NORTHWESTERN UNIVERSITY

Solution of Inverse Problem using Learning

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Electrical Engineering

By

Qiqin Dai

EVANSTON, ILLINOIS

September 2017

ABSTRACT

Solution of Inverse Problem using Learning

Qiqin Dai

In this dissertation, we start with the dictionary learning (DL) based single-frame super-resolution (SR) problem, where low resolution (LR) input frames are super-resolved to high resolution (HR) output frames. We propose to extend the previous single-frame SR methods to multiple-frames, i.e., estimating single HR output frame by multiple LR input frames, utilizing DL and motion estimation. Specifically, we adopt the use of bilevel dictionary learning which has been used for single-frame SR. It is extended to multiple frames by using motion estimation with sub-pixel accuracy. By simultaneously solving for a batch of patches from multiple frames, the proposed multiple-frame SR algorithms improve over single-frame SR. We then propose to unfold the iteration process in the LASSO solver to a feed-forward neural network and utilize KKT condition to refine the solution. The X-Ray fluorescence (XRF) image SR method is then investigated to address the trade-off between the spatial resolution of an XRF scan and the Signal-to-Noise Ratio (SNR) of each pixel's spectra. We propose to fuse an LR XRF image and a conventional HR RGB image into a product of HR XRF image. By learning the mapping from RGB signal to XRF signal, the LR XRF image is super-resolved to have the same spatial resolution as the HR RGB image. Finally, the XRF

image inpainting problem with adaptive sampling mask is investigated. A Convolutional Neural Network (CNN) is trained to generate adaptive binary sampling mask according to the RGB image. Then the XRF scanner scans a subset of the whole pixels according to the binary sampling mask, to speedup the scanning process. The sub-sampled XRF image is fused with the RGB image to reconstruct the full-sampled XRF image.

Table of Contents

CHAPTER 1

Introduction

The inverse problem is defined by a mapping between objects of interest, which is called parameters, and acquired information about these objects, which is called data or measurement. The mapping, or forward problem, is called the measurement operator (MO), denoted by M. The MO maps the parameter x to the measurement y, by

$$y = M(x). \tag{1.1}$$

Solving the inverse problem amounts to finding the parameters x based on the measurements y such that Equation 1.1 holds. Due to the lack of sufficient information in the measurements, solutions to inverse problems are usually non-unique and thus difficult to estimate. Priori knowledge of the parameters is usually needed to tackle the inherent ambiguity of inverse problems solutions.

In this dissertation, we formulate solutions to multiple-frame video super-resolution (SR), fast sparse coding inference, X-Ray Fluorescence (XRF) image SR, XRF image inpainting and adaptive sampling mask design using learning techniques to exploit the priori knowledge, instead of using human defined priori. For multiple-frame video SR, dictionary learning technique is applied to learn the non-linear mapping from low-resolution (LR) image patches to high-resolution (HR) image patches from large set of LR / HR training image patches. For XRF image SR, because there is not enough training data to find the mapping from LR XRF images to HR XRF images, instead, we utilize an HR conventional RGB image

as a SR guidance and learn the non-linear mapping from RGB spectrum to XRF spectrum. For XRF image inpainting, again there is not enough training data to learn the mapping from sub-sampled XRF images to full-sampled XRF images, so a conventional RGB image is utilized as an inpainting guidance and the non-linear mapping from RGB spectrum to XRF spectrum is learned. For the adaptive sampling mask generation, we propose to train the mask generation CNN along with the inpainting deep neural network, exploiting the adaptive sampling mask strategy through a pure data driven process.

1.1. Multiple-Frame Video Super-Resolution

Video super-resolution, namely estimating the high-resolution (HR) frames from low-resolution (LR) input sequences, has become one of the fundamental problems in image and video processing and has been extensively studied for decades. With the popularity of high-definition display devices, such as High-definition television (HDTV), or even Ultra-high-definition television (UHDTV), on the market, there is an avid demand for transferring LR videos into HR videos so that they are displayed on high resolution TV screens.

Figure 1.1 shows the degradation model relating the HR sequence to the LR sequence which is the input to the SR algorithms. The HR frames I_k^h are of size $LN_1 \times LN_2$ and the degraded LR frames $\tilde{I}_k^l$ are of size $N_1 \times N2$, where L represents the down-sampling factor. The original multiple HR frames are related through warping based on the motion fields. The HR frames are smoothed with a blur kernel, down-sampled and contaminated by additive Gaussian noise to generate the observed LR frames. The degradation model of the k^{th} frame is therefore given by

$$\tilde{I}_k^l = DBI_k^h + \epsilon_k, \tag{1.2}$$

Figure 1.1. Degradation and SR model. The original HR video frames are related to each other by motion fields. The HR sequence is then degraded to generate the observed sequence according to Equation (1.2). Our proposed dictionary based video SR algorithm estimates the HR sequence, as well as the motion field.

where I_k^h and $\tilde{I}_k^l$ are the HR and LR frames, respectively, written in lexicographical notation as vectors, B represents the blur matrix, D is the down-sampling matrix and ϵ_k represents the Gaussian noise vector. Although Equation (1.2) provides the relationship between the k^{th} HR and LR frames, we can find the relationship between any two frames $\tilde{I}_i^l$ and I_k^h via the motion model. In that sense, Equation (1.2) can be extended to

$$\tilde{I}_k^l = DBC(d_{i,k})I_i^h + \epsilon_{i,k}, \tag{1.3}$$

where $C(d_{i,k})$ is the warping matrix generated by the motion vectors $d_{i,k}$, mapping frame i into frame k, and $\epsilon_{i,k}$ captures both the mis-registration error and the Gaussian noise. For $i = k$, Equation (1.3) turns into Equation (1.2), since $C(d_{i,k})$ becomes the identity operator. Equation (1.3) provides the additional observations for the LR frame $\tilde{I}_k^l$, for various values of $i \neq k$. The objective of the multiple frame SR algorithm is to operate on the observed multiple LR frames $\tilde{I}_k^l$ provided by Equation (1.3) for various values of i and obtain an estimate of the HR frame I_k^h.

1.2. Learning Fast Approximations of Sparse Coding

Sparse coding (SC) is the problem of representing input signal as a linear combination of a small set of basis signals [**40**], where the weights associated with the basis signals are called sparse coefficients. Proven to be both robust to noise and useful in extracting high level features, SC has gained popularity over the last decade and benefits a wide range of signal processing applications, such as classification [**114**], clustering [**24**], compression [**20**], super-resolution [**117**] and denoising [**41**].

We are particularly interested in the ℓ_1-based sparse approximation problem, which is also called the LASSO problem [**103**]. The inference problem of sparse coding is, for a given input vector $x \in \mathbb{R}^n$, to find the optimal sparse code vector $z \in \mathbb{R}^m$ that minimizes an energy function that combines the squared ℓ_2 norm of the reconstruction error and an ℓ_1 norm sparsity penalty on the code, that is,

$$z^* = \arg\min_z \|x - Dz\|_2^2 + \lambda \|z\|_1 , \tag{1.4}$$

where D is $n \times m$ dictionary matrix whose columns are the basis vectors and λ is a coefficient controlling the sparsity penalty.

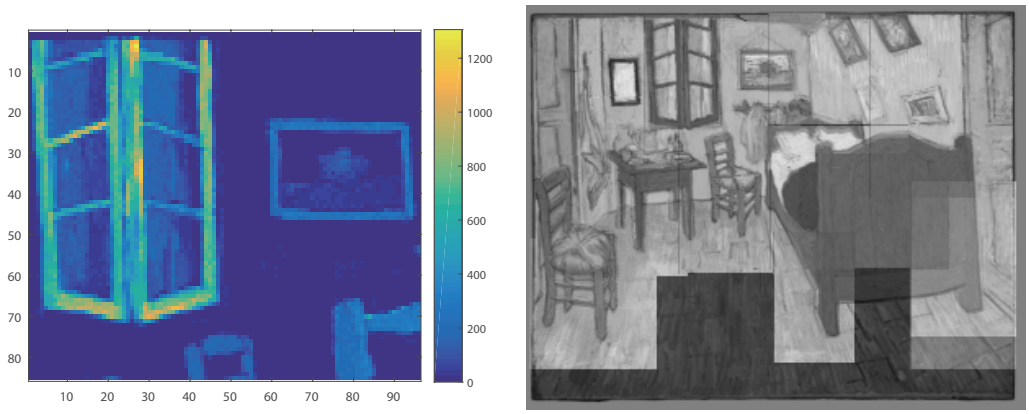
A major problem with ℓ_1-based sparse coding is that the inference algorithm is usually computational expensive, making it impractical for real-time applications. For example, for image compression, given an input image, the inference algorithm needs to compute the sparse coefficients for every patch in the image. Therefore, numerous efforts have been devoted to seeking efficient sparse coding solvers [**11**,**31**,**49**,**68**,**69**,**79**,**107**,**115**]. However, the optimization is still carried out iteratively with these algorithms, therefore the computation is still considerable.

1.3. XRF Image Super-Resolution

Over the last few years, X-Ray fluorescence (XRF) laboratory-based systems have evolved to lightweight and portable instruments thanks to technological advancements in both X-Ray generation and detection. Spatially resolved elemental information can be provided by scanning the surface of the sample with a focused or collimated X-ray beam of (sub) millimeter dimensions and analyzing the emitted fluorescence radiation, in a nondestructive in-situ fashion entitled Macro X-Ray Fluoresence (MA-XRF). The new generations of XRF spectrometers are used in the Cultural Heritage field to study the technology of manufacture, provenance, authenticity, etc, of works of art. Because of their fast non-invasive set up, we are able to study of large, fragile and location inaccessible art objects and archaeological collections. In particular, XRF has been extensively used to investigate historical paintings, by capturing the elemental distribution images of their complex layered structure. This method reveals the painting history from the artist creation to restoration processes [**4**, **7**].

As with other imaging techniques, high spatial resolution and high Signal-to-Noise Ratio (SNR) are desirable for XRF scanning systems. However, the acquisition time is usually limited resulting in a compromise between dwell time, spatial resolution, and desired image quality. In the case of scanning large scale mappings, a choice may be made to reduce the dwell time and increase the step size, resulting in low SNR XRF spectra and low spatial resolution XRF images.

An example of an XRF scan is shown in Figure 1.4 (a). Channel 636 corresponding to Cr Ka elemental X-Ray lines was extracted from a scan of Vincent Van Gogh's "Bedroom" painted in 1889 (housed at The Art Institute of Chicago, acc # 1926.417). The image is color coded for better visibility. This is an image out of 4096 channels that were simultaneously



(a) (b)

Figure 1.2. (a) XRF map showing the distribution of Cr Ka on a section of the "Bedroom", by Vincent Van Gogh, The Art Institute of Chicago, and (b) the automatic registration of 10 maps layered on top of the original resolution RGB image.

acquired by a Bruker M6 scanning energy dispersive XRF instrument. The image has a low resolution (LR) of 96×85 pixels, yet still took $1 - 2$ hour to acquire it. Given the fact that the paining has dimensions 73.6×92.3 cm, at least 10 such patches are needed to capture the whole painting. Much higher resolution would be desirable for didactic purposes to show curators, conservators, and the general public. This makes the acquisition process highly impractical and therefore impedes the use of XRF scanning instruments as high resolution widefield imaging devices. In Figure 1.4 (b) we also show an automatic registration of all 10 averaged XRF maps (across all channels) layered on top of the original RGB image.

1.4. XRF Image Inpainting

As illustrated in Section 1.3, XRF imaging techniques are popular these days and as with other imaging techniques, high spatial resolution and hight quality spectra is desirable for XRF scanning systems. However, the acquisition time is usually limited resulting in a compromise between dwell time, spatial resolution, and desired image quality. In the case of

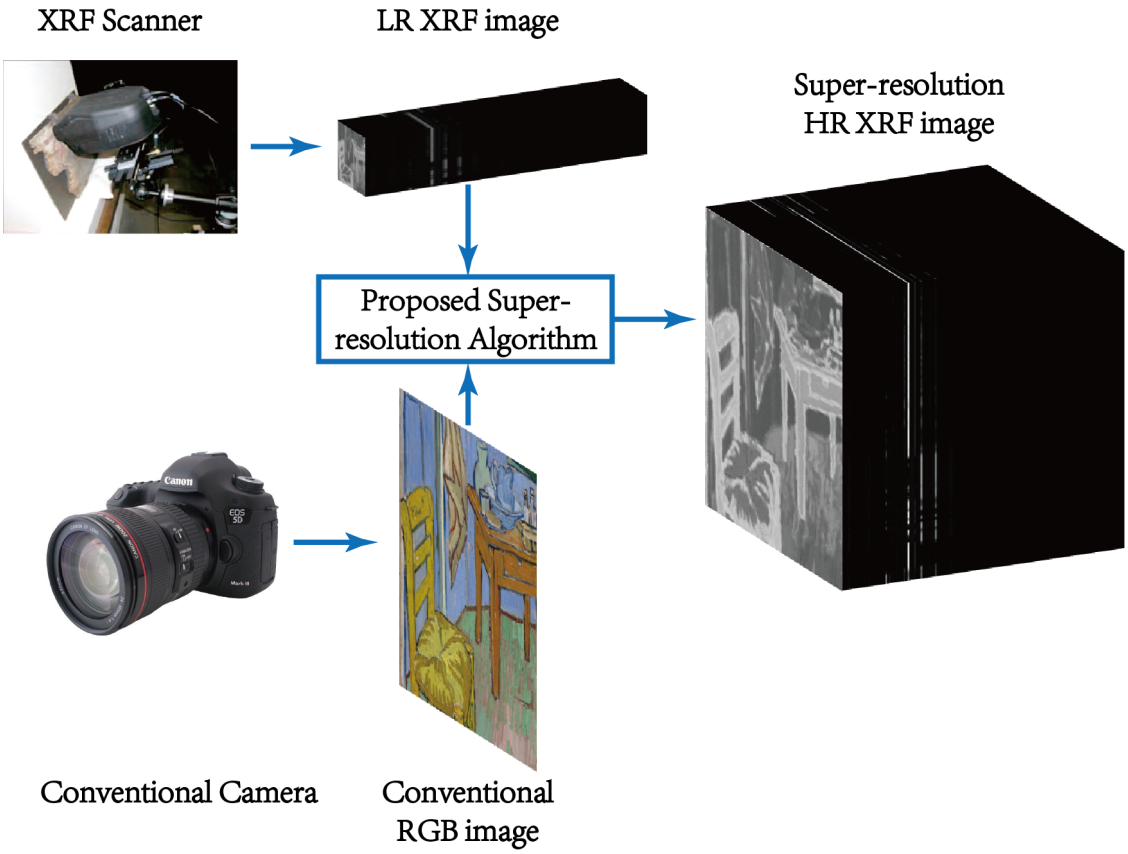
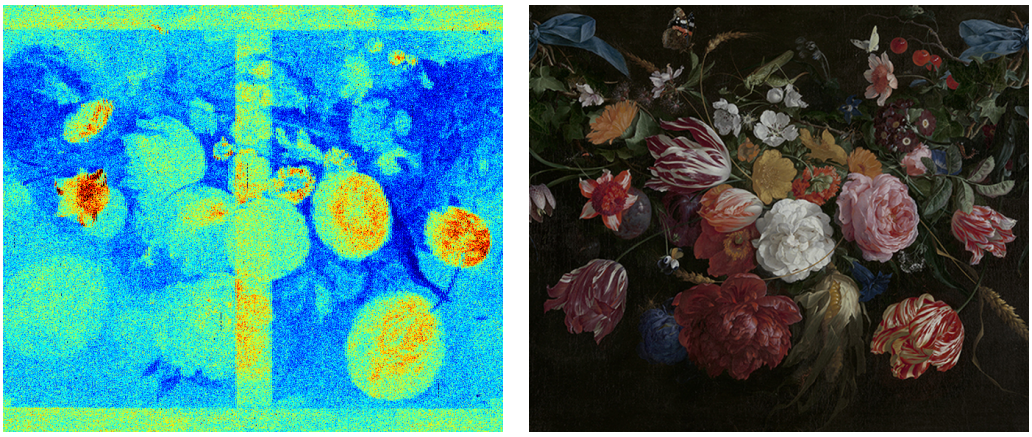


Figure 1.3. XRF images have high spectral resolution but low spatial resolution, whereas the opposite is true for conventional RGB images. The LR XRF image and the HR RGB image are fused to obtain an HR XRF image.

scanning large scale mappings, a choice may be made to reduce the dwell time and increase the step size, resulting in noisy XRF spectra and low spatial resolution XRF images.

An example of an XRF scan is shown in Figure 1.4 (a). Channel #582−602 corresponding to *Pb Lη* XRF emission line was extracted from a scan of Jan Davidsz. de Heem's "Bloemen en insecten" painted in 1645 (housed at Koninklijk Museum voor Schone Kunsten (KMKSA) Antwerp). The image is color coded for better visibility. This XRF image was collected by a home-built XRF spectrometer (courtesy of Prof. Koen Janssens), with 2048 channels in spectrum, and spatial resolution 680×580 pixels. This scan has a relative short dwell time,

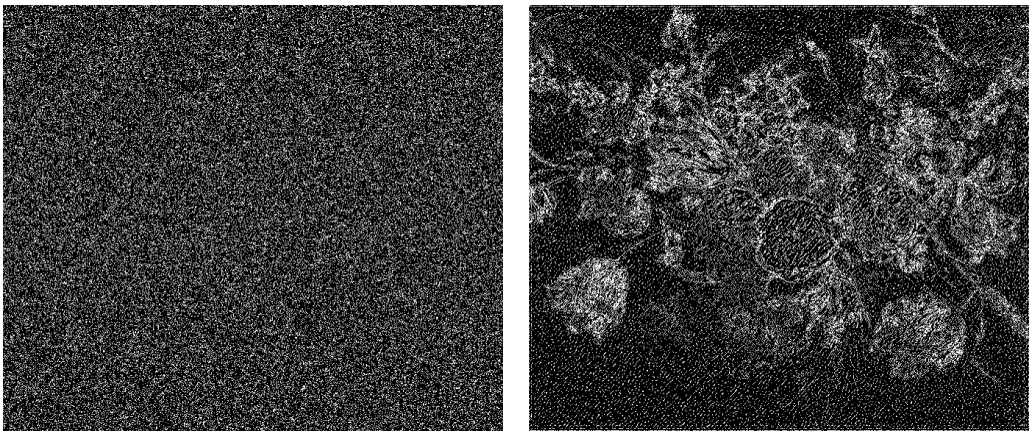


(a) (b)

Figure 1.4. (a) XRF map showing the distribution of *Pb Lη* XRF emission line (channel #582 - 602) of the "Bloemen en insecten" (ca 1645), by Jan Davidsz. de Heem, in the collection of Koninklijk Museum voor Schone Kunsten (KMKSA) Antwerp and (b) the HR RGB image.

resulting in low Signal-to-Noise Ratio (SNR), yet it still took 18 hours to acquire it. Many other XRF scanners with longer dwell time or slower scanning speed will need a longer acquisition time. Faster scanning speed will be desirable for promoting the popularity of the XRF scanning technique, since the slow acquisition process impedes the use of XRF scanning instruments as high resolution widefield imaging devices. The RGB image of the painting of resolution 680×580 pixels is shown in Figure 1.4 (b).

Image inpainting [**13**, **14**, **27**] is the process of recovering missing pixels in images. The XRF images are acquired through a raster scan process. We could therefore speed the scanning process up by skipping pixels and then utilizing an image inpainting technique to reconstruct the missing pixels. If we are to skip 80% of the pixels during acquisition (a 5x speedup), we could use a random sampling mask (shown in Figure 1.5 (a)) or we could design one utilizing the available RGB image (shown in Figure 1.5 (b)). The idea of the adaptive binary sampling mask is based on the assumption that the XRF image is highly correlated



(a) (b)

Figure 1.5. (a) Random binary sampling mask that skips 80% pixels and (b) Adaptive binary sampling mask that skips 80% pixels based on the input RGB images in Fig 1.4 (b).

with the RGB image. We would like to allocate more pixels to the informative parts of the image, such as high frequency textures, sharp edges and high contrast details, and spend fewer pixels for the non-informative parts of the image.

1.5. Outline of the Dissertation

In this dissertation we focus on solving the multiple-frame video SR problem, the fast sparse coding inference problem, the XRF image SR problem and the XRF Image Inpainting problem. We will make extensive use of the learning techniques throughout this Dissertation. The rest of this Dissertation is outlined as follows:

- In Chapter 2, we provide related work on multiple-frame video SR, fast sparse coding inference, XRF image SR and XRF image inpainting.
- In Chapter 3, we propose two multiple-frame super-resolution (SR) algorithms based on dictionary learning and motion estimation. First, we adopt the use of video bilevel dictionary learning which has been used for single-frame SR. It is extended

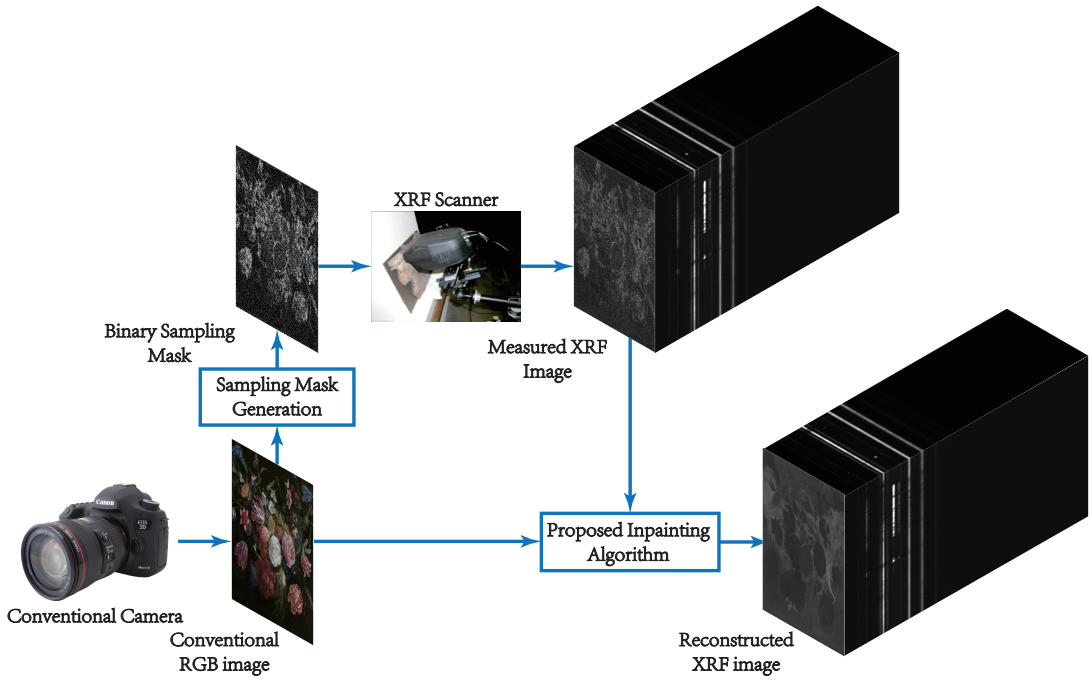


Figure 1.6. The proposed pipeline for the XRF image inpainting utilizing adaptive sampling mask. The binary adaptive sampling mask is generated based on the RGB image of the scan target. Then the XRF scanner sampled the target objected based on the binary sampling mask. Finally, the sub-sampled XRF image and the RGB image are fused to reconstruct the full-sampled XRF image.

to multiple frames by using motion estimation with sub-pixel accuracy. We propose a batch and a temporally recursive multi-frame SR algorithm, which improve over single frame SR. Finally, we propose a novel dictionary learning algorithm utilizing consecutive video frames, rather than still images or individual video frames, which further improves the performance of the video SR algorithms. Extensive experimental comparisons with state-of-the-art SR algorithms verify the effectiveness of our proposed multiple-frame video SR approach.

- In Chapter 4, we propose a KKT condition refined deep ℓ_1 encoder framework. We first adopt the use of neural networks as in previous works of fast approximations of sparse coding. The support of the sparse coefficients estimated by the neural networks is then utilized to retrieve the support of the final sparse coefficients. Finally the KKT condition is applied to obtain an accurate solution to the original ℓ_1-based sparse approximation problem. The additional support retrieval and KKT condition refinement are implemented in an efficient way. Extensive experimental comparisons with the previous fast neural networks approach verify the effectiveness of our proposed KKT condition refined ℓ_1 encoder framework.
- In Chapter 5, we propose an XRF image super-resolution method to address this trade-off, thus obtaining a high spatial resolution XRF scan with high SNR. We fuse a low resolution XRF image and a conventional RGB high-resolution image into a product of both high spatial and high spectral resolution XRF image. There is no guarantee of a one to one mapping between XRF spectrum and RGB color since, for instance, paintings with hidden layers cannot be detected in visible but can in X-ray wavelengths. We separate the XRF image into the visible and non-visible components. The spatial resolution of the visible component is increased utilizing the high-resolution RGB image while the spatial resolution of the non-visible component is increased using a total variation super-resolution method. Finally, the visible and non-visible components are combined to obtain the final result.
- In Chapter 6, we propose an XRF image inpainting approach to address the issue of long scanning time, thus speeding up the scanning process while still maintaining the possibility to reconstruct a high quality XRF image. The RGB image of the scanning target is utilized to generate the adaptive sampling mask. The XRF scanner is

then driven according to the adaptive sampling mask to scan a subset of the total image pixels. Finally we inpaint the scanned XRF image by fusing the RGB image to reconstruct the full scan XRF image. There is no guarantee of an one to one mapping between XRF spectrum and RGB color image since, for instance, paintings with hidden layers cannot be detected in the visible RGB image of the painting but can in XRF wavelengths. We separate the XRF image into its visible and non-visible components. The reconstruction of the visible component is achieved utilizing the RGB image while the reconstruction of the non-visible component is achieved using a total variation inpainting method. Finally, the visible and non-visible components are combined to obtain the final result.

- Finally we draw conclusion remarks in Chapter 7.

CHAPTER 2

Related Work

In this chapter we provide related literature on multiple-frame video SR, fast sparse coding inference, XRF image SR and XRF image inpainting.

2.1. Multiple-Frame Video Super-Resolution

SR techniques have been extensively studied in the literature. Detailed literature reviews of this topic can be found in [**18**, **60**, **83**, **85**]. With one class of approaches multiple observations are used in increasing the resolution of one image, as described in Equation (1.3). Such multiple observations can be due to global sub-pixel motion between the camera and the scene or due to the dynamics of the scene, i.e., the sub-pixel motion of individual objects in the scene. In the former case either multiple still cameras or one still camera which changes its position are used. The motion vectors $d_{i,k}$ in this case are constant for the whole frame but they typically represent more complicated motions than simple translation, such as rotation (e.g., [**105**]). In the latter case, one video camera is typically used, which might move as well resulting in global shifts amongst frames, but the additional information about the frame to be super-resolved is provided by the motion of objects, as is depicted in the neighboring frames. This is the case of video SR considered in this paper, in which case the motion vectors $d_{i,k}$ in Equation (1.3) depend on the pixel location. In designing video SR algorithms, the degradation matrix B is either considered known or is estimated from the data, along with the motion vectors $d_{i,k}$, the original HR frames, and the noise level,

either simultaneously [**12**, **73**, **92**, **93**], or sequentially [**94**, **102**]. Recently, Hung et al. [**53**] proposed a method based on codebooks derived from key-frames and achieved good SR performance on compressed videos. Zhou et al. [**126**] proposed to retrieve high-frequency details from complementary multi-frames by non-uniform interpolation, depending on registered LR frames with sub-pixel accuracy. They further improved the SR performance in [**125**] when the number of LR inputs is small by taking advantage of nonlocal self-similarity to fit local surfaces. Liu et al. [**73**, **74**] proposed to estimate the blur kernel, noise level, motion field and HR frames jointly by Maximum-a-Posteriori (MAP) inference. Ma et al. [**77**] presented an algorithm that extended the same idea to handle motion blur. Liao et al. [**71**] proposed to apply a traditional multi-frame SR method [**42**] to obtain SR drafts with different motion estimation parameters, and then to combine the SR drafts through a deep convolutional neural network (CNN).

Another class of SR approaches is represented by single frame SR, where a single observation is used to increase the resolution of one frame. Due to the limited LR information, example-based or learning approaches, such as dictionary learning (DL) approaches [**97**, **108**, **117**–**120**], showed recently promising single frame SR performance. These methods learn the non-linear mapping from an LR frame to the corresponding HR frame through an HR/LR training data set in the training phase and apply the learned non-linear mapping to an LR observation in the testing phase. DL approaches have also been utilized for deblurring [**82**] and denoising of images and image sequences [**41**, **88**]. For the SR of a still image using dictionary techniques, typically only one observation of an LR image is utilized. The mapping from an LR to an HR image, as depicted by Equation (1.2) is learned during training and is captured by the structures of two coupled, LR and HR, dictionaries. No explicit use of the degradation matrix B is made during the sparse coding based reconstruction of the HR

frame. Some methods [**117**–**120**] might include a back-projection step, thus using matrix B, as a final refinement step. However, based on our knowledge, the first work reported in the literature on the application of DL to video SR is the work in [**97**]. According to it, block-based motion estimation is performed among input LR keyframes and DL is only applied for single-frame SR when the motion compensation error is larger than a threshold. The approach reported in [**97**] however does not provide sub-pixel precision in motion estimation and does not utilize any of the advanced DL techniques. Later the work in [**70**] utilized the semi-coupled DL technique [**108**] to super-resolve each LR frame individually and performed a weighted fusion of the super-resolved HR frames by nonlocal similarity match [**46**]. However, the nonlocal similarity match is also block-based and do not fully exploit the sub-pixel shift information. Also the initial HR frames estimation by the semi-coupled DL and patch similarity match are performed sequentially, so the reconstruction error by the semi-coupled DL SR will not be minimized in the later SR steps.

2.2. Fast Sparse Coding Inference

With the development of deep learning, neural networks have been applied to approximate the solution of sparse coefficients [**48**, **51**, **61**, **109**]. In [**48**], the original Iterative Shrinkage and Thresholding Algorithm (ISTA) [**11**, **31**] is unfolded to a feed-forward neural network, called LISTA (Learned ISTA). In the training phase, the LISTA network parameters are optimized to produce the closest sparse coefficients to the original ISTA algorithm. In the testing phase, fast inference is obtained by this feed-forward neural network.

However, good reconstruction accuracy is not guaranteed by LISTA [**48**], because the sparse coefficients estimated by LISTA are not the optimal solutions and usually do not satisfy the KKT condition [**66**] of the original ℓ_1-based sparse approximation problem. So

for applications such as image compression [**20**], where the reconstruction accuracy is crucial, its performance will be questionable.

2.3. XRF Image Super-Resolution

While there is a large body of work on SR for either conventional RGB images [**9**, **34**, **100**, **117**, **118**] or hyperspectral images [**1**, **2**, **35**, **50**, **62**, **67**, **112**], little work has been done for SR on XRF images. XRF SR poses a particular challenge because the acquired spectrum signal usually has low SNR. In addition, correlations among spectral channels need to be preserved for the interpolated pixels. Finally, the large number of channels (4096 channels in Fig. 1.4) leads to a computation challenge, since super-resolving each channel slice by slice is computational expensive.

In our previous work on XRF image SR [**28**], a Dictionary Learning (DL) technique [**40**] with spatial smoothness constraint was applied to reduce the number of channels to be super-resolved by traditional SR methods. The performance was limited since SR based on the LR XRF image is rather challenging.

The low spatial resolution limitations of hyperspectral images have led researchers in image processing and remote sensing to attempt to fuse them with conventional high spatial resolution RGB images. This image fusion [**110**] style SR can be seen as a generalization of pan-sharpening [**5**, **47**], which enhances an LR color image by fusing it with a single-channel black-and-white ("panchromatic") image of higher resolution. Recently, matrix factorization has played an important role in enhancing the spatial resolution of hyperspectral imaging systems [**1**, **35**, **62**, **67**]. In [**62**], a sparse matrix factorization technique was proposed to decompose the LR hyperspectral image into a dictionary of basis vector and a set of sparse coefficients. The HR hyperspectral image was then reconstructed using the learned basis and

sparse coefficients computed from the HR RGB image. The SR performance is improved by imposing spatio-spectral sparsity [**1**], physical constraints [**67**] and structural prior [**35**]. Bayesian approaches [**50**, **112**] impose additional priors on the distribution of the image intensities and apply MAP inference. Non-parametric Bayesian dictionary learning is applied in [**2**] to obtain a spectral basis, and then obtain the HR image with Bayesian sparse coding.

In all these hyperspectral image SR methods [**1**, **2**, **35**, **50**, **62**, **67**, **112**], because the RGB spectrum is contained within the hyperspectral spectrum, the transformation from the hyperspectral signal to the RGB signal is linear and known. However, in XRF imaging, because the RGB spectrum (400 nm - 700 nm) is outside the XRF spectrum (0.03 nm - 6 nm, i.e., 0.2 KeV - 40 KeV), there is no direct transformation from the XRF signal to the RGB signal. Also the hidden part of the scanning object will be captured in the XRF image [**3**], while absent in the RGB image. According to our knowledge, no work has been done on XRF image SR, by modeling the input LR image as a combination of the visible and non-visible components, and increasing the spatial resolution of the visible component and non-visible component by fusing an HR conventional RGB image with implicit spectral transformation and using a standard total variation SR method, respectively. The physically grounded unmixing constraints in [**67**] on endmembers and abundances are extended in this paper to model the implicit transformation between the XRF spectrum and the RGB spectrum, as well as the visible / non-visible separation.

2.4. XRF Image Inpainting

Irregular sampling techniques have long been studied in the image processing and computer graphics fields to achieve compact representation of images. Such irregular sampling techniques, such as stochastic sampling [**26**], may have better anti-aliasing performance

compared with uniform sampling intervals if frequencies greater than the Nyquist limit are present. Further performance improvement can be obtained if the sampling distribution is not only irregular but also adaptive to the signal itself. The limited samples should be concentrated in those rich in detail parts of the image, so as to simulate human vision [**98**]. Several works have been reported in the literature on adaptive sampling techniques. An early significant work in this direction is made by Eldar *et al.* [**43**]. A farthest point strategy is proposed which permits progressive and adaptive sampling of an image. Later Rajesh *et al.* [**89**] proposed a progressive image sampling technique inspired by the lifting scheme of wavelet generation. A similar method is developed by Demaret *et al.* [**32**] by utilizing an adaptive thinning algorithm. Ramponi *et al.* [**91**] developed an irregular sampling method based on a measure of the local sample skewness. Lin *et al.* [**72**] viewed grey scale images as manifolds with density and sampled them according to the generalized Ricci curvature. Liu *et al.* [**75**] proposed an adaptive progressive image acquisition algorithm based on kernel construction.

Most of these irregular sampling and adaptive sampling techniques [**26**,**32**,**43**,**72**,**75**,**89**, **91**], need their own specific reconstruction algorithm to reconstruct the full sampled signal. Furthermore, all these sampling techniques are model based approaches, relying on pre-defined priors and according to our knowledge, no work has been done on utilizing machine learning techniques to design the adaptive sampling mask.

Inspired by the recent successes of convolutional neural networks (CNNs) [**65**,**101**] in high level computer vision tasks, deep neural networks (DNNs) emerged in addressing low level computer vision tasks as well [**21**,**34**,**45**,**54**,**55**,**59**,**86**,**96**,**106**,**113**,**121**]. For the task of image inpainting, Pathak *et al.* [**86**] presented an auto-encoder to perform context-based image inpainting. The inpainting performance is improved by introducing perceptual loss [**121**] and

on-demand learning [**45**]. Iliadis *et al.* [**54**] utilized a deep-fully-connected networks for video compressive sensing while also learning an optimal binary sampling mask [**55**]. However, the learned optimal binary sampling mask is not adaptive to the input video signals. According to our knowledge, no work has been made on generating the adaptive binary sampling mask for the image inpainting problem using deep learning.

While there is a large body of work on inpainting conventional RGB images [**13**, **14**, **27**, **44**, **45**, **86**, **95**, **121**, **127**], very little work has appeared in the literature on inpainting XRF images [**13**], and no work on fusing a conventional RGB image during the inpainting process. XRF image inpainting poses a particular challenge because the acquired spectrum signal usually has low SNR. In addition, the correlation among spectral channels needs to be preserved for the inpainted pixels. Finally, the large number of channels (2048 channels or 20 element maps in Figure 1.4) leads to a computational challenge, since inpainting each channel or element map slice by slice is computational expensive. In our previous work on spatial-spectral representation for XRF image super-resolution [**29**], the input LR XRF image is fused with an HR conventional RGB image to obtain an HR XRF output image. In detail, a linear mixing model [**80**, **87**] is applied to model the XRF spectrum of each pixel. The XRF signal is also modeled as a combination of the visible signal and the non-visible signal, because the hidden part of the painting is not visible in the conventional RGB image, while it can be captured in the XRF image [**3**], in other words, there is no direct one-to-one mapping between the visible RGB spectrum and the XRF spectrum. The spatial resolution of the visible component XRF signal is increased by fusing an HR conventional RGB image while the spatial resolution of the non-visible part is increased by using a standard total variation regularizer [**8**, **81**]. The proposed XRF image inpainting algorithm by fusing an

HR conventional RGB image can be regarded as an extension of our previous XRF SR approach.

CHAPTER 3

Sparse Representation Based Multiple Frame Video Super-Resolution

3.1. Introduction

In this chapter, we propose an approach for video SR, according to which multiple LR observations of an HR video frame are utilized according to Equation (1.3) for both designing coupled dictionaries connecting the sparse representation of LR and HR image frames, as well as for reconstructing an HR frame. We borrow two ideas from single frame SR, namely, bilevel coupled dictionary [**117**–**120**] and multiple-dictionaries [**37**, **108**], to be explained later. We incorporate them into a multiple frame SR framework, according to which the non-redundant information contained in LR frames which are typically related by sub-pixel shifts among them is utilized to generate an HR frame. We propose a multiple dictionary multiple frame video SR algorithm utilizing sub-pixel accurate motion estimation. With our proposed SR approach, the estimated optical flow is utilized to obtain multiple frame high accuracy registration and an HR frame is reconstructed from multiple LR frames. The moving parts in a scene can be super-resolved by the sub-pixel shift information while for the stationary parts, the SNR improves due to the multiple observation of the same scene. As far as registration error is concerned, we address it by adapting the weight parameter that enforces the similarity of multiple LR observations, so that our proposed algorithm has the ability to move between single frame bilevel coupled dictionary [**117**, **118**] SR approach and multi-frame SR approach, and perform at least as good as any of these two approaches.

The multiple frame SR performance is further improved by training dictionaries from consecutive video frames. Most dictionary learning techniques [**41**, **82**, **88**, **90**, **97**, **108**, **119**, **120**, **124**] use still images or individual video frames to train the dictionaries. However, this causes an inconsistency in multiple frame SR since we are super-resolving videos while the dictionaries are trained from still images. The proposed training from videos incorporates temporal information into the dictionaries, and makes the training and testing phases consistent. Although as a result the training phase becomes more complicated, the testing phase remains the same.

