Sparse Recovery Techniques for Hyperspectral Imaging

By

Hemant Kumar Aggarwal

under the supervision of

Dr. Angshul Majumdar

Indraprastha Institute of Information Technology-Delhi

August, 2016

Sparse Recovery Techniques for Hyperspectral Imaging

By

Hemant Kumar Aggarwal

submitted

in partial fulfillment of the requirements

for the award of the degree of

Doctor of Philosophy

to the

Indraprastha Institute of Information Technology-Delhi

Okhla Industrial Estate, Phase III

New Delhi, India - 110020

August, 2016

Indraprastha Institute of Information Technology-Delhi
Okhla Industrial Estate, Phase III
New Delhi-110020, India
August-2016

Certificate

This is to certify that the thesis titled **Sparse Recovery Techniques for Hyperspectral Imaging** being submitted by **Mr. Hemant Kumar Aggarwal** to the Indraprastha Institute of Information Technology-Delhi, for the award of the degree of **Doctor of Philosophy**, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

August, 2016
Angshul Majumdar

Indraprastha Institute of Information Technology-Delhi
New Delhi-110020

Indraprastha Institute of Information Technology-Delhi
Okhla Industrial Estate, Phase III
New Delhi-110020, India
August-2016

Declaration

This is to be certified that the dissertation entitled **Sparse Recovery Techniques for Hyperspectral Imaging** being submitted by **Mr. Hemant Kumar Aggarwal** to the **Indraprastha Institute of Information Technology-Delhi**, for the award of degree of **Doctor of Philosophy**, is a bonafide work carried out by me. This research work has been carried out under the supervision of **Dr. Angshul Majumdar**.

The study pertains to this dissertation has not been submitted in part or in full, to any other University or Institution for the award of any other degree.

Hemant Kumar Aggarwal
PhD student
IIIT-Delhi, India

Acknowledgement

I would like to express my sincere thanks to my supervisor **Dr. Angshul Majumdar** for providing me research direction, active monitoring, and timely feedback. I wish to thank him for his enthusiasm, highest priority to research, and the habit of never keeping the things pending during my entire duration of Ph.D. Without his valuable thoughts, recommendations, and patience, It would have been difficult for me to complete this dissertation. I would like to thank him for providing excellent research facilities.

I am grateful to the **Indraprastha Institue of Information Technology-Delhi** for providing excellent infrastructure and research environment. I would like to express gratitude to **Tata Consultancy Services** for providing me research fellowship that helped me financially throughout my PhD.

I would like to thank all my fellow colleagues namely Anupriya Gogna, Hemanta Kumar Mondal, Mohamad Ayatullah Maktumi, Monalisa Jena, Sathish V, Shiju S, and Wazir Singh for their help and support. I gratefully acknowledge the unselfish help given to me by them. I also wish to thank my colleague Naushad Ansari for several technical discussions throughout my Ph.D.

I am thankful to my parents, brother, and sister for their full support under all conditions not just during Ph.D. but throughout my life.

Hemant Kumar Aggarwal

Contents

List of Figures

Page

List of Tables

Page

Acronyms

BP Basis Pursuit

BPDN Basis Pursuit Denoising

BTES Binary Tree based Edge Sensing

CFA Color Filter Array

CLSR Collaborative Sparse Regression

DCT Discrete Cosine Transform

JSTV Joint Sparse Total Variation

LASSO Least Absolute Shrinkage and Selection Operator

LSMD Least-Square based Multispectral Demosaicing

MOD Method of Optimal Directions

MSFA Multispectral Filter Array

PSNR Peak Signal to Noise Ratio

SBJS Split Bregman based Joint Sparse

SBTV Split Bregman based Total Variation

SPGL1 Spectral Projected Gradient for L1

SRTV Total Variation Spatial Regularization

SSIM Structural Similarity Index

SVD Singular Value Decomposition

UMSFA Uniform Multispectral Filter Array

Abstract

Human vision is a powerful imaging system that can capture and interpret light energy coming from different sources although it is limited to visible light. There are various applications such as face recognition, medical imaging, agriculture, geology, surveillance, etc. that benefits by imaging several bands of the electromagnetic spectrum outside the visible range. The hyperspectral imaging techniques are capable of capturing hundreds of bands of the electromagnetic spectrum and thus, can be considered as the generalization of color imaging.

The focus of this dissertation is on modeling hyperspectral imaging problems as linear inverse problems and solving them by exploiting inherent data properties. These imaging problems often form an underdetermined system of the linear equations having infinitely many solutions; therefore, additional constraints based on prior knowledge about data can help in determining the solution uniquely.

This work aims at developing multispectral image acquisition and reconstruction techniques such that minimal changes are required in the hardware of compact digital cameras. A uniform multispectral filter array design has been proposed that satisfy both spatial consistency and spectral uniformity requirements. Based on proposed filter array pattern, an efficient demosaicing algorithm has been proposed to reconstruct the full multispectral image from severely under-sampled raw image such that reconstructed image has good visual quality.

This work also focuses on hyperspectral denoising problem. A novel spatio-spectral total-variation model has been proposed that gives a sparser representation of the sorted discrete gradient coefficients as compared to the band by band hyperspectral total-variation model. A general additive noise model was considered that accounts for not only Gaussian noise but also the sparse noise that includes impulse noise and line strips. The resulting optimization problem was solved using augmented-Lagrangian like the split-Bregman algorithm.

Another problem discussed in this work is the hyperspectral unmixing problem that is related to blind source separation problem in signal processing. A joint-sparse model along with total variation in the general noise model framework has been considered in formulating the problem as a linear sparse unmixing problem. Since a particular endmember may be present at several locations, therefore, abundance maps shows joint sparsity. Proposed joint-sparsity and total variation based unmixing algorithm have been compared with several related unmixing algorithms to empirically demonstrate its performance using visual quality as well as signal to noise ratio.

Further, A deep dictionary learning based approach has been proposed for hyperspectral image classification problem. The learning proceeds in a greedy fashion, therefore for each level we only need to learn a single layer of the dictionary. A comparative study with deep belief network and stacked autoencoder based techniques suggests that in the practical scenario, when the training data is limited, the proposed method outperforms these more established tools.

Chapter 1

Introduction

Imaging is the science and engineering of developing hardware and software solutions to capture light energy. Every object on earth can absorb, reflect and transmit some amount of radiations at certain wavelengths of electromagnetic spectrum depending on the properties of that object. This light energy coming from an object can be captured and processed using various imaging technologies. The captured light energy can be further processed, visualized, and analyzed depending on the application. This chapter describes what multidimensional images are, how to capture them, which specific imaging problems are discussed in this dissertation and what are specific research contributions.

1.1 Background

Image processing is active interdisciplinary research area at the intersection of various fields such as signal processing, optimization, computer science, calculus, and probability. Image processing acts as a building block for subjects like pattern recognition, computer vision, video processing. Image processing is ubiquitous as it has applications in geology, astrophysics, geography, military services, medicine, biology, remote sensing, environmental science, etc..

Images captured in more than one band of the electromagnetic spectrum are termed as multidimensional images. These multiband images can be thought of as a collection of the gray-scale images of a particular object taken at different wavelengths in parallel. Images can be captured at different wavelength regions depending on the application areas such as X-ray images, ultra-violet images, radar images, thermal images, color images, panchromatic images, etc. This work focuses on multidimensional images that are captured from near ultraviolet to infrared region between the wavelength range of around 400 nm to 2500 nm. A multispectral image has three to thirty bands whereas hyperspectral images have hundreds of bands with a very narrow spectral gap. Figure 1.1 shows a sample color image of 3 bands, a multispectral image with five bands, a hyperspectral image with 360 bands.

