &\geq f(\hat{x}) \\ \left[x^N - \left(y - \frac{C}{x^N + \varepsilon}\right)\right](x^N + \varepsilon) &\geq f(\hat{x}) \\ \left[x^N - \left(y - \frac{C}{x^N + \varepsilon}\right)\right](y + \varepsilon) &\geq f(\hat{x}) \\ x^N - \left(y - \frac{C}{x^N + \varepsilon}\right) &\geq \frac{f(\hat{x})}{y + \varepsilon} \\ x^N - x^{N+1} &> \frac{f(\hat{x})}{y + \varepsilon} \end{aligned}$$

If we take $\epsilon = \frac{f(\hat{x})}{2(y+\varepsilon)}$, then $x^N - x^{N+1} > 2\epsilon$, and we can thus obtain

$$\begin{aligned} & |x^{N+1} - \hat{x}| \\ =& |x^{N+1} - x^N + x^N - \hat{x}| \\ \geq& \left||x^{N+1} - x^N| - |x^N - \hat{x}|\right| \\ >& |2\epsilon - \epsilon| = \epsilon \end{aligned}$$

This is however a contradiction to (31), and thus $x^\ell$ converges to $\bar{x}_2$.

End of proof. □

Bibliography

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[2] Pablo Arias, Gabriele Facciolo, Vicent Caselles, and Guillermo Sapiro. A variational framework for exemplar-based image inpainting. *International Journal of computer Vision*, 93(3):319–347, 2011.

[3] Jean-François Aujol, Guy Gilboa, Tony Chan, and Stanley Osher. Structure-texture image decomposition‖modeling, algorithms, and parameter selection. *International Journal of Computer Vision*, 67(1):111–136, 2006.

[4] Anthony J Bell and Terrence J Sejnowski. The independent components of natural scenes are edge filters. *Vision research*, 37(23):3327–3338, 1997.

[5] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000.

[6] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[7] Hilton Bristow, Anders Eriksson, and Simon Lucey. Fast convolutional sparse coding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[8] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139, 2008.

[9] Antoni Buades, Triet M Le, Jean-Michel Morel, and Luminita A Vese. Fast cartoon+ texture image filters. *IEEE Transactions on Image Processing*, 19(8):1978–1986, 2010.

[10] Aeron M Buchanan and Andrew W Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[11] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[12] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[13] Emmanuel J Candès and David L Donoho. Ridgelets: A key to higher-dimensional intermittency? *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 357(1760):2495–2509, 1999.

[14] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011.

[15] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[16] Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted $\ell_1$ minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.

[17] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

[18] Derek Chan, Hylke Buisman, Christian Theobalt, and Sebastian Thrun. A noise-aware filter for real-time depth upsampling. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications8*, 2008.

[19] Tony Chan, Antonio Marquina, and Pep Mulet. High-order total variation-based image restoration. *SIAM Journal on Scientific Computing*, 22(2):503–516, 2000.

[20] Tony F Chan and Selim Esedoglu. Aspects of total variation regularized l 1 function approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817–1837, 2005.

[21] Tony F Chan and Jianhong Jackie Shen. *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. SIAM Press, 2005.

[22] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[23] Tao Chen, Kai-Kuang Ma, and Li-Hui Chen. Tri-state median filter for image denoising. *IEEE Transactions on Image processing*, 8(12):1834–1838, 1999.

[24] Yi-Lei Chen and Chiou-Ting Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[25] Yunjin Chen, Rene Ranftl, and Thomas Pock. Insights into analysis operator learning: From patch-based sparse models to higher order mrfs. *IEEE Transactions on Image Processing*, 23(3):1060–1072, 2014.

[26] Yunjin Chen, Wei Yu, and Thomas Pock. On learning optimized reaction diffusion processes for effective image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[27] Ronald R Coifman and David L Donoho. *Translation-invariant de-noising*. Springer, 1995.