Because our proposed SR method is a learning method, we do not explicitly model and estimate the blur kernel (matrix B in Equation (1.3)) in the sparse coding reconstruction of the HR frame in the SR testing phase. Clearly, in the training phase, the HR and LR patch pairs carry the blur information which will be incorporated into the resulting trained HR and LR dictionary pairs. To handle the potential mismatch of the blur kernel in training and testing phase, an idea similar to the one in [**82**] can be applied. Multiple blurred and downsampled versions of the same HR video will be used to train LH/HR dictionary pairs (assume there are N such pairs). All such dictionary pairs will then be used to reconstruct N HR videos from one LR observation during testing. A decision criterion can be adopted to decide which reconstruction is the preferred one. For example, from the N HR reconstructions N LR observations can be generated using the N different blur kernels. All these N LR generated observations will be compared against the actual observation and the one with the smallest error (say the k^{th} one) will determined which HR reconstruction (the k^{th} one) will be chosen. This way a blur identification is indirectly performed.

Based on the results reported in the literature [**18**, **85**], the quality of the multiple-frame SR critically depends on the accuracy of the motion estimates. The two important

characteristics of the motion field are that 1) it should have sub-pixel accuracy and 2) it should be dense. There is a plethora of techniques in the literature for estimating a dense motion field [**10**, **52**]. Optical flow techniques assume that the optical flow is preserved over time. This information is utilized to form the optical flow equation connecting spatial and temporal gradients. More recent optical flow algorithms [**39**, **99**] use a variational coarse-to-fine framework to handle large displacements.

In-depth and comprehensive experiments demonstrate that our proposed SR framework outperforms the state-of-the-art super resolution frameworks, such as, NE+NNLS [**15**], NE+LLE [**23**], ANR [**104**], SR-CNN [**34**], Enhancer [**57**] and Bayesian [**73**] on UHD (4K) sequences.

Our main contributions lie in the following three aspects:

- We extended the bilevel coupled dictionary learning based single frame SR method [**117**, **118**] from a single dictionary to multiple dictionaries (Section 3.2.1).
- We extended the bilevel coupled dictionary learning based single frame SR method [**117**, **118**] from a single frame to multiple frames by developing two approaches: a batch approach and a recursive approach (Section 3.2.2 3.2.3).
- We proposed and developed an approach for training the dictionaries from consecutive video frames instead from individual still images (Section 3.3).

This paper is an extension of our previous work [**30**]. The extension and additional contributions lie in the following aspects:

- We proposed a recursive multiple frame video super-resolution algorithm in Section 3.2.3 and the corresponding algorithm for training dictionaries from videos in Section 3.3.2.

- We utilized an adaptive weight parameter which depends on the mis-registration error (Equation 3.7).
- We introduced an iteration between motion estimation and HR frame estimation for both the batch approach and recursive approach.
- We illustrated a detailed algorithm for training dictionaries from videos.
- We introduced multiple SR steps for large upscale factors.
- We included more comprehensive experimental results.

The rest of the chapter is organized as follows. Section 3.2 presents our proposed dictionary based multiple-frame SR framework. Section 3.3 illustrates a novel dictionary training strategy, that of training from videos. Section 3.4 provides experimental results, and finally conclusions are drawn in Section 3.5.

3.2. Dictionary Based Multiple-Frame Super-Resolution Approach

Given the LR image sequence $\{I_1^l, \ldots, I_k^l, \ldots\}$, the goal of SR is to estimate the HR sequence $\{I_1^h, \ldots, I_k^h, \ldots\}$. Since each frame is primarily correlated with its neighbors and to also reduce computation, when we are super-resolving the k^{th} frame I_k^h, only the adjacent $(M + N)$ frames $I_{k-M}^l, \ldots, I_{k+N}^l$ are used. Clearly when $N = 0$, causal processing is performed.

In this section, we introduce two approaches to find the sparse representation of an LR patch y_k by incorporating motion information from the neighboring frames, namely, the batch approach and the temporally recursive approach. The core idea of these two approaches originates from the fact that image registration through motion compensation provides multiple observations of the same scene, enabling the SR algorithm to take advantage of the details lost in the k^{th} frame but present in past or future frames. For simplicity the super-resolution

framework will be derived for gray-scale images; however, it can be easily extended to handle color image.

3.2.1. Multiple Bilevel Dictionary Learning

The first task we address is the coupled learning of high and low resolution dictionaries over a large database of training HR images. Each HR image I_j^h in the training database is degraded by blur and noise and down-sampled, according to Equation (1.2), resulting in the corresponding LR image $\tilde{I}_j^l$. Each LR image $\tilde{I}_j^l$ is up-sampled using bicubic interpolation to become I_j^l, so that I_j^h and I_j^l have the same size. In the remaining part of the paper, when dealing with LR frames, we refer to I_j^l, which is the bicubically interpolated LR frame $\tilde{I}_j^l$. I_j^h and I_j^l are then divided into patches of size $W \times W$; the corresponding i^{th} patches out of L total patches are lexicographically ordered to form vectors x^i and y^i, respectively. In the dictionary learning phase, we aim at finding HR and LR dictionaries D^h and D^l such that the sparse representation of any HR patch over D^h is identical to that of the corresponding LR patch over D^l. In order to do so, Yang *et al.* [**117**, **118**] formulated the following bilevel optimization problem

$$\begin{aligned} \min_{D^h, D^l} \quad & \sum_{i=1}^{L} \left\| x^i - D^h z^i \right\|_2^2 \\ \text{s.t.} \quad & z^i = \arg\min_{\alpha^i} \left\| F y^i - F D^l \alpha^i \right\|_2^2 + \lambda \left\| \alpha^i \right\|_1 \\ & \left\| D^h(:,k) \right\|_2 \leq 1, \left\| D^l(:,k) \right\|_2 \leq 1, \ \forall k, \end{aligned} \tag{3.1}$$

where α^i contains the sparse representation of the i^{th} HR/LR patch, $\|.\|_2$ and $\|.\|_1$ represent the l_2 and the l_1 vector norms, respectively, λ is the regularization parameter which controls the sparsity of the sparse coefficient a^i, F is a linear operator which extracts features of the

LR patches, and $\left\|D^h(:,k)\right\|$ and $\left\|D^l(:,k)\right\|$ indicate the k^{th} column of matrices D^h and D^l, respectively.

In the testing phase, given an observed LR patch y, we first solve the following LASSO problem

$$z = \arg\min_{\alpha} \left\|Fy - FD^l\alpha\right\|_2^2 + \lambda \left\|\alpha\right\|_1, \tag{3.2}$$

and then the sparse coefficient z is applied to the HR dictionary D^h to obtain the HR patch x corresponding to y, that is,

$$x = D^h z. \tag{3.3}$$

With the bilevel dictionary learning technique, in the training phase, when updating the sparse coefficient z^i in the so referred to as the lower level, the optimization is consistent with the optimization in the testing phase (Equation (3.2)), thus guaranteeing good reconstruction accuracy. Improved SR results have been reported with this bilevel formulation in [**117**,**118**] compared to the previous formulation in [**119**,**120**].

Because of the diverse structures and textures in images of different styles, using a general coupled dictionary is often not good enough to super-resolve all variations in image patches. Considering the fact that image patches, according to their appearance, can be classified into different categories (such as textures, flat regions, edges, etc.), we train a coupled dictionary for each such category. The heuristic clustering strategy in [**108**] is integrated in our framework. More specifically, K-Means clustering is performed on sampled LR training patches y after applying the feature filter F. Let y_c^i be the i^{th} LR training patch belonging to cluster c, which has in total L_c patches, and x_c^i its corresponding HR training patch. The coupled dictionary $(D_c^l\ D_c^h)$ is then trained on $\{x_c^i, y_c^i\}_{i=1}^{L_c}$ based on Equation (3.1).

After learning the C coupled dictionaries $\{(D_1^l\ D_1^h), \ldots, (D_C^l\ D_C^h)\}$, during the testing phase, for a sample LR patch y, the most appropriate dictionary c^* is determined via

$$c^* = \underset{c=1\ldots C}{\arg\min} \|O_c - Fy\|_2^2 , \tag{3.4}$$

where O_c is the centroid of the columns of the c^{th} LR dictionary. Here, the Euclidean distance between the centroid and the LR patch is used as the similarity measure. The best dictionary pair $(D_{c^*}^l\ D_{c^*}^h)$ is then used to find the HR version of y (denoted by x) by solving Equation (3.2).

3.2.2. A Batch Multiple Frame Video Super-Resolution Algorithm

A dictionary based batch multiple-frame video SR algorithm is shown in Fig. 3.1 (when $M = N = 1$). The three consecutive LR frames are shown in pink while the HR frame corresponding to the middle LR frame is depicted in green. We want to fill in the patch x_k which is the HR version of the patch y_k in the k^{th} frame, by combining information from patch y_k, the motion compensated patches $y_{k-M}^{MC}, \ldots, y_{k-1}^{MC}, y_{k+1}^{MC}, \ldots, y_{k+N}^{MC}$ and the pre-trained multiple coupled dictionaries $(D_c^l\ D_c^h)$.

With this approach, we alternate optimizing for the motion field and the HR frames I_k^h. In the first iteration, the motion field is estimated based on the LR input frames $\left\{I_{k+j}^l\right\}_{j=-M}^{j=N}$. The optical flow method in [**39**] is applied to obtain the motion field with sub-pixel accuracy. Then the motion compensated versions of y_k are computed according to the motion field in the past and future frames, denoted by $\left\{y_{k+j}^{MC}\right\}_{j=-M}^{j=N, j\neq 0}$. To super-resolve y_k in the k^{th} frame, the most appropriate LR dictionary indexed by c^*, out of the C possible choices, is found via Equation (3.4). Then the best dictionary pair $(D_{c^*}^l\ D_{c^*}^h)$ is picked to find the HR version

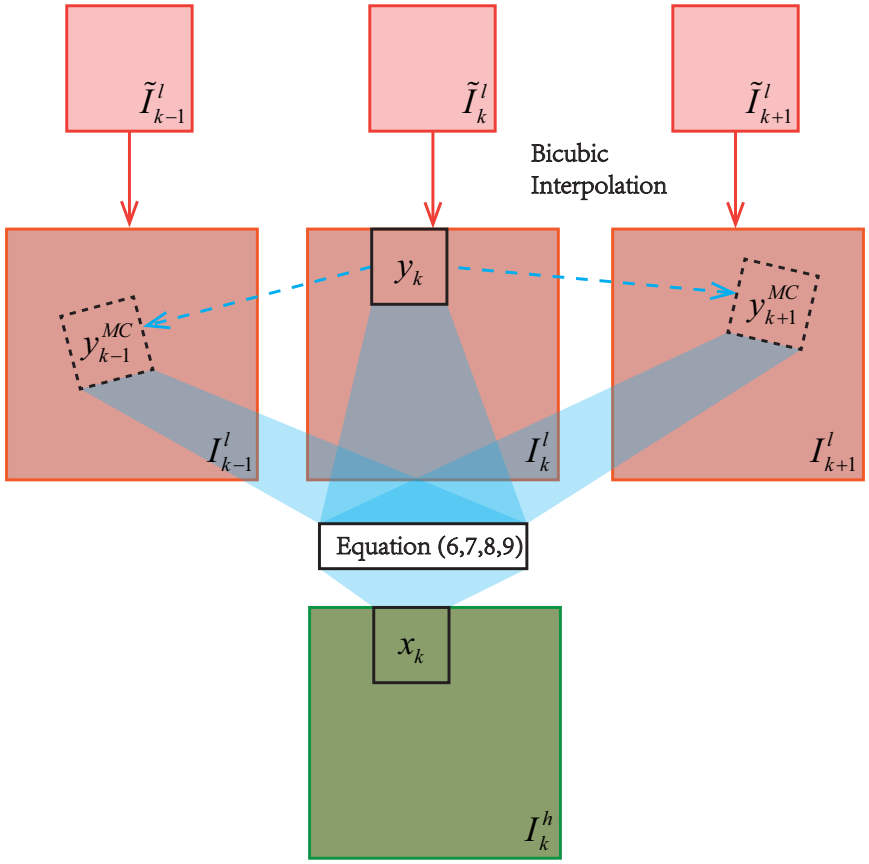


Figure 3.1. Batch approach (the figure is depicted for the case $M = N = 1$).

of y_k according to

$$\min_{\substack{\alpha_k,\alpha^{MC}_{k+j},\\ j=-M,\dots,N,j\neq 0}} \left\|Fy_k - FD^l_{c^*}\alpha_k\right\|_2^2 + \sum_{j=-M,j\neq 0}^{N} \left\|Fy^{MC}_{k+j} - FD^l_{c^*}\alpha^{MC}_{k+j}\right\|_2^2$$
$$+\lambda(\|\alpha_k\|_1 + \sum_{j=-M,j\neq 0}^{N} \left\|\alpha^{MC}_{k+j}\right\|_1) + \sum_{j=-M,j\neq 0}^{N} \gamma_j \left\|D^h_{c^*}\alpha_k - D^h_{c^*}\alpha^{MC}_{k+j}\right\|_2^2 \tag{3.5}$$

$$x_k = D^h_{c^*}\alpha_k, \tag{3.6}$$

where α_{k+j}^{MC} is the sparse representation of y_{k+j}^{MC}. The first two terms in Equation (3.5) ensure the fidelity to the LR observations (similar to Equation (3.2)). The middle two terms are l_1 regularizers promoting the sparse representation of the LR patches by the LR dictionaries and the last term enforces the similarity of the reconstructed HR patches from past and future frames to the current frame. Only α_k is used in Equation (3.6) to reconstruct the HR patch in the current frame. The regularization parameter λ is chosen experimentally, while the choice of the γ_j's is described below.

After the reconstruction of the HR frames $\left\{I_{k+j}^h\right\}_{j=-M}^{j=N}$ in the first iteration, we update the motion field based on these HR frames by applying the optical flow algorithm in [**39**], since it typically results in a higher accuracy motion field than the one resulting by using the LR frames $\left\{I_{k+j}^l\right\}_{j=-M}^{j=N}$. We can alternate updating the motion field and the HR frames $\left\{I_{k+j}^h\right\}_{j=-M}^{j=N}$ until convergence.

An important point to be taken into account is that the desired accuracy on motion estimation will not be reached if images have a lot of aliasing. Notice that the mis-registration error between y_k and y_{k+j}^{MC}, i.e., $e(k,k+j) = \left\|y_k - y_{k+j}^{MC}\right\|_2$, is proportional to $\left\|D_{c^*}^h \alpha_k - D_{c^*}^h \alpha_{k+j}^{MC}\right\|_2$. Therefore, γ_j in Equation (3.5) should be small when $e(k,k+j)$ is relatively large, and vice versa, in other words they are inversely proportional. The exponential function of the mis-registration is applied here to formalize this relationship, as in [**84**],

$$\gamma_j = \beta_1 \cdot \exp(-\beta_2 \cdot e(k,k+j)^2), \tag{3.7}$$

where β_1 and β_2 are adjusted experimentally. If the registration error is large, γ_j will become small and the proposed method in Equation (3.5) degenerates to a single frame super-resolution method, since we weakly enforce the similarity of the reconstructed HR patches in the temporal domain.

3.2.3. A Recursive Multiple Frame Video Super-Resolution Algorithm

In this section, we propose a novel temporally recursive algorithm for dictionary-based multiple-frame video SR. By using information from already super-resolved frames in the past, the recursive method provides efficient computation, reduced storage, high quality super-resolution results and no delay in processing.

As depicted in Figure (3.2), with the recursive approach, unlike the batch approach, only past frames are used in order to super-resolve y_k. This way the algorithm is temporally causal therefore there is no delay by waiting for future LR frames prior to super-resolving the current one. Because neighboring frames exhibit redundant information, using HR information from previously super-resolved frames can improve the quality of the current SR frame.

Given an LR patch y_k in the k^{th} frame, the most suitable LR dictionary indexed by c^* is first found via Equation (3.4). Like the iteration estimation process of the HR frames and motion field in the batch approach (Section 3.2.2), in the first iteration, the motion field is estimated by the optical flow method in [**39**] with sub-pixel accuracy based on the LR frames $\left\{I^l_{k-j}\right\}_{j=0}^{j=N}$. Motion compensated versions of y_k $\left(\left\{y^{MC}_{k-j}\right\}_{j=1}^{j=N}\right)$ are then found according to the motion field. Subsequently, their corresponding HR patches $\left(\left\{x^{MC}_{k-j}\right\}_{j=1}^{j=N}\right)$ are determined

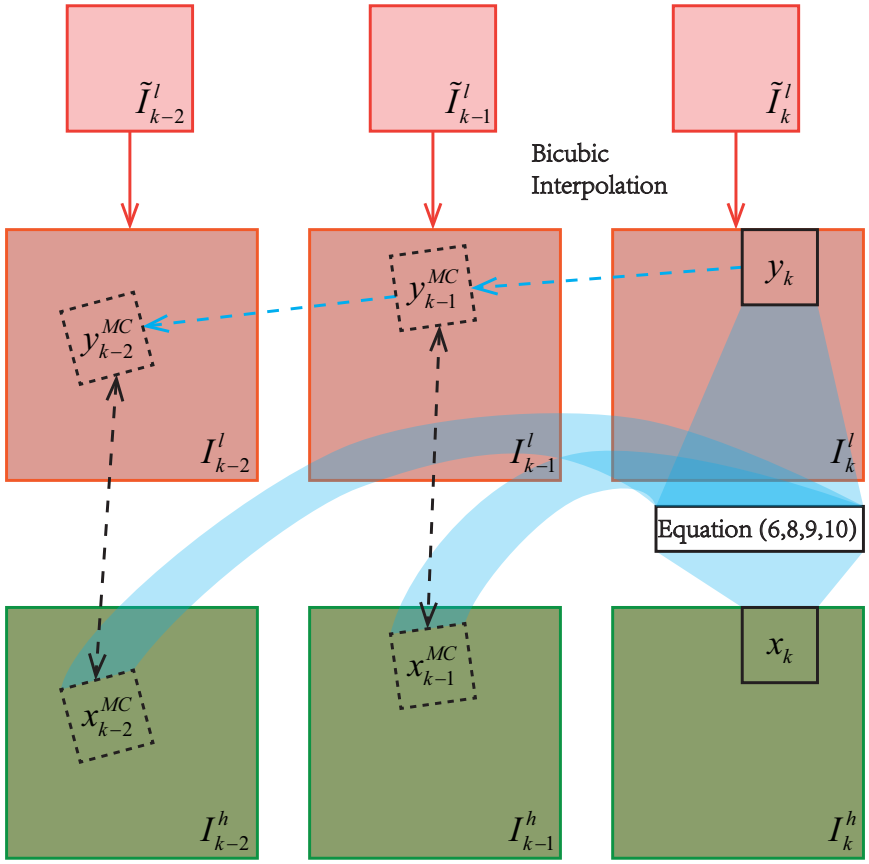


Figure 3.2. The recursive approach, (the figure is depicted for the case $N = 2$).

by the motion field as well and substituted into the following temporally recursive model

$$\min_{\alpha_k} \left\| F y_k - F D^l_{c^*} \alpha_k \right\|_2^2 + \lambda \left\| \alpha_k \right\|_1 + \sum_{j=1}^{N} \gamma_j \left\| D^h_{c^*} \alpha_k - x^{MC}_{k-j} \right\|_2^2 \tag{3.8}$$

The first term in the above equation ensures the fidelity to the data, i.e., the current LR observations, while the second term promotes the sparsity of the solution α_k. The last term enforces the similarity between the reconstructed HR patches of the current frame ($D^h_{c^*}\alpha_k$) and the previous reconstructed HR patches $\left(\left\{x^{MC}_{k-j}\right\}_{j=1}^{j=N}\right)$. Also γ_j is selected adaptively according to Equation (3.7). Similarly to the batch approach, the corresponding HR patch

x_k is obtained according to Equation (3.6). The reconstruction error will not propagate to future frames due to this adaptive weight. Assume that frame I_k^h has large reconstruction error in a certain region. Its motion compensated patches to frame $(k+1)$ will have large registration error, in which case γ_j will be small and Equation (3.8) will degenerate to a single frame super-resolution method. The reconstructed frame I_{k+1}^h will have smaller reconstruction error and will provide helpful HR information to reconstruct frame $(k+2)$, and so on.

Similarly to the batch approach, after the reconstruction of the HR frame I_k^h in the first iteration, a more accurate motion field can be estimated based on the HR frames $\left\{I_{k-j}^h\right\}_{j=0}^{j=N}$ by applying the optical flow algorithm in [**39**]. The motion field and the HR frames $\left\{I_{k-j}^h\right\}_{j=0}^{j=N}$ are updated in an alternate fashion until convergence.

Unlike the batch approach, with the use of motion compensated HR patches $\left(\left\{x_{k-j}^{MC}\right\}_{j=1}^{j=N}\right)$ from the super-resolved previous HR frames, only the coefficients α_k of the patches in the current frame are estimated, which significantly reduces both storage and computation.

3.3. Training Dictionaries from Videos

Typically, for the dictionary learning process, all training patches are sampled from still images or individual video frames. This causes some inconsistency in training and testing, since clearly we are trying to super-resolve videos while the dictionaries are trained from still images. Also the optimizations for training (Equation (3.1)) and testing (Equation (3.5) or Equation (3.8)) are not consistent.

We therefore propose two new dictionary training algorithms based on consecutive video frames and motion estimation for both the batch and recursive approaches. Both algorithms

are applied to each cluster (Section 3.2.1) separately, so in the following equation we omit the dependency on a particular cluster for simplifying the notation.

3.3.1. Video Training for the Batch Approach

As shown in Figure (3.3), during training, a number of consecutive video frames from the training videos are used. In the k^{th} training video sequence of total L_s video sequences, the original HR frames, $\{I^h_{k+j}\}^{j=N}_{j=-M}$, are degraded to obtain the LR frames $\{I^l_{k+j}\}^{j=N}_{j=-M}$. Motion estimation is then performed utilizing the $(M+N+1)$ frames to find the corresponding patches $\{y^{MC}_{k+j}\}^{j=N,j\neq 0}_{j=-M}$ $\left(\{x^{MC}_{k+j}\}^{j=N,j\neq 0}_{j=-M}\right)$ of y_k (x_k) in the past and future frames. Let L_p be the number of sampled patches in each scene. The coupled dictionary $(D^l\ D^h)$ for the batch multiple-frame video SR approach is then trained on $\left\{\left\{x^i_k, y^i_k, \{y^{i\ MC}_{k+j}\}^{j=N,j\neq 0}_{j=-M}\right\}^{i=L_p}_{i=1}\right\}^{k=L_s}_{k=1}$ according to the bilevel dictionary learning in Equation (3.9) above.

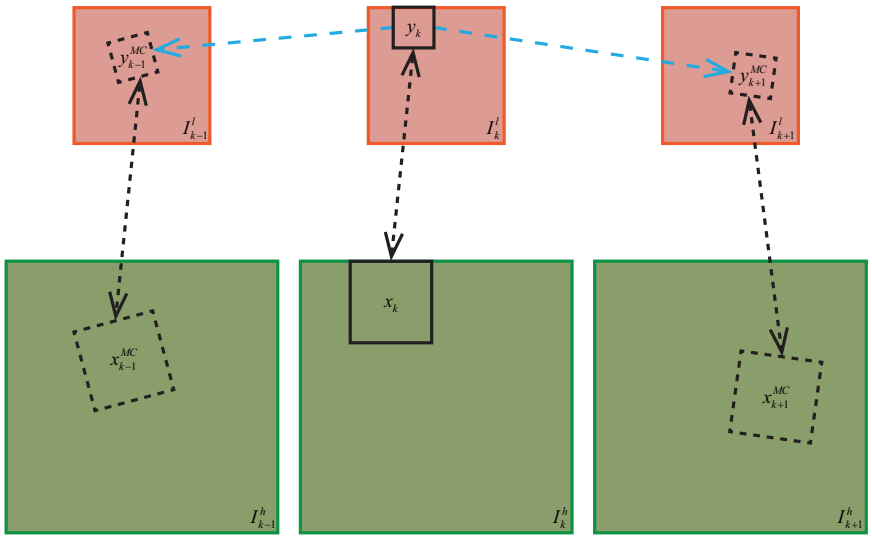


Figure 3.3. Batch approach: training from consecutive frames when $M = N = 1$.

$$
\begin{aligned}
\min_{D^h, D^l} \quad & \sum_{k=1}^{L_s} \sum_{i=1}^{L_p} \left\| x_k^i - D^h z_k^i \right\|_2^2 \\
\text{s.t.} \quad & z_k^i, z_{k+j}^{i\ MC}{}_{j=-M,\ldots,N,j\neq 0} = \arg\min_{\substack{\alpha_k^i, \alpha_{k+j}^{i\ MC}{}_{j=-M,\ldots,N,j\neq 0}}} \left\| F y_k^i - F D^l \alpha_k^i \right\|_2^2 + \sum_{j=-M,j\neq 0}^{N} \left\| F y_{k+j}^{i\ MC} - F D^l \alpha_{k+j}^{i\ MC} \right\|_2^2 \\
& \qquad + \lambda (\|\alpha_k^i\|_1 + \sum_{j=-M,j\neq 0}^{N} \left\| \alpha_{k+j}^{i\ MC} \right\|_1) + \sum_{j=-M,j\neq 0}^{N} \gamma_j \left\| D^h \alpha_k^i - D^h \alpha_{k+j}^{i\ MC} \right\|_2^2 \\
& \qquad \left\| D^h(:,k) \right\|_2 \leq 1, \left\| D^l(:,k) \right\|_2 \leq 1, \ \forall k
\end{aligned}
\tag{3.9}
$$

The objective function in Equation (3.9) is highly nonlinear and nonconvex. Similarly to [**117**, **118**], we alternate optimizations over D^h, D^l and $\left(z_k^i \ \left\{ z_{k+j}^{i\ MC} \right\}_{j=-M}^{j=N,j\neq 0} \right)$ while keeping the remaining of the terms fixed. When D^h and D^l are fixed, the optimization over $\left(z_k^i \ \left\{ z_{k+j}^{i\ MC} \right\}_{j=-M}^{j=N,j\neq 0} \right)$ becomes a standard LASSO problem as reformulated in Equation (3.10).

When $\left(z_k^i \ \left\{ z_{k+j}^{i\ MC} \right\}_{j=-M}^{j=N,j\neq 0} \right)$ and D^h are fixed, the optimization over D^l is reduced to

$$
\begin{aligned}
\min_{D^l} \quad & \sum_{k=1}^{L_s} \sum_{i=1}^{L_p} \Bigg(\left\| F y_k^i - F D^l z_k^i \right\|_2^2 \\
& \qquad + \sum_{j=-M,j\neq 0}^{N} \left\| y_{k+j}^{i\ MC} - F D^l z_{k+j}^{i\ MC} \right\|_2^2 \Bigg) \\
\text{s.t.} \quad & \left\| D^l(:,k) \right\|_2 \leq 1, \ \forall k,
\end{aligned}
\tag{3.13}
$$

$$
\min_{z_{k-M}^{i\ MC}, \cdots, z_k^i, \cdots, z_{k+N}^{i\ MC}} \left\| \begin{bmatrix} FD^l & & & & \\ & \ddots & & & \\ & & FD^l & & \\ & & & \ddots & \\ & & & & FD^l \\ -\gamma_{-M} D^h & & \gamma_{-M} D^h & & \\ & \ddots & \vdots & \ddots & \\ & & \gamma_N D^h & & -\gamma_N D^h \end{bmatrix} \begin{bmatrix} z_{k-M}^{i\ MC} \\ \vdots \\ z_k^i \\ \vdots \\ z_{k+N}^{i\ MC} \end{bmatrix} - \begin{bmatrix} F y_{k-M}^{i\ MC} \\ \vdots \\ F y_k^i \\ \vdots \\ F y_{k+N}^{i\ MC} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \right\|_2^2 + \lambda \left\| \begin{bmatrix} z_{k-M}^{i\ MC} \\ \vdots \\ z_k^i \\ \vdots \\ z_{k+N}^{i\ MC} \end{bmatrix} \right\|_1
\tag{3.10}
$$

which is a quadratically constrained quadratic program (QCQP) [**19**] that can be efficiently optimized using conjugate gradient descent [**68**]. The l_2 norm constraint can be satisfied by simply projecting each column onto the unit ball at each iteration according to Equation (3.14), that is,

$$D^l(:,k) = \frac{D^l(:,k)}{\max\left(1, \|D^l(:,k)\|_2\right)}. \tag{3.14}$$

Finally, when we fix $\left(z_k^i \ \left\{z_{k+j}^{i\ MC}\right\}_{j=-M}^{j=N,j\neq 0}\right)$ and D^l, by collecting terms containing D^h in both upper and lower levels, the optimization over D^h becomes

$$\begin{aligned} \min_{D^h} \quad & \sum_{k=1}^{L_s}\sum_{i=1}^{L_p}\left(\left\|x_k^i - D^h z_k^i\right\|_2^2 + \sum_{j=-M,j\neq 0}^{N} \gamma_j \left\|D^h(z_k^i - z_{k+j}^{i\ MC})\right\|_2^2 \right) \\ \text{s.t.} \quad & \left\|D^h(:,k)\right\|_2 \leq 1, \ \forall k, \end{aligned} \tag{3.15}$$

which is also a QCQP [**19**] and can be optimized by conjugate gradient descent [**68**]. The projection to the unit ball becomes

$$D^h(:,k) = \frac{D^h(:,k)}{\max\left(1, \|D^h(:,k)\|_2\right)}. \tag{3.16}$$

Algorithm 1 summarizes the complete procedure of our coupled dictionary learning algorithm for sequential video training.

Notice that the lower level optimization of Equation (3.9) in the training phase is consistent with the optimization in the testing phase of multiple-frame sequential SR in Equation (3.5). Therefore the training and testing phases are consistent and the accuracy in sequentially reconstructing one frame from multiple frames is guaranteed.

Algorithm 1. Coupled dictionary learning: training from video for batch approach

input: training patch sets $\left\{\left\{x_k^i, y_k^i, \{y_{k+j}^{i\ MC}\}_{j=-M}^{j=N,j\neq 0}\right\}_{i=1}^{i=L_p}\right\}_{k=1}^{k=L_s}$

1: Initialization: initialize $D^{l(0)}$ and $D^{h(0)}$ by Equation (3.1) based on $\left\{\{x_k^i, y_k^i\}_{i=1}^{i=L_p}\right\}_{k=1}^{k=L_s}$, $n = 0$

2: **repeat**

3: Update $\left(z_k^i, \{z_{k+j}^{i\ MC}\}_{j=-M}^{j=N,j\neq 0}\right)$ according to Equation (3.10);

4: Update $D^{l(n+1)}$ from $D^{l(n)}$ according to Equation (3.13);

5: Project the columns of $D^{l(n+1)}$ onto the unit ball according to Equation (3.14);

6: Update $D^{h(n+1)}$ from $D^{h(n)}$ according to Equation (3.15);

7: Project the columns of $D^{h(n+1)}$ onto the unit ball according to Equation (3.16);

8: n=n+1;

9: **until** convergence

output: coupled dictionaries $D^{l(n)}$ and $D^{h(n)}$.

To train multiple dictionaries, Algorithm 1 is applied to each cluster separately. Feature filter F is applied on the LR patch y_k^i to cluster each training patch set $\left\{x_k^i, y_k^i, \{y_{k+j}^{i\ MC}\}_{j=-M}^{j=N,j\neq 0}\right\}$.

3.3.2. Video Training for the Recursive Approach

Similarly to Section 3.3.1, a number of consecutive video frames are used in the training phase, as depicted in Figure (3.4). The original HR frames $\{I_{k-j}^h\}_{j=0}^{j=N}$, are degraded to obtain the LR frames $\{I_{k-j}^l\}_{j=0}^{j=N}$. The backwards corresponding patches $\{y_{k-j}^{i\ MC}\}_{j=1}^{j=N}$ $\left(\{x_{k-j}^{i\ MC}\}_{j=1}^{j=N}\right)$ to y_k^i (x_k^i) are obtained by motion estimation, performed on the LR frames. Let y_k^i be the i^{th} LR training patch, x_k^i the corresponding HR training patch to (y_k^i) and $x_{k-j}^{i\ MC}$ the motion compensated patch of x_k^i in the $(k-j)^{th}$ HR frame. We then train the coupled dictionary $(D^l\ D^h)$ for the recursive multiple-frame approach based on $\left\{\left\{y_k^i, \{x_{k-j}^{i\ MC}\}_{j=1}^{j=N}, x_k^i\right\}_{i=1}^{i=L_p}\right\}_{k=1}^{k=L_s}$ by the following bilevel optimization

$$
\begin{aligned}
\min_{D^h, D^l} \quad & \sum_{k=1}^{L_s} \sum_{i=1}^{L_p} \left\| x_k^i - D^h z_k^i \right\|_2^2 \\
\text{s.t.} \quad & z_k^i = \arg\min_{\alpha_k^i} \left\| F y_k^i - F D^l z_k^i \right\|_2^2 + \lambda \left\| z_k^i \right\|_1 \\
& \qquad + \sum_{j=1}^{N} \gamma_j \left\| x_{k-i}^{i\ MC} - D^h z_k^i \right\|_2^2 \\
& \left\| D^h(:,k) \right\|_2 \leq 1, \ \left\| D^l(:,k) \right\|_2 \leq 1, \ \forall k.
\end{aligned}
\tag{3.17}
$$

The optimization strategy from Section 3.3.1 can be applied here by alternating optimization over D^l, z_k^i, and D^l. When D^h and D^l are fixed, optimizing over z_k^i is a standard LASSO problem

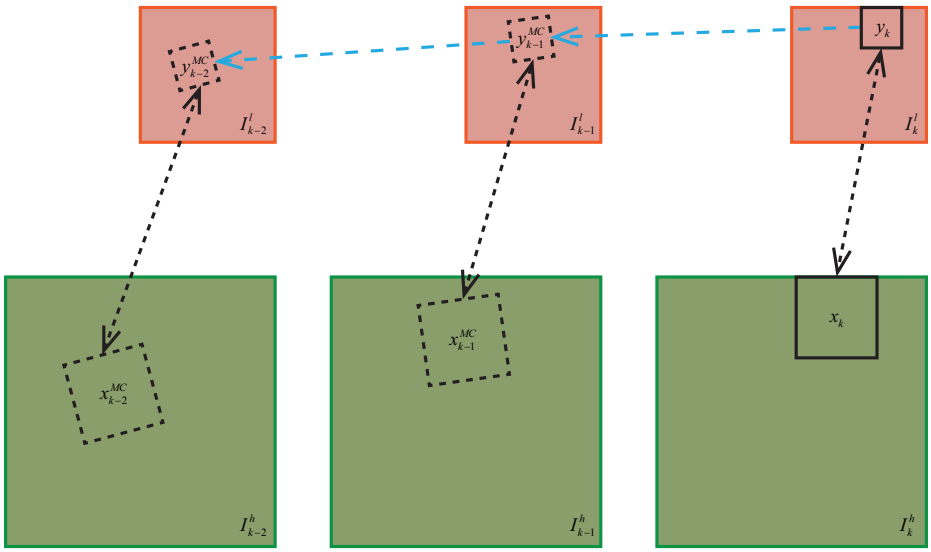


Figure 3.4. Recursive approach: training from consecutive frames when $N = 2$.

$$\min_{z_k^i} \left\| \begin{bmatrix} Fy_k^i \\ \\ \gamma_1 x_{k-1}^{i\ MC} \\ \vdots \\ \gamma_N x_{k-N}^{i\ MC} \end{bmatrix} - \begin{bmatrix} FD^l \\ \\ \gamma_1 D^h \\ \vdots \\ \gamma_N D^h \end{bmatrix} z_k^i \right\|_2^2 + \lambda \left\| z_k^i \right\|_1 . \tag{3.18}$$

In the next step, by fixing D^h and z_k^i, the optimization over D^l is reduced to

$$\begin{aligned} \min_{D^l} \quad & \sum_{k=1}^{L_s} \sum_{i=1}^{L_p} \left\| Fy_k^i - FD^l z_k^i \right\|_2^2 \\ \text{s.t.} \quad & \left\| D^l(:,k) \right\|_2 \leq 1,\ \forall k, \end{aligned} \tag{3.19}$$

which can be carried out by conjugate gradient descent [**68**] followed by projection onto the unit ball (Equation (3.14)).

Finally, the optimization over D^h is carried out by fixing D^l and z_k^i, and solving the following QCQP problem

$$\begin{aligned} \min_{D^h} \quad & \sum_{k=1}^{L_s} \sum_{i=1}^{L_p} \left(\left\| x_k^i - D^h z_k^i \right\|_2^2 + \sum_{j=1}^{N} \gamma_j \left\| x_{k-j}^{i\ MC} - D^h z_k^i \right\|_2^2 \right) \\ \text{s.t.} \quad & \left\| D^h(:,k) \right\|_2 \leq 1,\ \forall k, \end{aligned} \tag{3.20}$$

and then projecting onto the unit ball (Equation (3.14)).

The iterative procedure of the coupled dictionary learning algorithm for recursive video training is summarized in Algorithm 2.

Algorithm 2 can be applied on each cluster separately to train multiple dictionaries. Feature filter F on the LR patch y_k^i is utilized to cluster each training patch set $\left\{ y_k^i, \left\{ x_{k-j}^{i\ MC} \right\}_{j=1}^{j=N}, x_k^i \right\}$.

Algorithm 2. Coupled dictionary learning: train from video for recursive approach

input: training patch sets $\left\{\left\{y_k^i, \{x_{k-j}^{i\ MC}\}_{j=1}^{j=N}, x_k^i\right\}_{i=1}^{i=L_p}\right\}_{k=1}^{k=L_s}$

1: Initialization: initialize $D^{l(0)}$ and $D^{h(0)}$ by Equation (3.1) based on $\left\{\{y_k^i, x_k^i\}_{i=1}^{i=L_p}\right\}_{k=1}^{k=L_s}$, $n = 0$

2: **repeat**

3: Update z_k^i according to Equation (3.18);

4: Update $D^{l(n+1)}$ from $D^{l(n)}$ according to Equation (3.19);

5: Project the columns of $D^{l(n+1)}$ onto the unit ball according to Equation (3.14);

6: Update $D^{h(n+1)}$ from $D^{h(n)}$ according to Equation (3.20);

7: Project the columns of $D^{h(n+1)}$ onto the unit ball according to Equation (3.16);

8: n=n+1;

9: **until** convergence

output: coupled dictionaries $D^{l(n)}$ and $D^{h(n)}$.

3.4. Experimental Results

Our two proposed algorithms extend the bilevel dictionary learning [**117**, **118**] in two aspects: from single dictionary to multiple dictionaries and from single frame to multiple frames. We first show that each extension is beneficial by comparing the SR performances of single dictionary single frame SR (Bilevel), multiple dictionaries single frame SR (MDSF), single dictionary multiple frames SR (SDMF-B for the batch approach, SDMF-R for the recursive approach), multiple dictionary multiple frames SR (MDMF-B for the batch approach, MDMF-R for the recursive approach) and MDMF-B/MDMF-R with the proposed video training (MDMF-B-VT/MDMF-R-VT). We also compare the performance of the proposed algorithm with state-of-the-art video SR algorithms, such as Enhancer [**57**], Bayesian [**73**], Bayesian-MB [**77**] and DraftCNN [**71**].