There are applications that benefits by capturing some wavelengths of light outside the visible region. Multispectral images have been successfully used in the detection of oil spill on the sea surface [4, 5] using both active and passive remote sensing tech-

(a) *Color image* **(b)** *Multispectral image* **(c)** *Hyperspectral image*

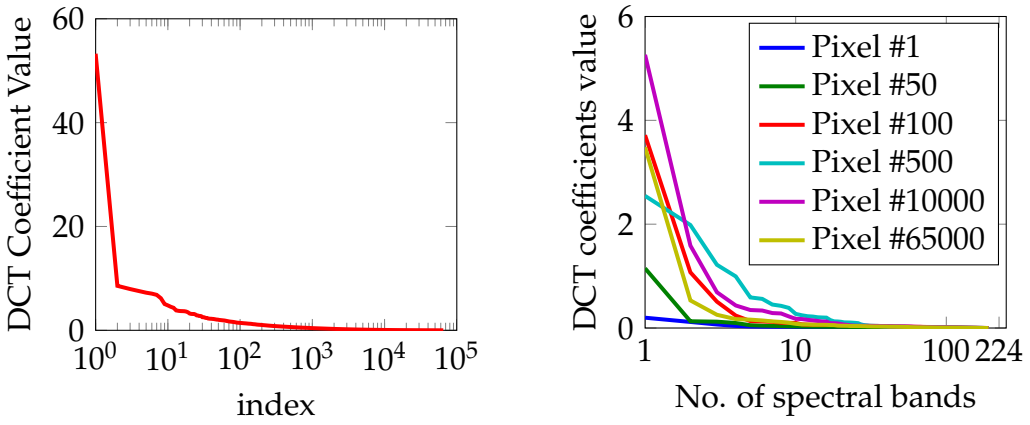
Figure 1.1: *Examples of multi-dimensional images*

niques. The infrared images help to tackle the impact of illumination changes in the face recognition problem [6–8]. Hyperspectral images are also used for imaging in the ultraviolet range which is challenging due to high scattering of light in below visible range. The applications include detection of colored dissolved organic matter in natural water. UV light is used in many applications such as small amount of surface contamination, scratches on the lenses, criminology, etc. Crop monitoring is an important application of remote sensing images [9, 10] in agriculture domain. Crop monitoring includes crop production, monitoring, checking for diseases, and soil moisture detection. Satellite images are extensively used in geological applications such as mineral exploration [11] application, underwater exploration, buried sediments analysis, etc.. A review of various applications of hyperspectral images in medicine domain can be found in [12]. Hyperspectral images are used in applications such as tumor identification, making surgical decisions, evaluating the health of dental structures etc.. Remote sensing images assist in planning of resource management, landscape management [13], monitoring the condition of habitats and species distribution.

Sparse Recovery

A signal is said to be sparse if it contains very few nonzero coefficients compared to its length. Similarly, a signal is said to be compressible if the absolute value of its sorted coefficients follows a power law decay. Figure 1.2 shows an example of the compressible representation of a hyperspectral image. Figure 1.2(a) shows the sparse representation of band 10 of WDC image of size 256×256 by using two-dimensional Discrete Cosine Transform (DCT). It can be observed that coefficient values decay exponentially and only around 100 out of a total of 65536 coefficients are significant. The image is compressible because of the existence of high spatial correlation among neighboring pixel values.

The image is compressible not only spatially but also spectrally. In a hyperspectral image, each pixel is captured over hundreds of bands having very narrow spectral gap; therefore, each pixel is spectrally correlated as well. Figure 1.2(b) shows the plot of sorted one-dimensional DCT coefficients for some pixels of WDC image over 191 bands. It can be observed that there are around ten significant coefficients compared to 191.



(a) *Spatial Sparsity* **(b)** *Spectral Sparsity*

Figure 1.2: *Spatial and spectral compressibility of hyperspectral images*

1.2 Multispectral Demosaicing

The term imaging refers to the process of capturing, processing, analyzing, and visualizing an image. Imaging is a very broad and active research area and this dissertation particularly focuses on three imaging problems namely multispectral demosaicing problem, hyperspectral denoising problem, and hyperspectral unmixing problem. Figure 1.3 shows the process of capturing a color image using single-sensor architecture.

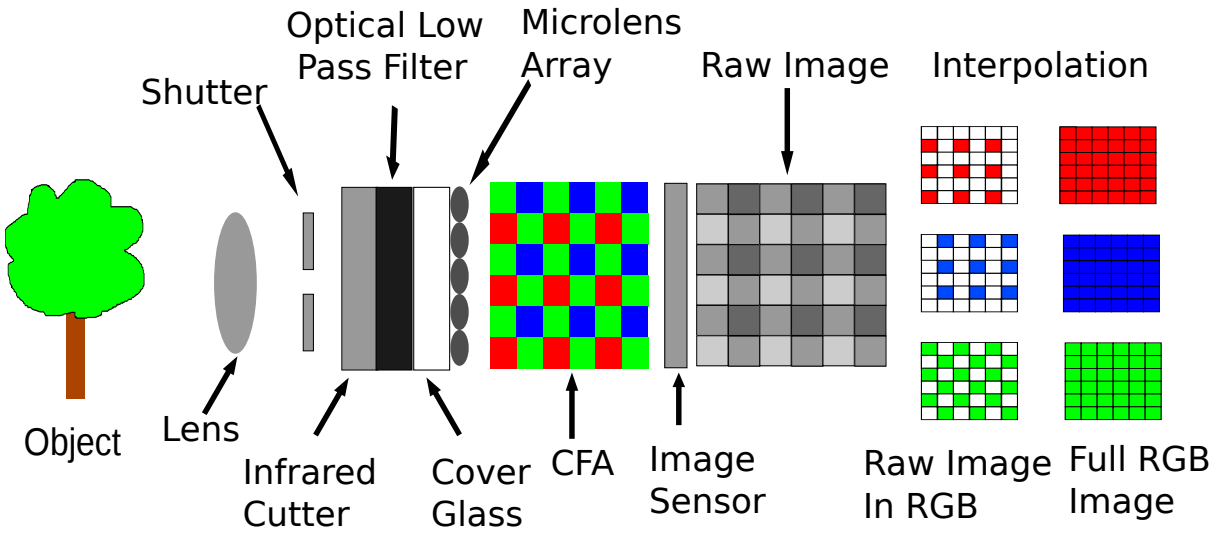


Figure 1.3: *Color imaging process using single-sensor architecture.*

When light enters through the lens of a point and shoot digital camera then initially there is a shutter inside it that allows light to enter inside the camera. If the shutter is open for a long time, then more light enters in the camera. After that, there is an infrared cutter that does not allow infrared light to penetrate into the camera. Followed

by the infrared cutter, there is an optical low pass filter that allows only the light with low frequency to enter because high-frequency light has high energy and can damage the hardware of the camera. Hence, both, very low and very high wavelengths are stopped to enter into the camera. The optical low pass filter is followed by a micro-lens array that focuses the incoming light onto the color filter array which allows a particular wavelength to hit a photo-diode on the image sensor. When the energy of incoming light photons is higher than that of band-gap energy of semiconductor (silicon), then the electron-hole pair is generated. These electrons generated inside the depletion region are utilized as signal charge. This electron charge is converted into the voltage by an active transistor inside the pixel.

Color imaging using a single sensor is possible because of the invention of Color Filter Array (CFA) that filters the light such that only one particular wavelength is captured at a pixel. The most popular CFA is Bayer filter [14] that capture 50% samples of green band and 25% samples for each of red and blue band. The green band is captured twice that of red and blue because our eyes are more sensitive to green band compared to the other two. Thus, at each pixel, only a single color intensity is captured such an image is termed as the raw image. This image is also called mosaiced image because it is a combination of three bands. The raw image has only one value at each pixel and remaining two have to be interpolated for each pixel. This process of interpolating the unknown pixel values is called color image demosaicing problem. There are many linear and non-linear demosaicing algorithms to reconstruct the full-color image from the raw image. This work extends the single-sensor color imaging architecture to multispectral imaging architecture.

There has been a lot of advancement in the various aspects of camera technology such as reduction in camera size, higher image resolution, more efficient imaging sensors, fast image processing algorithms, etc. Due to these advancements, the color imaging technology has become very popular in different application areas. Despite having more information content than color images, multispectral imaging is not widely used because they are not as accessible as color images due to the high cost of multispectral cameras. The main cause of high cost of such cameras is the use of several expensive imaging sensors and many moving optical and mechanical parts.

Many multi-spectral cameras have as many CCD-sensors as the number of bands to be captured. An example of a multispectral camera is shown in Fig. 1.4. This camera has six imaging sensors to capture six different wavelengths of light by using different filters. The size and cost of multi-spectral cameras increases with the number of CCD sensors required to capture a larger number of bands of electromagnetic spectrum. The main motivation of this work is to bring down the cost of the multi-spectral cameras by proposing a single-sensor architecture as used for color image acquisition. The use of multiple image sensors causes complexities such as pixel-to-pixel registration of each band so that all the imaging sensors capture same area as accurately as possible. Furthermore, multi-sensor cameras consume more power to operate due to complex circuitry as compared to single-sensor cameras. Therefore, it is desirable to have a single-sensor multi-spectral camera that can handle above mentioned limitations to a certain extent.

Figure 1.4: *A multispectral camera*

There is a vast literature on CFA and the corresponding demosaicing algorithms. Many single-sensor digital cameras capture color images using Bayer pattern [14] where 50% samples of green band and 25% each of red and the blue band are obtained. Many linear and non-linear demosaicing algorithms have been proposed in the literature for finding missing intensity values at each pixel. A review of color image demosaicing algorithms is mentioned in [15]. Linear time gradient-based bilinear interpolation technique have been proposed in [16] where it has been claimed to be superior to many non-linear algorithms. The design idea of Bayer pattern and many such algorithms is based on the property of the human visual system that human eye is more sensitive to the green band as compare to the red and blue band. The idea of Bayer pattern may not be directly generalized to develop filters and demosaicing algorithms for multi-spectral images since there are large number of bands each having some unique characteristics.