[28] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004.

[29] Zhen Cui, Hong Chang, Shiguang Shan, Bineng Zhong, and Xilin Chen. Deep network cascade for image super-resolution. In *European Conference on Computer Vision*. 2014.

[30] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transaction on Image Processing*, 16(8):2080–2095, 2007.

[31] Joachim Dahl, Per Christian Hansen, Søren Holdt Jensen, and Tobias Lindstrøm Jensen. Algorithms and software for total variation image reconstruction via first-order methods. *Numerical Algorithms*, 53(1):67–92, 2010.

[32] Fernando De La Torre and Michael J Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1-3):117–142, 2003.

[33] James Diebel and Sebastian Thrun. An application of markov random fields to range sensing. In *Conference on Neural Information Processing Systems*, 2005.

[34] Minh N Do and Martin Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on image processing*, 14(12):2091–2106, 2005.

[35] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, 2014.

[36] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*. 2014.

[37] W Dong, G Shi, and X Li. Nonlocal image restoration with bilateral variance estimation: a low-rank approach. *IEEE Transaction on Image Processing*, 22(2):700–711, 2013.

[38] Weisheng Dong, Guangming Shi, Xin Li, Yi Ma, and Feng Huang. Compressive sensing via nonlocal low-rank regularization. *IEEE Transaction on Image Processing*, 23(8):3618–3632, 2014.

[39] David L Donoho. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627, 1995.

[40] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[41] Michael Elad, Peyman Milanfar, and Ron Rubinstein. Analysis versus synthesis in signal priors. *Inverse problems*, 23(3):947, 2007.

[42] Anders Eriksson and Anton Van Den Hengel. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the $l_1$ norm. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[43] Raanan Fattal. Image upsampling via imposed edge statistics. In *ACM Transactions on Graphics (TOG)*, volume 26, page 95, 2007.

[44] Maryam Fazel. *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University, 2002.

[45] Maryam Fazel, Haitham Hindi, and Stephen P Boyd. A rank minimization heuristic with application to minimum order system approximation. In *American Control Conf. (ACC)*, 2001.

[46] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *IEEE International Conference on Computer Vision*, 2013.

[47] David J Field. Relations between the statistics of natural images and the response properties of cortical cells. *JOSA A*, 4(12):2379–2394, 1987.

[48] William T Freeman, Egon C Pasztor, and Owen T Carmichael. Learning low-level vision. *International journal of computer vision*, 40(1):25–47, 2000.

[49] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *IEEE International Conference on Computer Vision*, 2009.

[50] Jochen Gorski, Frank Pfeuffer, and Kathrin Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research*, 66(3):373–407, 2007.

[51] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *International Conference on Machine Learning*, 2010.

[52] Shuhang Gu, Nong Sang, and Fan Ma. Fast image super resolution via local regression. In *International Conference on Pattern Recognition*, 2012.

[53] Shuhang Gu, Qi Xie, Deyu Meng, Wangmeng Zuo, Xiangchu Feng, and Lei Zhang. Weighted nuclear norm minimization and its applications to low level vision. *International Journal of Computer Vision*, pages 1–26, 2016.

[54] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[55] Shuhang Gu, Wangmeng Zuo, Qi Xie, Deyu Meng, Xiangchu Feng, and Lei Zhang. Convolutional sparse coding for image super-resolution. In *IEEE International Conference on Computer Vision*, 2015.

[56] Bumsub Ham, Minsu Cho, and Jean Ponce. Robust image filtering using joint static and dynamic guidance. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[57] Simon Hawe, Martin Kleinsteuber, and Klaus Diepold. Analysis operator learning and its application to image reconstruction. *IEEE Transactions on Image Processing*, 22(6):2138–2150, 2013.

[58] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013.

[59] Li He, Hairong Qi, and Russell Zaretzki. Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[60] Heiko Hirschmüller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[61] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *ACM symposium on User interface software and technology*, pages 559–568, 2011.