3.4.1. Implementation Details

We performed an extensive set of experiments utilizing frames of a 4K video database [**56**]. There is a high demand of upscaling videos of low resolutions to 4K resolution (2160×3840) these days due the proliferation of 4K monitors. Upscaling of 1080P (1080×1920) or 540P (540×960) resolution to 4K videos is a representative example used in this paper, resulting in an upscale factor of 2 and 4, respectively. In detail, for upscale factor 2, there are in total 57 scenes in the 4K video database [**56**]. LR (1080×1920) frames result from the degradation of the original HR (2160×3840) frames by the Matlab function "imresize", which is experimentally found to represent a Gaussian blur kernel with variance approximately equal to 0.4, thus specifying the B matrix in Equation (1.2). 50 scenes are used for training and 7 for testing. In the training phase of these experiments, 800,000 patch sets are sampled from the center frame and the motion compensated neighboring frames for training the dictionary from videos, while the same 800,000 patches in center frames are used for training the dictionary from images. The patch size is 5×5 and no feature filter F is applied to the LR patches. The reason for not doing so is that we verified experimentally that by using for example four high-pass filters, as was done in [**117**, **118**], does not provide any sizeable advantage. In addition, four high-pass filter will increase the dimension of the LR dictionary atoms by a factor of four, thus increasing considerably the required computation. λ is chosen to be 0.02 by a parameter traversing experiment, as shown in Figure (3.5). β_1 and β_2 are chosen to be equal to 0.2 and $\frac{1}{3 \times max(e(k,k+j))}$ according to the convexity criteria in [**84**], respectively. Every dictionary for the SDSF, SDMF-B, and SDMF-R approaches has 512 atoms and the dictionary for the MDSF, MDMF-B and MDMF-R approaches has 8 subdictionaries with 512 atoms each. For the multiple-dictionary methods in the testing

phase, we solve the LASSO problem with only one sub-dictionary and the computation for assigning patches to each cluster (Equation (3.4)) is negligible, therefore the comparison is fair.

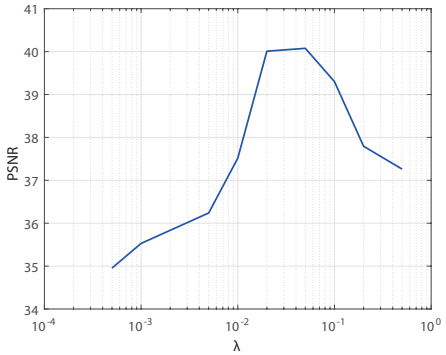


Figure 3.5. λ is traversed to find its optimal value. For each tested λ, we perform the multiple frame SR according to Equation (3.5) and compute its corresponding PSNR value.

In the testing phase of upscale factor 2, 6 consecutive video frames are super-resolved by each method. 5×5 patches are extracted with overlap of 4 pixels between adjacent patches. The multiple estimates of the same pixel from different overlapping patches are averaged to obtain the final result. For those multiple-frame batch SR methods, the current LR frame, together with one LR backward and one LR forward frames (i.e., $M = N = 1$), are utilized to estimate the current frame. For those multiple-frame recursive SR methods, the current LR frame, together with two LR/HR super-resolved backward frames (i.e., $N = 2$) are used to estimate the current frame. We tested a number of optical flow estimation algorithm [**10**]. Based on their comparison we are using the method in [**39**] in all reported experiments.

For an upscale scale factor of 4, similarly to [**117**], we found experimentally that utilizing the trained coupled dictionaries for an upscale factor of 2 and upscaling the frames twice

with an upscale factor of 2 in each step provides better SR results than training and testing directly with an upscale factor of 4.

For color video frames, we apply our video SR algorithm to the luminance channel only, since humans are more sensitive to illumination changes. The color layers (Cb, Cr) are upscaled using bicubic interpolation. The results of the various methods are evaluated in terms of PSNR (peak signal-to-noise ratio) and SSIM [**111**] on the luminance channel.

3.4.2. Effect of the Proposed Extensions

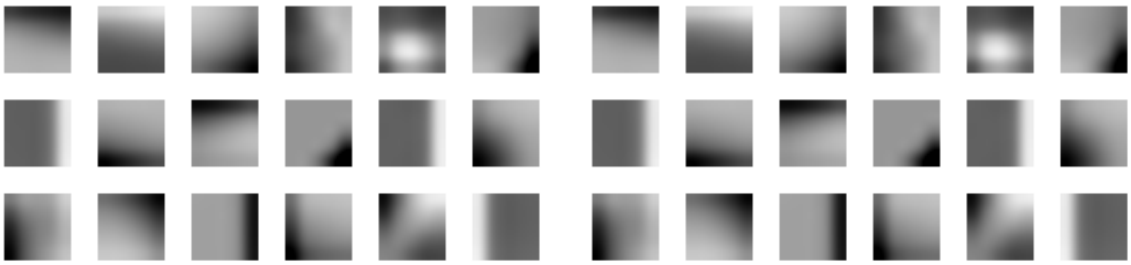
Our two proposed methods are based on the bilevel dictionary learning [**117**, **118**], which is a single dictionary single frame SR method. Since our methods extend it to use multiple dictionaries and multiple frames, we perform a controlled experiment for each extension here to show that the proposed extensions are effective. All multiple-frame SR methods utilize one iteration in updating the motion field and HR frames, since the effect of iteratively updating motion fields and HR frames will be discussed in Section 3.4.3.

	Bicubic	Bilevel	SDMF-B	SDMF-R	MDSF	MDMF-B	MDMF-R	MDMF-B-VT	MDMF-R-VT
Scene 2	45.27	46.12	46.81	46.41	46.79	47.66	46.86	**48.14**	47.41
Scene 8	38.18	39.94	40.08	40.32	40.34	40.59	40.60	40.98	**41.05**
Scene 18	41.43	43.04	43.41	43.69	43.37	43.92	44.19	44.32	**44.46**
Scene 25	44.40	46.69	47.52	47.68	47.37	48.45	47.83	**49.19**	48.59
Scene 33	40.22	42.95	43.08	43.55	43.27	43.68	44.05	**44.49**	44.48
Scene 45	42.43	43.72	44.07	44.18	44.05	44.49	44.28	44.60	**44.62**
Scene 48	33.90	36.10	36.20	35.66	36.55	36.67	36.07	**36.91**	36.64

Table 3.1. PSNR values (in dB) of the SR frame for various methods and test scenes (best results are shown in bold)

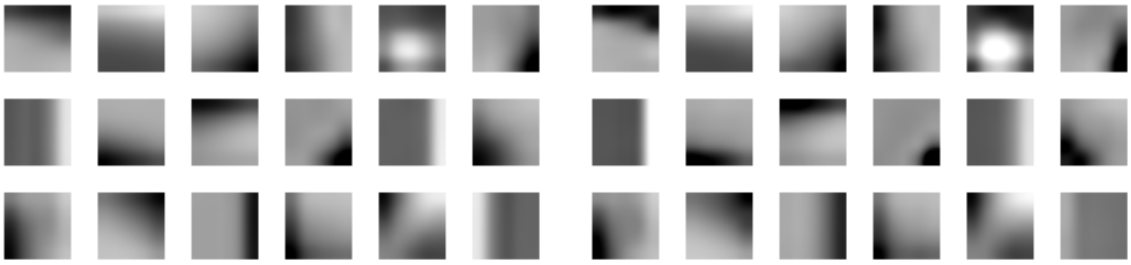
Table 3.1 shows the peak signal-to-noise ratio (PSNR) of the SR frames in dB (the dB values are averaged over 6 testing frames) for various algorithms and test sequences. The best results are shown in bold. From these experiments it is concluded that DL based multiple-frame SR methods (SDMF-B, SDMF-R, MDMF-B, MDMF-R) outperform single

LR/HR dictionary pair trained by Equation (3.1)



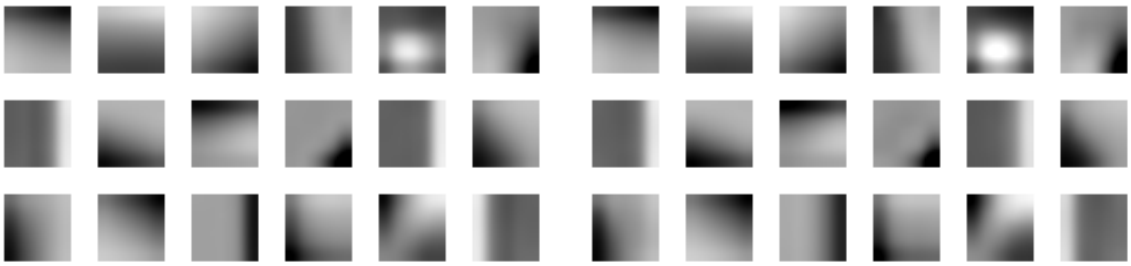
(a) D^l (b) D^h

LR/HR dictionary pair trained by Algorithm 1



(c) D^l (d) D^h

LR/HR dictionary pair trained by Algorithm 2



(e) D^l (f) D^h

Figure 3.6. Atoms of LR/HR dictionary pairs trained by three different dictionary learning algorithms. Each 5×5 atom is upscaled by a factor of 6 by bicubic interpolation for better visualization.

frame SR (Bilevel, MDSF). We can also see that multiple-dictionary SR methods perform better than single-dictionary SR methods, by comparing results of SDMF-B with MDMF-B

	Bicubic	MDSF	MDMF-B-VT			MDMF-R-VT		
			Iter=1	Iter=2	Convergence	Iter=1	Iter=2	Convergence
Scene 2	45.27	46.79	48.14	48.24	**48.26**	47.41	47.98	48.11
Scene 8	38.18	40.34	40.98	41.43	41.62	41.05	41.53	**41.72**
Scene 18	41.43	43.37	44.32	44.66	44.73	44.46	44.85	**44.94**
Scene 25	44.40	47.37	49.19	49.56	49.57	48.59	49.57	**49.68**
Scene 33	40.22	43.27	44.49	45.19	45.33	44.48	45.30	**45.44**
Scene 45	42.43	44.05	44.60	44.73	44.76	44.62	44.86	**44.91**
Scene 48	33.90	36.55	36.91	37.43	**37.56**	36.64	37.16	37.31

Table 3.2. PSNR values (in dB) of the SR frame for various methods, iterations and scenes.

and SDMF-R with MDMF-R. Finally, the proposed training dictionaries from video algorithms (Algorithm 1 and Algorithm 2), MDMF-B-VT and MDMF-R-VT, further improve the SR results over MDMF-B and MDMF-R.

We show in Figure (3.6) LR and HR dictionary atoms resulting from the various dictionary training approaches we have considered. 18 atoms from the D^l dictionary and the corresponding atoms in the D^h dictionary trained according to Equation (3.1) are shown respectively in Figure (3.6a) and (3.6b). The same 18 LR/HR atom pairs resulting from Algorithm 1 and Algorithm 2 are shown respectively in Figures (3.6c), (3.6d) and (3.6e), (3.6f). Notice that dictionaries D^l and D^h trained from Equation (3.1) is the initializations of D^l and D^h for Algorithms 1 and 2. As shown in Figure (3.6), sharper HR atoms result in general from our proposed training Algorithms 1 and 2 (compare Figure (3.6b), (3.6d) and (3.6f)).

In conclusion, our proposed multiple frames SR, utilizing multiple dictionaries and training dictionaries from videos are effective individually and their benefits in SR are cummulative, as the proposed MDMF-B-VT and MDMF-R-VT algorithms provide best SR results.

3.4.3. Effect of Iteration

The proposed batch approach (Section 3.2.2) and recursive approach (Section 3.3.2) by alternating optimizations update the motion fields and reconstructed HR frames I^h. To demonstrate the convergence of the iteration process, we calculate the normalized error $\left\|I^h_{p+1} - I^h_p\right\|^2_F / \left\|I^h_{p+1}\right\|^2_F$ (I^h_p is the reconstructed HR frame at the p^{th} iteration) at each iteration. This normalized error is shown in Figure (3.7) (left) for the batch approach (MDMF-B-VT) for two of the experiments, and the corresponding PSNR is in Figure (3.7) (right). As shown in Figure (3.7), the iteration process converges fast. Similar results are also observed with the recursive approach. In all experiments, we terminate the iteration when the normalized error is below the threshold of 5×10^{-7}.

We visualize the reconstruction error maps of a cropped region of the 6^{th} frame in scene 48 in Figure (3.8), which has a global panning motion of the background with the local motion

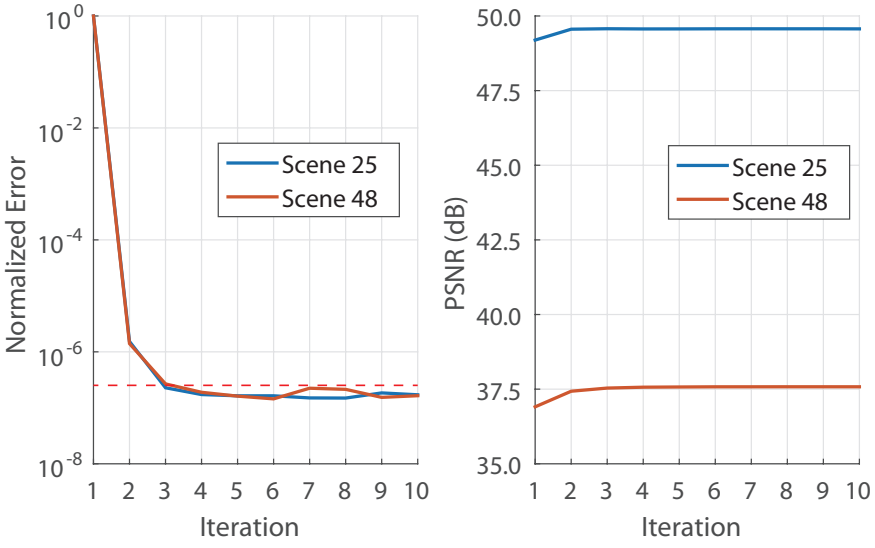


Figure 3.7. The iteration process of the batch approach. The normalized error $\left\|I^h_{p+1} - I^h_p\right\|^2_F / \left\|I^h_{p+1}\right\|^2_F$ of Scenes 25 and Scene 48 as a function of iteration is shown in the left image and the corresponding PSNR values are shown in the right image.

of the foreground. From the heat maps, the reconstruction error in the background texture region decreases as the iteration progresses, also the error in the handle in the foreground almost disappears at the final result.

More interestingly, as shown in Table 3.2, we observe that although the batch SR algorithm outperforms the recursive SR algorithm at iteration 1, their performance is comparable in the final iteration, illustrating that the batch approach is more robust to errors in motion estimation and that both approaches have similar performance when motion estimation is precise.

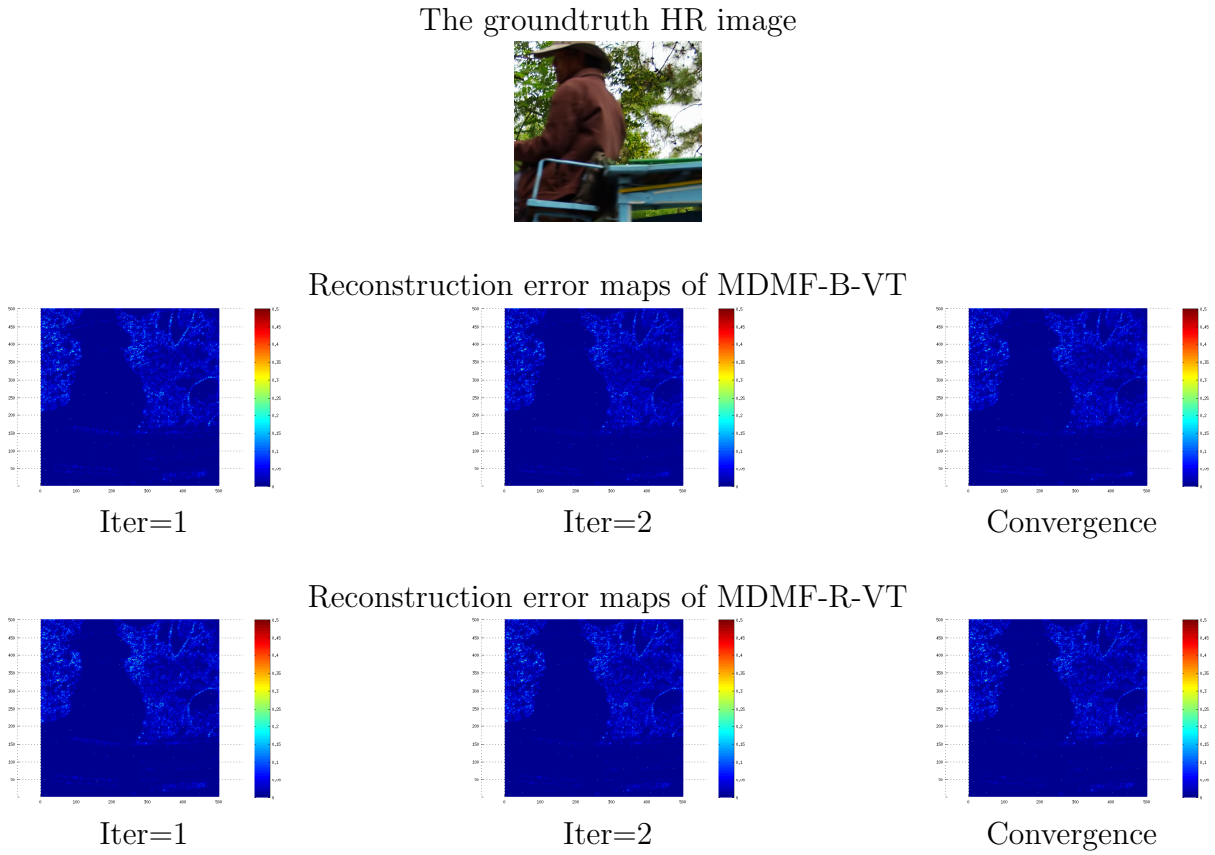


Figure 3.8. Reconstruction error maps of a cropped region in scene 48, with different iteration number.

Video	Bicubic	Bilevel [**117**, **118**]	NE+NNLS [**15**]	NE+LLE [**23**]	ANR [**104**]
Scene 2	44.87 0.9830	46.88 0.9879	46.85 0.9851	46.53 0.9834	46.91 0.9857
Scene 8	38.05 0.9738	39.95 0.9842	40.04 0.9824	41.08 0.9817	40.27 0.9832
Scene 18	40.81 0.9738	43.31 0.9849	43.18 0.9820	43.28 0.9816	43.45 0.9833
Scene 25	44.31 0.9917	47.44 0.9961	46.69 0.9938	47.38 0.9936	47.85 0.9952
Scene 33	39.42 0.9786	42.81 0.9904	42.70 0.9879	43.37 0.9889	43.59 0.9902
Scene 45	42.23 0.9718	44.11 0.9810	43.62 0.9772	43.89 0.9776	44.11 0.9791
Scene 48	33.81 0.9668	36.05 0.9808	35.78 0.9774	36.24 0.9785	36.39 0.9799
Average	40.50 0.9771	42.94 0.9865	42.69 0.9837	43.11 0.9836	43.22 0.9851
Video	SR-CNN [**34**]	Enhancer [**57**]	Bayesian [**73**]	MDMF-B-VT	MDMF-R-VT
Scene 2	47.41 0.9859	46.10 0.9854	46.23 0.9874	**48.26** **0.9882**	48.11 **0.9882**
Scene 8	41.08 0.9852	39.42 0.9823	39.73 0.9828	41.62 **0.9884**	**41.65** 0.9882
Scene 18	43.94 0.9844	42.93 0.9844	43.16 0.9842	44.74 0.9877	**45.04** **0.9884**
Scene 25	48.29 0.9955	46.17 0.9938	46.36 0.9954	**49.57** **0.9970**	49.07 0.9967
Scene 33	43.83 0.9907	43.05 0.9908	42.65 0.9900	**45.33** 0.9937	45.11 **0.9938**
Scene 45	44.32 0.9797	43.11 0.9764	43.59 0.9790	44.76 0.9812	**44.84** **0.9823**
Scene 48	37.48 0.9826	35.24 0.9751	35.27 0.9770	**37.57** **0.9846**	36.89 0.9821
Average	43.76 0.9863	42.29 0.9840	42.43 0.9851	**44.55** **0.9887**	44.39 0.9885

Table 3.3. PSNR values (in dB, top) and SSIM values (bottom) of experimental results comparing our proposed methods with the state-of-the-art methods for upscale factor 2 (best results are shown in bold).

3.4.4. Comparison with State-of-the-Art Results

In the previous Sections 3.4.2 and 3.4.3, we show that our extensions of single frame bilevel SR methods [**117**, **118**] are effective and the iterative updates of the motion field and HR frames improve the SR performance. Here we compare our proposed methods, MDMF-B-VT and MDMF-R-VT, with other state-of-the-art methods, including Bayesian [**73**] and

Video	Bicubic	Bilevel [**117**, **118**]	NE+NNLS [**15**]	NE+LLE [**23**]	ANR [**104**]
Scene 2	39.58 0.9648	40.50 0.9662	41.32 0.9691	41.12 0.9675	41.32 0.9691
Scene 8	32.13 0.9013	32.46 0.9099	33.00 0.9145	32.95 0.9187	32.81 0.9107
Scene 18	35.65 0.9122	36.37 0.9209	36.76 0.9243	36.82 0.9249	36.76 0.9243
Scene 25	36.10 0.9515	37.02 0.9546	37.90 0.9622	37.78 0.9607	37.49 0.9587
Scene 33	32.15 0.8899	33.44 0.9140	33.79 0.9157	33.94 0.9188	34.00 0.9206
Scene 45	36.13 0.9101	36.71 0.9155	37.12 0.9193	37.27 0.9211	37.35 0.9226
Scene 48	27.25 0.8514	28.03 0.8730	28.04 0.8710	28.20 0.8757	28.26 0.8780
Average	34.14 0.9116	34.93 0.9220	35.42 0.9252	35.44 0.9268	35.43 0.9263
Video	SR-CNN [**34**]	Enhancer [**57**]	Bayesian [**73**]	MDMF-B-VT	MDMF-R-VT
Scene 2	43.17 0.9703	40.62 0.9695	39.18 0.9660	**43.48** 0.9737	42.90 **0.9740**
Scene 8	33.40 0.9198	32.09 0.9121	31.73 0.8972	**33.48** **0.9266**	33.42 0.9250
Scene 18	37.50 0.9280	36.44 0.9308	35.70 0.9183	**37.68** 0.9331	37.65 **0.9341**
Scene 25	38.35 0.9633	37.44 0.9621	35.34 0.9473	**39.03** **0.9702**	38.75 0.9687
Scene 33	34.57 0.9230	34.67 0.9304	32.14 0.8945	**34.92** 0.9363	34.86 **0.9374**
Scene 45	37.90 0.9253	37.15 0.9267	35.76 0.9083	**38.42** **0.9340**	38.10 0.9316
Scene 48	28.73 0.8883	27.75 0.8679	26.76 0.8393	**28.75** **0.8921**	28.49 0.8842
Average	36.23 0.9311	35.17 0.9285	33.80 0.9101	**36.54** **0.9380**	36.31 0.9364

Table 3.4. PSNR values (in dB, top) and SSIM values (bottom) of experimental results comparing our proposed methods with the state-of-the-art methods for upscale factor 4 (best results are shown in bold).

a commercial software Enhancer [**57**], and six single frame SR methods including Bicubic, Bilevel [**117**, **118**], NE+NNLS [**15**], NE+LLE [**23**], ANR [**104**] and SR-CNN [**34**]. Two more state-of-the-art methods [**71**, **77**] will be compared in Section 3.4.5 with smaller spatial resolution because their implementation is extremely slow on 4K resolution.

According to Table 3.3 and Table 3.4, our proposed approaches (MDMF-B-VT and MDMF-R-VT) provide the best SR performance compared to all other methods for both

upscale factors of 2 and 4, demonstrating the effectiveness of the proposed algorithms. Although the Bayesian SR method [**73**] evaluates the blur kernel, noise level and super-resolved frames simultaneously, it requires the motion compensation of 30 consecutive frames in the backward and forward directions, which is computationally infeasible with 4K videos because of the memory and computational limitations. When we drop the consecutive frames from 30 to 3, the SR performance of [**73**] is not as good as ours. In Figure (3.9), we compare the visual quality of our upscaled images with the result produced by several recent state-of-the-art SR methods. We notice that all these SR methods produce sharper images than bicubic interpolation, however artifacts are introduced. Next we notice that our proposed method has fewer artifacts and shaper edges compared to all other methods.

3.4.5. Robustness to noise

In this section, we evaluate the noise robustness of different SR algorithms by adding Gaussian noise to the LR input frames. The center regions (480×640) of the original 4K frames are utilized as the HR ground truth, in order to compare with two more state-of-the-art video SR methods, Bayesian-MB [**77**] and DraftCNN [**71**]. The LR input frames (240×320) are obtained by spatially downsampling the HR frames by a factor of 2 and adding white Gaussian noise with variance 0.001. Different SR methods are applied to increase the spatial resolution by a factor of 2. We also show the experimental results with no additional Gaussian noise (noise variance 0).

As shown in Table 3.5, the SR performance of all methods is reduced when noise is added, as expected. The HR dictionaries for the dictionary learning based methods, Bilevel [**117**, **118**], MDMF-B-VT and MDMF-R-VT, are trained with noise free HR frames, so the reconstructed HR frames naturally contain less noise. The sparse coding problem

in SR testing phase is also proven to be robust to noise [**36**], so better SR performance is obtained by the dictionary learning based methods (Bilevel [**117**, **118**], MDMF-B-VT and MDMF-R-VT). By comparing the SR results of Bilevel [**117**, **118**] with MDMF-B-VT and MDMF-R-VT, we found out that better SR performance is obtained by utilizing multiple LR noisy input frames. The proposed MDMF-B-VT consistently outperforms MDMF-R-VT, since it estimates the sparse coefficients of 3 noisy LR patches simultaneously.

The average computation time for all SR algorithms to super-resolve 1 frame is also shown in Table 3.5. All experiments except Enhancer and Bayesian-MB are performed on a Linux workstation with an Intel Xeon E5-2630 processor with 2.4GHz and 64 GB RAM. The Enhancer and the Bayesian-MB algorithm were only available for the Windows operating system and were tested on a different workstation with Intel i7-6820 processor with 2.70GHz and 16 GB RAM. Notice that our proposed methods MDMF-B-VT and MDMF-R-VT can be sped up by a factor of 4 approximately if we only apply 1 iteration instead of 4 iterations. For MDMF-B-VT, the motion estimation takes 21.5s and the sparse coefficients inference of Equation (3.5) takes 14.4s on average for one iteration. For MDMF-R-VT, the motion estimation takes 22.1s and the sparse coefficients inference of Equation (3.8) takes 9.1s on average for one iteration. So our proposed methods can be further sped up by utilizing faster motion estimation methods and sparse coefficients inference algorithms.

We visually compare the SR results of our proposed methods with several other state-of-the-art SR methods, when white Gaussian noise with variance 0.001 is added to the LR input frames. We notice that the dictionary learning based methods, Bilevel [**117**, **118**], MDMF-B-VT and MDMF-R-VT, outperform others in suppressing the noise. The proposed MDMF-B-VT algorithm provides the sharpest HR frame with few artifacts.

The temporal continuity between adjacent super-resolved HR frames is compared in Figure (3.11) by visualizing the motion compensation error of two adjacent super-resolved HR frames by different SR algorithms. The optical flow estimation method in [**39**] is applied to estimate the motion field between two adjacent super-resolved HR frames, and the second frame is warped to the first one according to the computed motion field. The difference between the first frame and the warped second frame is visualized to compare the temporal smoothness of different SR algorithms. The main idea behind this is that if two adjacent super-resolved frames are temporally smooth, then an accurate motion field can be estimated and the resulting motion compensated difference will be small. In quantifying this difference we compute the RMSE (Root-Mean-Square Error). The smoothness of the motion field is of course also indicative of the temporal continuity between adjacent frames. One can imagine situations where the RMSE of the displaced frame difference is small but the motion field exhibits large variations. We therefore also compute the Total Variation (TV) of the estimated motion field vectors, in both the horizontal (VxTV) and vertical (VyTV) directions. In comparing the temporal smoothness of video frames, both the RMSE of the displaced frame difference and the TV of the motion field should be taken into account; the smaller such measures the higher the temporal smoothness. As shown in Figure (3.11), our proposed MDMF-B-VT method produces the smallest RMSE on the motion compensation error, as well as the smallest TV on the motion vector, demonstrating that it better explores the spatio-temporal correlation of consecutive frames. Notice that our proposed MDMF-R-VT method produces the second smallest RMSE on the motion compensation error while have larger TV on the motion vector compared to Bilevel [**117**, **118**], so its temporal smoothness is similar to Bilevel [**117**, **118**]. However, its SR performance is still 2.3 dB better than Bilevel [**117**, **118**] on average according to Table 3.5. It is also interesting to point out that

according to Table V, the single frame SR method SRCNN [**34**] outperforms the multiple frame SR method Enhancer [**57**] in terms of the averaged single frame PSNR and SSIM metrics, while Enhancer [**57**] has smaller motion compensation error of adjacent frames according to Figure (3.11), illustrating that multiple frame SR methods provide an advantage in terms of the temporal smoothness of the super-resolved HR frames.

Video	Bicubic		Bilevel [**117**, **118**]		NE+NNLS [**15**]		NE+LLE [**23**]		ANR [**104**]		SR-CNN [**34**]	
Noise Variance	0	0.001	0	0.001	0	0.001	0	0.001	0	0.001	0	0.001
Scene 2	45.53	32.68	47.78	37.51	46.57	33.31	46.66	32.85	47.16	32.91	46.67	36.13
	0.9806	0.7085	0.9875	0.8920	0.9843	0.7377	0.9833	0.7144	0.9854	0.7180	0.9835	0.8499
Scene 8	35.01	31.60	36.80	33.49	37.11	32.01	37.22	31.70	37.17	31.72	38.29	32.75
	0.9424	0.7462	0.9645	0.8708	0.9640	0.7728	0.9643	0.7550	0.9655	0.7578	0.9686	0.8401
Scene 18	41.65	32.38	44.72	36.68	44.22	33.18	44.53	32.74	44.89	32.80	44.65	35.48
	0.9780	0.7168	0.9902	0.8949	0.9876	0.7530	0.9874	0.7311	0.9892	0.7345	0.9873	0.8546
Scene 25	43.94	32.47	47.60	37.21	46.05	33.31	47.26	32.93	47.87	32.97	46.87	35.96
	0.9917	0.7246	0.9971	0.9090	0.9952	0.7628	0.9947	0.7416	0.9966	0.7447	0.9935	0.8698
Scene 33	35.94	31.97	39.65	34.62	40.46	32.85	41.21	32.58	41.36	32.60	40.88	33.27
	0.9606	0.7941	0.9839	0.9054	0.9809	0.8237	0.9830	0.8093	0.9842	0.8113	0.9834	0.8792
Scene 45	44.60	33.26	46.99	37.55	46.24	33.87	46.63	33.50	47.08	33.55	46.57	36.44
	0.9850	0.7584	0.9912	0.9092	0.9889	0.7808	0.9885	0.7613	0.9904	0.7647	0.9887	0.8768
Scene 48	34.96	31.62	36.88	33.47	36.74	31.98	37.25	31.84	37.38	31.89	**38.39**	32.77
	0.9660	0.7921	0.9788	0.9059	0.9764	0.8165	0.9782	0.8024	0.9796	0.8048	0.9809	0.8806
Average	40.23	32.28	42.92	35.79	42.48	32.93	42.97	32.59	43.27	32.63	43.19	34.69
	0.9720	0.7487	0.9847	0.8982	0.9825	0.7782	0.9828	0.7593	0.9844	0.7622	0.9837	0.8644
Computation time	-		15.0 s		75.9 s		13.1 s		1.5 s		2.6 s	

Video	Enhancer [**57**]		Bayesian [**73**]		Bayesian-MB [**77**]		DraftCNN [**71**]		MDMF-B-VT		MDMF-R-VT	
Noise Variance	0	0.001	0	0.001	0	0.001	0	0.001	0	0.001	0	0.001
Scene 2	46.91	33.95	45.34	30.26	43.95	32.56	**47.94**	32.88	**47.94**	**40.55**	47.91	40.13
	0.9849	0.7597	0.9821	0.5983	0.9836	0.7044	**0.9880**	0.7150	0.9877	**0.9539**	0.9875	0.9414
Scene 8	36.68	32.23	36.25	29.08	36.72	31.35	37.43	31.70	**38.80**	**35.61**	38.59	35.10
	0.9632	0.7875	0.9645	0.6528	0.9632	0.7448	0.9671	0.7556	0.9733	**0.9314**	**0.9734**	0.9209
Scene 18	45.41	33.77	42.69	29.74	42.17	32.37	45.80	32.80	**46.68**	39.47	46.39	38.95
	0.9909	0.7743	0.9879	0.5981	0.9859	0.7188	0.9919	0.7321	**0.9929**	0.9581	0.9926	0.9457
Scene 25	46.53	33.92	44.38	29.73	42.96	32.47	46.41	32.88	49.40	**39.99**	**50.13**	39.85
	0.9952	0.7828	0.9953	0.6112	0.9926	0.7284	0.9962	0.7395	0.9978	**0.9690**	**0.9981**	0.9584
Scene 33	39.53	32.97	38.29	29.40	38.26	31.73	39.03	32.08	**42.79**	**36.39**	41.19	34.77
	0.9832	0.8312	0.9792	0.7108	0.9782	0.7903	0.9846	0.7972	**0.9896**	**0.9488**	0.9870	0.9293
Scene 45	46.08	34.58	44.26	31.28	44.76	33.23	47.35	33.45	47.61	**40.10**	**47.76**	39.67
	0.9884	0.8075	0.9868	0.6645	0.9878	0.7531	0.9913	0.7603	**0.9918**	**0.9564**	**0.9918**	0.9480
Scene 48	35.95	32.18	35.67	29.23	35.39	30.95	36.34	31.52	38.10	**34.44**	37.35	32.93
	0.9692	0.8278	0.9755	0.7141	0.9637	0.7779	0.9745	0.7960	**0.9820**	**0.9416**	0.9784	0.9170
Average	42.44	33.37	40.98	29.82	40.60	32.09	42.90	32.47	**44.47**	38.08	44.19	37.34
	0.9821	0.7958	0.9816	0.6500	0.9793	0.7454	0.9848	0.7565	**0.9879**	0.9513	0.9870	0.9372
Computation time	1.7 s		114.2 s		76.0 s		2165.7 s		159.3 s		115.5 s	

Table 3.5. PSNR values (in dB, top) and SSIM values (bottom) of experimental results comparing our proposed methods with the state-of-the-art methods (best results are shown in bold) under different noise conditions.

3.5. Conclusion

In this paper we presented two novel video SR frameworks, the batch approach and the recursive approach, based on dictionary learning and motion estimation. According to them, the HR patches are estimated from multiple corresponding LR patches or previously super-resolved HR patches in multiple frames, making the dictionary-based reconstruction algorithm more accurate. The dictionary training algorithms that utilize multiple frames of the training videos further improved the SR performance by making the training and testing phases consistent. We performed experiments with 4K videos and showed that our methods outperform the state-of-the-art algorithms, based either on quantitative analysis or visual comparison.

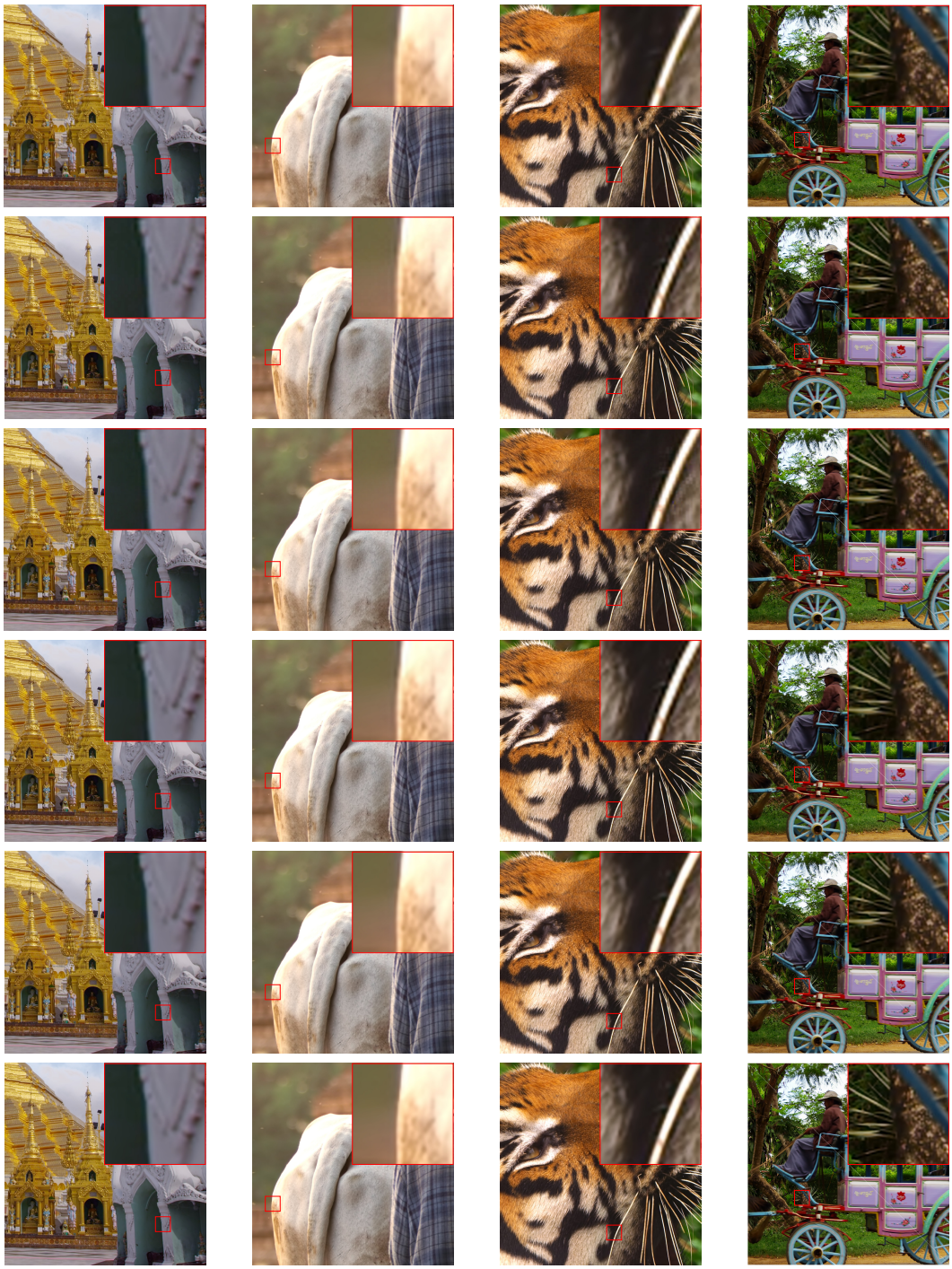


Figure 3.9. Visual Comparison of SR results. Column left to right is scene 8, scene 25, scene 33 and scene 48, respectively. Row top to bottom is Bicubic, Bilevel [**117**, **118**], Enhancer [**57**], Bayesian [**73**], proposed MDMF-B-VT and proposed MDMF-R-VT, respectively. Our proposed algorithms can generate natural-looking frames without noticeable visual artifacts. Because the testing frames have high resolution, results are better viewed in zoomed PDF.