A multispectral filter array design have been proposed in [17] which is based on the probability of appearance of each band to be used in target recognition. This design takes into account spectral consistency which is used to avoid optical cross-talk that causes some artifacts into the image. Authors of that paper also considered spatial uniformity which says that for doing interpolation, uniform sampling is better than random sampling. This filter design is a three step process consisting of finding binary tree based on the probability of appearance, determining pixel locations for interpolation using checkerboard selection, and finally combining the results to get the raw image. However, that proposed design of filter array is based on the prior knowledge of the probability of appearance of bands. Based on above filter array design, a generic demosaicing approach Binary Tree based Edge Sensing (BTES) algorithm has been proposed in [18] which is based on exploring edge correlation to do the interpolation of missing band values. That algorithm first determines which band to interpolate based on the probability of appearance of each band and then order of interpolation for each pixel is determined followed by image transforms to do the interpolation. A multi-spectral camera design has also been proposed in [19] which is based on extending color filter array design using color channel differences. This filter is a 3×2 multi-spectral filter on CCD-chip. Authors selected this design for doing fast bi-linear interpolation to find unknown intensity values. This technique is based

spectral channel differences which can be considered as a smoothing operation after bi-linear interpolation. This algorithm does not make use of the spectral-correlation among bands of multi-spectral imagery.

Multispectral images are captured using a multi-sensor architecture that uses a separate sensor to capture each band. Since for each band, there is a separate sensor, therefore, the quality of acquired image is very high, and there is no need for image interpolation algorithm. Due to the use of numerous hardware and optical-mechanical parts, the size and cost of these imaging devices are very high. When several bands corresponding to the same scene are captured by different sensors, then there is the probability of error in the measurement process in the sense that individual pixels may not be corresponding to the same spatial location. This problem is often termed as the image to image registration problem.

This work aims at addressing the problem of reducing size and cost of hand-held multispectral cameras by utilizing single-sensor based architecture which is used in most compact digital cameras for color imaging.

The use of single-sensor can reduce the size and cost of the multispectral camera. It poses several challenges. There is no standard optical filter array to capture multiple bands using the single-sensor architecture similar to Bayer filter for color imaging. Bayer filter is not directly extensible for multi-spectral imaging because that has been designed according to the sensitivity of the human visual system. In the case of multispectral images, we shall not use the filters that have been designed for color images since we are trying to capture beyond visible range. Therefore, there is need of Multispectral Filter Array (MSFA) that allows us to take measurements of several bands. Let there exists some MSF array which allows capturing multispectral images using single-sensor then if we capture four bands then we will have only 25% samples, on the other hand, with five bands we will only have 20% samples of each band. Recovering a full image from such a small number of samples is a challenging problem. Figure 1.5 shows how to extend color imaging architecture for multispectral imaging.

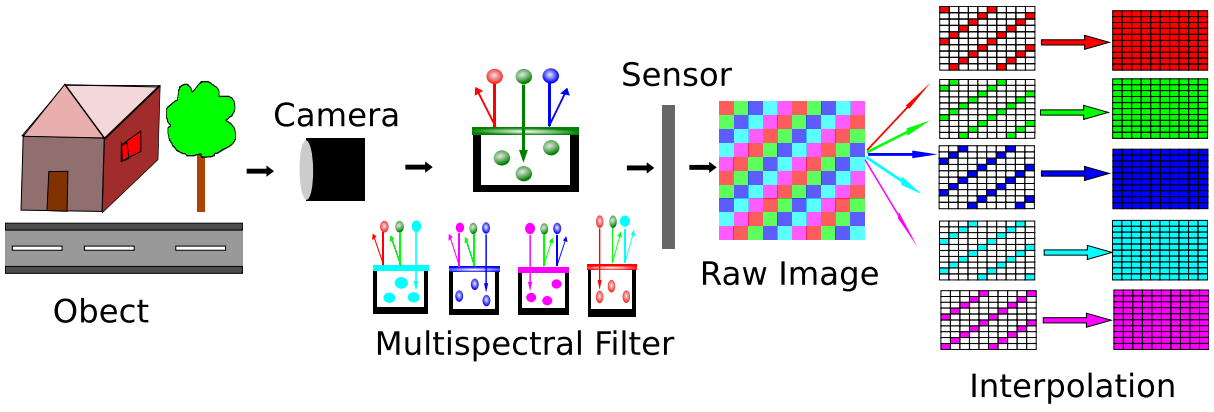


Figure 1.5: *Multispectral Demosaicing Process*

A minimal change in the existing hardware has been proposed. In particular, the CFA need to be replaced with MSFA that will allow capturing multiple wavelengths of light. There are several challenges in achieving it such as the MSFA should be

generic enough to be adapted to different multispectral imaging requirements such as the number of bands and the spectral gap. The MSFA should not be dependent on the properties of the human visual system since multispectral imaging also spans invisible region.

After obtaining the raw multispectral image from MSFA, an interpolation algorithm is required to obtain the full multispectral image from the raw image as shown in Fig. 1.5. This work also focuses towards developing reconstruction algorithms for multispectral imaging.

1.3 Hyperspectral Denoising

A problem that often occur in image acquisition process is the denoising problem. An image is said to be noisy if it contains undesired or missing information. This unwanted information can be in the form of a random signal that causes a change in actual intensity value at some or all pixels in the image. Image denoising is a classical problem as images often get corrupted by noise at any level of processing from image acquisition to image archiving in memory. Various kinds of noise may be simultaneously present in an image such as Gaussian noise, random-valued impulse noise, salt and pepper noise, shot noise, line strips, etc. Figure 1.6 shows an example of denoising problem. Here the original image is corrupted by three kinds of noise, and the captured image is the noisy image. The denoising problem is to recover the original image given the observed noisy image. When the image is corrupted by several kinds of noise, then the problem is called mixed-noise reduction problem.

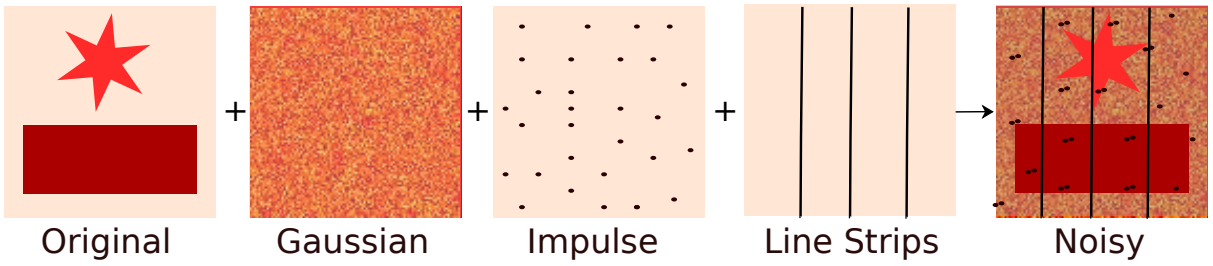


Figure 1.6: *Mixed noise reduction problem*

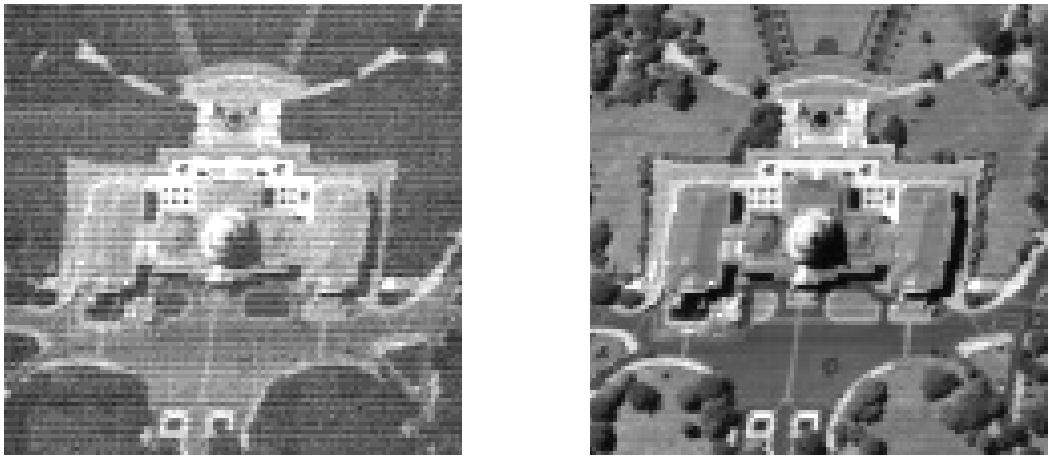
The most common type of noise is Gaussian noise that corrupts each pixel in the image by some amount. Random valued impulse noise corrupts few pixels in the image but corrupts them heavily by some random amount. Salt-and-pepper noise is an extreme case of random-valued impulse noise in which corrupted pixels values become either zero or one. Shot noise and line strips mostly occur in satellite images. In the case of shot noise, a particular pixel gets corrupted in all the bands. The line-strip problem is the presence of partial or full horizontal or vertical lines in the image depending on the type scanners.