[62] Hui Ji, Chaoqiang Liu, Zuowei Shen, and Yuhong Xu. Robust video denoising using low rank matrix completion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[63] Shuiwang Ji and Jieping Ye. An accelerated gradient method for trace norm minimization. In *International Conference on Machine Learning*, pages 457–464, 2009.

[64] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4):1742–1755, 2012.

[65] Qifa Ke and Takeo Kanade. Robust $l_1$ norm factorization in the presence of outliers and missing data by alternative convex programming. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[66] Charles Kervrann. Pewa: Patch-based exponentially weighted aggregation for image denoising. In *Conference on Neural Information Processing Systems*, 2014.

[67] Martin Kiechle, Simon Hawe, and Martin Kleinsteuber. A joint intensity and depth co-sparse analysis model for depth map super-resolution. In *IEEE International Conference on Computer Vision*, 2013.

[68] Jin-Hwan Kim, Chul Lee, Jae-Young Sim, and Chang-Su Kim. Single-image deraining using an adaptive nonlocal means filter. In *IEEE International Conference on Image Processing*, 2013.

[69] Kwang In Kim and Younghee Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1127–1133, 2010.

[70] Dilip Krishnan, Terence Tay, and Rob Fergus. Blind deconvolution using a normalized sparsity measure. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[71] Nojun Kwak. Principal component analysis based on l1-norm maximization. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 30(9):1672–1680, 2008.

[72] HyeokHyen Kwon, Yu-Wing Tai, and Stephen Lin. Data-driven depth map refinement via multi-scale sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[73] HyeokHyen Kwon, Yu-Wing Tai, and Stephen Lin. Data-driven depth map refinement via multi-scale sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[74] Edwin H Land and John J McCann. Lightness and retinex theory. *JOSA*, 61(1):1–11, 1971.

[75] Anat Levin, Boaz Nadler, Fredo Durand, and William T Freeman. Patch complexity, finite pixel correlations and optimal denoising. In *European Conference on Computer Vision*. 2012.

[76] Anat Levin and Yair Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1647, 2007.

[77] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding and evaluating blind deconvolution algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[78] Liyuan Li, Weimin Huang, IY-H Gu, and Qi Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transaction on Image Processing*, 13(11):1459–1472, 2004.

[79] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep joint image filtering. In *European Conference on Computer Vision*, 2016.

[80] Yu Li and Michael S Brown. Single image layer separation using relative smoothness. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[81] Zhouchen Lin, Arvind Ganesh, John Wright, Leqin Wu, Minming Chen, and Yi Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. In *Intl. Workshop on Comp. Adv. in Multi-Sensor Adapt. Processing*, 2009.

[82] Zhouchen Lin, Risheng Liu, and Zhixun Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In *Conference on Neural Information Processing Systems*, 2011.

[83] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[84] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, 2010.

[85] Canyi Lu, Changbo Zhu, Chunyan Xu, Shuicheng Yan, and Zhouchen Lin. Generalized singular value thresholding. *arXiv preprint arXiv:1412.2231*, 2014.

[86] Si Lu, Xiaofeng Ren, and Feng Liu. Depth enhancement via low-rank matrix completion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[87] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3397–3405, 2015.

[88] Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, 2012.

[89] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *IEEE International Conference on Computer Vision*, 2009.

[90] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Trans. on Image Processing,*, 17(1):53–69, 2008.

[91] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.

[92] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

[93] mark schmidt. minfunc, 2013. http://mloss.org/software/view/529/.

[94] Deyu Meng and Fernando De La Torre. Robust matrix factorization with unknown noise. In *IEEE International Conference on Computer Vision*, 2013.

[95] Yves Meyer. *Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures*, volume 22. American Mathematical Soc., 2001.

[96] Peyman Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Processing Magazine*, 30(1):106–128, 2013.