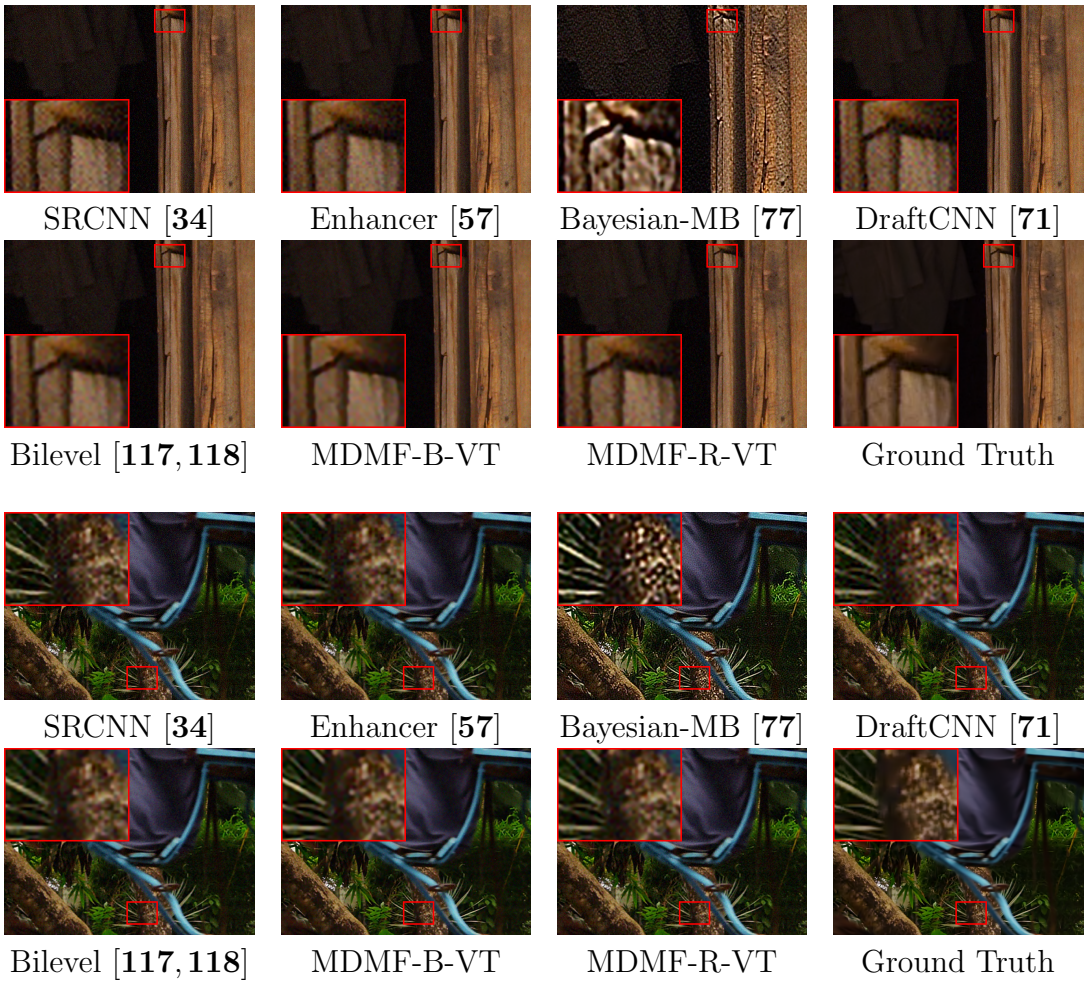


Figure 3.10. Visual Comparison of SR results of different SR methods when Gaussian noise variance equals to 0.001. Our proposed algorithms suppress the noise and generate the closest HR frames to the Ground Truth HR frames. Readers are suggested to zoom in to see the details.

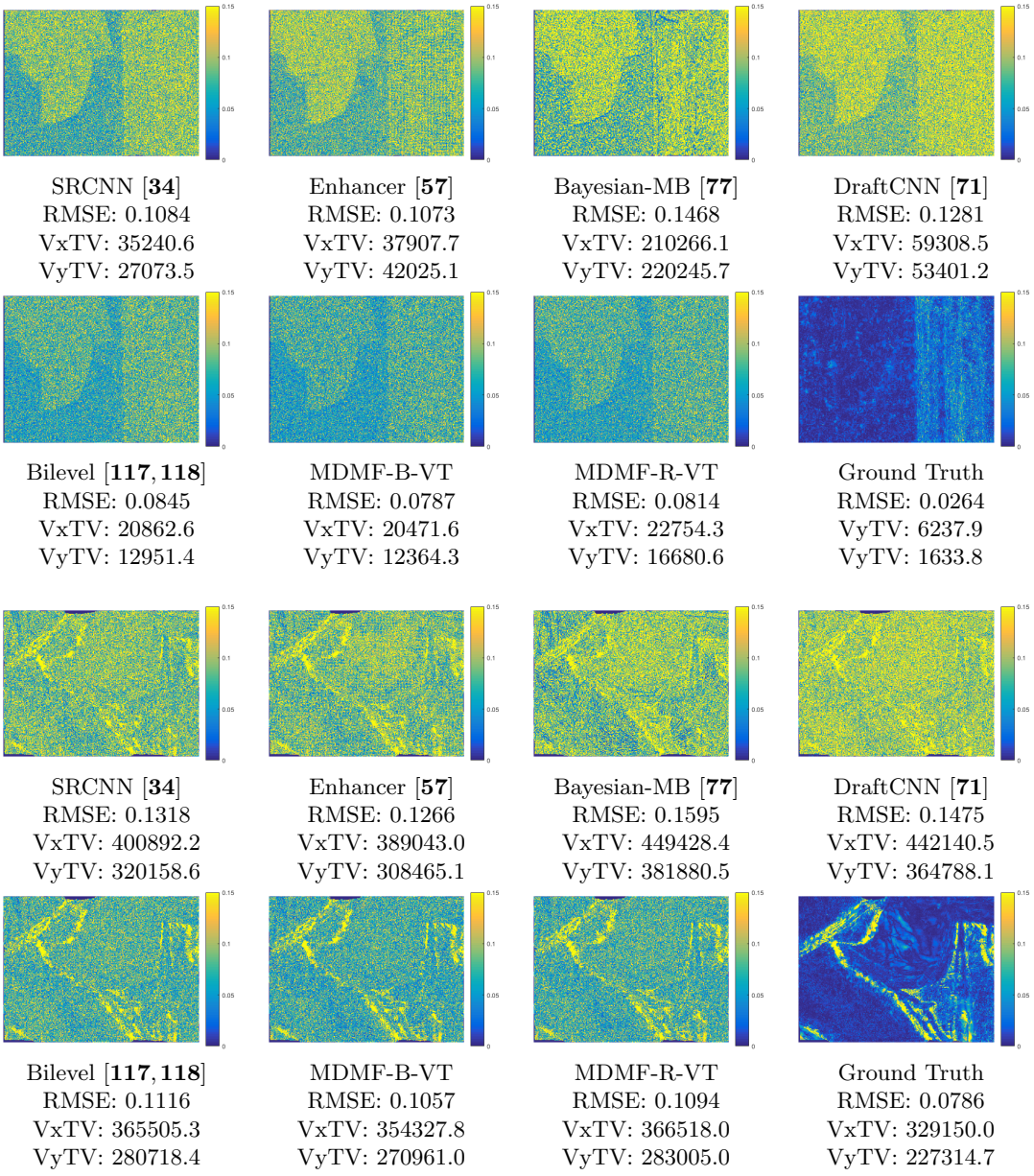


Figure 3.11. Visual Comparison of the motion compensation error of the SR results by different SR methods when Gaussian noise variance equals to 0.001. Our proposed algorithms have the smallest motion compensation error from both the error head map and RMSE metric, illustrating the advantages in temporal smoothness of the super-resolved frames.

CHAPTER 4

KKT Condition Refined Deep ℓ_1 Encoders

4.1. Introduction

In this chapter, we propose a KKT condition refined Learned ISTA (KKT-LISTA) framework. First we utilize the LISTA network to have an initial estimate of the position (support) and sign of the non-zero coefficients. The support is then refined by nearest neighbor retrieval from the support bank computed by the original ISTA algorithm. Finally the KKT condition with the known support is utilized to obtain accurate sparse coefficients. The additional computation for support retrieval and KKT refinement is acceptable so we still hold a computation advantage compared to those iterative optimization algorithms [**11**, **31**, **49**, **68**, **69**, **79**, **107**, **115**]. In-depth and comprehensive experiments prove that our proposed KKT-LISTA outperforms the original LISTA in both optimization accuracy and applicability.

4.2. Neural Network Implementation of Sparse Coding

ISTA [**11**, **31**, **107**] is a popular algorithm for sparse code inference. To solve the problem in Equation (1.4), ISTA performs the following iteration until convergence

$$z(k+1) = h_\theta(Wx + Sz(k)) \qquad z(0) = \mathbf{0}, \tag{4.1}$$

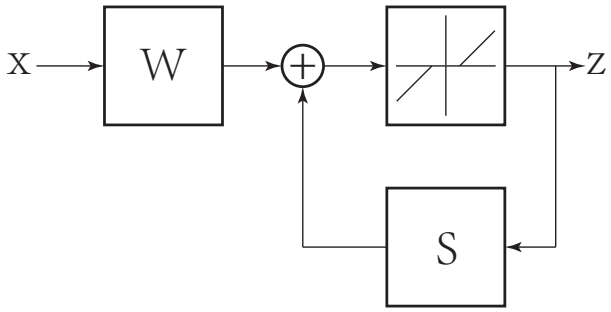


Figure 4.1. Block diagram of the ISTA algorithm for sparse coding.

where $W = \frac{1}{L}D^T$, L is an upper bound on the largest eigenvalue of D^TD, $S = I - \frac{1}{L}D^TD$, and the shrinkage function h_θ is defined as $[h_\theta(Y)]_i = sign(Y_i)(|Y_i| - \theta_i)+$. In standard ISTA, all thresholds are set to $\theta_i = \lambda/L$. This iteration process is illustrated in Figure (4.1).

In [**48**], the iteration process of ISTA is unfolded into a feed-forward neural network, called LISTA, with finite layers. As demonstrated in Figure (4.2), LISTA was proposed to efficiently approximate the sparse coefficients z of the input signal x as it would be estimated by solving Equation (1.4) for a given dictionary D. The network has a finite number of recurrent stages, each of which updates the intermediate sparse code according to Equation (4.1). Different from the ISTA algorithm, the network parameters W, S and thresholds θ are learned from training data using a back-propagation algorithm, instead of directly determined by D and λ. In the training phase, the following energy function is minimized

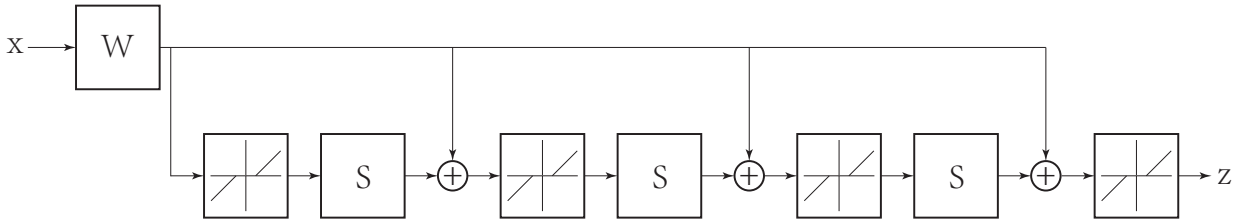


Figure 4.2. LISTA diagram.

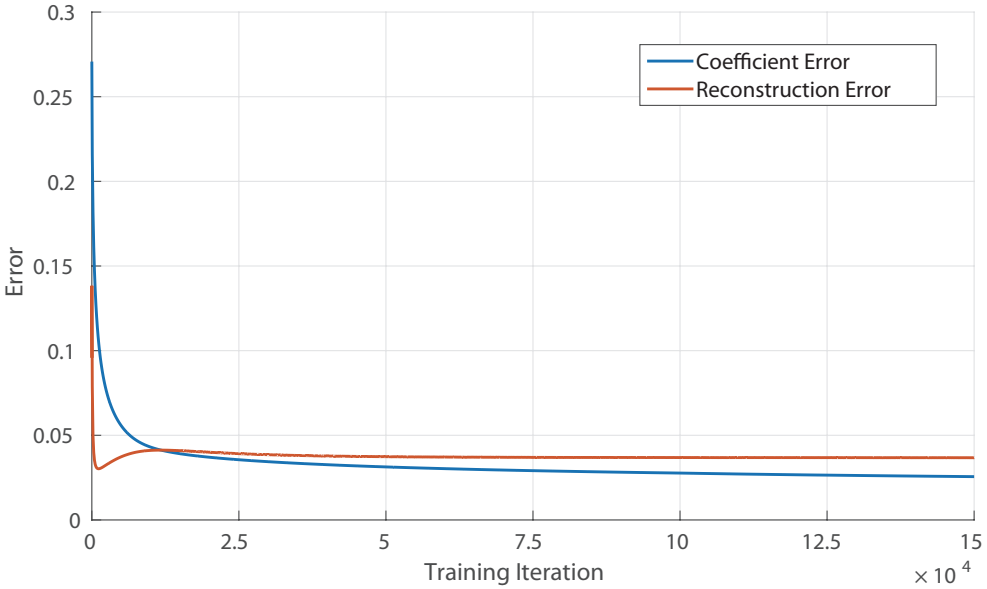


Figure 4.3. LISTA diagram.

$$\frac{1}{N}\sum_{j=1}^{N}\|L(x_j, W, S, \theta) - z_j^{gt}\|_2^2 \tag{4.2}$$

where x_j is the j^{th} training signal, z_j^{gt} is its corresponding sparse coefficients solved by the ISTA algorithm, $L(x_j, W, S, \theta)$ is the LISTA network function, and N is the total number of training signals. In the testing phase, an approximation of the sparse coefficients can be obtained with a fixed number of recurrent stages.

However, although $z_j^{net} = L(x_j, W, S, \theta)$ is a moderate approximation to z_j^{gt}, the reconstruction accuracy ($\frac{1}{N}\sum_{j=1}^{N}\|Dz_j^{net} - x_j\|_2^2$) is usually low. As shown in Figure (4.3), as iteration increases during training, the reconstruction error is not minimized even though the coefficient error $\frac{1}{N}\sum_{j=1}^{N}\|z_j^{net} - z_j^{gt}\|_2^2$ is decreasing. So for tasks such as image restoration, denoising and compression, where the reconstruction accuracy is important, the performance of LISTA networks will not be satisfactory.

4.3. KKT Condition Refined Deep ℓ_1 Encoders

Since the origin LISTA network will not provide accurate sparse coefficients, we propose to utilize the KKT condition of the original problem (Equation (1.4)) to refine the sparse coefficients. The KKT conditions could be obtained by taking the derivative of Equation (1.4) with respect to z. This tell us that any SC solution $\hat{z}$ must satisfy

$$D^T(x - D\hat{z}) = \lambda s, \tag{4.3}$$

where $s = \partial \|\hat{z}\|_1$, a subgradient of the ℓ_1 norm evaluated at $\hat{z}$, which equals

$$s_i = \begin{cases} +1, & \hat{z}_i > 0 \\ -1, & \hat{z}_i < 0 \\ [-1, +1], & \hat{z}_i = 0 \end{cases} \qquad i = 1, ..., m, \tag{4.4}$$

where $\hat{z}_i$ is the i^{th} element of $\hat{z}$.

We then define the non-zero coefficients set, which is the set of index of the non-zero coefficients of $\hat{z}$, that is,

$$E = \{i \in \{1, ..., n\} : \hat{z}_i \neq 0\}. \tag{4.5}$$

If D_E, the part of the dictionary corresponding to the set E, has full column rank (i.e., the basis vectors corresponding to the non-zero coefficients set are linearly independent), there is an unique SC solution satisfying

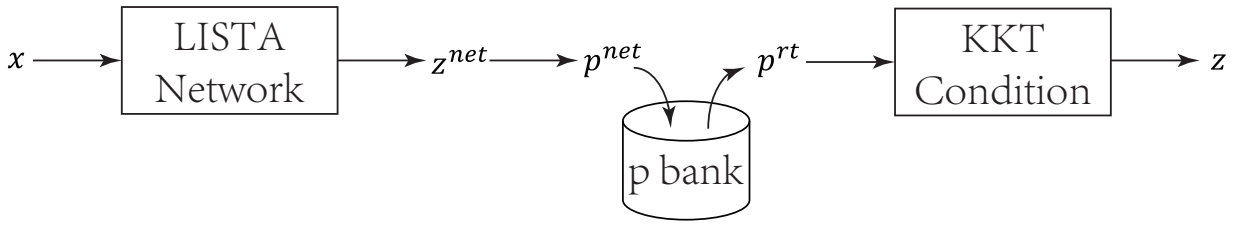


Figure 4.4. Block diagram of the proposed KKT-LISTA.

$$\begin{aligned} \hat{z}_E &= (D_E^T D_E)^{-1}(D_E^T x - \lambda s_E), \\ \hat{z}_{-E} &= 0. \end{aligned} \tag{4.6}$$

If we know the non-zero coefficients set E (position of non-zero coefficients) and s_E (sign of non-zero coefficients), the sparse coefficients $\hat{z}$ (both $\hat{z}_E$ and $\hat{z}_{-E}$) can be solved in closed form by Equation (4.6). The column dimension of matrix D_E equals to the number of non-zero coefficients, which is usually small, so the computation of $(D_E^T D_E)^{-1}$ is fast. Here we define an $m \times 1$ support indicator vector p to incorporate both the position and sign of the non-zero coefficients of z

$$p_i = \begin{cases} +1, & z_i > 0 \\ -1, & z_i < 0 \\ 0, & z_i = 0 \end{cases} \qquad i = 1, ..., m. \tag{4.7}$$

Although the sparse coefficients z^{net} estimated by the LISTA network usually do not satisfy the KKT condition (Equation (4.6)), they provide a moderate estimation of the position of non-zero coefficients and their signs, in other words, p^{net} computed from z^{net} is close to p^{gt} computed from z^{gt}. The algorithm that directly applies the position and sign of the non-zero coefficients of z^{net} to the KKT condition is named R-KKT-LISTA.

The closed form solution (Equation (4.6)) by the KKT condition depends entirely on the support indicator vector p, so a better estimation of p will result in a better solution. p^{net} estimated by the LISTA network can be improved by a simple retrieval step. Let $p_j^{gt}, j = 1, ..., M$ (p bank) be the support indicator vector computed from M total sparse coefficients $z_j^{gt}, j = 1, ..., M$. During retrieval, Locality-Sensitive Hashing (LSH) [**6**] can be applied to find the closet p^{rt} vector to p^{net} in Hamming distance from the p bank in $O(1)$ time. The retrieved p^{rt} outperforms p^{net} because it is computed by the original ISTA algorithm [**11**], instead of the fast approximation LISTA [**48**], so a more accurate solution can be computed by the KKT condition (Equation (4.6)).

The proposed pipeline of KKT-LISTA is summarized in Figure (4.4); given an input signal x, the LISTA network is first applied to obtain the initial estimation of the sparse coefficient z^{net}. Then the support indicator vector p^{net} is computed, and the nearest neighbor of p^{net} among the p bank is retrieved. Finally the KKT condition is applied to compute the exact sparse coefficients z. One should notice that we are not seeking to produce an approximate sparse code for all possible input signals, but only for input signals which have the same distribution as our training signals. In the LISTA network training and the closest support retrieval step, we are concentrated on the solution of restricted problem of interest, not the general problem. However, by collecting large enough training data, which is consistent to the testing data, good sparse coefficients inference performance can be obtained.

4.4. Experimental Results

In this section, the methods ISTA, LISTA, R-KKT-LISTA and the proposed KKT-LISTA are compared on the original ℓ_1 sparse coding problem, as well as the image compression problem.

4.4.1. ℓ_1 sparse coding problem

	$m = 25$			
Methods	$\|x - Dz\|_2^2$	$\|z\|_1$	$\|x - Dz\|_2^2 + \lambda\|z\|_1$	time(s)
LISTA	0.0447 (0.0154)	1.26 (-0.04)	0.120 (0.013)	0.09
R-KKT-LISTA	0.0504 (0.0211)	4.65 (3.35)	0.330 (0.223)	0.17
KKT-LISTA	0.0332 (0.0039)	1.46 (0.16)	0.119 (0.012)	0.19
ISTA	0.0293	1.30	0.107	6.52
	$m = 50$			
Methods	$\|x - Dz\|_2^2$	$\|z\|_1$	$\|x - Dz\|_2^2 + \lambda\|z\|_1$	time(s)
LISTA	0.0433 (0.0231)	1.17 (-0.07)	0.113 (0.019)	0.09
R-KKT-LISTA	0.0289 (0.0087)	3.28 (2.04)	0.226 (0.132)	0.20
KKT-LISTA	0.0265 (0.0063)	1.39 (0.15)	0.110 (0.016)	0.23
ISTA	0.0202	1.24	0.094	7.09
	$m = 100$			
Methods	$\|x - Dz\|_2^2$	$\|z\|_1$	$\|x - Dz\|_2^2 + \lambda\|z\|_1$	time(s)
LISTA	0.0449 (0.0296)	1.09 (-0.09)	0.110 (0.024)	0.11
R-KKT-LISTA	0.0201 (0.0048)	1.67 (0.49)	0.120 (0.034)	0.26
KKT-LISTA	0.0241 (0.0088)	1.34 (0.16)	0.104 (0.018)	0.28
ISTA	0.0153	1.18	0.086	7.68

Table 4.1. Experimental result comparing different methods in reconstruction error, ℓ_1 error and overall error.

In this experiment we compare the performance of different methods on solving the original ℓ_1 sparse coding problem. The training data-set consists of 100,000 image patches of size 5×5 pixels, randomly selected from the Myanmar 4K Footage Database [**56**]. The testing data-set consists of 1,000 different image patches sampled from the same database. The patches with small standard deviation were discarded. $\lambda = 0.06$ is used in Equation (1.4).

In the training phase, the dictionary is first trained by the method in [**79**]. We considered three cases, one with $m = 25$ (complete dictionary), another with $m = 50$ (2 times

(a). 31.91 dB in PSNR. (b). 32.94 dB in PSNR.

Figure 4.5. Reconstruction of LISTA (a) and KKT-LISTA (b) encoded images.

over-complete dictionary) and the third with $m = 100$ (4 times over-complete dictionary). After the dictionaries are trained, the ground truth sparse coefficient z^{gt} is solved by ISTA. The three layer LISTA networks are trained according to the back propagation process as illustrated in [**48**] until convergence. The p bank is composed of 100,000 p^{gt} vectors computed from z^{gt}. The LSH data structure is created by the E2LSH [**6**] method for fast nearest neighbor retrieval. We found experimentally that larger p bank size than 100,000 has little benefits to performance while consumes more memory on the LSH data structure.

In the testing phase, z is estimated according to our proposed framework in Figure (4.4). We also compute the sparse coefficients estimate by the R-KKT-LISTA method (without the support indicator vector refinement) for comparison.

The statistics of the mean reconstruction error ($\|x - Dz\|_2^2$), the mean ℓ_1 sparsity penalty ($\|z\|_1$), the mean overall energy ($\|x - Dz\|_2^2 + \lambda\|z\|_1$) and computation time over 1,000 testing signals are shown in Table (4.1). The difference between those approximation methods and the ground truth ISTA is shown in brackets. Compared to LISTA, our proposed KKT-LISTA

significantly decreases the reconstruction error $\|x - Dz\|_2^2$, which is important for such tasks as signal restoration. The overall energy is also decreased, by comparing the difference between LISTA and ISTA to the difference between KKT-LISTA and ISTA, the difference decreases from 0.013 to 0.012 (7.7%) when $m = 25$, from 0.019 to 0.016 (15.8%) when $m = 50$, from 0.024 to 0.018 (25.0%) when $m = 100$, respectively. By examining the results of R-KKT-LISTA, we found out that without the support indicator vector retrieval step, the application of the KKT condition refinement will not have smaller overall error compared to LISTA. As for the computation time, our proposed KKT-LISTA approximately doubles the computation time of LISTA, because of the additional p bank retrieval step and the computation of Equation (4.6). However, it is still much faster than ISTA by approximately 30 times.

4.4.2. Application to Image Compression

Compression of still images is an active and matured field of research. By SC, the original signal can be represented efficiently by the sparse coefficients under some given basis. By storing the non-zero coefficients, the original signal can be compressed. The proposed KKT-LISTA sparse coefficients inference algorithm can be applied to speed up the encoding phase. In this experiment, our proposed KKT-LISTA is compared with LISTA in terms of the quality of the reconstructed image.

In the encoding phase, the image is broken into 5×5 patches with 1 pixel overlap with the adjacent patches. For each patch, the sparse coefficients are inferred by the proposed KKT-LISTA, as well as LISTA for comparison. The KKT-LISTA framework and LISTA networks are the same as the ones in Section 4.4.1 with $m = 100$. In the decoding phase, the

dictionary trained in Section 4.4.1 for $m = 100$ is applied to reconstruct each image patch, $x = Dz$. The overlapping pixels are averaged to produce the final results.

As shown in Figure 4.5(a) and Figure 4.5(b), the proposed KKT-LISTA outperforms LISTA in terms of both visual quality and quantitative PSNR (Peak Signal to Noise Ratio).

4.5. Conclusion

In this paper, we present a KKT condition refined LISTA framework to solve the ℓ_1-based sparse approximation problem. The support of the sparse coefficients is initially estimated by the LISTA network and refined by nearest neighbor retrieval. Finally the KKT condition is applied to solve the accurate sparse coefficients. Experimental results show that our proposed framework results in both smaller reconstruction error and overall energy for the ℓ_1-based sparse approximation problem.

CHAPTER 5

Spatial-Spectral Representation for X-Ray Fluorescence Image Super-Resolution

5.1. Introduction

In this chapter, we propose a super-resolution (SR) approach to obtain high resolution (HR) XRF images, with the aid of a conventional HR RGB image, as shown in Fig. 1.3. Our proposed XRF image SR algorithm can also be applied to spectral images obtained by any other raster scanning methods, such as Scanning Electron Microscope (SEM), Energy Dispersive Spectroscopy (EDS) and Wavelength Dispersive Spectroscopy (WDS). We model the spectrum of each pixel using a linear mixing model [**80**, **87**]. Since there is no direct one-to-one mapping between the visible RGB spectrum and the XRF spectrum, because the hidden part of the paining is not visible in the conventional RGB image, but it can be captured in the XRF image [**3**], we model the XRF signal as a combination of the visible signal (on the surface) and the non-visible signal (hidden under surface), as shown in Fig. 5.1. For super-resolving the visible XRF signal we follow a similar approach to previous research in [**1**, **2**, **35**, **50**, **62**, **67**, **112**]. A coupled XRF-RGB dictionary is learned to explore the correlation between XRF and RGB signals. The RGB dictionary is applied to obtain the sparse representation of the HR RGB input image, resulting in an HR coefficient map. Then the XRF dictionary is applied on the HR coefficient map to reconstruct the HR XRF image. For the non-visible part, we increase its spatial resolution using a standard total variation regularizer [**8**, **81**]. Finally, the HR visible and the HR non-visible XRF signals

are combined to obtain the final HR XRF result. We do not explicitly separate the input LR XRF image into visible and non-visible parts in advance. Instead, we formulate the whole SR problem as an optimization problem. By alternatively optimizing over the coupled XRF-RGB dictionary and the visible / non-visible HR coefficient maps, the fidelity of the estimated HR output to both the LR XRF and HR RGB input signals is improved, thus resulting in a better SR output. Both synthetic and real experiments show the effectiveness of our proposed method, in terms of reconstruction error and visual sharpness of the SR result, compared to other methods, such as bicubic interpolation, the total variation only SR method [**8**, **81**] and hyperspectral image SR methods [**1**, **35**, **67**].

The paper is organized as follows. We formulate the XRF image SR problem in Section 5.2, while the proposed method is described in Section 5.3. In Section 5.4, we provide the experimental results with both synthetic data and real data to evaluate the approach. The paper is concluded in Section 5.5.

5.2. Problem Formulation

As shown in Fig. 5.1, we are seeking the estimation of an HR XRF image $\bar{Y} \in \mathbb{R}^{W \times H \times B}$ that has both high spatial and high spectral resolution, with W, H and B the image width, image height and number of spectral bands, respectively. We have two inputs: an LR XRF image $\bar{X} \in \mathbb{R}^{w \times h \times B}$ with lower spatial resolution $w \times h$, $w \ll W$ and $h \ll H$; and a conventional HR RGB image $\bar{I} \in \mathbb{R}^{W \times H \times b}$ with high spatial resolution, but a small number of spectral bands, $b \ll B$. The input LR XRF image $\bar{X}$ can be separated into two parts: the visible component $\bar{X}_v \in \mathbb{R}^{w \times h \times B}$ and the non-visible component $\bar{X}_{nv} \in \mathbb{R}^{w \times h \times B}$. We propose to estimate the HR visible component $\bar{Y}_v \in \mathbb{R}^{W \times H \times B}$ by fusing the conventional HR RGB

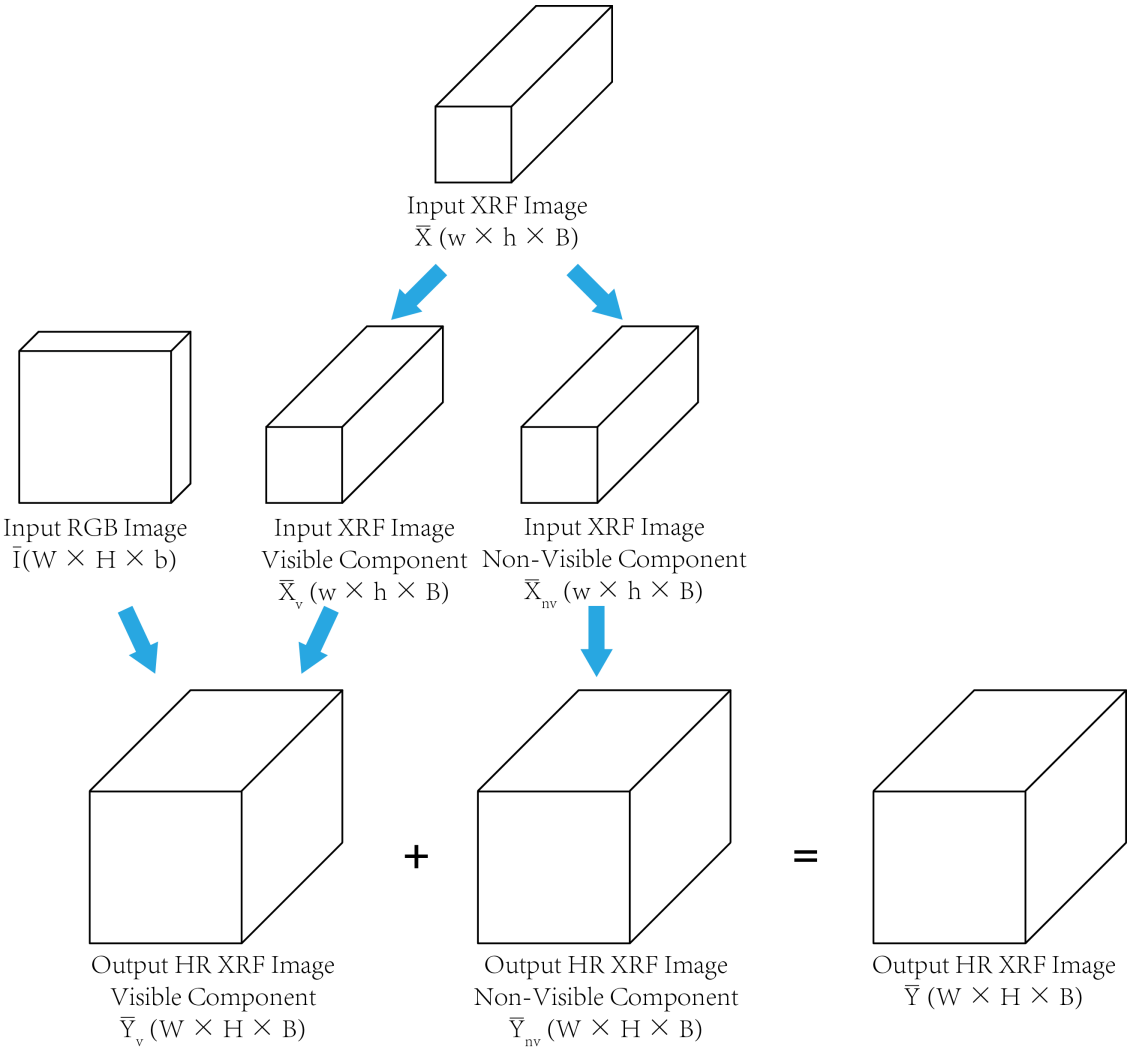


Figure 5.1. Proposed pipeline of spatial-spectral representation for X-ray fluorescence image super-resolution. The visible component of input XRF image is fused with the input RGB image to obtain the visible component of HR XRF image. The non-visible component of the input XRF image is super-resolved to obtain the non-visible component of HR XRF image. The HR visible and non-visible component of output XRF image are combined to obtain the final output.

input image $\bar{I}$ with the visible component of the input LR XRF image $\bar{X}_v$ and estimate the HR non-visible component $\bar{Y}_{nv} \in \mathbb{R}^{W \times H \times B}$ by using standard total variation SR methods.

To simplify notation, the images cubes are written as matrices, i.e. all pixels of an image are concatenated, such that every column of the matrix corresponds to the spectral response at a given pixel, and every row corresponds to the lexicographically ordered image in a specific spectral band. Accordingly, the image cubes are written as $Y \in \mathbb{R}^{B\times N_h}$, $X \in \mathbb{R}^{B\times N_l}$, $I \in \mathbb{R}^{b\times N_h}$, $X_v \in \mathbb{R}^{B\times N_l}$, $X_{nv} \in \mathbb{R}^{B\times N_l}$, $Y_v \in \mathbb{R}^{B\times N_h}$ and $Y_{nv} \in \mathbb{R}^{B\times N_h}$, where $N_h = W \times H$ and $N_l = w \times h$. We therefore have

$$X = X_v + X_{nv}, \tag{5.1}$$

$$Y = Y_v + Y_{nv}, \tag{5.2}$$

according to the visible / non-visible component separation model as shown in Fig. 5.1.

Let us denote by $y_v \in \mathbb{R}^B$ and $y_{nv} \in \mathbb{R}^B$ the one-dimensional spectra at a given spatial location of $\bar{Y}_v$ and $\bar{Y}_{nv}$, that is, representing a column of Y_v and Y_{nv}, according to the linear mixing model [**16**, **63**], they can be described as

$$y_v = \sum_{j=1}^{M} d_{v,j}^{xrf}\alpha_{v,j}, \quad Y_v = D_v^{xrf}A_v, \tag{5.3}$$

$$y_{nv} = \sum_{j=1}^{M} d_{nv,j}^{xrf}\alpha_{nv,j}, \quad Y_{nv} = D_{nv}^{xrf}A_{nv}, \tag{5.4}$$

where $d_{v,j}^{xrf}$ and $d_{nv,j}^{xrf}$ represent respectively the endmembers for the visible and non-visible components, then $D_v^{xrf} \equiv [d_{v\ 1}^{xrf}, d_{v\ 2}^{xrf}, \ldots, d_{v\ M}^{xrf}] \in \mathbb{R}^{B\times M}$, $D_{nv}^{xrf} \equiv [d_{nv\ 1}^{xrf}, d_{nv\ 2}^{xrf}, \ldots, d_{nv\ M}^{xrf}] \in \mathbb{R}^{B\times M}$. $\alpha_{v,j}$ and $\alpha_{nv,j}$ are the corresponding per-pixel abundances. Equation 5.3 holds for a specific column y_v of matrix Y_v (say the k^{th} column). We take the corresponding $\alpha_{v,j,j=1,\ldots,M}$

and stack them into a $M \times 1$ column vector, this vector then becomes the k^{th} column of the matrix $A_v \in \mathbb{R}^{M \times N_h}$. In a similar manner we construct matrix $A_{nv} \in \mathbb{R}^{M \times N_h}$. The endmembers D_v^{xrf} and D_{nv}^{xrf} act as a basis dictionary to represent Y_v and Y_{nv} in a lower-dimensional space $\mathbb{R}^M$ and $rank\{Y_v\} \leq M,\ rank\{Y_{nv}\} \leq M$.

The visible and non-visible components of the input LR XRF image X_v and X_{nv}, respectively, are a spatially downsampled version of Y_v and Y_{nv}, respectively, that is

$$X_v = Y_v S = D_v^{xrf} A_v S, \tag{5.5}$$

$$X_{nv} = Y_{nv} S = D_{nv}^{xrf} A_{nv} S, \tag{5.6}$$

where $S \in \mathbb{R}^{N_h \times N_l}$ is the downsampling operator that describes the spatial degradation from HR to LR.

Similarly, the HR conventional RGB image I can be described by the linear mixing model [**16**, **63**],

$$I = D^{rgb} A_v, \tag{5.7}$$

where $D^{rgb} \in \mathbb{R}^{b \times M}$ is the RGB dictionary. Notice that the same abundances matrix A_v is used in Equations 5.3 and 5.5. This is because the visible component of the scanning object is captured by both the XRF and the conventional RGB images. The matrix A_v encompasses the spectral correlation between the visible component of the XRF and the conventional RGB images.