Images are corrupted by noise due to several reasons including fluctuations in power supply, dark current, non-uniformity of detector response, etc.. A real hy-

perspectral image may get corrupted by several kinds of noise including Gaussian noise, random-valued impulse noise, salt-and-pepper noise, horizontal and vertical deadlines, etc. Therefore, hyperspectral denoising is a mixed noise reduction problem consisting of a mixture of Gaussian noise and sparse noise. The term sparse noise [20] refers to the noise that corrupts only a few pixels in the image but corrupts them heavily. The sparse noise includes random-valued impulse noise, salt-and-pepper noise, horizontal and vertical dead-lines.

The noise that occurs due to conditions like poor lighting, dark current or sensor noise is found to obey Poisson distribution and is approximately modeled as additive Gaussian noise. This noise enters into the system while capturing an image. Horizontal line strips often occur in images captured by whisk-broom kind of sensors that have rotating mirrors perpendicular to the flight direction. Vertical line strips mostly occur in images taken by the push-broom type of sensors which capture scene along the flight direction. Shot noise occurs due to some defective pixels. Random fluctuations in the power supply of satellite's sensor often corrupt these images by random-valued impulse noise. Images also become noisy due to dark current and non-uniformity of detector response.

Figure 1.7 shows a portion of band 134 and band 132 of WDC image as an example of a noisy and a clean image. It can be observed that band 134 have some undesirable information in the form of some horizontal lines whereas band 132 is comparatively very sharp. The aim of hyperspectral denoising is to reduce the unwanted information from corrupted bands.



(a) *Band 134* **(b)** *Band 132*

Figure 1.7: *Real noisy hyperspectral image*

If there is noise in a hyperspectral image, then the best option is to recapture it, but often this is not possible because of several reasons. It might be not possible to reorient the satellite to that geographical location also the cycle time of satellite may be very long. Instantly, it might not be possible to check image for the presence of noise because there is a time gap between the image being captured and received on the Earth. Also, sometimes denoising become essential if some non-repeatable natural

phenomenon is begin imaged.

Hyperspectral denoising is an important pre-processing step in various applications of hyperspectral images such as terrain classification and target detection. Hyperspectral denoising is a classical well studied problem [21–26]. There are studies such as [27–29] which consider mixed noise reduction from gray-scale images. These studies analyze mixture of only Gaussian and salt-and-pepper noise whereas we address a realistic scenario by taking into account more general noise and attempt to solve this problem for hyperspectral images. A recent low-rank matrix recovery (LRMR) based denoising approach [20] can reduce mixed noise from hyperspectral images. The low-rank based model is a global model which, in the context of hyperspectral images, exploits spectral correlation whereas total variation is a local model that utilizes spatial correlation within a band.

1.4 Hyperspectral Unmixing

If an observed signal consists of several source signals then the problem of identifying each component signal from that mixture is called source separation problem. This problem occurs in many application areas such as speech processing, medical imaging, remote sensing, etc. In the context of hyperspectral images, the source separation problem is called the unmixing problem. Hyperspectral unmixing is a classical, important, and challenging problem in remote sensing. It is a problem of identifying endmembers and their fractional abundances present at every pixel in a hyperspectral image. The term endmember refers to various materials that may be directly or indirectly present in a hyperspectral image. The term direct-presence indicates the existence of pure pixels, and indirect presence refers to mixed pixels. A pixel in a satellite image corresponds to an extensive spatial area on earth as demonstrated in Fig. 1.8. This spatial region constituting that pixel may be covered by a single object or multiple

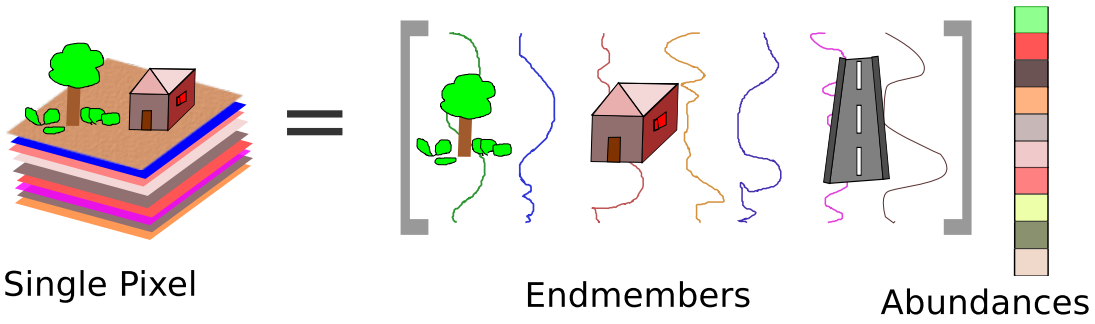


Figure 1.8: *The linear unmixing model for one pixel*

objects. If the area covered by a pixel forms a single object then such a pixel is called pure pixel; otherwise, it is called mixed pixel. The term fractional abundance indicates the percentage of a particular endmember present at a pixel. Thus, abundance map shows the distribution of a particular endmember over a region. The pure pixels have the fractional abundance of one whereas mixed pixels have fractional abundance

between zero and one. The unmixing problem can be categorized as linear or non-linear. Figure 1.8 shows the linear unmixing problem for a pixel of a hyperspectral image. The linear unmixing problem considers each pixel as a linear combination of several available endmembers. The non-linear unmixing problem also accounts for the interaction between various endmembers constituting that pixel.

Hyperspectral unmixing has applications in various domains such as geology, agriculture [30], environmental studies, biology [31], etc. The abundance maps are often used as feature vectors [32] in several image processing and pattern recognition related applications of hyperspectral images. Hyperspectral unmixing is also utilized in denoising [33], data fusion [34], and super-resolution [35] related applications.

If a hyperspectral image is of very high resolution, then its constituent endmembers shall be considered at micro level such as chemical composition of the pixel. These images require the unmixing problem to be handled at the micro level. However in this work, we are interested in the macro level decomposition of a pixel into its constituent components. An overview of hyperspectral unmixing algorithms has been discussed in [36].

Often hyperspectral images are corrupted by some kinds of noise as discussed in the previous section; therefore, it is desirable to do unmixing of hyperspectral images even when they are corrupted by one or several of these kinds of noise. This problem of unmixing in the presence of mixed noise can be approached by first applying a denoising algorithm followed by the unmixing algorithm. This work directly recovers the abundance map in the presence of mixed noise. There are studies such as [26, 33] that also perform unmixing in the presence of noise. This work is different from these existing methods in both the noise model and the solution approach.

This work is based on linear mixing model for unmixing as shown in Fig. 1.9, however, there are various nonlinear models for the hyperspectral unmixing whose survey can be found in [37]. The work focuses on the sparse unmixing problem in which each pixel can be represented as linear combination of few endmembers out of hundreds of available endmembers as shown in the Fig. 1.9. Columns of the abundance matrix in this Figure shows sparse coefficients corresponding to each pixel.

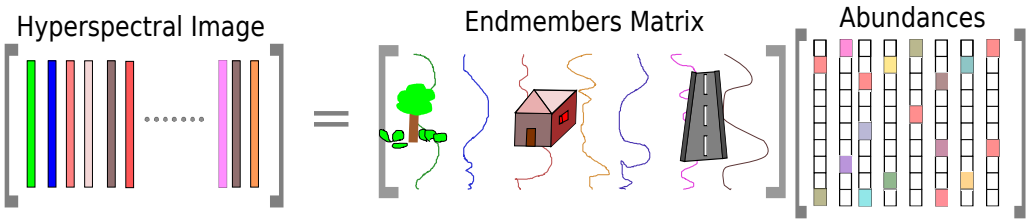


Figure 1.9: *Sparse linear unmixing model*

There are algorithms such as pixel purity index (PPI) [38] and N-FINDER [39] which require the presence of pure pixels in the image. However this pure pixel assumption may not be true always, and therefore, this work proposes to do unmixing in the absence of pure pixels. Hyperspectral unmixing approaches can be categorized as the one that utilizes existing spectral libraries and others that try to estimate endmem-

ber spectral signatures using non-negative matrix factorization based techniques such as [40]. This work is based on utilizing existing spectral libraries available for many materials in different categories of endmembers such as artificial, minerals, soils, etc.

This work intends to utilize the mixed noise model proposed in [41] that was later employed for denoising in [20]. This model allows us to formulate the linear hyperspectral unmixing problem that explicitly account for both Gaussian and sparse noise. The total number of endmembers available from different spectral libraries (e.g. the USGS library) are enormous, but only a few of these endmembers are present in a given hyperspectral image. At every pixel, a subset of the endmembers (present in the whole image) is present. This observation can be modeled as joint-sparse [42] regularization on abundance maps. Natural images often exhibit high spatial correlation implying that pixels having the same spectral signature may be present in the neighborhood. This observation can be modeled as total-variation [43] regularization on abundance maps. Thus, this work proposes a hyperspectral unmixing algorithm that utilizes generic noise model and explores both joint sparsity and spatial smoothness of abundance maps. The resulting optimization problem is solved using the split-Bregman [44] based technique. Our work improves over the state of the art sparse regression based unmixing techniques sparse regression (SR) [45] and its variants total variation spatial regularization (SRTV) [46] and collaborative sparse regression (CLSR) [47].