[97] Leon Mirsky. A trace inequality of john von neumann. *Monatshefte für Mathematik*, 79(4):303–306, 1975.

[98] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Conference on Neural Information Processing Systems*, 2007.

[99] Karthik Mohan and Maryam Fazel. Iterative reweighted algorithms for matrix rank minimization. *The Journal of Machine Learning Research*, 13(1):3441–3473, 2012.

[100] Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

[101] Yadong Mu, Jian Dong, Xiaotong Yuan, and Shuicheng Yan. Accelerated low-rank visual recovery by random projection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[102] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, 2012.

[103] Feiping Nie, Heng Huang, and Chris HQ Ding. Low-rank matrix recovery via efficient schatten p-norm minimization. In *AAAI*, 2012.

[104] Tae-Hyun Oh, Hyeongwoo Kim, Yu-Wing Tai, Jean-Charles Bazin, and In So Kweon. Partial sum minimization of singular values in rpca for low-level vision. In *IEEE International Conference on Computer Vision*, 2013.

[105] Neal Parikh, Stephen P Boyd, et al. Proximal algorithms.

[106] Jaesik Park, Hyeongwoo Kim, Yu-Wing Tai, Michael S Brown, and Inso Kweon. High quality depth map upsampling for 3d-tof cameras. In *IEEE International Conference on Computer Vision*, 2011.

[107] Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Conference on Signals, Systems and Computers*, 1993.

[108] T Peleg and M Elad. A statistical prediction model based on sparse representations for single image super-resolution. *IEEE trans. on image processing*, 23(6):2569–2582, 2014.

[109] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990.

[110] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. In *ACM transactions on graphics*, volume 23, pages 664–672, 2004.

[111] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[112] Ron Rubinstein, Alfred M Bruckstein, and Michael Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

[113] Ron Rubinstein, Tomer Peleg, and Michael Elad. Analysis k-svd: a dictionary-learning algorithm for the analysis sparse model. *IEEE Transactions on Signal Processing*, 61(3):661–677, 2013.

[114] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[115] Salakhutdinov Ruslan and Nathan Srebro. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *Conference on Neural Information Processing Systems*, 2010.

[116] Uwe Schmidt and Stefan Roth. Shrinkage fields for effective image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[117] Yiyuan She. An iterative algorithm for fitting nonconvex penalized generalized linear models with grouped predictors. *Computational Statistics & Data Analysis*, 56(10):2976–2990, 2012.

[118] Xiaoyong Shen, Chao Zhou, Li Xu, and Jiaya Jia. Mutual-structure for joint filtering. In *IEEE International Conference on Computer Vision*, 2015.

[119] Nathan Srebro, Tommi Jaakkola, et al. Weighted low-rank approximations. In *International Conference on Machine Learning*, 2003.

[120] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[121] Yu-Wing Tai, Shuaicheng Liu, Michael S Brown, and Stephen Lin. Super resolution using edge prior and single image detail synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[122] Marshall F Tappen, Ce Liu, Edward H Adelson, and William T Freeman. Learning gaussian conditional random fields for low-level vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[123] Radu Timofte, Vincent De, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *IEEE International Conference on Computer Vision*, 2013.

[124] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[125] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *IEEE International Conference on Computer Vision*, 1998.

[126] Ivana Tosic and Sarah Drewes. Learning joint intensity-depth sparse representations. *IEEE Transactions on Image Processing*, 23(5):2122–2132, 2014.

[127] Javier S Turek, Irad Yavneh, and Michael Elad. On mmse and map denoising under sparse representation modeling over a unitary dictionary. *IEEE Transactions on Signal Processing*, 59(8):3526–3535, 2011.

[128] Naiyan Wang and Dit-Yan Yeung. Bayesian robust matrix factorization for image and video processing. In *IEEE International Conference on Computer Vision*, 2013.