The physically grounded constraints in [**67**] are shown to be effective, so we propose to impose similar constraints, by making full use of the fact that the XRF endmembers are XRF

spectra of individual materials, and the abundances are proportions of those endmembers. Consequently, they are subject to the following constraints:

$$0 \leq D_{v,ij}^{xrf} \leq 1, \forall i, j \tag{5.8a}$$

$$0 \leq D_{nv,ij}^{xrf} \leq 1, \forall i, j \tag{5.8b}$$

$$0 \leq D_{ij}^{rgb} \leq 1, \forall i, j \tag{5.8c}$$

$$A_{v,ij} \geq 0, \forall i, j \tag{5.8d}$$

$$A_{nv,ij} \geq 0, \forall i, j \tag{5.8e}$$

$$\mathbf{1^T}(A_v + A_{nv}) = \mathbf{1^T}, \tag{5.8f}$$

where $D_{v,ij}^{xrf}$, $D_{nv,ij}^{xrf}$, D_{ij}^{rgb}, $A_{v,ij}$ and $A_{nv,ij}$ are the (i, j) elements of matrices D_v^{xrf}, D_{nv}^{xrf}, D^{rgb}, A_v and A_{nv}, respectively, $\mathbf{1^T}$ demotes a row vector of 1's compatible with the dimensions of A_v and A_{nv}. Equations 5.8a, 5.8b and 5.8c enforce the non-negative, bounded spectrum constraints on endmembers, Equations 3.6d and 3.6e, enforce the non-negative constraints on abundances, and Equation 5.8e enforces the visible component abundances and non-visible component abundances for every pixel to sum up to one.

5.3. Proposed Solution

In order to solve the XRF image SR problem, we need to estimate A_v, A_{nv}, D^{rgb}, D_v^{xrf} and D_{nv}^{xrf} simultaneously. Utilizing Equations 5.1, 5.5, 5.6, 5.7 and 5.8, we can form the following constrained least-squares problem:

$$\min_{\substack{A_v, A_{nv}, D^{rgb}, \\ D_v^{xrf}, D_{nv}^{xrf}}} \|X - D_v^{xrf} A_v S - D_{nv}^{xrf} A_{nv} S\|_F^2 \tag{5.9a}$$

$$+ \|I - D^{rgb} A_v\|_F^2 + \lambda \|\nabla(D_{nv}^{xrf} A_{nv})\|_F^2$$

$$\text{s.t.} \quad 0 \leq D_{v\ ij}^{xrf} \leq 1, \forall i, j \tag{5.9b}$$

$$0 \leq D_{nv\ ij}^{xrf} \leq 1, \forall i, j \tag{5.9c}$$

$$0 \leq D_{ij}^{rgb} \leq 1, \forall i, j \tag{5.9d}$$

$$A_{v\ ij} \geq 0, \forall i, j \tag{5.9e}$$

$$A_{nv\ ij} \geq 0, \forall i, j \tag{5.9f}$$

$$\mathbf{1^T}(A_v + A_{nv}) = \mathbf{1^T}, \tag{5.9g}$$

$$\|A_v + A_{nv}\|_0 \leq s, \tag{5.9h}$$

with $\|\cdot\|_F$ denoting the Frobenius norm, and $\|\cdot\|_0$ the ℓ_0 norm, i.e., the number of non-zero elements of the given matrix. The first term in Equation 5.9a represents a measure of the fidelity of the observed XRF data X, the second term the fidelity to the observed RGB image I and the third term in Equation 5.9a is the total variation (TV) regularizer. It is defined as

$$\begin{aligned}
&\|\nabla(D_{nv}^{xrf} A_{nv})\|_F^2 \\
&= \sum_{i=1}^{H-1} \sum_{j=1}^{W-1} \|D_{nv}^{xrf} \bar{A}_{nv}(i, j, :) - D_{nv}^{xrf} \bar{A}_{nv}(i+1, j, :)\|_2^2 \\
&\qquad + \|D_{nv}^{xrf} \bar{A}_{nv}(i, j, :) - D_{nv}^{xrf} \bar{A}_{nv}(i, j+1, :)\|_2^2 \\
&= \|D_{nv}^{xrf} A_{nv} G\|_F^2
\end{aligned} \tag{5.10}$$

where $\bar{A}_{nv} \in \mathbb{R}^{W \times H \times M}$ is the $3D$ volume version of A_{nv} and $\bar{A}_{nv}(i,j,:) \in \mathbb{R}^{M}$ is the non-visible component abundance of pixel (i,j). $G \in \mathbb{R}^{N_h \times ((W-1)(H-1))}$ is the horizontal/vertical first order difference operator. To estimate the HR visible component abundance A_v, the HR RGB image I can provide spatial details. However, to estimate the HR non-visible component abundance A_{nv}, there is no additional spatial information, so we need the TV regularizer (Equation 5.10) to impose spatial smoothness on the non-visible component. The TV regularizer parameter λ controls the spatial smoothness of the reconstructed non-visible component, $Y_{nv} = D_{nv}^{xrf} A_{nv}$.

The constraint Equations 5.9e, 5.9f, 5.9g together restrict the abundances of visible and non-visible components, and also act as a sparsity prior on the per-pixel abundances, since they bound the ℓ_1 norm of the combined abundances $(A_v + A_{nv})$ to be 1 for all pixels. The last constraint Equation 5.9h is an optional constraint, which further enforces the sparsity of the combined abundance $(A_v + A_{nv})$.

The optimization in Equation 5.9 is non-convex and difficult to solve if we optimize over all the parameters A_v, A_{nv}, D^{rgb}, D_v^{xrf} and D_{nv}^{xrf} directly. We found it effective to alternatively optimize over these parameters. Also because Equation 5.9 is highly non-convex, good initialization is needed to start the local optimization. A similar approach as the coupled dictionary learning technique in [**119**, **120**] is applied here to initialize these parameters.

5.3.1. Initialization

Let $I^l \in \mathbb{R}^{b \times N_l}$ and $A_v^l \in \mathbb{R}^{M \times N_l}$ be the spatially downsampled RGB image I and visible component abundance A_v, we have

$$I^l = IS, \tag{5.11}$$

$$A_v^l = A_v S. \tag{5.12}$$

Then the coupled dictionary learning technique in [**119**, **120**] can be utilized to initialize D^{rgb} and D_v^{xrf} by

$$\begin{aligned} \min_{D^{rgb}, D_v^{xrf}} \quad & \|I^l - D^{rgb} A_v^l\|_F^2 + \|X - D_v^{xrf} A_v^l\|_F^2 \\ & + \beta \sum_{k=1}^{N_l} \|A_v^l(:,k)\|_1, \\ \text{s.t.} \quad & \|D^{rgb}(:,k)\|_2 \leq 1, \forall k, \\ & \|D_v^{xrf}(:,k)\|_2 \leq 1, \forall k, \end{aligned} \tag{5.13}$$

where $\|\cdot\|_1$ is the ℓ_1 vector norm, parameter β control the sparseness of the coefficients in A_v^l $A_v^l(:,k)$, $D^{rgb}(:,k)$ and $D_v^{xrf}(:,k)$ denote the k^{th} column of matrix A_v^l, D^{rgb}, and D_v^{xrf}, respectively. Details of the optimization can be found in [**119**, **120**]. D^{rgb} and D_v^{xrf} are initialized using Equation 5.13 and D_{nv}^{xrf} is initialized to be equal to D_v^{xrf}. A_v is initialized by upsampling A_v^l computed in Equation 5.13, while A_{nv} is set to be zero at initialization.

5.3.2. Optimization Scheme

We propose to alternatively optimize over all the parameters in Equation 5.9a. First we optimize over A_v and A_{nv} by fixing all other parameters,

$$
\begin{aligned}
\min_{A_v, A_{nv}} \quad & \|X - D_v^{xrf} A_v S - D_{nv}^{xrf} A_{nv} S\|_F^2 \\
& + \|I - D^{rgb} A_v\|_F^2 + \lambda \|\nabla (D_{nv}^{xrf} A_{nv})\|_F^2 \\
\text{s.t.} \quad & A_{v\ ij} \geq 0, \forall i, j \\
& A_{nv\ ij} \geq 0, \forall i, j \\
& \mathbf{1^T}(A_v + A_{nv}) = \mathbf{1^T}, \\
& \|A_v + A_{nv}\|_0 \leq s.
\end{aligned}
\tag{5.14}
$$

PALM (proximal alternating linearized minimization) algorithm [**17**] is utilized to optimize over A_v and A_{nv} by a projected gradient descent method. For Equation 5.14, the following three steps are iterated for $q = 1, 2, \ldots$ until convergence:

$$V_v^q = A_v^{q-1} - \frac{1}{d_v} D^{rgb^T} (D^{rgb} A_v^{q-1} - I) \tag{5.15a}$$

$$
\begin{aligned}
V_{nv}^q = & A_{nv}^{q-1} \\
& - \frac{1}{d_{nv}} (D_{nv}^{xrf^T} (D_{nv}^{xrf} A_{nv}^{q-1} S - (X - D_v^{xrf} A_v^{q-1} S)) S^T \\
& + \lambda D_{nv}^{xrf^T} D_{nv}^{xrf} A_{nv} G G^T)
\end{aligned}
\tag{5.15b}
$$

$$\{A_v^q, A_{nv}^q\} = prox_{A_v, A_{nv}}(V_v^q, V_{nv}^q), \tag{5.15c}$$

where $d_1 = \gamma_1 \|D^{rgb} D^{rgb^T}\|_F$, $d_2 = \gamma_2 \|D_{nv}^{xrf} D_{nv}^{xrf^T}\|_F$ are non-zero step size constants, and $prox_{A_v, A_{nv}}$ is the proximal operator that project V_v^q, V_{nv}^q onto the constraints of Equation 5.14. The proximal projection is computational inexpensive because it just sets negative entries of V_v^q and V_{nv}^q to zero and scales every column of V_v^q and V_{nv}^q simultaneously to equal one in ℓ_1 norm. Notice that in Equation 5.15a, only the gradient of the second term in Equation 5.14

is utilized to update V_v^q, because we want the visible component coefficients A_v to be determined by the RGB image I only, instead of being determined jointly by the RGB image I and the XRF image X.

Second, we optimize over D^{rgb} solving the following constrained least-squares problem:

$$\begin{aligned} \min_{D^{rgb}} \quad & \|I - D^{rgb}A_v\|_F^2 \\ \text{s.t.} \quad & 0 \leq D_{ij}^{rgb} \leq 1, \forall i, j. \end{aligned} \tag{5.16}$$

Likewise, Equation 5.16 is minimized by iterating the following steps until convergence:

$$E^q = D^{rgb^{q-1}} - \frac{1}{d_{rgb}}(D^{rgb^{q-1}}A_v - I){A_v}^T \tag{5.17a}$$

$$D^{rgb^q} = prox_{D^{rgb}}(E^q), \tag{5.17b}$$

with $d_{rgb} = \gamma_3\|A_v{A_v}^T\|_F$ again a non-zero step size constant and $prox_{D^{rgb}}$ the proximal operator that projects E^q onto the constraint of Equation 5.16. The proximal operator this time is also computationally inexpensive since it just truncates the entries of E^q to 0 from below and to 1 from above.

Similarly, D_v^{xrf} is then optimized by solving

$$\begin{aligned} \min_{D_v^{xrf}} \quad & \|(X - D_{nv}^{xrf}A_{nv}S) - D_v^{xrf}A_vS\|_F^2 \\ \text{s.t.} \quad & 0 \leq {D_v^{xrf}}_{ij} \leq 1, \forall i, j, \end{aligned} \tag{5.18}$$

using the following iteration steps:

$$
\begin{aligned}
U^q = {} & D_v^{xrf^{q-1}} \\
& - \frac{1}{d_v^{xrf}}(D_v^{xrf^{q-1}} A_v S - (X - D_{nv}^{xrf} A_{nv} S)) S^T {A_v}^T
\end{aligned}
\tag{5.19a}
$$

$$
D_v^{xrf^q} = prox_{D_v^{xrf}}(U^q), \tag{5.19b}
$$

where $d_v^{xrf} = \gamma_4 \| A_v {A_v}^T \|_F$ is the non-zero step size constant and $prox_{D_v^{xrf}}$ is the proximal operator which project U^q onto the constraints of Equation 5.18. It is the same as the proximal operator in Equation 5.17b.

Finally, we optimize D_{nv}^{xrf} by solving the following problem,

$$
\begin{aligned}
\min_{D_{nv}^{xrf}} \quad & \| (X - D_v^{xrf} A_v S) - D_{nv}^{xrf} A_{nv} S \|_F^2 \\
& + \lambda \| \nabla (D_{nv}^{xrf} A_{nv}) \|_F^2 \\
\text{s.t.} \quad & 0 \leq {D_{nv}^{xrf}}_{ij} \leq 1, \forall i, j.
\end{aligned}
\tag{5.20}
$$

Likewise, the following two steps are iterated until convergence:

$$
\begin{aligned}
L^q = {} & D_{nv}^{xrf^{q-1}} \\
& - \frac{1}{d_{nv}^{xrf}}(D_{nv}^{xrf^{q-1}} A_{nv} S - (X - D_v^{xrf} A_v S)) S^T {A_{nv}}^T \\
& - \lambda D_{nv}^{xrf} A_{nv} G G^T {A_{nv}}^T
\end{aligned}
\tag{5.21a}
$$

$$
D_{nv}^{xrf^q} = prox_{D_{nv}^{xrf}}(L^q), \tag{5.21b}
$$

where $d_{nv}^{xrf} = \gamma_5 \| A_n v A_n v^T \|_F$ again is a non-zero step size constant, $prox_{D_{nv}^{xrf}}$ is the proximal operator projecting L^q onto the constraints of Equation 5.20, which is the same proximal

operator as the ones in Equations 5.17b and 5.19b. The complete optimization scheme is illustrated in Algorithm 3. According to Equations 5.2, 5.3, 5.4, the HR XRF output image Y can be computed by

$$Y = Y_v + Y_{nv} = D_v^{xrf} A_v + D_{nv}^{xrf} A_{nv}. \tag{5.22}$$

Algorithm 3. Proposed Optimization Scheme of Equation 5.9

input: LR XRF image X, HR conventional RGB image I.

Initialize $D^{rgb(0)}$, $D_v^{xrf(0)}$ and $A_v^{l\,(0)}$ by Equation (5.13);
Initialize $D_{nv}^{xrf(0)} = D_v^{xrf(0)}$;
Initialize $A_v{}^{(0)}$ by upsampling $A_v^{l\,(0)}$;
Initialize $A_{nv}{}^{(0)} = \mathbf{0}$;
$n = 0$;
repeat
  Estimate $A_v{}^{(n+1)}$ and $A_{nv}{}^{(n+1)}$ with Equation 5.15;
  Estimate $D^{rgb(n+1)}$ with Equation 5.17;
  Estimate $D_v^{xrf(n+1)}$ with Equation 5.19;
  Estimate $D_{nv}^{xrf(n+1)}$ with Equation 5.21;
  n=n+1;
until convergence

output: HR XRF image
$Y = D_v^{xrf} A_v + D_{nv}^{xrf} A_{nv}$.

5.4. Experimental Results

To verify the performance of our proposed SR method, we have performed extensive experiments on both synthetic and real XRF images. The basic parameters of the proposed SR method are set as follows: the number of atoms in the dictionaries D^{rgb}, D_{nv}^{xrf} and D_v^{xrf} is $M = 50$ for synthetic experiments and $M = 200$ for real experiments; $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 = 1.01$, which only affects the speed of convergence; parameter β in Equation 5.13

is set to 0.02 and λ in Equation 5.9 is set to 0.1. The optional constraint in Equation 5.9h is not applied here.

5.4.1. Error Metrics

As a primary error metric we use, the root mean squared error (RMSE) between the estimated HR XRF image Y and the ground truth image Y^{gt} is computed

$$RMSE = \sqrt{\frac{\|Y - Y^{gt}\|_F^2}{BN_h}}. \tag{5.23}$$

The peak-signal-to-noise ratio (PSNR) is reported as well,

$$PSNR = 20 \log_{10} \frac{\max(Y^{gt})}{RMSE}, \tag{5.24}$$

where $\max(Y^{gt})$ denoting the maximum entry of Y^{gt}.

The spectral angle mapper (SAM, [**122**]) in degrees is also utilized, which is defined as the angle in $\mathbb{R}^B$ between an estimated pixel and the ground truth pixel, averaged over the whole image,

$$SAM = \frac{1}{N_h} \sum_{j=1}^{N_h} \arccos \frac{Y(:,j)^T Y^{gt}(:,j)}{\|Y(:,j)\|_2 \|Y^{gt}(:,j)\|_2}. \tag{5.25}$$

5.4.2. Comparison Methods

In order to compare over results with the hyperspectral image SR method GSOMP [**1**], CSUSR [**67**] and NSSR [**35**], the linear degradation matrix P mapping the XRF spectrum to its corresponding RGB representation needs to be estimated first. Since these methods do not estimate this linear transformation,

$$I^l = PX, \tag{5.26}$$

where $I^l \in \mathbb{R}^{b \times N_l}$ is defined in Equation 5.11 and $X \in \mathbb{R}^{B \times N_l}$ is the input LR XRF image. Although this linear transformation model does not hold for XRF and its corresponding RGB images, we are trying to find the best approximation P so that we can apply the mentioned above hyperspectral image SR methods. The best approximation P can be computed by the following least-squares problem

$$\min_{P} \quad \|PX - I^l\|_F^2. \tag{5.27}$$

The Trust-Region-Reflective Least Squares algorithm [**25**] can be utilized to estimate P.

Besides the above mentioned hyperspectral image SR methods, we also propose two baseline methods to compare against, since SR for XRF images is still an open problem. Baseline method #1 only uses LR XRF image as input, increasing its spatial resolution by the same TV regularizer as in Equation 5.9, by solving

$$\min_{A, D^{xrf}} \|X - D^{xrf}AS\|_F^2 + \lambda \|\nabla(D^{xrf}A)\|_F^2 \tag{5.28a}$$

$$\text{s.t.} \quad 0 \leq D_{ij}^{xrf} \leq 1, \forall i, j \tag{5.28b}$$

$$A_{ij} \geq 0, \forall i, j \tag{5.28c}$$

$$\mathbf{1^T}A = \mathbf{1^T}, \tag{5.28d}$$

$$\|A\|_0 \leq s, \tag{5.28e}$$

which is a special case of the proposed optimization problem in Equation 5.9. A detailed optimization scheme can be found in Appendix A.1. After solving for D^{xrf} and A, the HR output XRF image Y can be reconstructed by

$$Y = D^{xrf}A. \tag{5.29}$$

Baseline method #2 does not model the input LR XRF image as a combination of visible and non-visible components, and increases its spatial resolution with a conventional HR RGB image, by solving

$$\min_{A,D^{xrf},D^{rgb}} \|I - D^{rgb}A\|_F^2 + \|X - D^{xrf}AS\|_F^2 \tag{5.30a}$$

$$\text{s.t.} \quad 0 \leq D_{ij}^{xrf} \leq 1, \forall i, j \tag{5.30b}$$

$$0 \leq D_{ij}^{rgb} \leq 1, \forall i, j \tag{5.30c}$$

$$A_{ij} \geq 0, \forall i, j \tag{5.30d}$$

$$\mathbf{1^T}A = \mathbf{1^T}, \tag{5.30e}$$

$$\|A\|_0 \leq s, \tag{5.30f}$$

which is also a special case of the proposed optimization problem in Equation 5.9. Detailed optimization scheme can be found in Appendix A.2. After solving for D^{rgb}, D^{xrf} and A, the HR output XRF image Y can be reconstructed by Equation 5.29.

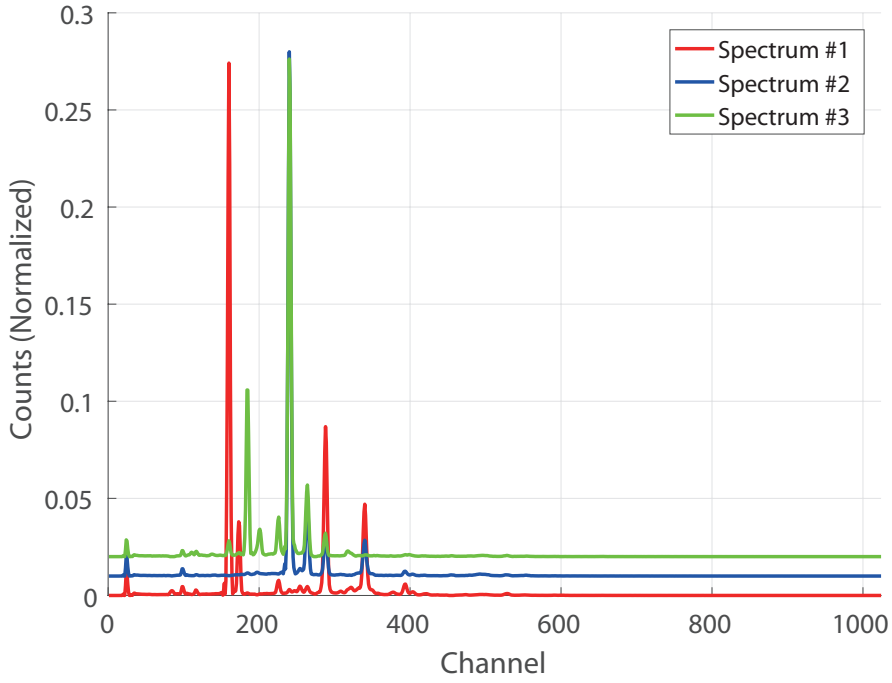
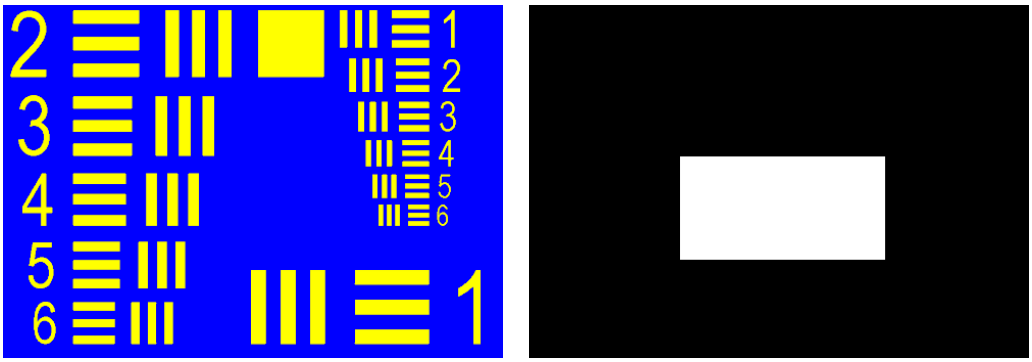


Figure 5.2. Three noise free spectra used to synthesize the HR XRF image. Spectra # 2 and # 3 are shifted vertically (by 0.01 and 0.02, respectively) for visualization purposes.

5.4.3. Synthetic Experiments

We compare the SR results for different methods with a synthetic experiment first. We combined 3 noise free spectra with a significant amount of spectral overlap (1024×1), an HR airforce target image ($345 \times 490 \times 3$) as the visible image and a rectangle image ($345 \times 490 \times 3$) as the non-visible image to simulate the ground truth HR XRF image Y^{gt} ($345 \times 490 \times 1024$). The 3 noise free spectra, HR airforce target image and the rectangle image are shown in Figs. 5.2, 5.3 (a) and 5.3 (b), respectively. In detail, we assume that the yellow foreground of the airforce target image corresponds to spectrum # 1, the blue background of the airforce image corresponds to spectrum #2 and the white foreground of the rectangle image corresponds to spectrum #3. The LR XRF image X ($69 \times 98 \times 1024$)

was obtained by spatially down-sampling Y^{gt} by a factor of 5 in each dimension and adding Gaussian noise to it with SNR 35dB.



(a) (b)

Figure 5.3. (a) The airforce image is utilized as the visible component. (b) The rectangle image is utilized as the non-visible component.

The RMSE, PSNR and SAM metrics were computed between the SR results of different methods described in Section 5.4.2 and the HR ground truth Y^{gt}. The default parameters of methods GSOMP [**1**], CSUSR [**67**] and NSSR [**35**] in their original paper were applied in our synthetic experiments. Optimal parameter λ of Baseline #1 method (Equation 5.28) and the proposed method (Equation 5.9) was found experimentally. To make fair comparisons, the number of atoms in the dictionary is set to be 50 for all methods (GSOMP [**1**], CSUSR [**67**], NSSR [**35**], Baseline # 2 and the proposed method) that utilize dictionaries. As shown in Table 5.1, our proposed method has the smallest RMSE, highest PSNR and smallest SAM. By comparing Baseline #1 method with the proposed method, the benefit of utilizing an HR RGB image can be validated. By comparing Baseline #2 method with the proposed method, it can be seen that a better and more realistic model that assumes the XRF signal is a combination of visible component and non-visible component is beneficial to obtain better SR results. The traditional hyperspectral image SR methods (GSOMP [**1**], CSUSR [**67**] and NSSR [**35**]) rely on an accurate linear degradation model from hyperspectral to RGB

signals. When the degradation model is not accurate, their performance is inferior to our proposed method. Baseline #2 can be considered an extension of CSUSR [**67**], in that we learn the coupled RGB and XRF dictionaries simultaneously and we do not utilize the linear degradation model, which is a more flexible approach and produces better SR performance than CSUSR [**67**].

Metric	GSOMP [**1**]	CSUSR [**67**]	NSSR [**35**]	Baseline #1	Baseline #2	Proposed
RMSE	3.42	0.70	3.85	2.03	0.59	**0.50**
PSNR	37.51	51.36	36.46	42.03	52.83	**54.12**
SAM	22.78	3.19	18.46	8.38	2.10	**2.00**

Table 5.1. Experimental results on synthetic data comparing different SR methods discussed in Section 5.4.2 in terms of RMSE, PSNR and SAM. Best results are shown in bold.

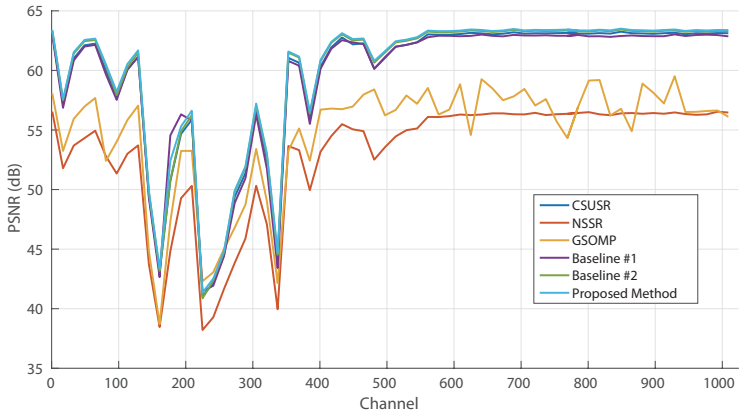


Figure 5.4. The average PSNR curves as a function of the channels of the spectral bands for the SR method.

Fig. 5.4 shows the average PSNR curves as a function of the channels of the spectral bands for the test methods. It can be seen that the hyperspectral SR methods GSOMP [**1**] and NSSR [**35**] produce inferior SNR over all spectral channels. All other methods perform well for spectral bands outside the range [100 400] and our proposed method constantly outperforms all other methods. For spectral bands in the range [100 400], both methods'

performance decreases because of the overlapping spectra, as shown in Fig. 5.2. Notice that Baseline #1 slightly outperforms the proposed method around channel #200, which is because there is a peak for both the non-visible (Spectrum #3 in Fig. 5.2) and visible component spectrum (Spectrum #1 in Fig. 5.2) around channel #200. The proposed method makes errors in separating these two peaks, resulting in worse performance than Baseline #1 which avoids explicit visible/non-visible decomposition.

We compare the visual quality of different SR methods on the region of interest of channel # 210 - 230 in Fig. 5.5. Because GSOMP, CSUR, and NSSR hyperspectral image SR methods [**1**,**35**,**67**] rely on an accurate linear degradation model from hyperspectral to RGB signals, SR results are poor. Baseline method #1 did not utilize the HR RGB image in SR and so failed to reconstruct fine details. Baseline method #2 assumes one-to-one mapping between RGB and XRF signals, thus it produced artifacts in the region where the visible and non-visible components overlap. Our proposed method produced the SR result closest to ground truth. Notice that the non-visible component (rectangle) is more blurry than the visible component (airforce target), since it is super-resolved by a TV regularizer and does not use any HR RGB image information.

5.4.4. Real Experiments

For our first real experiment, the real data was collected by a Bruker M6 scanning energy dispersive XRF instrument, with 4096 channels in spectrum. Studies from XRF image #3 scanned from Vincent Van Gogh's "Bedroom" (Fig. 1.4) are presented here. The HR RGB image is aligned to the LR XRF image utilizing feature points matching.

We first validated that the proposed method in Equation (5.9) can accurately represent the XRF spectrum, and that the reconstructed spectral signal has a higher SNR compared to the original spectral signal.

As shown in Fig. 5.6, our proposed approach provides accurate reconstruction of the original signal. The XRF dictionary is trained from all spectral signals of the XRF image based on minimizing the Euclidean distance between the reconstructed and the original signals. As a result, noise is reduced, and the reconstructed signal has higher SNR compared to the original signal.

For our first real experiment, HR ground truth was not available to assess the quality of the reconstructed HR XRF images. This is because all XRF maps we had access to were low resolution and noisy. We compare the visual quality of different SR methods on the region of interest of channel # 611 - 657, corresponding to CrK XRF peak, in Fig. 5.7. Hyperspectral SR method GSOMP [**1**] produced a noisy output in (c), because it relies on an accurate degradation model from XRF signal to RGB signal. Hyperspectral SR methods CSUSR [**67**] and NSSR [**35**] update the XRF dictionary to ensure the fidelity to LR input, so they produce less noise as compared to GSOMP [**1**]. However, they either create non existing content in (d) or lose existing content in (e), in the towel regions. Baseline method #1 creates a blurry SR result, since it does not utilize an HR RGB image. Also it fails to resolve the fine detail in the towel region. Baseline method #2 produced visually satisfactory SR results, but failed to reconstruct the line between the wall and the floor. This is because of the one-to-one mapping assumption incorrectly maps brown pixels in the table and the line between the wall and the floor to the same XRF spectra. Our proposed method in (h) produces both a visually satisfactory result as well as strong similarity with the original LR input (a).

For our second real experiment, the real data was collected by a home-built X-ray fluorescence spectrometer (courtesy of Prof. Koen Janssens), with 2048 channels in spectrum. Studies from XRF image scanned from Jan Davidsz. de Heem's "Bloemen en insecten" (ca 1645), in the collection of Koninklijk Museum voor Schone Kunsten (KMKSA) Antwerp, are presented here. The original XRF image has dimension $680 \times 580 \times 2048$. We first spatially downsample the original XRF image by factor 5 and obtain the input LR XRF image with dimension $136 \times 116 \times 2048$. Then different SR methods are applied to increase the spatial resolution of the LR XRF image by factor 5. Notice that because the original HR XRF image is noisy and blurry, it is different from the HR ground truth. However, we can still use it as a reference to compute the RMSE, PSNR and SAM metrics to quantitatively compare the performance of different SR methods. We can also use it as a reference to visually compare different SR results with the original HR XRF image.

As shown in Table 5.2, our proposed method provides the closest reconstruction compared to all other methods. The traditional hyperspectral image SR methods (GSOMP [**1**] and NSSR [**35**]) produce considerably greater reconstruction error. Baseline #2 does not assume linear transformation model from XRF spectrum to RGB and updates the XRF and RGB endmembers simultaneously, resulting in better SR results. Baseline method #1 produces SR results more similar to the original HR XRF image compared to Baseline method #2, since both SR results of Baseline method #1 and the original HR XRF image are blurry.

Metric	GSOMP [**1**]	CSUSR [**67**]	NSSR [**35**]	Baseline #1	Baseline #2	Proposed
RMSE	75.18	70.20	79.72	70.35	70.43	**69.83**
PSNR	42.70	55.66	49.70	56.06	54.93	**56.19**
SAM	32.60	12.30	25.81	11.60	12.98	**11.32**

Table 5.2. Experimental results on "Bloemen en insecten" comparing different SR methods discussed in Section 5.4.2 in terms of RMSE, PSNR and SAM. Best results are shown in bold.

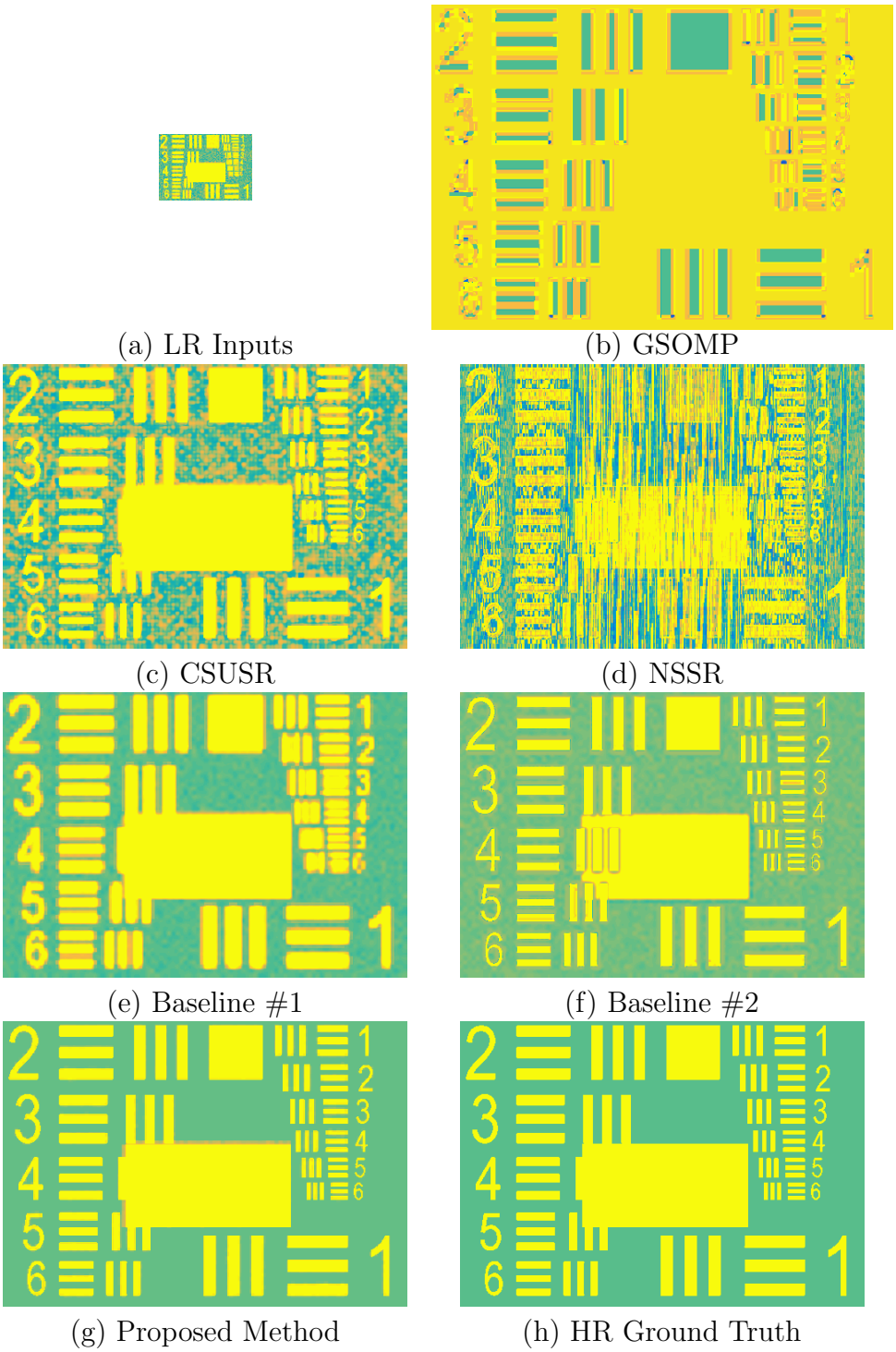
Finally, our proposed method produces a result most similar to the input HR XRF image, demonstrating the effectiveness of our proposed method.

The visual quality of different SR methods on the region of interest of channel #582 - 602, corresponding to $Pb\ L\eta$ XRF emission line, is compared in Fig. 5.8. Notice that the two long rectangles in the origin HR XRF image (h) are the stretcher bars under the canvas, which is not visible on the RGB image. Hyperspectral SR methods CSUSR [**67**] and GSOMP [**1**] in (c) and (d) produce noisy results and produce visible artifacts in many regions again. Baseline method #1 in (e) improves SNR compared to the origin HR XRF image. However, its SR result is blurry and fails to resolve the details on the flowers. Baseline method #2 in (f) utilizes HR RGB image as input, so its SR result is sharp and many details are resolved. However, because it does not model the non-visible component of the XRF image, it fails to precisely reconstruct the two hidden stretcher bars. Also when compared to the origin HR XRF image (h), it produces many artifacts, such as the textures of the flower in the middle, edges and stems. Our proposed method in (g) successfully reconstructs the non-visible stretcher bars. The reconstructed stretcher bars are blurry compared to other objects, because it does not utilize any information from the HR RGB image. More details are resolved by our proposed method. When compared to the origin HR image (h), we can conclude that those resolved details have high fidelity to the original HR image (h). The SNR is also improved by our proposed method.

5.5. Conclusion

In this paper we presented a novel XRF image SR framework based on fusing an HR conventional RGB image. The XRF spectrum of each pixel is represented by an endmembers dictionary, as well as the RGB spectrum. We also decomposed the input LR XRF image

into visible and non-visible components, making it possible to find the non-linear mapping from RGB spectrum to XRF spectrum. The non-visible component is super-resolved using a standard total variation regularizer. The HR visible XRF component and HR non-visible XRF component are combined to obtain the final HR XRF image. Both synthetic and real experiments show the effectiveness of our proposed method. Due to the extreme high spectral dimensions of the problem, the proposed algorithm is computationally demanding. Our future work includes its speedup with the use of parallel computation. We also plan to mount an HR RGB camera on the XRF scanner and calibrate the relative position of the XRF beam and the HR RGB camera, so that we can capture the HR RGB image and LR XRF image simultaneously and have accurate alignment of these two signals.



(a) LR Inputs (b) GSOMP

(c) CSUSR (d) NSSR

(e) Baseline #1 (f) Baseline #2

(g) Proposed Method (h) HR Ground Truth

Figure 5.5. Visualization of the SR result of the synthetic experiment. Region of interest of channel #210 - 230 is selected. (a) is the LR input XRF image. (b), (c), (d), (e), (f), (g) are the SR result of GSOMP [**1**], CSUSR [**67**], NSSR [**35**], Baseline #1, Baseline #2 and proposed method, respectively. (h) is the HR ground truth image.