1.5 Hyperspectral Classification

Data classification is a well studied problem in machine learning. It primarily involve grouping the data items into classes based on the similarity among data items. Data classification task can be broadly categorized as supervised classification or unsupervised classification. When there exists training data to learn a classifier then it is called supervised classification whereas when no training data is available then grouping of data is done in an unsupervised manner popularly known as clustering. There are many approaches to perform classification such as decision trees, support vector machine (SVM), neural networks (NN), k-nearest neighbor (KNN), etc..

KNN is one of the simplest classifier. It takes a test sample and identifies K-nearest training samples. The class label can be decided using the majority voting technique. Figure 1.10 describe the working of KNN classifier on two-class classification problem. It can be observed from Fig. 1.10 that KNN classifier is dependent on the choice of k.

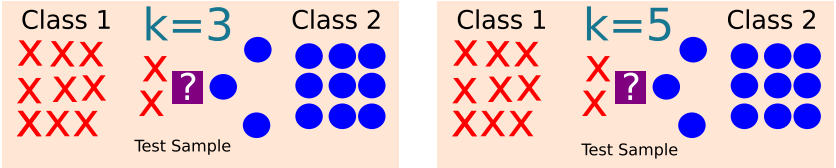


Figure 1.10: *K-nearest neighbor classifier*

When k=3 then test sample gets assigned to class 1 whereas when k=5 then test sample

gets assigned to class 2.

The nearest subspace classifier (NSC) can be thought of as an extension of KNN classifier. It assumes that all the data samples belonging to a particular class lies in a subspace. Therefore, each test sample is assigned a class label based on the distance of the test sample from different training subspaces. Figure 1.11 shows an example of how nearest subspace classifier works. In this example, the distance of given test sample is calculated from both the subspaces and since it is near subspace 1 therefore, it gets assigned to class 1 (red cross).

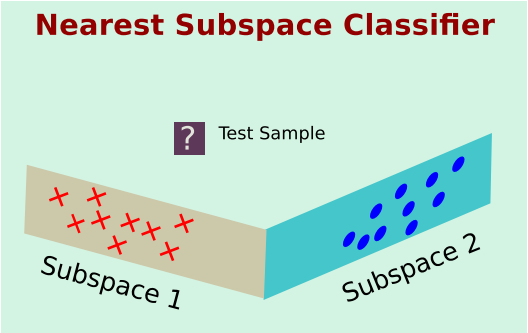


Figure 1.11: *Nearest subspace classifier*

The sparse regression based classifier does not make any such assumption and it tries to find a sparse representation of each test sample using all the training samples as described in the Fig. 1.12. All the training samples from different classes are used as a dictionary to find the sparse representation of each test sample. The sparse coefficients of each test sample corresponding to the training data of each class are then used to find the approximation of the test sample. The label of a class whose training data gives best approximation of a test sample, gets assigned to that test sample.

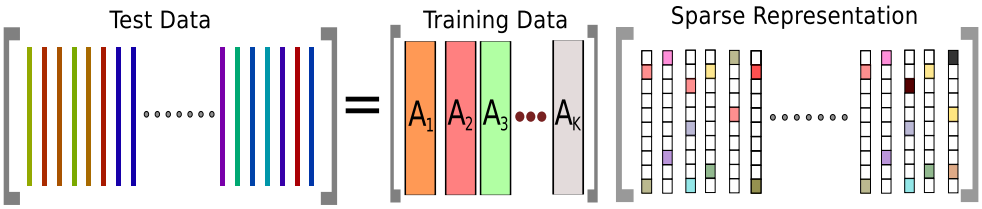


Figure 1.12: *Sparse regression based classifier*

The classification is performed either on the raw data or sometimes features are extracted from the data which are then classified. The features are often extracted with the intention of reducing the dimensionality, reducing the effect of noise, increase the class separability, increasing the speed of classification algorithm, etc..

The hyperspectral image classification is an important application that finds use in many areas such as land-cover change detection, urban planning, environmental monitoring, etc.. It is different from ordinary image classification problems where several images are available and each image belong to a single class. In hyperspectral

image classification, a large image of the study area is available in which each pixel is required to be assigned a particular class label. Each pixel of a satellite image covers a large ground area that make it challenging to manually collect lot of training data therefore hyperspectral image classification become challenging due to limited training data.

If the images are of very high spatial resolution then pixels of different classes can have very similar spectral signature that makes it difficult to classify at pixel level. Often super-pixel based classification or object based classification techniques are found suitable in that situation to reduce the dimensionality of the dataset. If the images are of very low spatial resolution then a pixel may consist of several classes then unmixing techniques are often applied to find the fraction of different materials present in that pixel and then label of the class with highest abundance may be assigned to that pixel.

Feature extraction is often required to reduce the high dimensionality of hyperspectral datasets. Various feature extraction techniques has been proposed in literature for hyperspectral feature extraction including principle component analysis (PCA), variations of wavelet transforms, dictionary learning based features, etc.. Different kinds of band ratios such as normalized difference vegetation index, normalized difference water index, Soil Adjusted Vegetation Index, normalized difference built-up index are often used to extract vegetation, water, soil, and urban areas respectively.

1.6 Research Contributions

This thesis focuses on four inter-related problems. The first one is on acquisition of multi-spectral images. From a signal processing driven perspective, we propose filter design and reconstruction techniques for single sensor multi-spectral cameras. The second problem is denoising of hyper-spectral images; this is a low-level image processing operation which usually follows acquisition. The third problem is that of unmixing. This is a unique step in hyper-spectral imaging which succeeds preprocessing (denoising) and precedes image analysis. The fourth and final problem is that of hyper-spectral classification- which is usually an automated image analysis problem.

The research contributions are summarized as follows:

- **Multispectral Demosaicing :** A generic filter array design has been proposed to capture multi-spectral images using hypothetical single-sensor multi-spectral cameras. The design idea is based on the uniform sampling of intensity values from each band irrespective of spectral properties of any particular band. A reconstruction technique has also been proposed to interpolate unknown intensity values of other bands at each pixel. The proposed method was evaluated on both color and multispectral image datasets. Quantitative evaluation of the proposed technique was done using peak signal to noise ratio.
- **Hyperspectral Denoising :** The denoising problem has been formulated as a mixed noise reduction problem. A general noise model has been considered

which accounts for not only Gaussian noise but also sparse noise. The inherent structure of hyperspectral images has been exploited by utilizing two approaches. The first method simultaneously exploits sparsity along spatial dimension using 2D DCT and spectral sparsity using 1D DCT whereas the second procedure considers 2D-total variation along the spatial dimension and 1D-total variation along the spectral dimension. In both the cases, the denoising problem has been formulated as an optimization problem whose solution has been derived using the split-Bregman approach. Experimental results demonstrate that proposed algorithm reduces a significant amount of noise from real noisy hyperspectral images. The proposed algorithms have been compared with existing state-of-the-art techniques. The quantitative and qualitative results demonstrate the superiority of proposed algorithms using peak signal to noise ratio, structural similarity, and the visual quality.

- **Hyperspectral Unmixing :** The hyperspectral unmixing problem is considered in a general scenario that includes the presence of mixed noise. The unmixing model explicitly takes into account both Gaussian noise and sparse noise. The unmixing problem has been formulated to exploit joint-sparsity of abundance maps. A total-variation based regularization has also been utilized for modeling smoothness of abundance maps. The split-Bregman technique has been employed to derive an algorithm for solving resulting optimization problem. Detailed experimental results on both synthetic and real hyperspectral images demonstrate the advantages of proposed technique.

- **Hyperspectral Classification :** The hyperspectral classification problem is considered from the dictionary learning perspective. A general deep dictionary learning framework has been proposed that learn the dictionary in a greedy fashion by avoiding the need for back-propagation. We compare our approach to the deep belief network (DBN) and stacked autoencoder (SAE) based techniques on standard hyperspectral classification datasets.

The research outcomes have been disseminated through publications in journals and conferences listed at page 91.

1.7 Dissertation Organization

Chapter 2 gives a description of some important basic concepts that are building blocks of this thesis. It includes a brief description of what is compressed sensing and its variants such as blind compressed sensing and Kronecker compressed sensing. Concepts related to dictionary learning and joint-sparsity are also discussed along with low-rank minimization and total variation minimization.

Chapter 3 discusses the multispectral demosaicing problem which is an extension of the color image demosaicing problem. This chapter also describes how demosaicing problem can be formulated as parameter estimation problem by simultaneously

utilizing nearby band information. Since there is no standard filter array to capture multispectral images, therefore, a description of proposed uniform multispectral filter array is also contained in this chapter.