[129] Shenlong Wang, D Zhang, Yan Liang, and Quan Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[130] Shenlong Wang, Lei Zhang, and Yan Liang. Nonlocal spectral prior model for low-level vision. In *Asian Conference on Computer Vision*, 2012.

[131] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[132] Zhou Wang and David Zhang. Progressive switching median filter for the removal of impulse noise from highly corrupted images. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 46(1):78–80, 1999.

[133] Joachim Weickert. *Anisotropic diffusion in image processing*, volume 1. 1998.

[134] Brendt Wohlberg. Efficient convolutional sparse coding. In *ICASSP*, 2014.

[135] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[136] Yuan Xie, Shuhang Gu, Yan Liu, Wangmeng Zuo, Wensheng Zhang, and Lei Zhang. Weighted schatten $p$-norm minimization for image denoising and background subtraction. *arXiv preprint arXiv:1512.01003*, 2015.

[137] Li Xu, Qiong Yan, Yang Xia, and Jiaya Jia. Structure extraction from texture via relative total variation. *ACM Transactions on Graphics (TOG)*, 31(6):139, 2012.

[138] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *European Conference on Computer Vision*. 2014.

[139] Chih-Yuan Yang and Ming-Hsuan Yang. Fast direct super-resolution by simple functions. In *IEEE International Conference on Computer Vision*, 2013.

[140] Jianchao Yang, Zhe Lin, and Scott Cohen. Fast image super-resolution based on in-place example regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[141] Jianchao Yang, Zhaowen Wang, Zhe Lin, Xianbiao Shu, and Thomas Huang. Bilevel sparse coding for coupled feature spaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[142] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[143] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[144] Qingxiong Yang, Ruigang Yang, James Davis, and David Nistér. Spatial-depth super resolution for range images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[145] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Robert Fergus. Deconvolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[146] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *IEEE International Conference on Computer Vision*, 2011.

[147] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730. 2012.

[148] Debing Zhang, Yao Hu, Jieping Ye, Xuelong Li, and Xiaofei He. Matrix completion by truncated nuclear norm regularization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[149] Zhengdong Zhang, Arvind Ganesh, Xiao Liang, and Yi Ma. Tilt: transform invariant low-rank textures. *International Journal of Computer Vision*, 99(1):1–24, 2012.

[150] Qian Zhao, Deyu Meng, Zongben Xu, Wangmeng Zuo, and Lei Zhang. Robust principal component analysis with complex noise. In *International Conference on Machine Learning*, 2014.

[151] Yinqiang Zheng, Guangcan Liu, Shigeki Sugimoto, Shuicheng Yan, and Masatoshi Okutomi. Practical low-rank matrix approximation under robust $l_1$ norm. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[152] Leon Wenliang Zhong and James T Kwok. Fast stochastic alternating direction method of multipliers. In *International Conference on Machine Learning*, 2013.

[153] Mingyuan Zhou, Haojun Chen, Lu Ren, Guillermo Sapiro, Lawrence Carin, and John W Paisley. Non-parametric bayesian dictionary learning for sparse image representations. In *Conference on Neural Information Processing Systems*, 2009.

[154] Xiaowei Zhou, Can Yang, Hongyu Zhao, and Weichuan Yu. Low-rank modeling and its applications in image analysis. *arXiv preprint arXiv:1401.3409*, 2014.

[155] Jiejie Zhu, Liang Wang, Ruigang Yang, James E Davis, and Zhigeng Pan. Reliability fusion of time-of-flight depth and stereo geometry for high quality depth maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1400–1414, 2011.

[156] Song Chun Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.

[157] Yu Zhu, Yanning Zhang, and Alan L Yuille. Single image super-resolution using deformable patches. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[158] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *IEEE International Conference on Computer Vision*, 2011.

[159] Wangmeng Zuo, Dongwei Ren, Shuhang Gu, Liang Lin, and Lei Zhang. Discriminative learning of iteration-wise priors for blind deconvolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.