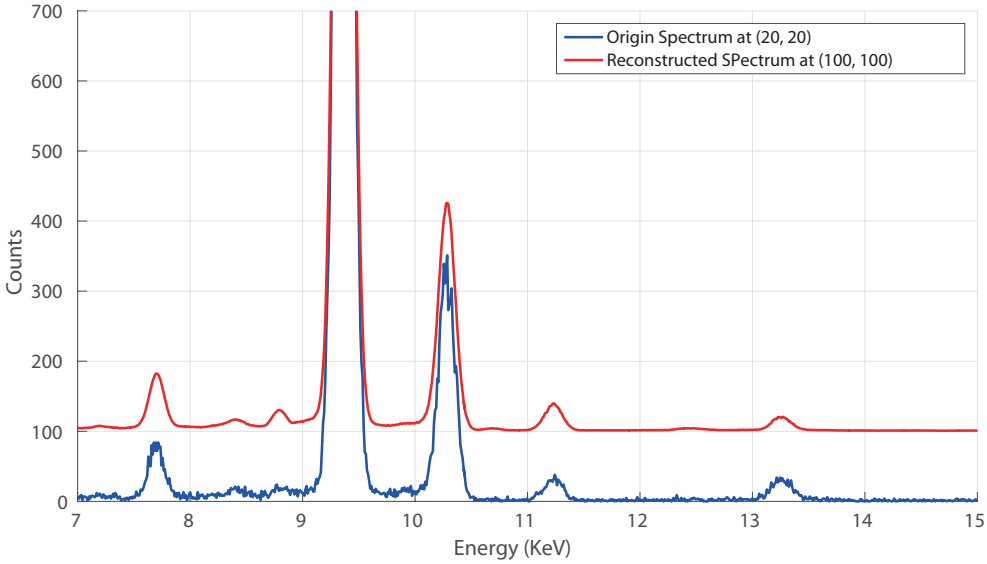
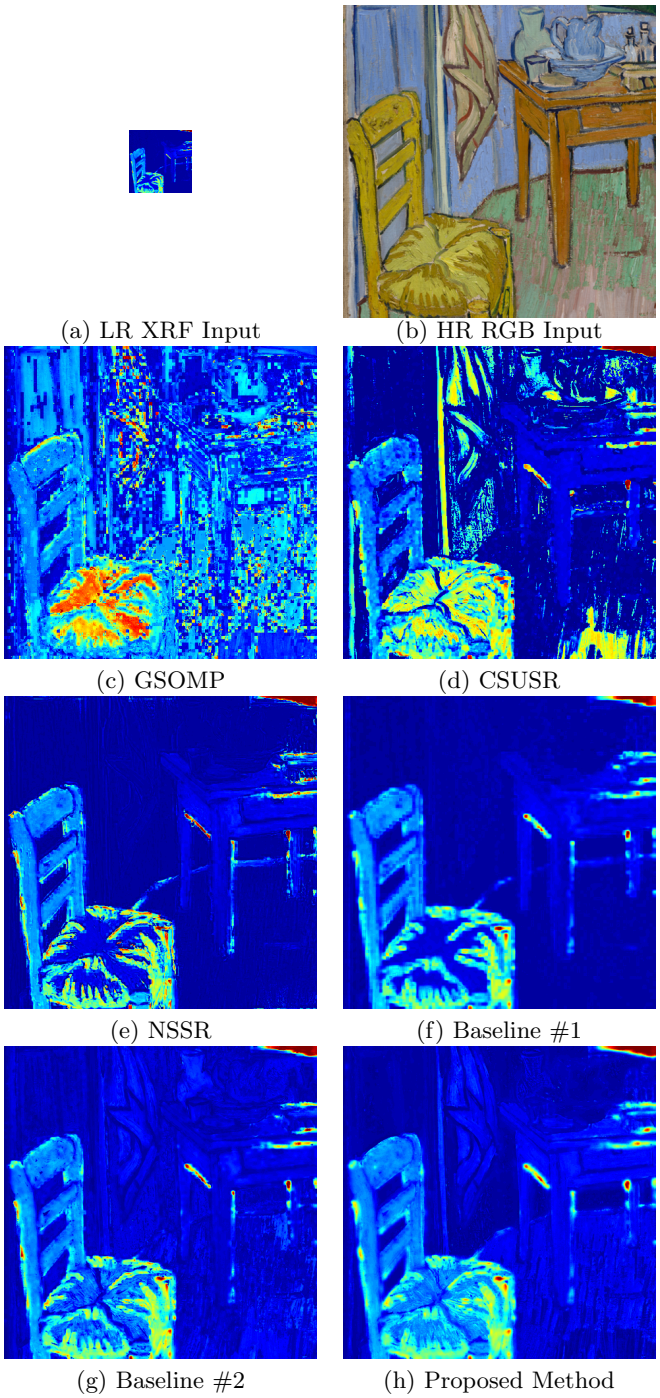


Figure 5.6. The reconstruction of a spectrum using our proposed method. The reconstructed spectrum is shifted vertically (100 counts) for visualization purposes.



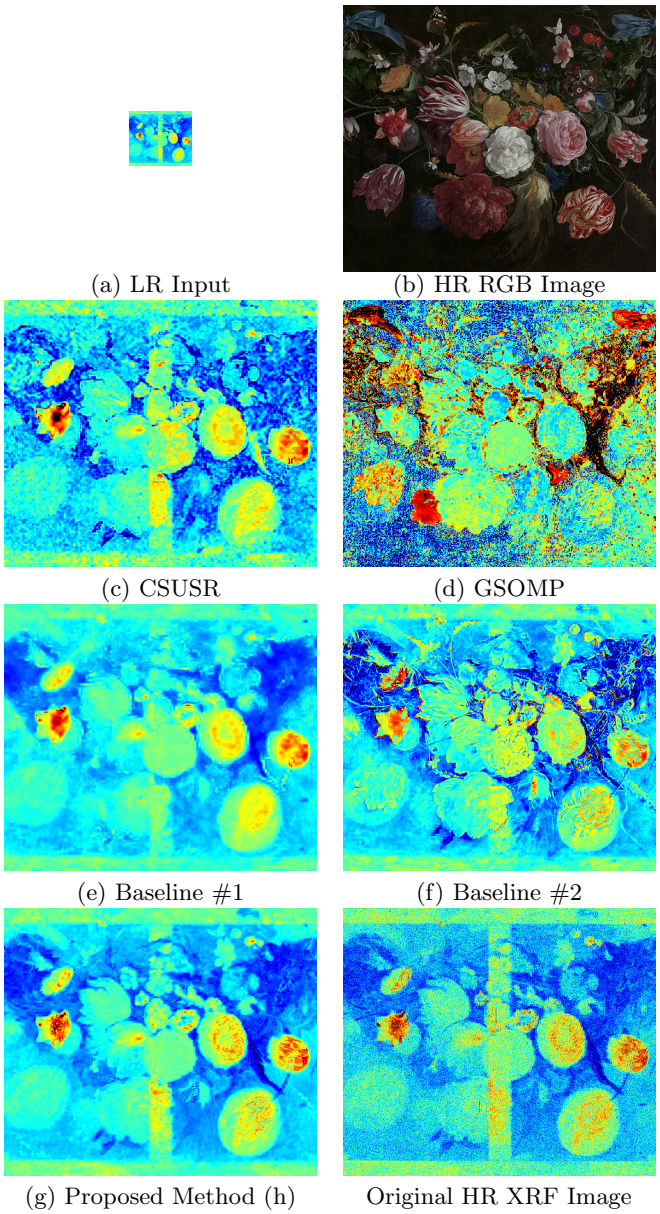
(a) LR XRF Input (b) HR RGB Input

(c) GSOMP (d) CSUSR

(e) NSSR (f) Baseline #1

(g) Baseline #2 (h) Proposed Method

Figure 5.7. Visualization of the SR result of the real experiment on the "Bedroom". Region of interest of related to CrK peak (channel #611 - 657) is selected. (a) is the LR input XRF image and (b) is the HR input RGB image. (c), (d), (e), (f), (g), (h) are the SR result of GSOMP [**1**], CSUSR [**67**], NSSR [**35**], Baseline #1, Baseline #2 and proposed method, respectively.



(a) LR Input (b) HR RGB Image

(c) CSUSR (d) GSOMP

(e) Baseline #1 (f) Baseline #2

(g) Proposed Method (h) Original HR XRF Image

Figure 5.8. Visualization of the SR result of the DeHeem real experiment on the "Bloemen en insecten". Region of interest of related to *Pb Lη* XRF emission line (channel #582 - 602) is selected. (a) is the LR input XRF image and (b) is the HR input RGB image. (c), (d), (e), (f), (g) are the SR result of CSUSR [**67**], GSOMP [**1**], Baseline #1, Baseline #2 and proposed method, respectively. (h) is the original HR XRF image. Readers are suggested to zoom in in order to compare the details of different results.

CHAPTER 6

X-Ray Fluorescence Image Inpainting Utilizing Adaptive Sampling Mask

6.1. Introduction

In this chapter, we propose an image inpainting approach to speedup the acquisition time of the XRF images, with the aid of a conventional RGB image, as shown in Figure 1.6. The proposed XRF image inpainting algorithm can also be applied to spectral images obtained by any other raster scanning processes, such as Sanning Electron Microscope (SEM), Energy Dispersive Spectroscopy (EDS) sand Wavelength Dispersive Spectroscopy (WDS). First, the conventional RGB image of the scanning target is applied to generate the adaptive sampling mask. Then the XRF scanner will scan the corresponding pixels according to the binary sampling mask. The speedup in acquisition is achieved since many pixels will be skipped during the acquisition process. Finally, the subsampled XRF image is fused with the conventional RGB image to reconstruct the full scan XRF image, as an image inpainting algorithm. For the fusion based XRF image inpainting algorithm, similar to our previous super-resolution (SR) approach [**28**, **29**], we model the spectrum of each pixel using a linear mixing model [**80**, **87**]. Because the hidden part of the paining is not visible in the conventional RGB image, but it can be captured in the XRF image [**3**], there is no direct one-to-one mapping between the visible RGB spectrum and the XRF spectrum. We model the XRF signal as a combination of the visible signal (on the surface) and the non-visible signal (hidden under surface), as shown in Figure 6.1. To inpaint the visible component XRF

signal, we follow a similar approach in hyper-spectral image SR [**1**, **2**, **35**, **50**, **62**, **67**, **112**]. A coupled XRF-RGB dictionary pair is learned to explore the correlation between XRF and RGB signals. The RGB dictionary is then applied to obtain the sparse representation of the RGB input image, resulting in a full-sampled coefficient map. Then the XRF dictionary could be applied on the full-sampled coefficient map to reconstruct the XRF image. Different from those hyperspectral image SR approaches, we experimentally found that for the inpainting problem, a spatial adaptive total variation regularizer [**8**, **81**] is needed to produce smooth XRF output image. For the non-visible part, we inpaint its missing pixels using a standard total variation regularizer. Finally, the reconstructed visible and non-visible XRF signals are combined to obtain the final XRF reconstruction result. The input subsampled XRF image is not explicitly separated into visible and non-visible parts in advance. Instead, the whole inpainting problem is formulated as an optimization problem. By alternatively optimizing over the coupled XRF-RGB dictionary and the visible / non-visible full-sampled coefficient maps, the fidelity of the estimated full-sampled output to both the subsampled XRF and RGB input signals is improved, thus resulting in a better inpainting output. Both synthetic and real experiments show the effectiveness of our proposed method, in terms of both reconstruction error and visual quality of the inpainting result, compared to other methods [**44**, **95**, **127**].

This chapter is organized as follows. We introduce the adaptive sampling mask design in Section 6.2. We describe the XRF image inpainting problem in Section 6.3. In Section 6.4, we provide the experimental results with both synthetic data and real data to evaluate the approach. The paper is concluded in Section 6.5.

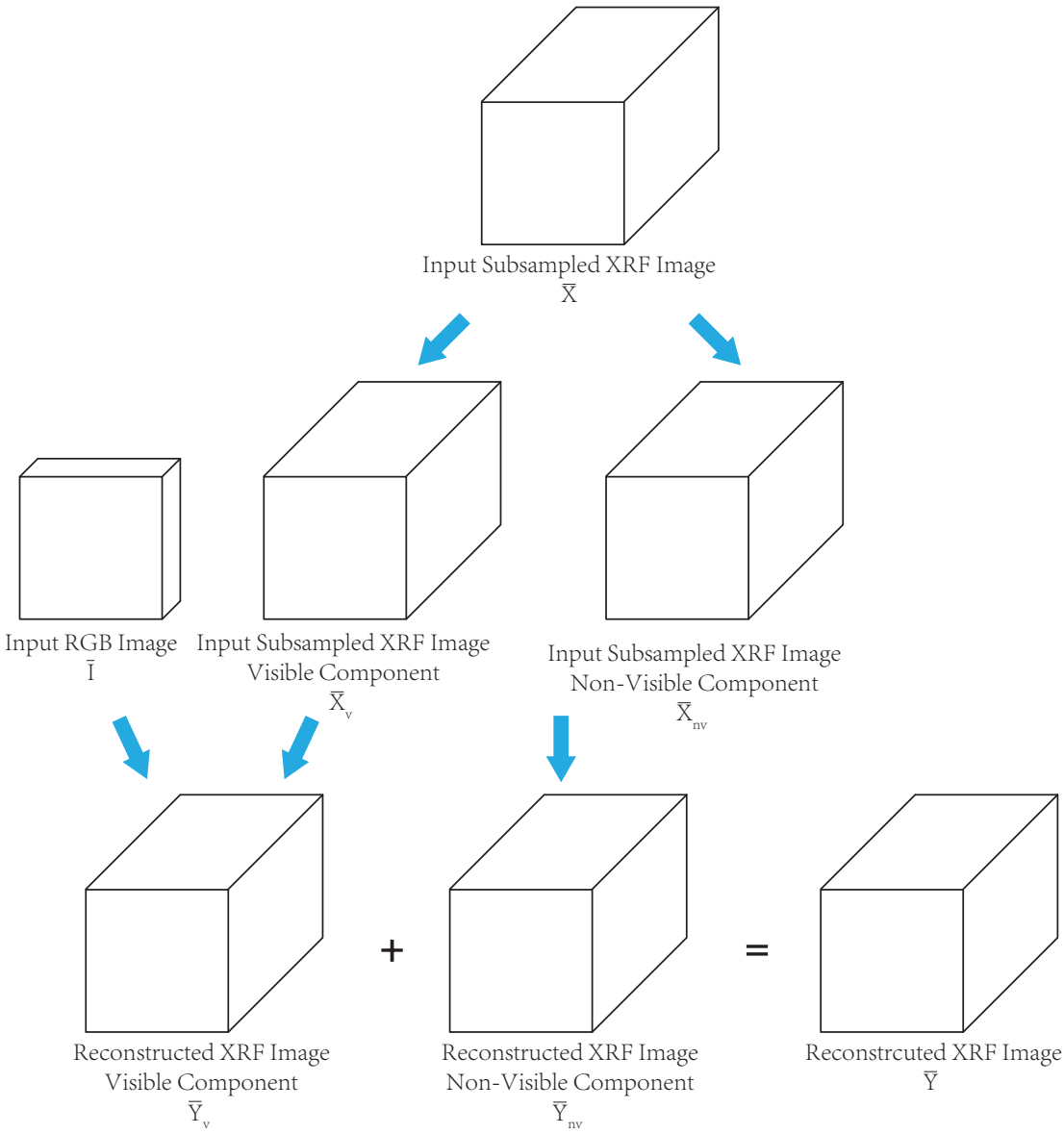


Figure 6.1. Proposed pipeline of XRF image inpainting. The visible component of the input subsampled XRF image is fused with the input RGB image to obtain the visible component of the reconstructed XRF image. The non-visible component of the input XRF image is super-resolved to obtain the non-visible component of the reconstructed XRF image. The reconstructed visible and non-visible component of output XRF image are combined to obtain the final output.

6.2. Adaptive Sampling Mask Generation utilizing Convolutional Neural Network

In this section, we present our proposed adaptive sampling mask generation using a CNN, in other words, we describe the details of the "Sampling Mask Generation" block in Figure 1.6. We first formulate the problem of adaptive sampling mask design, followed by the presentation of the overall network architecture consisting of both the inpainting network and the mask generation network.

6.2.1. Problem Formulation

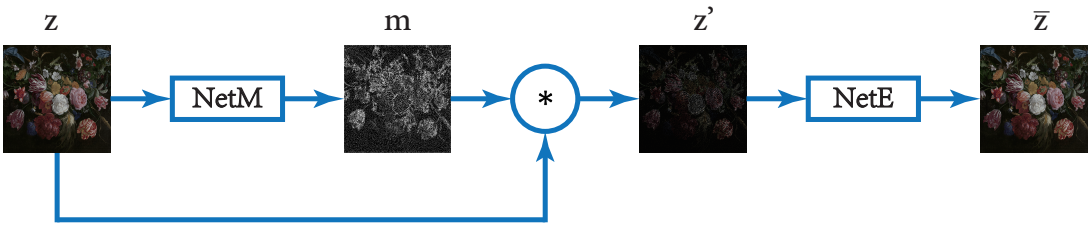


Figure 6.2. Pipeline for adaptive sampling mask generation utilizing CNN.

As shown in Figure 6.2, we denote by z an input original image. Our mask generation network $NetM$ produces a binary sampling mask $m = NetM(z, c)$, where $c \in [0\ 1]$ is the predefined sampling percentage. The entries of m are equal to 1 for the sampled pixles and 0 otherwise. The corrupted image z' is obtained by

$$z' = z \odot m = z \odot NetM(z, c), \tag{6.1}$$

where $\odot$ is the element-wise product operation. The reconstructed image $\bar{z}$ is obtained by the inpainting network $NetE$,

$$\bar{z} = NetE(z') = NetE(z \odot NetM(z, c)). \tag{6.2}$$

The overall pipeline is shown in Figure 6.2. We could regard the whole pipeline (Equation 6.2) as one network, with input z and output $\bar{z}$, and perform an end to end training. If we simultaneously optimize the mask generation network $NetM$ and the inpainting network $NetE$ according to the following loss function,

$$\mathcal{L}(z) = \|z - \bar{z}\|_2 = \|z - NetE(z \odot NetM(z, c))\|_2, \tag{6.3}$$

$NetM$ will perform an optimized adaptive sampling strategy according to the input image, and $NetE$ will also perform optimized image inpainting. After the mask has been generated by the network $NetM$, we can replace the inpainting network $NetE$ with other image inpainting algorithms. The detailed network architecture of $NetE$ and $NetM$ are discussed in the following two subsections 6.2.2 and 6.2.3, respectively.

6.2.2. Deep Learning Network Architecture for Inpainting Network

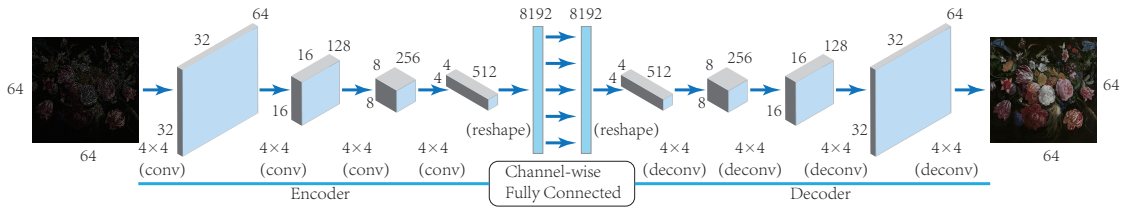


Figure 6.3. Network architecture for the image inpainting network ($NetE$). The inpainting framework is an autoencoder network with the encoder and decoder connected by a channel-wise fully-connected layer.

The network architecture in [**45**] is used for the inpainting network, as shown in Figure 6.3. The network is an encoder-decoder pipeline. The encoder takes a corrupted image

z' of size 64×64 as input and encodes it in the latent feature space. The decoder takes the feature representation and outputs the resorted image $\bar{z} = NetE(z')$. The encoder and decoder are connected through a channel-wise fully-connected layer. For the encoder, four convolutional layers are utilized. A batch normalization layer [**58**] is placed after each convolutional layer to accelerate the training speed and stabilize the learning process. The Leaky Rectified Linear Unit (LeakyReLU) activation [**78**, **116**] is used in all layers in the encoder.

The convolutional layers in the encoder only connect all the feature map together, but there are no direct connections among different locations within each specific feature map. Fully-connected layers are then applied to handle this information propagation. To reduce the number of parameters in the fully connected layers, a channel-wise fully-connected layer is used to connect the encoder and decoder, as in [**86**]. The channel-wise fully connected layer is designed to only propagate information within activations of each feature map. This significantly reduces the number of parameters in the network and accelerates the training process.

The decoder consists of four deconvolutional layers [**38**, **76**, **123**], each of which is followed by a rectified linear unit (ReLU) activation except the output layer. Tanh function is used in the output layer to restrict the pixel range of the output image. The series of up-convolutions and non-linearities conducts a non-linear weighted upsampling of the feature produced by the encoder and generates a higher resolution image of the target size (64×64).

6.2.3. Deep Learning Network Architecture for Mask Generation Network

According to our knowledge, no prior work has been reported on generating the adaptive binary sampling mask utilizing CNN. The desired mask generation network $NetM$ should satisfy the following criteria:

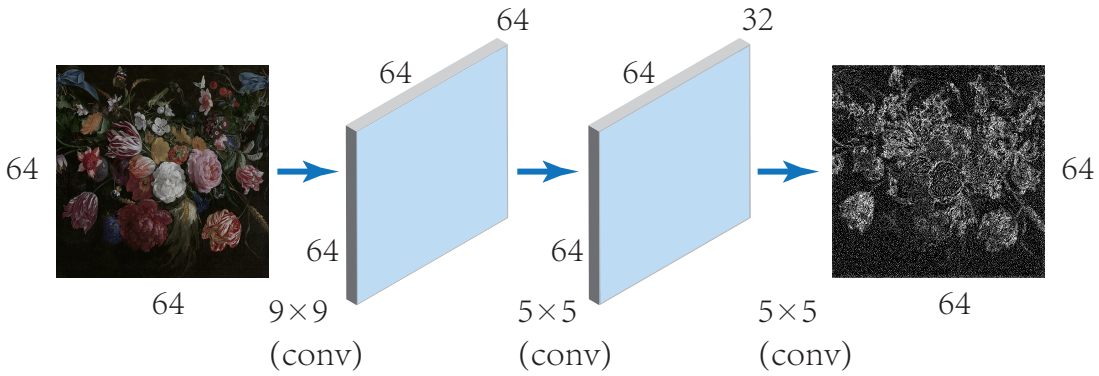


Figure 6.4. Network architecture for the mask generation network ($NetM$). Three layers of convolution are used to estimate the binary sampling mask.

- The output image should have the same spatial resolution as the input image.
- The network architecture should be fully convolutional to handle arbitrary input sizes.
- The output image should be binary.
- The output image should have a certain percentage c of 1's.

A network architecture similar to SRCNN [**34**] is applied here, as shown in Figure 6.4. The network $NetM$ consists of three convolutional layers, to handle an arbitrary input size. To keep the spatial dimensions at each layers the same, the images are paded with zeros. Each convolutional layer is followed by a ReLU activation except the output layer.

Let us denote by L_{ij} the $(i,j)^{th}$ element of L which is mapped to the range [0 1] with mean value equal to c, by the mapping F defined as

$$D_{ij} = F(L_{ij}) = \begin{cases} c + c \times \tanh(L_{ij} - \bar{L}), & \text{if } c \leq 0.5 \\ c + (1-c) \times \tanh(L_{ij} - \bar{L}), & \text{if } c > 0.5 \end{cases}, \tag{6.4}$$

where D_{ij} is the $(i,j)^{th}$ element of matrix $D \in \mathbf{R}^2$, and $\bar{L}$ is the mean of L. Then $D_{ij} \in [0\ 1]$ and the mean value of D, denoted by $\overline{D}$, will be approximately equal to c if the distribution of L is symmetric with respect to $\bar{L}$ ($\overline{D} \approx c$). Then the Bernoulli distribution $Ber()$ is applied to the binarization of the values of D, that is,

$$B_{ij} = Ber(D_{ij}) = \begin{cases} 1, & p = D_{ij} \\ \\ 0, & p = 1 - D_{ij} \end{cases}, \tag{6.5}$$

Notice that

$$\overline{B} \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} E(B_{ij}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} D_{ij} = c, \tag{6.6}$$

where N^2 is the total number of pixels of L and $E(B_{ij})$ is the expected value of B_{ij}. Therefore, B is binary matrix with mean value equal to c, implying that it has c percent of 1s.

Since applying the function $Ber(F(\cdot))$ on the input L will make the output of the network be binary and have c percent of 1s, we then make it the last layer activation function. Notice that function $Ber(D)$ is not continuous and its derivatives do not exist, making the back propagation optimization during training impractical. We use its expected value D to approximate it during training and apply the original function $Ber(D)$ during testing.

6.3. Spatial-Spectral Representation for X-Ray Fluorescence Image Inpainting

In this section, we propose the XRF image inpainting algorithm by fusing with it a conventional RGB image, providing the details of the "Proposed Inpainting Algorithm" block in Figure 1.6. The proposed fusion style inpainting approach has similarities with our previous fusion style SR approach [**29**]. We first formulate the XRF image inpainting problem, then demonstrate our proposed solution to this inpainting problem.

6.3.1. Problem Formulation

As shown in Figure 6.1, we are seeking the estimation of a reconstructed XRF image $\bar{Y} \in \mathbb{R}^{W\times H\times B}$ that is full-sampled, with W, H and B the image width, height and number of spectral bands, respectively. We have two inputs: a sub-sampled XRF image $\bar{X} \in \mathbb{R}^{W\times H\times B}$ with the known binary sampling mask $\bar{S} \in \mathbb{R}^{W\times H}$. $\overline{X(i,j,:)}$ is equal to the zero vector if not sampled, i.e., corresponding to $\overline{S(i,j)} = 0$; and a conventional RGB image $\bar{I} \in \mathbb{R}^{W\times H\times b}$ with the same spatial resolution as the target XRF image $\bar{Y}$, but a small number (equal to 3) of spectral bands, $b \ll B$. The input sub-sampled XRF image $\bar{X}$ can be separated into two parts: the visible component $\bar{X}_v \in \mathbb{R}^{W\times H\times B}$ and the non-visible component $\bar{X}_{nv} \in \mathbb{R}^{W\times H\times B}$, with the same binary sampling mask $\bar{S}$ as $\bar{X}$. We propose to estimate the fully sampled visible component $\bar{Y} \in \mathbb{R}^{W\times H\times B}$ by fusing the conventional RGB image $\bar{I}$ with the visible component of the input sub-sampled XRF image $\bar{X}_v$ and the fully sampled non-visible component $\bar{Y}_{nv} \in \mathbb{R}^{W\times H\times B}$ by using standard total variation inpainting methods.

To simplify notation, the image cubes are written as matrices, i.e., all pixels of an image are concatenated, such that every column of the matrix corresponds to the spectral response at a given pixel, and every row corresponds to a lexicographically ordered spectral band. Those un-sampled pixels are skipped in this matrix representation. Accordingly, the image cubes are written as $Y \in \mathbb{R}^{B\times N_h}$, $X \in \mathbb{R}^{B\times N_s}$, $I \in \mathbb{R}^{b\times N_h}$, $X_v \in \mathbb{R}^{B\times N_s}$, $X_{nv} \in \mathbb{R}^{B\times N_s}$, $Y_v \in \mathbb{R}^{B\times N_h}$, $Y_{nv} \in \mathbb{R}^{B\times N_h}$, where $N_h = W \times H$ and $N_s = W \times H \times c$ is the number of sampled XRF pixels. We therefore have

$$X = X_v + X_{nv}, \tag{6.7}$$

$$Y = Y_v + Y_{nv}, \tag{6.8}$$

according to the visible / non-visible component separation models as shown in Figure 6.1.

Let us denote by $y_v \in \mathbb{R}^B$ and $y_{nv} \in \mathbb{R}^B$ the one-dimensional spectra at a given spatial location of $\bar{Y}_v$ and $\bar{Y}_{nv}$, respectively. That is, a column of Y_v and Y_{nv} is represented , according to the linear mixing model [**16**, **63**], described as

$$y_v = \sum_{j=1}^{M} d_{v,j}^{xrf} \alpha_{v,j}, \quad Y_v = D_v^{xrf} A_v, \tag{6.9}$$

$$y_{nv} = \sum_{j=1}^{M} d_{nv,j}^{xrf} \alpha_{nv,j}, \quad Y_{nv} = D_{nv}^{xrf} A_{nv}, \tag{6.10}$$

where $d_{v,j}^{xrf}$ and $d_{nv,j}^{xrf}$ are column vectors representing respectively the endmembers for the visible and non-visible components, M is the total number of endmembers, $D_v^{xrf} \equiv [d_{v\ 1}^{xrf}, d_{v\ 2}^{xrf}, \ldots, d_{v\ M}^{xrf}] \in \mathbb{R}^{B \times M}$, $D_{nv}^{xrf} \equiv [d_{nv\ 1}^{xrf}, d_{nv\ 2}^{xrf}, \ldots, d_{nv\ M}^{xrf}] \in \mathbb{R}^{B \times M}$, and $\alpha_{v,j}$ and $\alpha_{nv,j}$ are the corresponding per-pixel abundances. Equation 6.8 holds for a specific column y_v of matrix Y_v (say the k^{th} column). We take the corresponding $\alpha_{v,j,j=1,\ldots,M}$ and stack them into an $M \times 1$ column vector. This vector then becomes the k^{th} column of the matrix $A_v \in \mathbb{R}^{M \times N_h}$. In a similar manner we construct matrix $A_{nv} \in \mathbb{R}^{M \times N_h}$. The endmembers D_v^{xrf} and D_{nv}^{xrf} act as basis dictionaries representing Y_v and Y_{nv} in a lower-dimensional space $\mathbb{R}^M$, with $rank\{Y_v\} \leq M, and\ rank\{Y_{nv}\} \leq M$.

The visible X_v and non-visible X_{nv} components of the input sub-sampled XRF image are spatially sub-sampled versions of Y_v and Y_{nv}, respectively, that is

$$X_v = Y_v S = D_v^{xrf} A_v S, \tag{6.11}$$

$$X_{nv} = Y_{nv}S = D_{nv}^{xrf} A_{nv} S, \tag{6.12}$$

where $S \in \mathbb{R}^{N_h \times N_s}$ is the sub-sampling operator that describes the spatial degradation from the fully sampled XRF image to the sub-sampled XRF image.

Similarly, the input RGB image I can be described by the linear mixing model [**16**, **63**],

$$I = D^{rgb} A_v, \tag{6.13}$$

where $D^{rgb} \in \mathbb{R}^{b \times M}$ is the RGB dictionary. Notice that the same abundance matrix A_v is used in Equations 6.9 and 6.11. This is because the visible component of the scanning object is captured by both the XRF and the conventional RGB images. Matrix A_v encompasses the spectral correlation between the visible component of the XRF and the RGB images.

The physically grounded constraints in [**67**] are shown to be effective in our previous work [**29**], so we propose to impose similar constraints, by making full use of the fact that the XRF endmembers are XRF spectra of individual materials, and the abundances are proportions of those endmembers. Consequently, they are subject to the following constraints:

$$0 \leq D_{v,ij}^{xrf} \leq 1, \forall i, j \tag{6.14a}$$

$$0 \leq D_{nv,ij}^{xrf} \leq 1, \forall i, j \tag{6.14b}$$

$$0 \leq D_{ij}^{rgb} \leq 1, \forall i, j \tag{6.14c}$$

$$A_{v,ij} \geq 0, \forall i, j \tag{6.14d}$$

$$A_{nv,ij} \geq 0, \forall i, j \tag{6.14e}$$

$$\mathbf{1^T}(A_v + A_{nv}) = \mathbf{1^T}, \tag{6.14f}$$

where $D_{v,ij}^{xrf}$, $D_{nv,ij}^{xrf}$, D_{ij}^{rgb}, $A_{v,ij}$ and $A_{nv,ij}$ are the (i, j) elements of matrices D_v^{xrf}, D_{nv}^{xrf}, D^{rgb}, A_v and A_{nv}, respectively, $\mathbf{1^T}$ demotes a row vector of 1's compatible with the dimensions of A_v and A_{nv}. Equations 6.14a, 6.14b and 6.14c enforce the non-negative, bounded spectrum constraints on endmembers, Equations 6.14d and 6.14e, enforce the non-negative constraints on abundances, and Equation 6.14e enforces the constraint that the visible component abundances and non-visible component abundances for every pixel sum up to one.

6.3.2. Proposed Solution

To solve the XRF image inpainting problem, we need to estimate A_v, A_{nv}, D^{rgb}, D_v^{xrf} and D_{nv}^{xrf} simultaneously. Utilizing Equations 6.7, 6.11, 6.12, 6.13 and 6.14, we can formulate the following constrained least-squares problem:

$$\min_{\substack{A_v, A_{nv}, D^{rgb}, \\ D_v^{xrf}, D_{nv}^{xrf}}} \|X - D_v^{xrf} A_v S - D_{nv}^{xrf} A_{nv} S\|_F^2 + \gamma \|\nabla_I (D_v^{xrf} A_v)\|_F^2 + \lambda \|\nabla (D_{nv}^{xrf} A_{nv})\|_F^2 + \|I - D^{rgb} A_v\|_F^2 \tag{6.15a}$$

$$\text{s.t.} \quad 0 \leq D_{v\ ij}^{xrf} \leq 1, \forall i, j \tag{6.15b}$$

$$0 \leq D_{nv\ ij}^{xrf} \leq 1, \forall i, j \tag{6.15c}$$

$$0 \leq D_{ij}^{rgb} \leq 1, \forall i, j \tag{6.15d}$$

$$A_{v\ ij} \geq 0, \forall i, j \tag{6.15e}$$

$$A_{nv\ ij} \geq 0, \forall i, j \tag{6.15f}$$

$$\mathbf{1^T}(A_v + A_{nv}) = \mathbf{1^T}, \tag{6.15g}$$

$$\|A_v + A_{nv}\|_0 \leq s, \tag{6.15h}$$

with $\|\cdot\|_F$ denoting the Frobenius norm, and $\|\cdot\|_0$ the ℓ_0 norm, i.e., the number of non-zero elements of the given matrix. The first term in Equation 6.15a represents a measure of the fidelity to the sub-sampled XRF data X, the second term is the total variation (TV) regularizer of the visible component, the third term is the TV regularizer of the non-visible component and the last term is the fidelity to the observed RGB image I. The TV regularizer of the visible component $\nabla_I(D_v^{xrf} A_v)$ is defined as

$$
\begin{aligned}
&\|\nabla_I(D_v^{xrf}A_v)\|_F^2 \\
&= \sum_{i=1}^{H-1}\sum_{j=1}^{W-1} w_{i,j}^{down}\|D_v^{xrf}\bar{A}_v(i,j,:) - D_v^{xrf}\bar{A}_v(i+1,j,:)\|_2^2 \\
&\qquad + w_{i,j}^{right}\|D_v^{xrf}\bar{A}_v(i,j,:) - D_v^{xrf}\bar{A}_v(i,j+1,:)\|_2^2 \\
&= \|D_v^{xrf}A_v P(I)\|_F^2
\end{aligned}
\tag{6.16}
$$

where $\bar{A}_v \in \mathbb{R}^{W\times H\times M}$ is the $3D$ volume version of A_v and $\bar{A}_v(i,j,:) \in \mathbb{R}^M$ is the non-visible component abundance of pixel (i,j). $w_{i,j}^{down}$ and $w_{i,j}^{right}$ are the adaptive TV weights in the vertical and horizontal directions, respectively, that is,

$$
w_{i,j}^{down} = e^{-\alpha\|\bar{I}(i,j,:)-\bar{I}(i+1,j,:)\|_2^2}, \tag{6.17}
$$

$$
w_{i,j}^{right} = e^{-\alpha\|\bar{I}(i,j,:)-\bar{I}(i,j+1,:)\|_2^2}, \tag{6.18}
$$

where $\bar{I}(i,j,:)$ is the RBG image pixel at position (i,j). $P(I) \in \mathbb{R}^{N_h\times((W-1)(H-1))}$ in Equation 6.16 is the adaptive horizontal/vertical first order difference operator, determined by the input RGB image I, according to Equation 6.17, 6.18. Equations 6.17, 6.18 indicate that the TV regularizer of the visible component is adaptive to the conventional RGB image $\bar{I}$. When the difference between two adjacent RGB pixels is small, a strong spatial smoothness constraint is placed on their corresponding XRF pixels, and vice versa. This adaptive TV regularizer is one of the main differences between this fusion based XRF image inpainting algorithm and our previous fusion based XRF image SR algorithm [**29**]. We found out that such TV regularizer on the visible component is essential for the inpainting problem, otherwise the inpainting results are not satisfactory. For the SR approach, we do not need such a TV regularizer on the visible component. The SR degradation model assumes that

the LR measured XRF image is a weighted sum of all the pixels in the target HR XRF image, so there is an implicit spatial smoothness constraint imposed by the LR XRF image. However, for the XRF image inpainting problem, we sub-sample the XRF image to obtain the measurement, so that many pixels are not sampled at all, making the reconstruction more difficult than for the SR problem. Also the mapping from RGB to XRF pixels is one to many, meaning that utilizing the RGB image can not guarantee a spatially smooth XRF reconstruction.

To estimate the HR visible component abundance A_v, the HR RGB image I can provide details in the spatial domain. However, to estimate the HR non-visible component abundance A_{nv}, there is no additional spatial information, so we need the TV regularizer (Equation 6.19) to impose spatial smoothness on the non-visible component. The TV regularizer for the non-visible component $\nabla(D_{nv}^{xrf} A_{nv})$ in Equation 6.15 is defined as

$$\begin{aligned} &\|\nabla(D_{nv}^{xrf} A_{nv})\|_F^2 \\ &= \sum_{i=1}^{H-1} \sum_{j=1}^{W-1} \|D_{nv}^{xrf} \bar{A}_{nv}(i,j,:) - D_{nv}^{xrf} \bar{A}_{nv}(i+1,j,:)\|_2^2 \\ &\qquad + \|D_{nv}^{xrf} \bar{A}_{nv}(i,j,:) - D_{nv}^{xrf} \bar{A}_{nv}(i,j+1,:)\|_2^2 \\ &= \|D_{nv}^{xrf} A_{nv} Q\|_F^2, \end{aligned} \tag{6.19}$$

where $\bar{A}_{nv} \in \mathbb{R}^{W \times H \times M}$ is the $3D$ volume version of A_{nv} and $\bar{A}_{nv}(i,j,:) \in \mathbb{R}^M$ is the non-visible component abundance of pixel (i,j). $Q \in \mathbb{R}^{N_h \times ((W-1)(H-1))}$ is the horizontal/vertical first order difference operator. There is no additional spatial information to estimate the non-visible component abundance A_{nv} of the full sampled XRF image, so an homogeneous TV regularizer (Equation 6.9) is imposed. The TV regularization parameters γ and λ in

Equation 6.15 control the spatial smoothness of the reconstructed visible and non-visible components, respectively.

The constraint Equations 6.15e, 6.15f, 6.15g together restrict the abundances of visible and non-visible components, and also act as a sparsity prior on the per-pixel abundances, since they bound the ℓ_1 norm of the combined abundances $(A_v + A_{nv})$ to be 1 for all pixels. The last constraint Equation 6.15h is an optional constraint, which further enforces the sparsity of the combined abundance $(A_v + A_{nv})$.

The optimization in Equation 6.15 is non-convex and difficult to carry out if we are to optimize over all the parameters A_v, A_{nv}, D^{rgb}, D_v^{xrf} and D_{nv}^{xrf} directly. We found it effective to alternatively optimize over these parameters. Also because Equation 6.15 is highly non-convex, good initialization is needed. A similar approach to the coupled dictionary learning technique in [**119**, **120**] is applied here to initialize these parameters.