Chapter 4 describes hyperspectral denoising problem formulated as a sparse recovery problem. First, a mixed noise model is described followed by brief literature review. Proposed noise reduction algorithms based on 3D-discrete cosine transform and 3D-spatio-spectral total variation are described. This chapter also presents comparative results with some standard denoising algorithms.

Chapter 5 is based on source separation problem in the context of hyperspectral images. First, the unmixing problem is described that how it can be modeled as a linear inverse problem. A description of terminology specific to the unmixing problem is also described. A solution based on joint-sparsity and total variation is the presented followed by detailed synthetic and real dataset experiments.

Chapter 6 introduces deep dictionary learning for hyperspectral image classification. First, the problem is described followed by some literature review. The greedy dictionary learning approach is then detailed followed by experimental results on standard hyperspectral image classification datasets. Finally, chapter 7 concludes the thesis with some limitations and future work.

Chapter 2

Preliminaries

2.1 Bregman Iteration for Basis Pursuit

This section describes a particular algorithm for the ℓ_1-minimization. This algorithm is based on the Bregman regularization. First, the Bregman distance is described followed by Bregman iterative algorithm and finally the split-Bregman algorithm is described. The Bregman iterative algorithm and split-Bregman technique are detailed in [44, 48].

The Bregman distance based on a convex function f is defined as

$$D_f^p(y,x) = f(y) - f(x) - p^T(y-x) \tag{2.1}$$

where $p \in \partial f(v)$ is some subgradient in the subdifferential set of f at y. The Bregman distance is geometrically demonstrated in Fig. 2.1 for the convex function f. It is the distance between the function value at the point y and the value of its first order Taylor expansion at the point x. The Bregman distance in Eq. (2.1) is not a distance in the

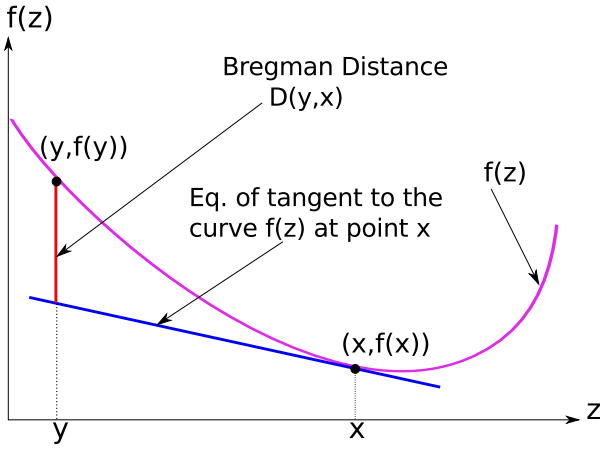


Figure 2.1: *Bregman distance at the point y between the convex function f and its first order Taylor expansion at the point x.*

usual sense since it is not symmetric i.e. $D_f^p(y,x) \neq D_f^p(x,y)$ but it has other properties of a metric such as non-negativity. The non-negativity can be proved from the convexity of the function f. We know that for a convex function f, following holds:

$$f(y) \geq f(x) + p^T(y-x) \quad \forall x, y \in \text{domain(f)}$$

on rearrangement, we get

$$D_f^p(y,x) = f(y) - f(x) - p^T(y-x) \geq 0$$

Therefore, the Bregman distance is non-negative.

Consider the Basis Pursuit (BP) problem

$$\min_x \|x\|_1 \quad \text{such that} \quad y = Ax \tag{2.2}$$

this problem can be solved using linear programming solvers but they are not suitable for matrices that are dense and large-scale. Sometimes matrix A is formed by the rows of orthogonal transforms for which fast operations for Ax and A^Tx exists. Moreover, observations in y are often contaminated by noise therefore instead of solving the problem (2.2), we can relax the constraint and solve following Basis Pursuit Denoising (BPDN) problem

$$\min_x \|x\|_1 + \frac{\lambda}{2}\|y - Ax\|_2^2. \tag{2.3}$$

There are various algorithms such as Spectral Projected Gradient for L1 (SPGL1) [49], StOMP [50], ISTA [51] for solving the BPDN problem or the related Least Absolute Shrinkage and Selection Operator (LASSO) problem.

Bregman in [52] suggested to solve following problem iteratively to get the solution of the equation (2.3)

$$x^{k+1} = \arg\min_x D_J^p(x, x^k) + \frac{\lambda}{2}\|y - Ax\|_2^2 \tag{2.4}$$

where $J(x) = \|x\|_1$, it can be re-written as

$$x^{k+1} = \arg\min_x J(x) - J(x^k) - \langle p^k, x - x^k\rangle + \frac{\lambda}{2}\|y - Ax\|_2^2.$$

From the optimality of x^{k+1}, we have $0 \in \partial J(x^{k+1}) - p^k - \lambda A^T(y - Ax^{k+1})$ we have,

$$p^{k+1} = p^k + \lambda A^T(y - Ax^{k+1})$$

Thus, Bregman iterative algorithm is summarized in Algorithm 1. At each step, this algorithm in version 1 requires to solve an optimization problem containing a subdifferential term and therefore it is a costly step. The Bregman iterative algorithm [48] proposed equivalent but simplified version 2 of this algorithm shown in Algorithm 2.

Algorithm 1 Bregman Iteration Version 1

input: $x^0 = 0$, $p^0 = 0$
for $k = 0, 1, \ldots$ **do**
  $x^{k+1} = \arg\min_x D_J^{p^k}(x, x^k) + \frac{\lambda}{2}\|y - Ax\|_2^2$
  $p^{k+1} = p^k + \lambda A^T(y - Ax^{k+1})$
end for

Algorithm 2 Bregman Iteration Version 2

input: $x^0 = 0$, $y^0 = 0$
for $k = 0, 1, \ldots$ **do**
  $y^{k+1} = y + (y^k - Ax^{k+1})$
  $x^{k+1} = \arg\min_x J(x) + \frac{\lambda}{2}\|y^{k+1} - Ax\|_2^2$
end for

To show the equivalence of algorithms 1 and 2, Let x^k and $\bar{x}^k$ are solutions to the problems in version 1 and 2. at iteration 0, both the algorithms solve the same problem

$$\min_x J(x) + \frac{\lambda}{2}\|y - Ax\|_2^2$$

because initially $D_J^{p^0} = J(x)$ and $y^1 = y$. But since this problem may have more than one solution therefore it is assumed that both may have resulted in different solutions x^1 and $\bar{x}^1$ respectively. It has been shown in [53] that gradient $A^T(y - Ax)$ is constant for all optimal solutions i.e $A^T(y - Ax^1) = A^T(y - A\bar{x}^1)$. Thus we have,

$$p^1 = p^0 + \lambda A^T(y - Ax^1) = \lambda A^T(y - Ax^1) = \lambda A^T(y - A\bar{x}^1) = \lambda A^T(y^1 - A\bar{x}^1)$$

Now by induction, we can show that $p^k = \lambda A^T(y^k - A\bar{x}^k)$. Consider p^{k+1} as in version 1:

$$\begin{aligned} p^{k+1} = p^k + \lambda A^T(y - Ax^{k+1}) &= p^k + \lambda A^T(y - A\bar{x}^{k+1}) \\ &= \lambda A^T(y^k - A\bar{x}^k) - \lambda A^T(A\bar{x}^{k+1} - y) \\ &= \lambda A^T(y^k - A\bar{x}^k - A\bar{x}^{k+1} + y) \\ &= \lambda A^T(y + (y^k - A\bar{x}^k) - A\bar{x}^{k+1}) \\ &= \lambda A^T(y^{k+1} - A\bar{x}^{k+1}) \end{aligned}$$

Thus we get $p^k = \lambda A^T(y^k - A\bar{x}^k)$. This value of p^k is used in proving that both the objective functions in version 1 and 2 are same up-to a constant. Consider the objective function in version 1:

$$\begin{aligned} \arg\min_x D_J^{p^k}(x, x^k) + \frac{1}{2}\|Ax - f\|_2^2 &= J(x) - J(x^k) - \langle p^k, x - x^k\rangle + \frac{1}{2}\|Ax - f^{k+1}\|_2^2 \\ &= J(x) - \langle p^k, x\rangle + \frac{1}{2}\|Ax - f^{k+1}\|_2^2 + C_1 \\ &= J(x) - \langle A^T(f^k - A\bar{x}^k), x\rangle + \frac{1}{2}\|Ax - f^{k+1}\|_2^2 + C_1 \\ &= J(x) - \langle f^k - A\bar{x}^k, Ax\rangle + \frac{1}{2}\|Ax - f^{k+1}\|_2^2 + C_1 \\ &= J(x) + \frac{1}{2}\|Ax - (f + (f^k - A\bar{x}^k))\|_2^2 + C_2 \\ &= J(x) + \frac{1}{2}\|Ax - f^{k+1}\|_2^2 + C_2. \end{aligned}$$

Thus both the Algorithms 1 and 2 are equivalent up-to a constant. The detailed convergence analysis ensuring to get optimal solution in finite number of steps, has been discussed in [48]. Here we restate the Theorem that shows the solution of BPDN problem in (2.3) by Bregman iterative algorithm can result in a solution to original BP problem in (2.2).

Theorem 1. *Suppose an iterate x^k from line 3 of algorithm 1 satisfies $y = Ax^k$; then x^k is a solution of the basis pursuit problem* (2.2).

Proof. By the non-negativity of Bregman Distance, we have

$$\begin{aligned} D_J^{p^k}(x, x^k) &\geq 0 \\ J(x) - J(x^k) - \langle p^k, x - x^k \rangle &\geq 0 \\ J(x^k) &\leq J(x) - \langle p^k, x - x^k \rangle \\ J(x^k) &\leq J(x) - \langle \lambda A^T(y^k - Ax^k), x - x^k \rangle \\ J(x^k) &\leq J(x) - \lambda \langle A^T(y^k - Ax^k), x - x^k \rangle \\ J(x^k) &\leq J(x) - \lambda \langle y^k - Ax^k, Ax - Ax^k \rangle \\ J(x^k) &\leq J(x) - \lambda \langle y^k - y, Ax - y \rangle \end{aligned}$$

Therefore, x^k satisfies $J(x^k) \leq J(x)$ for any x satisfying $y = Ax$, hence x^k is an optimal solution of the basis pursuit problem in (2.2). □

The Bregman iteration has several benefits over traditional continuation methods. It converges very quickly specially for the problems containing ℓ_1-regularization term. The convergence proof has been discussed in the appendix of [44]. Another advantage is that is does not require to update the parameter λ in every iteration. Thus we can choose to work with a constant λ that works for the sub-problem. Bregman iteration also avoids problems of numerical instabilities that occur as $\lambda \to \infty$ that occur with continuation methods.

2.2 The Split-Bregman Algorithm

This section describes how the split-Bregman algorithm [44] can be used to solve basis pursuit problem (2.2). Here we just repeat the problem for convenience:

$$\min_x \|x\|_1 \quad \text{such that} \quad y = Ax$$

as discussed in previous section, Algorithm 2 can be utilized to solve this problem with $y^{k+1} = y + (y^k - Ax^{k+1})$ as follows:

$$x^{k+1} = \arg\min_x J(x) + \frac{\lambda}{2}\|y^{k+1} - Ax\|_2^2.$$

Next we describe The split-Bregman algorithm to solve the above problem. By using a new variable d such that $d = x$, this problem can be re-written as

$$\min_x \|d\|_1 + \frac{\mu}{2}\|y^{k+1} - Ax\|_2^2 \quad \text{subject to } d = x$$

Converting it to unconstrained problem, we get

$$\min_{x,d} \|d\|_1 + \frac{\mu}{2}\|y^{k+1} - Ax\|_2^2 + \frac{\lambda}{2}\|d - x\|_2^2$$

let $J(x,d) = \|d\|_1 + \frac{\mu}{2}\|y^{k+1} - Ax\|_2^2$, we have

$$\min_{x,d} J(x,d) + \frac{\lambda}{2}\|d - x\|_2^2$$

applying Bregman iteration to this problem, we get following iterative procedure:

$$(x^{k+1}, d^{k+1}) = \arg\min_{x,d} D_J^p(x, x^k, d, d^k) + \frac{\lambda}{2}\|d - x\|_2^2 \tag{2.5}$$

and the sub-gradients can be updated as

$$p_x^{k+1} = p_x^k + \lambda(d^{k+1} - x^{k+1}) = \lambda \sum_{j=1}^{k+1}(d^j - x^j)$$

$$p_d^{k+1} = p_d^k - \lambda(d^{k+1} - x^{k+1}) = \lambda \sum_{j=1}^{k+1}(x^j - d^j)$$

define the variable b^k as $b^{k+1} = b^k + (x^{k+1} - d^{k+1})$ with $b^0 = 0$ then

$$b^{k+1} = b^0 + (x^1 - d^1) + \cdots + (x^k - d^k) + (x^{k+1} - d^{k+1}) = \sum_{i=1}^{k+1}(x^i - d^i)$$

Thus $p_x^{k+1} = -\lambda b^{k+1}$ and $p_d^{k+1} = \lambda b^{k+1}$. The main advantage of this strategy is that it combines both the subgradients p_x^k and p_d^k by some constant multiple of the variable b^k.

The objective function in Eq. (2.5) can be expressed as:

$$\begin{aligned}
&D_J^p(x, x^k, d, d^k) + \frac{\lambda}{2}\|d - x\|_2^2 \\
=&J(x,d) - J(x^k, d^k) - \langle p_x^k, x - x^k\rangle - \langle p_d^k, d - d^k\rangle + \frac{\lambda}{2}\|d - x\|_2^2 \\
=&J(x,d) - J(x^k, d^k) - \langle -\lambda b^k, x - x^k\rangle - \langle \lambda b^k, d - d^k\rangle + \frac{\lambda}{2}\|d - x\|_2^2 \\
=&J(x,d) - \langle -\lambda b^k, x - x^k\rangle - \langle \lambda b^k, d - d^k\rangle + \frac{\lambda}{2}\|d - x\|_2^2 + C_1 \\
=&J(x,d) + \lambda\langle b^k, x - d\rangle + \frac{\lambda}{2}\|d - x\|_2^2 + C_2 \\
=&J(x,d) + \frac{\lambda}{2}\left\|d - x - b^k\right\|_2^2 - \frac{\lambda}{2}\left\|b^k\right\|_2^2 + C_2 \\
=&J(x,d) + \frac{\lambda}{2}\left\|d - x - b^k\right\|_2^2 + C_3 \\
=&\|d\|_1 + \frac{\mu}{2}\|y^{k+1} - Ax\|_2^2 + \frac{\lambda}{2}\left\|d - x - b^k\right\|_2^2 + C_3
\end{aligned}$$

The benefit of this procedure is that it decouples the variable d in ℓ_1 and ℓ_2 part and the variable x appears in differentiable ℓ_2-norm terms only. The problem in Eq. (2.5) can be expressed as

$$\begin{aligned}(x^{k+1}, d^{k+1}) &= \arg\min_{x,d} \|d\|_1 + \frac{\mu}{2}\|y^{k+1} - Ax\|_2^2 + \frac{\lambda}{2}\left\|d - x - b^k\right\|_2^2 \\ b^{k+1} &= b^k + x^{k+1} - d^{k+1}\end{aligned} \tag{2.6}$$

The algorithm to solve BP problem in Eq. (2.2) can be summarized as

Algorithm 3 The Split-Bregman Algorithm

1: input: $x^0 = 0, b^0 = 0, y^0 = 0$
2: **while** not converge **do**
3: **for** $k = 0, 1, \ldots$ **do**
4: $x^{k+1} = \arg\min_{x} \frac{\mu}{2} \|y^{k+1} - Ax\|_2^2 + \frac{\lambda}{2} \left\|d - x - b^k\right\|_2^2$
5: $d^{k+1} = \arg\min_{,d} \|d\|_1 + \frac{\lambda}{2} \left\|d - x - b^k\right\|_2^2$
6: $b^{k+1} = b^k + x^{k+1} - d^{k+1}$
7: **end for**
8: $y^{k+1} = y + (y^k - Ax^{k+1})$
9: **end while**

2.3 Dictionary Learning

Compressed sensing requires a signal to be sparse in some sparsifying transform domain. Most of the practical signals have the sparse representation in some fixed data independent transform domain such as Fourier, Wavelet, DCT, etc. However, it is possible that a signal does not have sparse representation in some existing domain then it may be required to learn the sparsifying transform for that signal. Dictionary learning is the technique to learn a sparsifying transform domain from existing training data to get sparse representation of certain family of signals.

A dictionary is an over-complete basis i.e. it has more columns than rows. The columns of a dictionary are often referred to as atoms. These atoms may be the linear combinations of other atoms in the dictionary. The benefit of a dictionary over the basis is that a dictionary may result in even sparser representation as compared to a fixed basis. However it should be remembered that the dictionary is not ubiquitous unlike the fixed transforms. Dictionaries are learned to solve a problem for a particular class of signals, they are not intended to be generalizable to others.