6.3.2.1. Initialization. Let $Y^{(0)} \in \mathbb{R}^{B \times N_h}$ be the initialization of Y. Such initialization can be obtained by utilizing some standard image inpainting algorithms [**44**, **127**] to inpaint the sub-sampled XRF image slice by slice. Then the coupled dictionary learning technique in [**119**, **120**] can be utilized to initialize D^{rgb} and D_v^{xrf} by

$$\begin{aligned} \min_{D^{rgb}, D_v^{xrf}} \quad & \|I - D^{rgb} A_v\|_F^2 + \|Y^{(0)} - D_v^{xrf} A_v\|_F^2 \\ & + \eta \sum_{k=1}^{N_l} \|A_v(:,k)\|_1, \\ \text{s.t.} \quad & \|D^{rgb}(:,k)\|_2 \leq 1, \forall k, \\ & \|D_v^{xrf}(:,k)\|_2 \leq 1, \forall k, \end{aligned} \tag{6.20}$$

where $\|\cdot\|_1$ is the ℓ_1 vector norm, parameter β controls the sparseness of the coefficients in A_v, A_v(:,k), $D^{rgb}(:,k)$ and $D_v^{xrf}(:,k)$ denote the k^{th} column of matrices A_v, D^{rgb}, and D_v^{xrf},

respectively. Details of the optimization can be found in [**119**, **120**]. D^{rgb} and D_v^{xrf} are initialized using Equation 6.20 and D_{nv}^{xrf} is initialized to be equal to D_v^{xrf}. A_v is initialized by Equation 6.20 as well, while A_{nv} is set equal to zero at initialization.

6.3.2.2. Optimization Scheme. We propose to alternatively optimize over all the parameters in Equation 6.8a. First we optimize over A_v and A_{nv} by fixing all other parameters,

$$\begin{aligned}
\min_{A_v, A_{nv}} \quad & \|X - D_v^{xrf} A_v S - D_{nv}^{xrf} A_{nv} S\|_F^2 \\
& + \gamma \|\nabla_I (D_v^{xrf} A_v)\|_F^2 + \lambda \|\nabla (D_{nv}^{xrf} A_{nv})\|_F^2 \\
& + \|I - D^{rgb} A_v\|_F^2 \\
\text{s.t.} \quad & A_{v\ ij} \geq 0, \forall i, j \\
& A_{nv\ ij} \geq 0, \forall i, j \\
& \mathbf{1^T}(A_v + A_{nv}) = \mathbf{1^T}, \\
& \|A_v + A_{nv}\|_0 \leq s.
\end{aligned} \tag{6.21}$$

PALM (proximal alternating linearized minimization) algorithm [**17**] is utilized to optimize over A_v and A_{nv} by a projected gradient descent method. For Equation 6.21, the following three steps are iterated for $q = 1, 2, \ldots$ until convergence:

$$V_v^q = A_v^{q-1} - \frac{1}{d_v} D^{rgb^T} (D^{rgb} A_v^{q-1} - I) \tag{6.22a}$$

$$\begin{aligned}
V_{nv}^q = & A_{nv}^{q-1} \\
& - \frac{1}{d_{nv}} (D_{nv}^{xrf^T} (D_{nv}^{xrf} A_{nv}^{q-1} S - (X - D_v^{xrf} A_v^{q-1} S)) S^T \\
& + \lambda D_{nv}^{xrf^T} D_{nv}^{xrf} A_{nv} Q Q^T)
\end{aligned} \tag{6.22b}$$

$$\{A_v^q, A_{nv}^q\} = prox_{A_v, A_{nv}}(V_v^q, V_{nv}^q), \tag{6.22c}$$

where $d_1 = \beta_1 \| D^{rgb} {D^{rgb}}^T \|_F$, $d_2 = \beta_2 \| D_{nv}^{xrf} {D_{nv}^{xrf}}^T \|_F$ are non-zero step size constants, and $prox_{A_v, A_{nv}}$ is the proximal operator that project V_v^q, V_{nv}^q onto the constraints of Equation 6.21. The proximal projection is computational inexpensive because it just sets negative entries of V_v^q and V_{nv}^q to zero and scales every column of V_v^q and V_{nv}^q simultaneously to equal one in ℓ_1 norm. Notice that in Equation 6.22a, only the gradient of the second term in Equation 6.13 is utilized to update V_v^q, because we want the visible component coefficients A_v to be determined by the RGB image I only, instead of being determined jointly by the RGB image I and the XRF image X, to obtain spatially sharp estimation of the visible component coefficient A_v.

Second, we optimize over D^{rgb} solving the following constrained least-squares problem:

$$\begin{aligned} \min_{D^{rgb}} \quad & \| I - D^{rgb} A_v \|_F^2 \\ \text{s.t.} \quad & 0 \leq D_{ij}^{rgb} \leq 1, \forall i, j. \end{aligned} \tag{6.23}$$

Likewise, Equation 6.23 is minimized by iterating the following steps until convergence:

$$E^q = {D^{rgb}}^{q-1} - \frac{1}{d_{rgb}} ({D^{rgb}}^{q-1} A_v - I) {A_v}^T \tag{6.24a}$$

$${D^{rgb}}^q = prox_{D^{rgb}}(E^q), \tag{6.24b}$$

with $d_{rgb} = \beta_3 \| A_v {A_v}^T \|_F$ again a non-zero step size constant and $prox_{D^{rgb}}$ the proximal operator that projects E^q onto the constraint of Equation 6.23. The proximal operator this time is also computationally inexpensive since it just truncates the entries of E^q to 0 from below and to 1 from above.

Similarly, D_v^{xrf} is then optimized by solving

$$\begin{aligned} \min_{D_v^{xrf}} \quad & \|(X - D_{nv}^{xrf} A_{nv} S) - D_v^{xrf} A_v S\|_F^2 \\ & + \gamma \|\nabla_I (D_v^{xrf} A_v)\|_F^2 \\ \text{s.t.} \quad & 0 \leq {D_v^{xrf}}_{ij} \leq 1, \forall i, j, \end{aligned} \tag{6.25}$$

using the following iteration steps:

$$\begin{aligned} U^q = {} & {D_v^{xrf}}^{q-1} \\ & - \frac{1}{d_v^{xrf}} ({D_v^{xrf}}^{q-1} A_v S - (X - D_{nv}^{xrf} A_{nv} S)) S^T {A_v}^T \\ & - \gamma D_v^{xrf} A_v P(I) P(I)^T {A_v}^T \end{aligned} \tag{6.26a}$$

$$D_v^{xrfq} = prox_{D_v^{xrf}}(U^q), \tag{6.26b}$$

where $d_v^{xrf} = \beta_4 \|A_v {A_v}^T\|_F$ is the non-zero step size constant and $prox_{D_v^{xrf}}$ is the proximal operator which project U^q onto the constraints of Equation 6.17. It is the same as the proximal operator in Equation 6.24b.

Finally, we optimize D_{nv}^{xrf} by solving the following problem,

$$\begin{aligned} \min_{D_{nv}^{xrf}} \quad & \|(X - D_v^{xrf} A_v S) - D_{nv}^{xrf} A_{nv} S\|_F^2 \\ & + \lambda \|\nabla (D_{nv}^{xrf} A_{nv})\|_F^2 \\ \text{s.t.} \quad & 0 \leq {D_{nv}^{xrf}}_{ij} \leq 1, \forall i, j. \end{aligned} \tag{6.27}$$

Likewise, the following two steps are iterated until convergence:

$$
\begin{aligned}
L^q = {} & {D_{nv}^{xrf}}^{q-1} \\
& - \frac{1}{d_{nv}^{xrf}}({D_{nv}^{xrf}}^{q-1} A_{nv} S - (X - D_v^{xrf} A_v S)) S^T {A_{nv}}^T \\
& - \lambda D_{nv}^{xrf} A_{nv} Q Q^T {A_{nv}}^T
\end{aligned} \tag{6.28a}
$$

$$
D_{nv}^{xrf\,q} = prox_{D_{nv}^{xrf}}(L^q), \tag{6.28b}
$$

where $d_{nv}^{xrf} = \beta_5 \| A_n v A_n v^T \|_F$ again is a non-zero step size constant, $prox_{D_{nv}^{xrf}}$ is the proximal operator projecting L^q onto the constraints of Equation 6.19, which is the same proximal operator as the ones in Equations 6.16b and 6.18b. The complete optimization scheme is illustrated in Algorithm 4. According to Equations 6.8, 6.9, 6.10, the reconstructed full-sampled XRF output image Y can be computed by

$$
Y = Y_v + Y_{nv} = D_v^{xrf} A_v + D_{nv}^{xrf} A_{nv}. \tag{6.29}
$$

6.4. Experimental Results

In the experimental results section, we will first show the advantages of our proposed adaptive sampling mask generation CNN in RGB image inpainting task, followed by the performance of the proposed fusion based XRF inpainting algorithm in XRF image inpainting task.

6.4.1. Adaptive Sampling Mask for RGB Image Inpainting

For the RGB image inpainting task, we show the benefits of our proposed adaptive sampling mask, compared to the random sampling mask.

Algorithm 4. Proposed Optimization Scheme of Equation 6.15

input: Sub-sampled XRF image X, A conventional RGB image I.

Initialize $Y^{(0)}$ by inpainting X slice by slice;
Initialize $D^{rgb^{(0)}}$, $D_v^{xrf^{(0)}}$ and ${A_v}^{(0)}$ by Equation 6.20;
Initialize $D_{nv}^{xrf^{(0)}} = D_v^{xrf^{(0)}}$;
Initialize ${A_{nv}}^{(0)} = \mathbf{0}$;
$n = 0$;
repeat
  Estimate ${A_v}^{(n+1)}$ and ${A_{nv}}^{(n+1)}$ with Equation 6.21;
  Estimate $D^{rgb^{(n+1)}}$ with Equation 6.23;
  Estimate $D_v^{xrf^{(n+1)}}$ with Equation 6.25;
  Estimate $D_{nv}^{xrf^{(n+1)}}$ with Equation 6.27;
  n=n+1;
until convergence

output: Full-sampled XRF image
$Y = D_v^{xrf} A_v + D_{nv}^{xrf} A_{nv}$.

6.4.1.1. Datasets. To train our proposed adaptive sampling mask generation CNN in Section 6.2, ImageNet [**33**] database, without any of the accompanying labels, is used. We randomly select $1,000,000$, 100 and 100 images as the training, validation and test set, respectively. All the images are selected randomly among all categorizes, to capture as diverse image structures as possible. All the images are cropped to have spatial resolution 64×64.

6.4.1.2. Implementation Details. Our proposed adaptive sampling mask generation CNN (Section 6.2) is implemented in Torch. ADAM [**64**] is applied as the stochastic gradient descent solver for optimization. We use the same hyper-parameters suggest in [**45**] and batch size equal to 100 during the training. We pick $c = 0.2$ for the sampling percentage parameter. A 20% sampling percentage roughly speedup the XRF image scanning procedure by a factor of 5. 400 epochs are applied during the training process.

In the training, we first initialize the inpainting network $NetE$ according to [**45**]. Random sampling mask with $c = 0.2$ is utilized to corrupt the input RGB image. The mask generation

network $NetM$ is initialized randomly. We then train the whole network architecture in Figure 6.2. The learning rate of the mask generation network $NetM$ is set to be 0.0002 during training. The learning rate of the inpainting network $NetE$ is set to be 0, i.e., we fix $NetE$ when training $NetM$. If we optimize $NetM$ and $NetE$ simultaneously, although the whole network structure in Figure 6.2 will become optimal in reconstructing the input image z, $NetM$ and $NetE$ will be coupled with each other. Notice that the channel-wise fully connected layer in $NetE$ (Figure 6.3) is able to learn the high-level feature mapping, making $NetE$ be able to perform semantic image inpainting. However, we would like to utilize other general image inpainting algorithms to perform the inpainting reconstruction, not only $NetE$, and to make the adaptive sampling mask be general to as many image inpainting algorithms as possible. By fixing $NetE$, which is pre-trained by random sampling masks, $NetM$ is then constrained to be optimized to general image inpainting problem instead of semantic image inpainting problem. Better adaptive sampling mask generation CNN for general image inpainting problem could be trained by utilizing this training procedure.

6.4.1.3. Performance on ImageNet Testing Images. To compare our adaptive sampling mask with the random sampling mask, we apply both sampling to corrupt those 100 testing images from ImageNet database. The sampling rate is $c = 0.2$ for both sampling masks. For image inpainting algorithms, $NetE$ Inpainting [**45**], Harmonic Inpainting [**22**], Mumford-Shah Inpainting [**44**] and BPFA inpainting [**127**], are used to reconstruct the full-sampled RGB images.

The average PSNR and the average SSIM [**111**] metric over all 100 test images are shown in Figure 6.5 and Figure 6.6, respectively. First, we observe that the adaptive sampling mask outperform random sampling mask consistently over all inpainting reconstruction

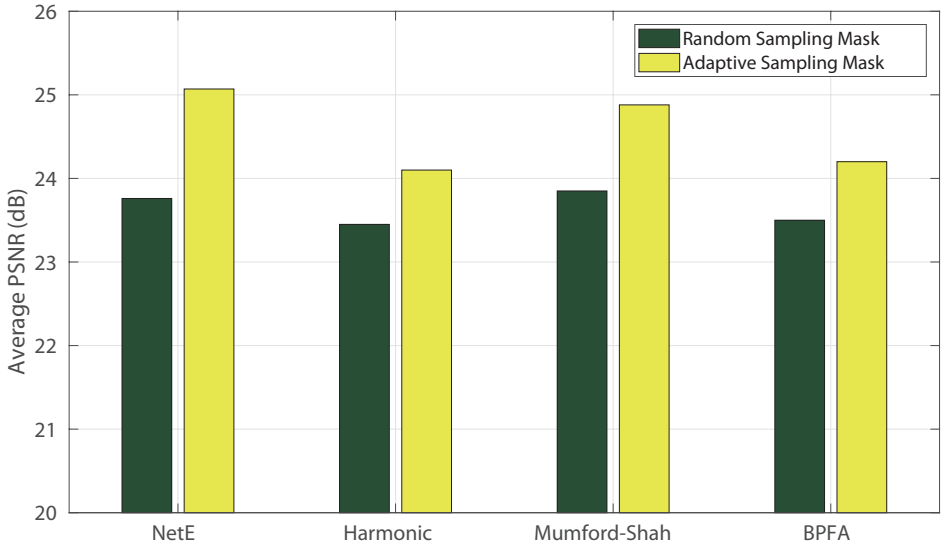


Figure 6.5. Average PSNR over all 100 test images from ImageNet database for several inpainting methods between random sampling mask and adaptive sampling mask.

algorithm by around 1 dB in PSNR and 0.02 in SSIM, showing the effectiveness of our proposed adaptive sampling mask generation network. Furthermore, we observe that the DNN based inpainting network $NetE$ demonstrates the highest PSNR and SSIM values and the largest improvement from random sampling to adaptive sampling among all the inpainting algorithms.

The visual quality comparison of the adaptive sampling mask and random sampling mask is shown in Figure 6.7. 3 test images are picked from the total 100 image testing set. The advantages of the proposed adaptive sampling mask over the random sampling mask can be observed by comparing the reconstruction results of the same inpainting algorithm over these two sampling strategies. For the test image #39, the adaptive sampling mask are able to capture the white dots in the red hat, resulting accurate reconstruction results of those white dots. In test image #89, adaptive sampling mask spends more sampling pixels in the

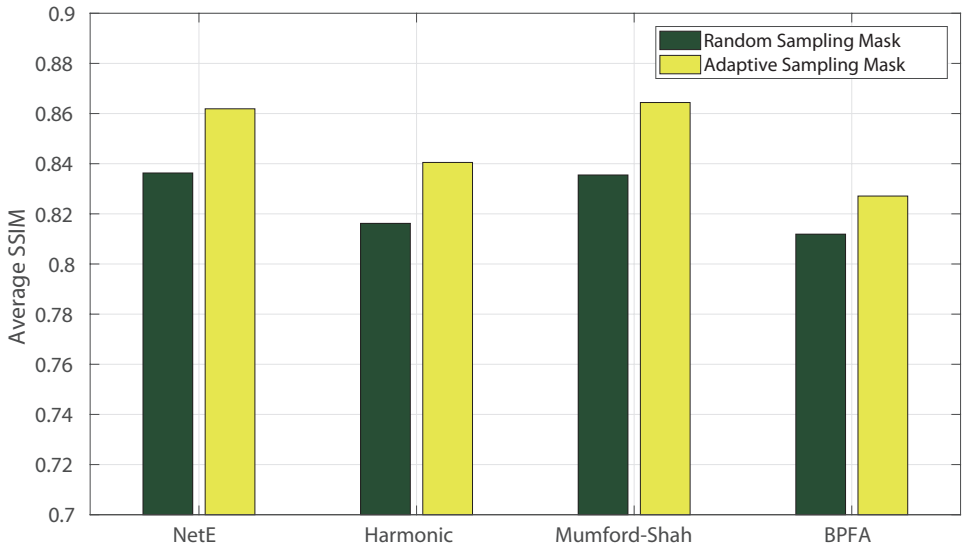


Figure 6.6. Average SSIM over all 100 test images from ImageNet database for several inpainting methods between random sampling mask and adaptive sampling mask.

high-frequency foreground object, resulting a better reconstruction of the foreground object. The reconstruction of the low-frequency background object is not as good as the random sampling mask since the number of the sampled pixels in background area is small. In test image #91, the adaptive sampling mask samples densely on the contour of the bird, resulting better reconstruction result of the contour. The advantages of the adaptive sampling mask over random sampling mask is consistent over all the inpainting reconstruction algorithms.

6.4.1.4. Performance on Painting Images. We also tested our proposed adaptive sampling mask on painting images. As shown in Figure 6.8 (a) and Figure 6.9 (b), two RGB images of the painting "Bloemen en insecten" and the part of the painting "Bedroom" are tested. The "Bloemen en insecten" image has spatial resolution 580×680 and part of the "Bedroom" image has spatial resolution 475×475. Random sampling mask and adaptive sampling mask is generated, as shown in Figure 6.8(b), Figure 6.9 (b) and Figure 6.8(c),

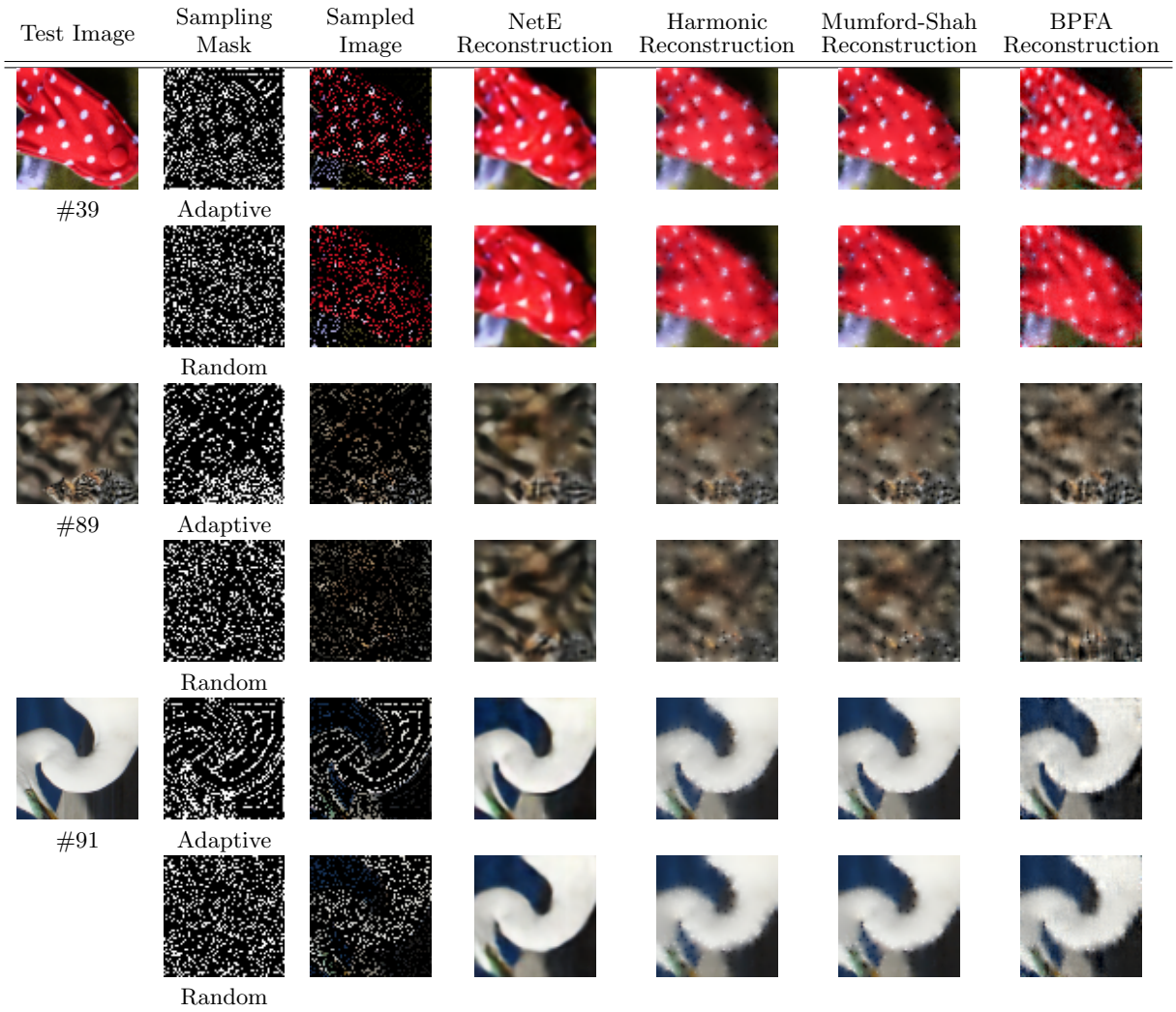


Figure 6.7. Visual Comparison of the reconstructed images using adaptive sampling mask and random sampling mask. The first column is the input test image and the second column is the sampling mask, either adaptive or random, the third column is the sampled image obtained by the sampling mask and the rest columns are the reconstruction result of $NetE$ Inpainting [**45**], Harmonic Inpainting [**22**], Mumford-Shah Inpainting [**44**] and BPFA inpainting [**127**], respectively.

Figure 6.9 (c). Harmonic Inpainting [**22**], Mumford-Shah Inpainting [**44**] algorithms are utilized to reconstructed the sampled RGB images, and the reconstruction results are shown in Figure 6.8 (d)-(g) and Figure 6.9 (d)-(g), with the PSNR values. $NetE$ Inpainting [**45**] is

not applied here for the inpainting reconstruction because $NetE$ is trained to inpaint RGB images with spatial resolution 64×64. The network structure shown in Figure 6.3 is not fully convolutional as there is the channel-wise fully connected layer in the middle. BPFA [**127**] inpainting is not applied here for the inpainting reconstruction because the algorithm is extremely slow on large scale images. By comparing the column of random sampling and the column of adaptive sampling, it can be concluded that our proposed adaptive sampling mask outperform the random sampling mask, in both visual quality of the reconstructed images and the PSNR values.

6.4.2. Adaptive Sampling Mask for X-Ray Fluorescence Image Inpainting

In the previous section (Section 6.4.1), we demonstrate the effectiveness of our proposed adaptive sampling mask in RGB image inpainting problem. To further verify the performance of the adaptive sampling mask and evaluate the performance of our proposed fusion based inpainting algorithm (Section 6.3), we have performed experiments on both synthetic and real XRF images. The basic parameters of the proposed SR method are set as follows: the number of atoms in the dictionaries D^{rgb}, D_{nv}^{xrf} and D_{v}^{xrf} is $M = 50$ for synthetic experiments and $M = 200$ for real experiments; $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 1.01$, which only affects the speed of convergence; parameter λ and γ in Equation 6.15 are set to be 0.1; parameter α in Equation 6.17 and Equation 6.18 is set to be 16 and η in Equation 6.20 is set to be 0.02. The optional constraint in Equation 6.15h is not applied here.

6.4.2.1. Error Metrics. As a primary error metric we use, the root mean squared error (RMSE) between the estimated full-sampled XRF image Y and the ground truth image Y^{gt} is computed Harmonic Inpainting [**22**], Mumford-Shah Inpainting [**44**] and BPFA inpainting [**127**]. Notice that to inpaint all the channels in the sub-sampled XRF image,

$$RMSE = \sqrt{\frac{\|Y - Y^{gt}\|_F^2}{BN_h}}. \tag{6.30}$$

The peak-signal-to-noise ratio (PSNR) is reported as well,

$$PSNR = 20 \log_{10} \frac{\max(Y^{gt})}{RMSE}, \tag{6.31}$$

where $\max(Y^{gt})$ denoting the maximum entry of Y^{gt}.

The spectral angle mapper (SAM, [**122**]) in degrees is also utilized, which is defined as the angle in $\mathbb{R}^B$ between an estimated pixel and the ground truth pixel, averaged over the whole image,

$$SAM = \frac{1}{N_h} \sum_{j=1}^{N_h} \arccos \frac{Y(:,j)^T Y^{gt}(:,j)}{\|Y(:,j)\|_2 \|Y^{gt}(:,j)\|_2}. \tag{6.32}$$

6.4.2.2. Comparison Methods. According to our knowledge, no work has been done on solving XRF (or Hyperspectral) image inpainting problem by fusing a conventional RGB image. So we can only compare with those traditional image inpainting algorithms, such as Harmonic Inpainting [**22**], Mumford-Shah Inpainting [**44**] and BPFA inpainting [**127**]. Harmonic Inpainting and Mumford-Shah inpainting methods are for image inpainting so we have to inpaint the XRF image channel by channel. BPFA inpainting [**127**] is able to inpaint multiple channels simultaneously.

6.4.2.3. Synthetic Experiment. We evaluate the inpainting results for different methods with a synthetic experiment first. We combined 3 noise free spectra with a significant amount of spectral overlap (50×1), an airforce target image ($345 \times 490 \times 3$) as the visible image and a rectangle image ($345 \times 490 \times 3$) as the non-visible image to simulate the ground truth HR XRF image Y^{gt} ($345 \times 490 \times 1024$). The 3 noise free spectra, the airforce target image and

the rectangle image are shown in Figs. 6.10, 6.11 (a) and 6.11 (b), respectively. In detail, we assume that the yellow foreground of the airforce target image corresponds to spectrum # 1, the blue background of the airforce image corresponds to spectrum #2 and the white foreground of the rectangle image corresponds to spectrum #3. The sampling rate c is set to be 0.2 for the inpainting problem. The adaptive sampling mask is generated according to the RGB image in Figure 6.11 (a). Mumford-Shah Inpainting algorithm [**44**] is utilized to initialize the reconstruction slice by slice.

The RMSE, PSNR and SAM metrics were computed between the inpainting results of different inpainting algorithms, under either random sampling mask or adaptive sampling mask. The default parameters of methods Harmonic Inpainting [**22**], Mumford-Shah Inpainting [**44**] and BPFA inpainting [**127**] in their original paper were applied in our synthetic experiments. As shown in Table 6.1, our proposed fusion inpainting method with adaptive sampling has the smallest RMSE, highest PSNR and smallest SAM. By comparing the reconstruction performance of the same inpainting algorithm with different sampling strategies, it can been seen that adaptive sampling usually outperforms random sampling. By comparing the reconstruction performance of the same sampling strategy with different inpainting algorithms, our proposed fusion based inpainting algorithm outperforms all the other inpainting algorithms. It can also be observed that the advantages of the adaptive sampling and the fusion based inpainting are additive, as the combination of the adaptive sampling and the fusion style inpainting has the best reconstruction performance.

We further validate the effectiveness of our proposed fusion based inpainting algorithm in Figure 6.12. The iteration process of the reconstructed full-sampled XRF image with respect to RMSE of both random sampling and adaptive sampling is shown here. Since Mumford-Shah inpainting [**44**] algorithm is used as initialization, we visualize its RMSE

Metric	Harmonic Random	Harmonic Adaptive	BPFA Random	BPFA Adaptive	Mumford-Shah Random	Mumford-Shah Adaptive	Proposed Random	Proposed Adaptive
RMSE	0.1166	0.1006	0.3334	0.3409	0.1107	0.0945	0.0981	**0.0808**
PSNR	18.67	19.94	15.44	15.94	19.11	20.49	20.17	**30.43**
SAM	5.55	4.74	33.91	36.83	4.50	3.68	4.63	**3.89**

Table 6.1. Experimental results on synthetic data comparing different inpainting methods, under both random and adaptive sampling strategies, discussed in Section 6.4.2.2 in terms of RMSE, PSNR and SAM. Best results are shown in bold.

as dashed horizontal lines, as baseline. It can be seen that for both random sampling and adaptive sampling, the proposed fusion based inpainting algorithm has smaller RMSE as the number of iteration increases. The RGB image of the inpainting target provides positive contribution to the inpainting reconstruction, according to our formulation in Section 6.3. Again, the adaptive sampling strategy has advantages over random sampling strategy, for both Mumford-Shah inpainting [**44**] algorithm and our proposed fusion based inpainting algorithm.

We compare the visual quality of different inpainting methods under both random sampling mask and adaptive sampling mask on the channel #6 in Figure 6.13. Because Harmonic Inpainting [**22**], Mumford-Shah Inpainting [**44**] and BPFA inpainting [**127**] do not utilize RGB image as guidance during the inpainting reconstruction, their inpainting results are not as sharp as our proposed fusion based inpainting algorithm. The appropriate reconstruction of the non-visible component of Figure 6.13 (i) and (j) shows that our proposed fusion based inpainting algorithm is able to handle hidden part of the scanning objects. Detailed validation on the non-visible component modeling can be found in [**29**]. Notice that the non-visible component in Figure 6.13 (i) and (j) is more blurry than the visible component (airforce target), since it is reconstructed by a TV regularizer and does not use any RGB image information. The sampling mask is not optimized for the non-visible component in

Figure 6.13 (j) as well. Notice that in Figure 6.13 (b), dense sampling pixels are applied on the high frequency region of the visible components, and the non-visible component are sampled sparsely.

6.4.2.4. Real Experiment. For our second real experiment, the real data was collected by a home-built X-ray fluorescence spectrometer (courtesy of Prof. Koen Janssens), with 2048 channels in spectrum. Studies from XRF image scanned from Jan Davidsz. de Heem's "Bloemen en insecten" (ca 1645), in the collection of Koninklijk Museum voor Schone Kunsten (KMKSA) Antwerp, are presented here. We utilize the super-resolved XRF image in our previous work [**29**] as the ground truth. The ground truth XRF image has dimension $680 \times 580 \times 2048$. We first extract 20 region of interest (ROI) spectrally and work on the extracted 20 XRF ROI, to decrease the spectral dimension from 2048 to 20. We have to decrease the spectral dimension so as to compare with other inpainting algorithms, since some algorithm [**22**, **44**] reconstruct the sub-sampled XRF image slice by slice and large spectral dimension will make the computational time very long. The sampling ratio c is set to be 0.2 again. Both random sampling strategy and adaptive sampling strategy are applied and analyzed. Then different inpainting methods are applied to reconstruct those sub-sampled XRF image.

Metric	Harmonic Random	Harmonic Adaptive	BPFA Random	BPFA Adaptive	Mumford-Shah Random	Mumford-Shah Adaptive	Proposed Random	Proposed Adaptive
RMSE	0.0195	0.0193	0.0176	0.0221	0.0184	0.0179	0.0168	**0.0160**
PSNR	34.19	34.30	35.29	33.56	34.70	34.93	35.48	**42.61**
SAM	2.18	2.29	2.01	2.26	1.99	1.92	1.81	**1.79**

Table 6.2. Experimental results on the "Bloemen en insecten" data comparing different inpainting methods, under both random and adaptive sampling strategies, discussed in Section 6.4.2.2 in terms of RMSE, PSNR and SAM. Best results are shown in bold.

As shown in Table 6.2, our proposed fusion based inpainting algorithm with the proposed adaptive sampling mask provides the closest reconstruction to the ground truth XRF image compared to all other methods. Our proposed algorithm utilize a conventional RGB image as guidance, which is full-sampled and has high contrast (Figure 6.8 (a)), resulting a better inpainting performance. By comparing the difference between column "Mumford-Shah Random" and column "Proposed Random", and the difference between column "Mumford-Shah Adaptive" and column "Proposed Adaptive", it can be concluded that the benefit gained by our proposed fusion based inpainting is large when the adaptive sampling mask is applied. For example, there is a $0.78dB$ improvement in PSNR by applying our proposed fusion based inpainting when random sampling mask is applied, while there is a $7.68dB$ improvement in PSNR when adaptive sampling mask is applied. This is because the adaptive sampling mask sampled the XRF image efficiently for the visible component and the fusion inpainting propagate the measured XRF pixels properly.

The iteration process of our proposed fusion inpainting algorithm, similar to Figure 6.12, is shown in Figure 6.14. Notice that at the beginning iterations of our proposed fusion inpainting algorithm, the RMSE is higher than the Mumford-Shah inpainting algorithm. This is because we decompose the inpainting result of Mumford-Shah inpainting algorithm by sparse representation, according to Equation 6.20. Due to complexity of the "Bloemen en insecten" data, we loose some accuracy at the first few iterations. However, with the iteration going on, the RMSE of both random sampling and adaptive sampling decreases and becomes smaller than the RMSE of Mumford-Shah inpainting algorithm.

The visual quality of different inpainting algorithms and sampling strategies on channel #16, corresponding to $Pb\ L\eta$ XRF emission line, is compared in Figure 6.15. The adaptive sampling mask is generated according to the RGB image in Figure 6.8 (a). The same random

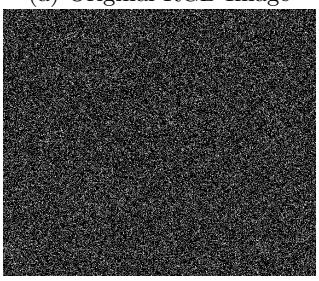
and adaptive sampling masks as the sampling masks in Figure 6.8 (b) and Figure 6.8 (c) are applied here. Notice that the two long rectangles in the ground truth XRF image Figure 6.15 (a) are the stretcher bars under the canvas, which is not visible on the RGB image. The reconstruction results (c) (e) (g) of both Harmonic Inpainting [**22**], Mumford-Shah Inpainting [**44**] and BPFA inpainting [**127**] based on the adaptive sampling mask are sharper than those results (b) (d) (f) based on the random sampling mask. This is because the majority of the XRF signal in the "Bloemen en insecten" data have correlation to the RGB signal, and the adaptive sampling mask, which is optimal to the RGB image, would also be optimal to the visible component of the XRF signal. The proposed fusion inpainting algorithm further improves the contrast and resolves more fine details in (i). The reconstructed stretcher bars in all cases are blurry compared to other objects, because it does not utilize any information from the RGB image, both in sampling and inpainting. When compared to the ground truth image (a), we can conclude that those resolved details have high fidelity to the ground truth image (a).

6.5. Conclusion

In this paper we presented a novel adaptive sampling mask generation algorithm based on CNN and a novel XRF image inpainting framework based on fusing a conventional RGB image. For the adaptive sampling mask generation, we trained the mask generation network $NetM$ along with the inpainting network $NetE$, to obtain optimal binary sampling mask based on the input RGB image. For the fusion based XRF image inpainting algorithm, the XRF spectrum of each pixel is represented by an endmembers dictionary, as well as the RGB spectrum. The input sub-sampled XRF image is decomposed into visible and non-visible components, making it possible to find the non-linear mapping from RGB spectrum to XRF

spectrum. Total variation regularizer is applied on both visible and non-visible components, to ensure the spatial smoothness of the reconstructed XRF image. The reconstructed full-sampled visible XRF component and the full-sampled non-visible XRF component are combined to obtain the final full-sampled XRF image. Both synthetic and real experiments show the effectiveness of our proposed methods.

(a) Original RGB Image



(b) Random Sampling Mask

(c) Adaptive Sampling Mask

(d) Harmonic Reconstruction of Randomly Sampled Image PSNR: 26.60 dB

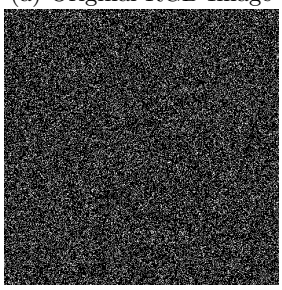
(e) Harmonic Reconstruction of Adaptively Sampled Image PSNR: 28.37 dB

(f) Mumford-Shah Reconstruction of Randomly Sampled Image PSNR: 26.97 dB

(g) Mumford-Shah Reconstruction of Adaptively Sampled Image PSNR: 29.05 dB

Figure 6.8. Visualization of the Inpainting result of the "Bloemen en insecten" painting. (a) is the original RGB image. (b) and (c) is the random sampling mask and the adaptive sampling mask, respectively. (d) and (f) is the reconstruction results of the randomly sampled image, using Harmonic and Mumford-Shah inpainting algorithms. (e) and (g) is the reconstruction results of the adaptively sampled image, using Harmonic and Mumford-Shah inpainting algorithms. Readers are suggested to zoom in in order to compare the details of different results.

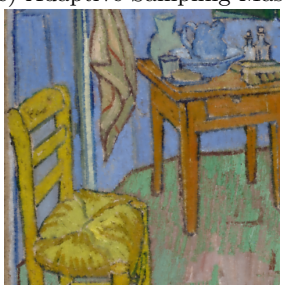
(a) Original RGB Image



(b) Random Sampling Mask

(c) Adaptive Sampling Mask

(d) Harmonic Reconstruction of Randomly Sampled Image
PSNR: 25.42 dB



(e) Harmonic Reconstruction of Adaptively Sampled Image
PSNR: 26.00 dB

(f) Mumford-Shah Reconstruction of Randomly Sampled Image
PSNR: 25.79 dB

(g) Mumford-Shah Reconstruction of Adaptively Sampled Image
PSNR: 26.53 dB

Figure 6.9. Visualization of the Inpainting result of part of the "Bedroom" painting. (a) is the original RGB image. (b) and (c) is the random sampling mask and the adaptive sampling mask, respectively. (d) and (f) is the reconstruction results of the randomly sampled image, using Harmonic and Mumford-Shah inpainting algorithms. (e) and (g) is the reconstruction results of the adaptively sampled image, using Harmonic and Mumford-Shah inpainting algorithms. Readers are suggested to zoom in in order to compare the details of different results.

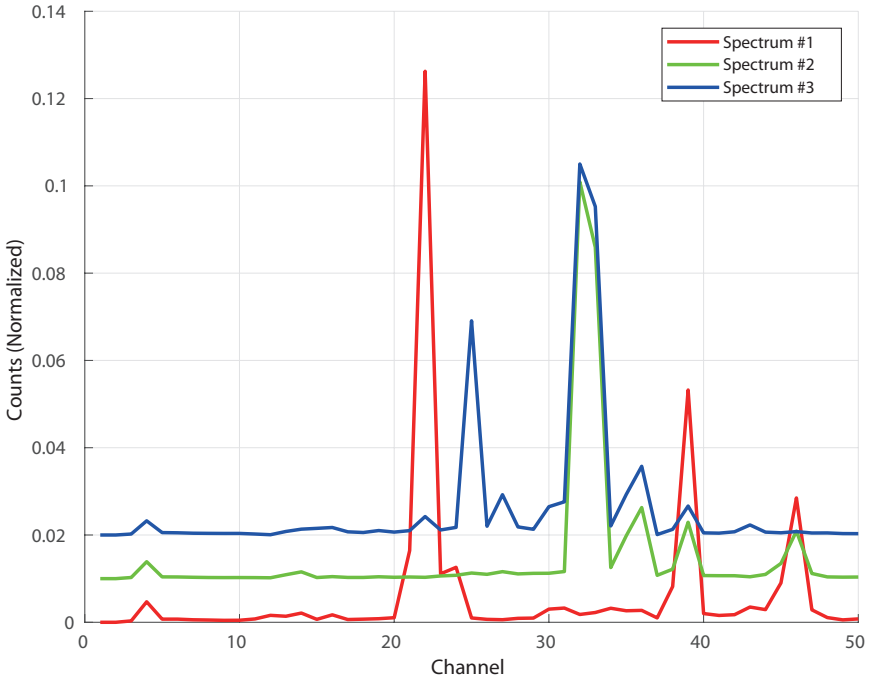
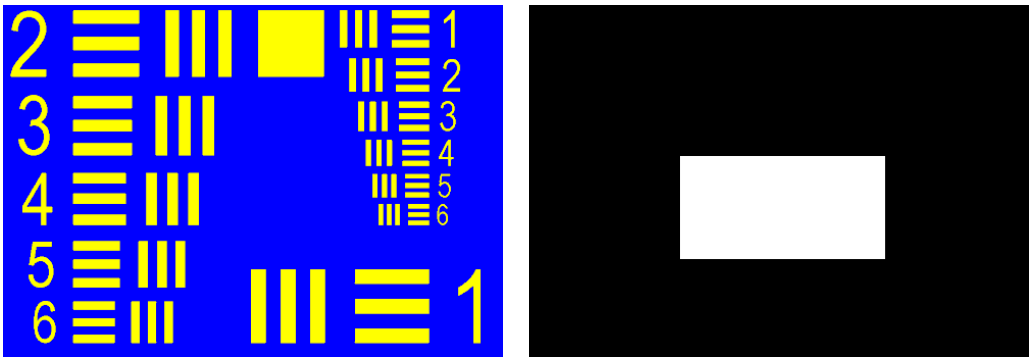


Figure 6.10. Three noise free spectra used to synthesize the full-sampled XRF image. Spectra # 2 and # 3 are shifted vertically (by 0.01 and 0.02, respectively) for visualization purposes.



(a) (b)

Figure 6.11. (a) The airforce image is utilized as the visible component. (b) The rectangle image is utilized as the non-visible component.

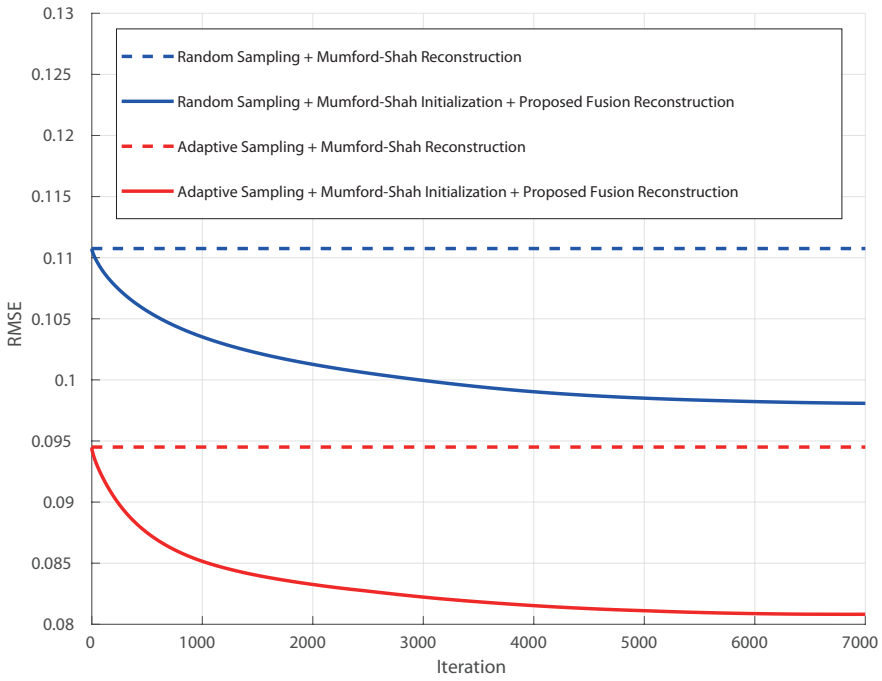
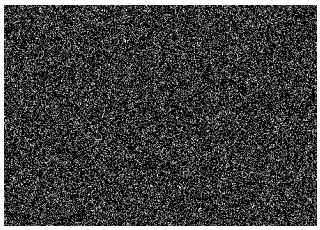
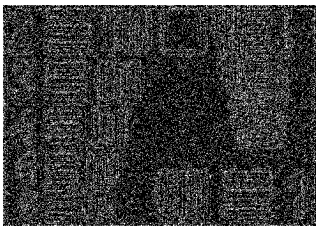


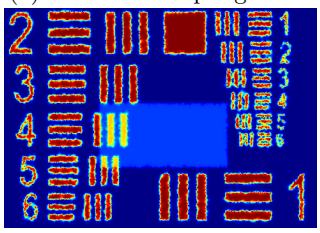
Figure 6.12. The iteration process of our proposed fusion based inpainting algorithm on the synthetic data. Mumford-Shah inpainting algorithm is utilized as initialization of our proposed algorithm. The iteration process of both random sampling and adaptive sampling shows that our proposed fusion based inpainting algorithm minimize the RMSE during the iteration.



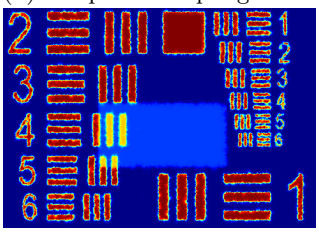
(a) Random Sampling Mask



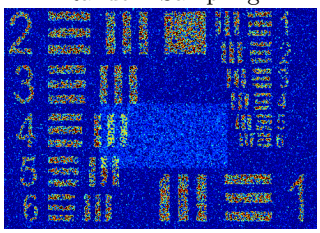
(b) Adaptive Sampling Mask



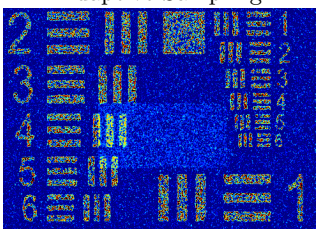
(c) Harmonic Reconstruction Random Sampling



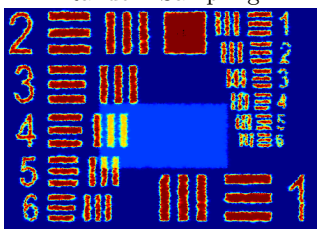
(d) Harmonic Reconstruction Adaptive Sampling



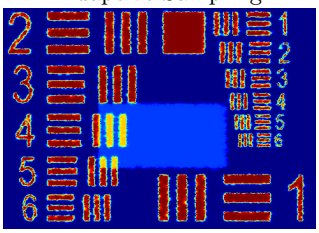
(e) BPFA Reconstruction Random Sampling



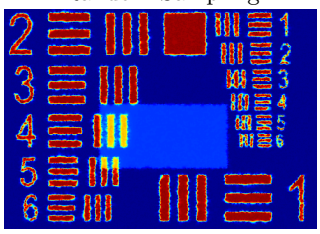
(f) BPFA Reconstruction Adaptive Sampling



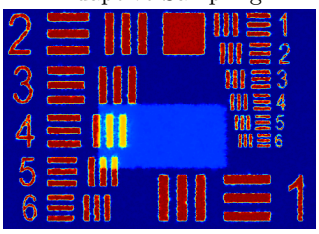
(g) Mumford-Shah Reconstruction Random Sampling



(h) Mumford-Shah Reconstruction Adaptive Sampling



(i) Proposed Reconstruction Random Sampling



(j) Proposed Reconstruction Adaptive Sampling

Figure 6.13. Visualization of the Inpainting result of the synthetic experiment. Channel #6 is selected. (a) is the random sampling mask. (b) is the adaptive sampling mask. (c)-(j) are the reconstruction results of different inpainting algorithms, for both randomly sampled XRF image and adaptively sampled XRF image. Readers are suggested to zoom in in order to compare the details of different results.

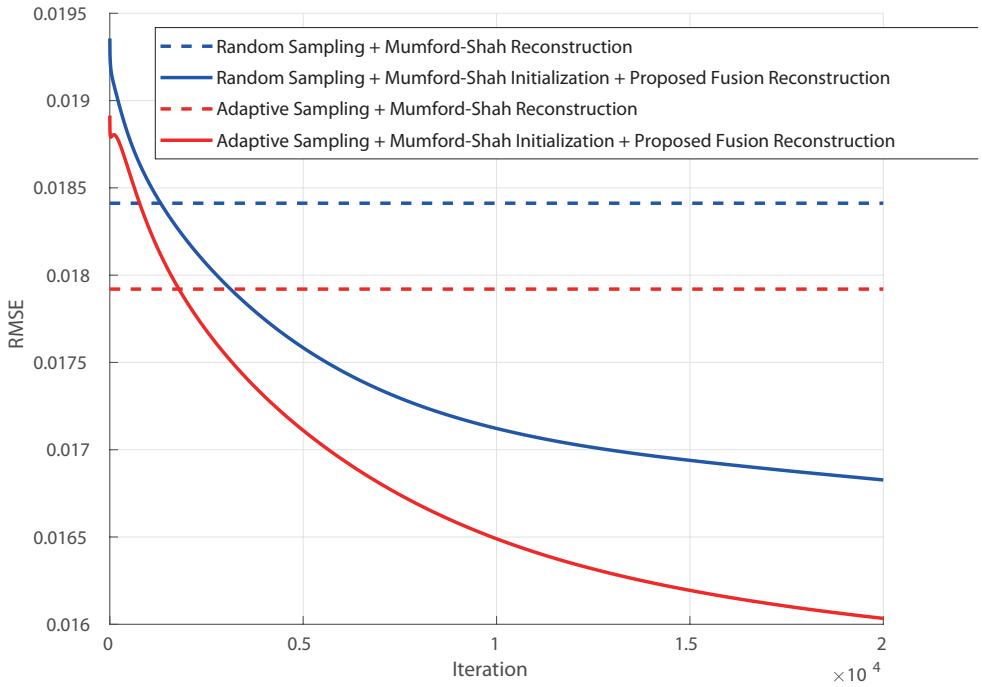
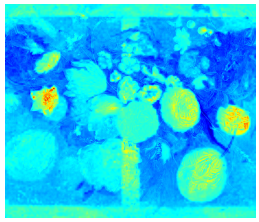
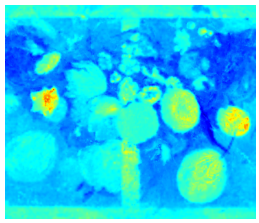


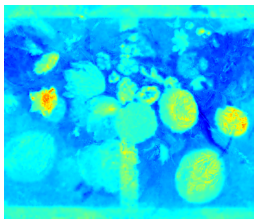
Figure 6.14. The iteration process of our proposed fusion inpainting algorithm on the "Bloemen en insecten" data. Mumford-Shah inpainting algorithm is utilized as initialization of our proposed algorithm. The iteration process of both random sampling and adaptive sampling shows that our proposed fusion inpainting algorithm minimize the RMSE during the iteration.



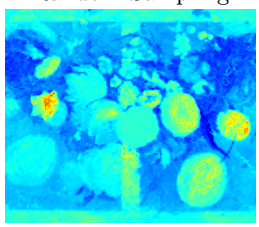
(a) Ground Truth XRF Image



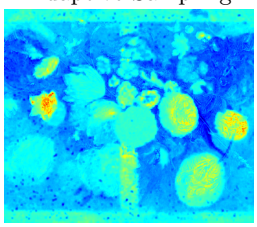
(b) Harmonic Reconstruction Random Sampling



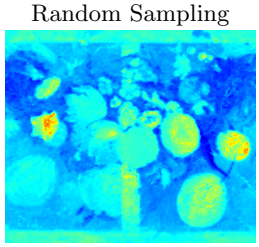
(c) Harmonic Reconstruction Adaptive Sampling



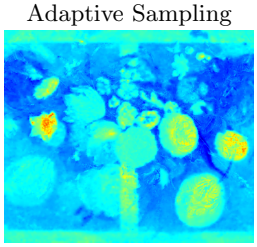
(d) BPFA Reconstruction Random Sampling



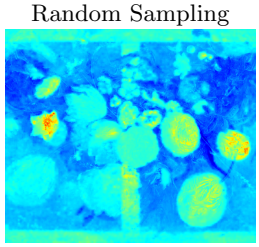
(e) BPFA Reconstruction Adaptive Sampling



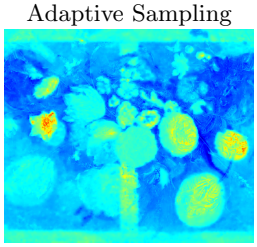
(f) Mumford-Shah Reconstruction Random Sampling



(g) Mumford-Shah Reconstruction Adaptive Sampling



(h) Proposed Reconstruction Random Sampling



(i) Proposed Reconstruction Adaptive Sampling

Figure 6.15. Visualization of the Inpainting result on the "Bloemen en insecten" data. Channel #16 related to *Pb* $L\eta$ XRF emission line is selected. (a) is the ground truth XRF image. The random and adaptive sampling masks are the same as the sampling masks in Figure 6.8 (b) and Figure 6.8 (c), respectively. (b)-(i) are the reconstruction results of different inpainting algorithms, for both randomly sampled XRF image and adaptively sampled XRF image. Readers are suggested to zoom in in order to compare the details of different results.

CHAPTER 7

Conclusions

In this dissertation we started with the inverse problem. We introduced the multiple-frame video super-resolution problem, the fast sparse coding inference problem, the X-Ray Fluorescence image super-resolution problem and the X-Ray Fluorescence image inpainting problem. Several related works were discussed. The previous coupled-dictionary learning based single-frame image super-resolution methods were then extended to multiple-frame, utilizing motion estimation by optical flow algorithms. We also extended previous work from single dictionary to multiple dictionaries. Improvement on the objective and subjective quality assessment has also been presented, showing the effectiveness of our proposed algorithm. We then propose to use deep network to speedup the sparse coding inference process with the KKT condition. We then proceeded to the X-Ray Fluorescence image super-resolution problem. Because there is not enough data to learning the priori knowledge, we proposed to fuse the low-resolution input X-Ray Fluorescent image with a high-resolution conventional RGB image. The nonlinear mapping from RGB spectrum to X-Ray Fluorescence spectrum is learned, by modeling the input X-Ray Fluorescence image as a combination of visible component and non-visible component. Both synthetic and real experiment show the effectiveness of our proposed method. Finally, we proposed to the X-Ray Fluorescence image inpainting problem. CNN is utilized to obtain the adaptive sampling mask based on the RGB image of the scanning object. The adaptively sub-sampled is then fused with a conventional RGB image to reconstruct the full-sampled XRF image. Extensive experimental

results demonstrated the effectiveness of our proposed adaptive sampling strategy and fusion based inpainting algorithm.

References

[1] Naveed Akhtar, Faisal Shafait, and Ajmal Mian. Sparse spatio-spectral representation for hyperspectral image super-resolution. In *Computer Vision–ECCV 2014*, pages 63–78. Springer, 2014.

[2] Naveed Akhtar, Faisal Shafait, and Ajmal Mian. Bayesian sparse representation for hyperspectral image super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3631–3640, 2015.

[3] Matthias Alfeld, Wout De Nolf, Simone Cagno, Karen Appel, D Peter Siddons, Anthony Kuczewski, Koen Janssens, Joris Dik, Karen Trentelman, Marc Walton, et al. Revealing hidden paint layers in oil paintings by means of scanning macro-xrf: a mock-up study based on rembrandt's an old man in military costume. *Journal of Analytical Atomic Spectrometry*, 28(1):40–51, 2013.

[4] Matthias Alfeld, Joana Vaz Pedroso, Margriet van Eikema Hommes, Geert Van der Snickt, Gwen Tauber, Jorik Blaas, Michael Haschke, Klaus Erler, Joris Dik, and Koen Janssens. A mobile instrument for in situ scanning macro-xrf investigation of historical paintings. *Journal of Analytical Atomic Spectrometry*, 28(5):760–767, 2013.

[5] Luciano Alparone, Lucien Wald, Jocelyn Chanussot, Claire Thomas, Paolo Gamba, and Lori Mann Bruce. Comparison of pansharpening algorithms: Outcome of the 2006 grs-s data-fusion contest. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(10):3012–3021, 2007.

[6] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 459–468. IEEE, 2006.

[7] Anila Anitha, Andrei Brasoveanu, Marco Duarte, Shannon Hughes, Ingrid Daubechies, Joris Dik, Koen Janssens, and Matthias Alfeld. Restoration of x-ray fluorescence images of hidden paintings. *Signal Processing*, 93(3):592–604, 2013.

[8] S Derin Babacan, Rafael Molina, and Aggelos K Katsaggelos. Total variation super resolution using a variational approach. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 641–644. IEEE, 2008.

[9] S Derin Babacan, Rafael Molina, and Aggelos K Katsaggelos. Variational bayesian super resolution. *Image Processing, IEEE Transactions on*, 20(4):984–999, 2011.

[10] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[11] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 693–696. IEEE, 2009.

[12] Stefanos P Belekos, Nikolaos P Galatsanos, and Aggelos K Katsaggelos. Maximum a posteriori video super-resolution using a new multichannel image prior. *Image Processing, IEEE Transactions on*, 19(6):1451–1464, 2010.

[13] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[14] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE transactions on image processing*, 12(8):882–889, 2003.

[15] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. *Proceedings of the 23rd British Machine Vision Conference (BMVC)*, pages 135.1–135.10, 2012.

[16] José M Bioucas-Dias, Antonio Plaza, Nicolas Dobigeon, Mario Parente, Qian Du, Paul Gader, and Jocelyn Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 5(2):354–379, 2012.

[17] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

[18] Sean Borman and Robert L Stevenson. Super-resolution from image sequences-a review. In *Circuits and Systems, Midwest Symposium on*, pages 374–374. IEEE Computer Society, 1998.

[19] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[20] Ori Bryt and Michael Elad. Compression of facial images using the k-svd algorithm. *Journal of Visual Communication and Image Representation*, 19(4):270–282, 2008.

[21] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2392–2399. IEEE, 2012.

[22] Tony F Chan and Jianhong Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449, 2001.

[23] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2004.

[24] Bin Cheng, Jianchao Yang, Shuicheng Yan, Yun Fu, and Thomas S Huang. Learning with-graph for image analysis. *Image Processing, IEEE Transactions on*, 19(4):858–866, 2010.

[25] Thomas F Coleman and Yuying Li. A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 6(4):1040–1058, 1996.

[26] Robert L Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics (TOG)*, 5(1):51–72, 1986.

[27] 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.

[28] Qiqin Dai, Emeline Pouyet, Oliver Cossairt, Marc Walton, Francesca Casadio, and Aggelos Katsaggelos. X-ray fluorescence image super-resolution using dictionary learning. In *Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 2016 IEEE 12th*, pages 1–5. IEEE, 2016.

[29] Qiqin Dai, Emeline Pouyet, Oliver Cossairt, Marc Walton, and Aggelos Katsaggelos. Spatial-spectral representation for x-ray fluorescence image super-resolution. *IEEE Transactions on image processing*.

[30] Qiqin Dai, Seunghwan Yoo, Armin Kappeler, and Aggelos K Katsaggelos. Dictionary-based multiple frame video super-resolution. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 83–87. IEEE, 2015.

[31] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *arXiv preprint math/0307152*, 2003.

[32] Laurent Demaret, Nira Dyn, and Armin Iske. Image compression by linear splines over adaptive triangulations. *Signal Processing*, 86(7):1604–1616, 2006.

[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[34] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014*, pages 184–199. Springer, 2014.

[35] Weisheng Dong, Fazuo Fu, Guangming Shi, Xun Cao, Jinjian Wu, Guangyu Li, and Xin Li. Hyperspectral image super-resolution via non-negative structured sparse representation. 2016.

[36] Weisheng Dong, Xin Li, Lei Zhang, and Guangming Shi. Sparsity-based image denoising via dictionary learning and structural clustering. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 457–464. IEEE, 2011.

[37] Weisheng Dong, D Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *Image Processing, IEEE Transactions on*, 20(7):1838–1857, 2011.

[38] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546, 2015.

[39] Marius Drulea and Sergiu Nedevschi. Total variation regularization of local-global optical flow. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 318–323. IEEE, 2011.

[40] M. Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer New York, 2010.

[41] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.

[42] Michael Elad and Yacov Hel-Or. A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur. *IEEE Transactions on Image Processing*, 10(8):1187–1193, 2001.

[43] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997.

[44] Selim Esedoglu and Jianhong Shen. Digital inpainting based on the mumford–shah–euler image model. *European Journal of Applied Mathematics*, 13(04):353–370, 2002.

[45] Ruohan Gao and Kristen Grauman. From one-trick ponies to all-rounders: On-demand learning for image restoration. *arXiv preprint arXiv:1612.01380*, 2016.

[46] Xinbo Gao, Qian Wang, Xuelong Li, Dacheng Tao, and Kaibing Zhang. Zernike-moment-based image super resolution. *IEEE Transactions on Image Processing*, 20(10):2738–2747, 2011.

[47] Andrea Garzelli, Filippo Nencini, Luciano Alparone, Bruno Aiazzi, and Stefano Baronti. Pan-sharpening of multispectral images: a critical review and comparison. In *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, volume 1. IEEE, 2004.

[48] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 399–406, 2010.

[49] Elaine T Hale, Wotao Yin, and Yin Zhang. Fixed-point continuation for \ell_1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008.

[50] Russell C Hardie, Michael T Eismann, and Gregory L Wilson. Map estimation for hyperspectral image resolution enhancement using an auxiliary sensor. *Image Processing, IEEE Transactions on*, 13(9):1174–1184, 2004.

[51] John R Hershey, Jonathan Le Roux, and Felix Weninger. Deep unfolding: Model-based inspiration of novel deep architectures. *arXiv preprint arXiv:1409.2574*, 2014.

[52] Berthold K Horn and Brian G Schunck. Determining optical flow. In *1981 Technical Symposium East*, pages 319–331. International Society for Optics and Photonics, 1981.

[53] Edson Mintsu Hung, Ricardo L de Queiroz, Fernanda Brandi, Karen França de Oliveira, and Debargha Mukherjee. Video super-resolution using codebooks derived from key-frames. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(9):1321–1331, 2012.

[54] Michael Iliadis, Leonidas Spinoulas, and Aggelos K Katsaggelos. Deep fully-connected networks for video compressive sensing. *arXiv preprint arXiv:1603.04930*, 2016.

[55] Michael Iliadis, Leonidas Spinoulas, and Aggelos K Katsaggelos. Deepbinarymask: Learning a binary mask for video compressive sensing. *arXiv preprint arXiv:1607.03343*, 2016.

[56] Harmonic Inc. Harmonic 4k footage, 2014.

[57] Infognition. Video enhancer, 2010.

[58] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[59] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and Aggelos K Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE Transactions on Computational Imaging*, 2(2):109–122, 2016.

[60] Aggelos K Katsaggelos, Rafael Molina, and Javier Mateos. Super resolution of images and video. *Synthesis Lectures on Image, Video, and Multimedia Processing*, 1(1):1–134, 2007.

[61] Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. Fast inference in sparse coding algorithms with applications to object recognition. *arXiv preprint arXiv:1010.3467*, 2010.

[62] Rei Kawakami, John Wright, Yu-Wing Tai, Yasuyuki Matsushita, Moshe Ben-Ezra, and Katsushi Ikeuchi. High-resolution hyperspectral imaging via matrix factorization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2329–2336. IEEE, 2011.

[63] Nirmal Keshava and John F Mustard. Spectral unmixing. *Signal Processing Magazine, IEEE*, 19(1):44–57, 2002.

[64] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[66] Harold W Kuhn. Nonlinear programming: a historical view. In *Traces and Emergence of Nonlinear Programming*, pages 393–414. Springer, 2014.

[67] Charis Lanaras, Emmanuel Baltsavias, and Konrad Schindler. Hyperspectral super-resolution by coupled spectral unmixing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3586–3594, 2015.

[68] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006.

[69] Yingying Li and Stanley Osher. Coordinate descent optimization for ℓ_1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Problems and Imaging*, 3(3):487–503, 2009.

[70] Meiyu Liang, Junping Du, and Linghui Li. Learning-based video superresolution reconstruction using spatiotemporal nonlocal similarity. *Mathematical Problems in Engineering*, 2015, 2015.

[71] Renjie Liao, Xin Tao, Ruiyu Li, Ziyang Ma, and Jiaya Jia. Video super-resolution via deep draft-ensemble learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 531–539, 2015.

[72] A Shu Lin, B Zhongxuan Luo, C Jielin Zhang, and D Emil Saucan. Generalized ricci curvature based sampling and reconstruction of images. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pages 604–608. IEEE, 2015.

[73] Ce Liu and Deqing Sun. A bayesian approach to adaptive video super resolution. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 209–216. IEEE, 2011.

[74] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):346–360, 2014.

[75] Jianxiong Liu, Christos Bouganis, and Peter YK Cheung. Kernel-based adaptive image sampling. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 1, pages 25–32. IEEE, 2014.

[76] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[77] Ziyang Ma, Renjie Liao, Xin Tao, Li Xu, Jiaya Jia, and Enhua Wu. Handling motion blur in multi-frame super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5224–5232, 2015.

[78] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, 2013.

[79] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM, 2009.

[80] Dimitris Manolakis, Christina Siracusa, and Gary Shaw. Hyperspectral subpixel target detection using the linear mixing model. *Geoscience and Remote Sensing, IEEE Transactions on*, 39(7):1392–1409, 2001.

[81] Antonio Marquina and Stanley J Osher. Image super-resolution by tv-regularization and bregman iteration. *Journal of Scientific Computing*, 37(3):367–382, 2008.

[82] Ryo Nakagaki and Aggelos K Katsaggelos. A vq-based blind image restoration algorithm. *Image Processing, IEEE Transactions on*, 12(9):1044–1053, 2003.

[83] Kamal Nasrollahi and Thomas B Moeslund. Super-resolution: a comprehensive survey. *Machine vision and applications*, 25(6):1423–1468, 2014.

[84] Min Kyu Park, Aggelos K Katsaggelos, and Moon Gi Kang. Regularized high-resolution image reconstruction considering inaccurate motion information. *Optical Engineering*, 46(11):117004–117004, 2007.

[85] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang. Super-resolution image reconstruction: a technical overview. *Signal Processing Magazine, IEEE*, 20(3):21–36, 2003.

[86] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[87] Carle M Pieters and Peter AJ Englert. *Remote geochemical analysis, elemental and mineralogical composition*, volume 1. 1993.

[88] Matan Protter and Michael Elad. Image sequence denoising via sparse and redundant representations. *Image Processing, IEEE Transactions on*, 18(1):27–35, 2009.

[89] Siddavatam Rajesh, K Sandeep, and RK Mittal. A fast progressive image sampling using lifting scheme and non-uniform b-splines. In *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, pages 1645–1650. IEEE, 2007.

[90] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3501–3508. IEEE, 2010.

[91] Giovanni Ramponi and Sergio Carrato. An adaptive irregular sampling algorithm and its application to image coding. *Image and Vision Computing*, 19(7):451–460, 2001.

[92] C Andrew Segall, Aggelos K Katsaggelos, Rafael Molina, and Javier Mateos. Bayesian resolution enhancement of compressed video. *Image Processing, IEEE Transactions on*, 13(7):898–911, 2004.

[93] C Andrew Segall, Rafael Molina, and Aggelos K Katsaggelos. High-resolution images from low-resolution compressed video. *Signal Processing Magazine, IEEE*, 20(3):37–48, 2003.

[94] Qi Shan, Zhaorong Li, Jiaya Jia, and Chi-Keung Tang. Fast image/video upsampling. In *ACM Transactions on Graphics (TOG)*, volume 27, page 153. ACM, 2008.

[95] Jianhong Shen and Tony F Chan. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2002.

[96] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

[97] Byung Cheol Song, Shin-Cheol Jeong, and Yanglim Choi. Video super-resolution algorithm using bi-directional overlapped block motion compensation and on-the-fly dictionary training. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(3):274–285, 2011.

[98] Mehrdad Soumekh. Multiresolution dynamic image representation with uniform and foveal spiral scan data. *IEEE transactions on image processing*, 7(11):1627–1635, 1998.

[99] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439. IEEE, 2010.

[100] Jian Sun, Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[101] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[102] Hiroyuki Takeda, Peyman Milanfar, Matan Protter, and Michael Elad. Super-resolution without explicit subpixel motion estimation. *Image Processing, IEEE Transactions on*, 18(9):1958–1975, 2009.

[103] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[104] Radu Timofte, Vivek De, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1920–1927. IEEE, 2013.

[105] RY Tsai and Thomas S Huang. Multiframe image restoration and registration. *Advances in computer vision and Image Processing*, 1(2):317–339, 1984.

[106] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[107] Cédric Vonesch and Michael Unser. A fast iterative thresholding algorithm for wavelet-regularized deconvolution. In *Optical Engineering+ Applications*, pages 67010D–67010D. International Society for Optics and Photonics, 2007.

[108] Shenlong Wang, D Zhang, Yan Liang, and Quan Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2216–2223. IEEE, 2012.

[109] Zhangyang Wang, Qing Ling, and Thomas S Huang. Learning deep ℓ_0 encoders. *arXiv preprint arXiv:1509.00153*, 2015.

[110] Zhijun Wang, Djemel Ziou, Costas Armenakis, Deren Li, and Qingquan Li. A comparative analysis of image fusion methods. *Geoscience and Remote Sensing, IEEE Transactions on*, 43(6):1391–1402, 2005.

[111] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[112] Qi Wei, Nicolas Dobigeon, and Jean-Yves Tourneret. Bayesian fusion of hyperspectral and multispectral images. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 3176–3180. IEEE, 2014.

[113] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1392, 2013.

[114] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[115] Tong Tong Wu and Kenneth Lange. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, pages 224–244, 2008.

[116] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[117] Jianchao Yang, Zhaowen Wang, Zhe Lin, Scott Cohen, and Thomas Huang. Coupled dictionary training for image super-resolution. *Image Processing, IEEE Transactions on*, 21(8):3467–3478, 2012.

[118] Jianchao Yang, Zhaowen Wang, Zhe Lin, Xianbiao Shu, and Thomas Huang. Bilevel sparse coding for coupled feature spaces. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2360–2367. IEEE, 2012.

[119] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[120] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *Image Processing, IEEE Transactions on*, 19(11):2861–2873, 2010.

[121] Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*, 2016.

[122] Roberta H Yuhas, Alexander FH Goetz, and Joe W Boardman. Discrimination among semi-arid landscape endmembers using the spectral angle mapper (sam) algorithm. 1992.

[123] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[124] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730. Springer, 2012.

[125] Fei Zhou, Shu-Tao Xia, and Qingmin Liao. Nonlocal pixel selection for multisurface fitting-based super-resolution. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(12):2013–2017, 2014.

[126] Fei Zhou, Wenming Yang, and Qingmin Liao. Interpolation-based image super-resolution using multisurface fitting. *IEEE Transactions on Image Processing*, 21(7):3312–3318, 2012.

[127] Mingyuan Zhou, Haojun Chen, John Paisley, Lu Ren, Lingbo Li, Zhengming Xing, David Dunson, Guillermo Sapiro, and Lawrence Carin. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transactions on Image Processing*, 21(1):130–144, 2012.

APPENDIX A

Appendices

A.1. Optimization scheme for Baseline #1

Let $A^l \in \mathbb{R}^{M \times N_l}$ be the spatially downsampled abundance A. The dictionary learning technique in [**79**] can be applied to initialize D^{xrf} and A^l by solving

$$\begin{aligned} &\min_{D^{xrf}, A^l} \ \|X - D^{xrf} A^l\|_F^2 + \beta \textstyle\sum_{k=1}^{N_l} \|A^l(:,k)\|_1, \\ &\text{s.t.} \quad \|D^{xrf}(:,k)\|_2 \leq 1, \forall k. \end{aligned} \tag{A.1}$$

D^{xrf} is initialized using Equation A.1 and A is initialized by upsampling A^l computed in Equation A.1.

Similar to the optimization scheme of our proposed method (Equation 5.14), Equation 5.28 can be alternatively optimized. First we optimize over A by fixing D^{xrf},

$$\min_{A} \ \|X - D^{xrf} AS\|_F^2 + \lambda \|\nabla(D^{xrf} A)\|_F^2 \tag{A.2a}$$

$$\text{s.t.} \quad A_{ij} \geq 0, \forall i, j \tag{A.2b}$$

$$\mathbf{1^T} A = \mathbf{1^T}, \tag{A.2c}$$

$$\|A\|_0 \leq s, \tag{A.2d}$$

PALM is utilized to optimize over A. For Equation A.2, the following two steps are iterated until convergence:

$$
\begin{aligned}
V^q &= A^{q-1} \\
&\quad - \frac{1}{d}({D^{xrf}}^T({D^{xrf}}A^{q-1}S - X)S^T \\
&\quad + \lambda {D^{xrf}}^T D^{xrf} AGG^T)
\end{aligned} \tag{A.3a}
$$

$$
A^q = prox_A(V^q), \tag{A.3b}
$$

where $d_2 = \gamma_2 \|D^{xrf}{D^{xrf}}^T\|_F$ are non-zero step size constants, and $prox_A$ is the proximal operator that project V^q onto the constraints of Equation A.2.

We then optimize over D^{xrf} solving the following constrained least-squares problem:

$$
\begin{aligned}
\min_{D^{xrf}} \quad & \|X - D^{xrf}AS\|_F^2 \\
\text{s.t.} \quad & 0 \leq D_{ij}^{xrf} \leq 1, \forall i, j,
\end{aligned} \tag{A.4}
$$

using the following iteration steps:

$$
\begin{aligned}
U^q &= {D^{xrf}}^{q-1} \\
&\quad - \frac{1}{d^{xrf}}({D^{xrf}}^{q-1}AS - X)S^TA^T
\end{aligned} \tag{A.5a}
$$

$$
{D^{xrf}}^q = prox_{D^{xrf}}(U^q), \tag{A.5b}
$$

where $d^{xrf} = \gamma_4 \|AA^T\|_F$ is the non-zero step size constant and $prox_{D^{xrf}}$ is the proximal operator which project U^q onto the constraints of Equation A.4.

The complete optimization scheme is demonstrated in Algorithm 5.

Algorithm 5. Proposed Optimization Scheme of Equation 5.28

input: LR XRF image X.

Initialize $D^{xrf^{(0)}}$ and $A^{l^{(0)}}$ by Equation (A.1);
Initialize $A^{(0)}$ by upsampling $A^{l^{(0)}}$;
$n = 0$;
repeat
  Estimate $A^{(n+1)}$ with Equation A.3;
  Estimate $D^{xrf^{(n+1)}}$ with Equation A.6;
  n=n+1;
until convergence

output: HR XRF image
$Y = D^{xrf}A$.

A.2. Optimization scheme for Baseline #2

For Equation 5.30, A, D^{xrf} and D^{rgb} can be initialized by Equation 5.13. We then alternatively optimize the unknowns in Equation 5.30. We first update A based on the RGB image by fixing all other parameters,

$$\min_{A} \quad \|I - D^{rgb}A\|_F^2 \tag{A.6a}$$

$$\text{s.t.} \quad A_{ij} \geq 0, \forall i, j \tag{A.6b}$$

$$\mathbf{1^T}A = \mathbf{1^T}, \tag{A.6c}$$

$$\|A\|_0 \leq s, \tag{A.6d}$$

utilizing the following iteration steps:

$$V^q = A^{q-1} - \frac{1}{d} D^{rgb^T}(D^{rgb}A^{q-1} - I) \tag{A.7a}$$

$$A^q = prox_A(V^q), \tag{A.7b}$$

where $d = \gamma_1 \|D^{rgb}D^{rgb^T}\|_F$ is non-zero step size constants, and $prox_A$ is the proximal operator that project V^q onto the constraints of Equation A.6.

We then update D^{rgb}

$$\begin{aligned} \min_{D^{rgb}} \quad & \|I - D^{rgb}A\|_F^2 \\ \text{s.t.} \quad & 0 \leq D_{ij}^{rgb} \leq 1, \forall i, j. \end{aligned} \tag{A.8}$$

by the following iteration steps:

$$E^q = D^{rgb^{q-1}} - \frac{1}{d_{rgb}}(D^{rgb^{q-1}}A - I)A^T \tag{A.9a}$$

$$D^{rgb^q} = prox_{D^{rgb}}(E^q), \tag{A.9b}$$

with $d_{rgb} = \gamma_3 \|AA^T\|_F$ again a non-zero step size constant and $prox_{D^{rgb}}$ the proximal operator that projects E^q onto the constraint of Equation A.8.

Finally we update D^{xrf}

$$\begin{aligned} \min_{D^{xrf}} \quad & \|X - D^{xrf}AS\|_F^2 \\ \text{s.t.} \quad & 0 \leq D_{ij}^{xrf} \leq 1, \forall i, j, \end{aligned} \tag{A.10}$$

using the following iteration steps:

$$U^q = D^{xrf^{q-1}} - \frac{1}{d^{xrf}}(D^{xrf^{q-1}}AS - X)S^TA^T \tag{A.11a}$$

$$D^{xrf^q} = prox_{D^{xrf}}(U^q), \tag{A.11b}$$

where $d^{xrf} = \gamma_4\|AA^T\|_F$ is the non-zero step size constant and $prox_{D^{xrf}}$ is the proximal operator which project U^q onto the constraints of Equation A.10.

The complete optimization scheme is summarized in Algorithm 6.

Algorithm 6. Proposed Optimization Scheme of Equation 5.30

---

input: LR XRF image X, HR conventional RGB image I.

1: Initialize $D^{rgb^{(0)}}$, $D^{xrf^{(0)}}$ and $A^{l^{(0)}}$ by Equation (5.13);
   Initialize $A^{(0)}$ by upsampling $A^{l^{(0)}}$;
   $n = 0$;
2: **repeat**
3: Estimate $A^{(n+1)}$ with Equation A.7;
4: Estimate $D^{rgb^{(n+1)}}$ with Equation A.9;
5: Estimate $D^{xrf^{(n+1)}}$ with Equation A.11;
6: n=n+1;
7: **until** convergence

output: HR XRF image
$Y = D^{xrf}A$.

This dissertation was typeset with LaTeX 2_ε[1] by the author.

[1]The macros used in formatting this dissertation are based on those written by Miguel A. Lerma, (Mathematics, Northwestern University) which have been further modified by Debjit Sinha (EECS, Northwestern University) to accommodate electronic dissertation formatting guidelines.