Dictionary learning has been discussed in various articles including [54–56]. Training data is required to learn a dictionary. Suppose we are given the data $X \in \mathbb{R}^{m \times n}$ such that each column x_i represents a training sample of size m, thus we have total

n training samples. Dictionary learning aims at learning a dictionary and the sparse representation of these signals in Y. The dictionary learning problem can be expressed as

$$\min_{D\in\mathcal{C},\alpha\in\mathbb{R}^{k\times n}} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\|x_i - D\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1\right) \tag{2.7}$$

where the constraint set $\mathcal{C}$ is defined as

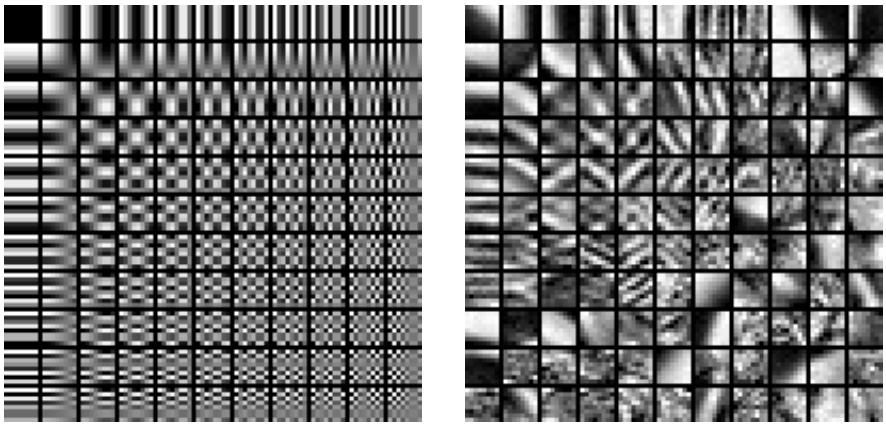
$$\mathcal{C} = \{D \in \mathbb{R}^{m\times k}\ s.t\ \forall j = 1,\ldots,k,\ d_j^T d_j \leq 1\}.$$

Various algorithms have been proposed in literature to solve the dictionary learning problem. The algorithms like Method of Optimal Directions (MOD) [57] solve for dictionary and sparse representation alternately. The K-SVD algorithm [58] takes a total of K rank-one approximations to determine all K atoms of the dictionary sequentially. The rank-one approximations are done using Singular Value Decomposition (SVD).

Figure 2.2 shows two over-complete dictionaries. Each atom d_i of the over-complete 1D-DCT dictionary $D^{1d} \in \mathbb{R}^{8\times 11}$ was generated using following equation as described in [56]:

$$d_k = cos(i-1)(k-1)\pi/11, \quad i = 1,\ldots,8.$$

The 2D-DCT dictionary $D^{2d} \in \mathbb{R}^{64\times 121}$ was then generated using $D^{2d} = D^{1d} \otimes D^{1d}$. Figure 2.2(a) shows each of the 121 atoms of the generated 2D-DCT dictionary. This



(a) *DCT dictionary* **(b)** *KSVD dictionary*

Figure 2.2: *Example of dictionary learning*

representation is generated independent of data and is suitable for certain classes of signals. The learned dictionaries are data dependent and are generated for a specific application related data. Figure 2.2(b) shows an example of dictionary learned by using K-SVD algorithm on the parrot image of size 512×768. A total of $40,000$ patches of size 8×8 were extracted from the image to learn the dictionary of size 64×121. A limitation of dictionary based methods is that, it is difficult to learn a dictionary for large size signals. Also these dictionaries are slow compared to fixed orthogonal transforms for which fast FFT based implementations exists.

2.4 Joint-Sparsity

The problem of recovering a sparse vector x from its lower dimensional projections y can be expressed as:

$$y = Ax + \eta, \quad \eta \rightarrow N(0, \sigma^2) \tag{2.8}$$

where η is noise. This problem is called single measurement vector (SMV) recovery problem. There are applications where multiple measurement vectors (MMV) are available and the problem of MMV recovery can be defined as follows:

$$Y = AX + N \tag{2.9}$$

where $A \in \mathbb{R}^{m \times n}$ and $X \in \mathbb{R}^{n \times L}$ and $Y \in \mathbb{R}^{m \times L}$. The matrices Y, X are called multiple measurement matrix and source matrix respectively. Here L represents total number of multiple measurement vectors. This problem (2.9) can be decomposed into several single measurement vector (SMV) problems as:

$$b^\ell = Ax^\ell \quad \ell = 1, \ldots, L$$

where $X = [x^1, \ldots, x^L]$ and $B = [b^1, \ldots, b^L]$. The MMV recovery can be formulated as mixed norm minimization problem:

$$\min_X \|X\|_{2,1} \quad \text{subject to} \quad \|Y - AX\|_F \leq \epsilon, \quad \|X\|_{2,1} = \sum_{i=1}^{n} \|X^{i\rightarrow}\|_2 \tag{2.10}$$

where $\|.^{i\rightarrow}\|_2$ denote the ℓ_2-norm of i^{th} row. MMV problem has been discussed in [42, 59, 60]. The ℓ_2-norm over the rows enforces the selected rows to have non-zero values for all the elements; the sum over the ℓ_2-norms promotes selection of only a few rows.

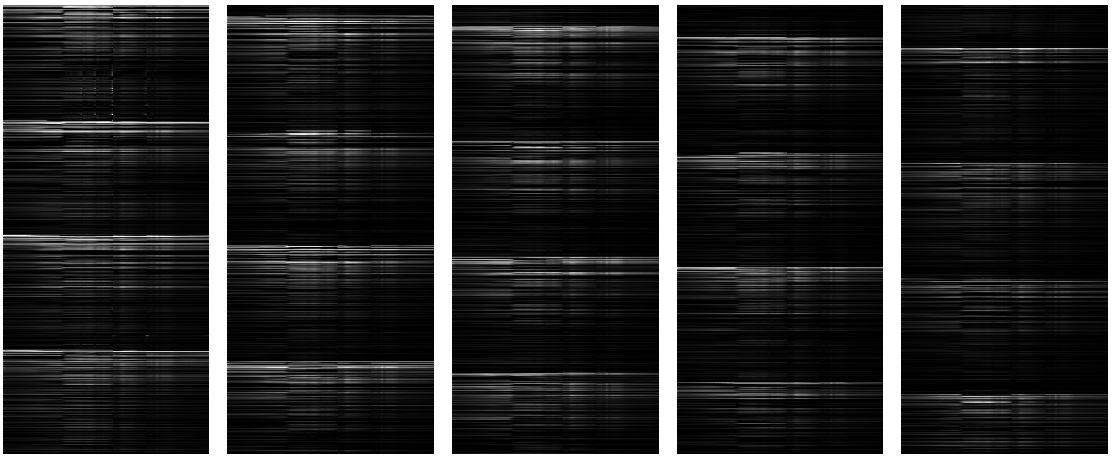


Figure 2.3: *Example of joint sparsity of WDC image. Each of the five matrices have 1000 rows and 191 columns. Leftmost matrix corresponds to DCT coefficients of first 1000 pixels. The second matrix corresponds to pixels 1001 to 2000. Similarly rightmost matrix corresponds to pixels 4001 to 5000.*

Figure 2.3 shows an example of joint sparsity on a real hyperspectral image. First, the image was represented as matrix form in which each column is a band. Thus a

hyperspectral image of size 256/$times$256/$times$191 corresponds to matrix $X$of size 65536×191. This was then multiplied by 2D-DCT to get the DCT coefficients Z such that $Z = DX$. The white horizontal lines corresponds to DCT coefficient value of different bands. It can be observed that hyperspectral images have joint-sparse structure.

Chapter 3

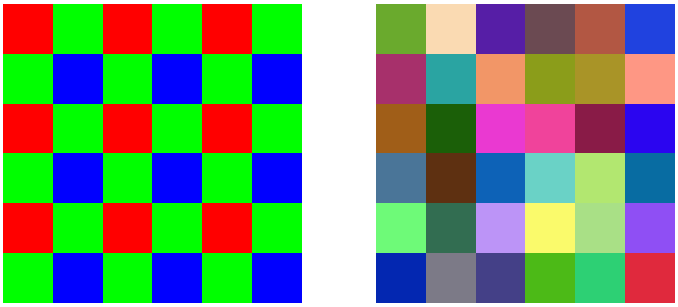
Multispectral Demosaicing

This chapter describes the demosaicing problem for multispectral imaging. The multispectral demosaicing problem is an extension of the color image demosaicing problem that is used in the design of single-sensor low-cost cameras. First, a brief literature review is described followed by proposed uniform multispectral filter array design, and its properties. Then a multispectral demosaicing algorithm is described to reconstruct full multispectral image followed by experimental results. This work considers two aspects of multi-spectral camera design namely- design of multi-spectral filter to capture multiple bands using single-sensor and the demosaicing technique to interpolate missing intensity values.

3.1 Literature Review

This section provides a brief review of some existing Multispectral Filter Array (MSFA) and demosaicing algorithms based on single-sensor architecture. Unlike Bayer pattern, there is no popular MSFA to acquire multispectral images. In literature, few MSFA designs have been proposed to capture multiple bands using single-sensor architecture. Traditional CFA for color imaging is Bayer filter [14] which is used in many compact digital cameras. It is shown in Fig. 3.1(a). Bayer filter allows us to capture a color image using single image sensor. It is placed on the top of the image sensor such that only one intensity value is captured at every pixel. This filter captures 50% samples of green band and 25% each of red and blue and. The idea behind this approach is that human eye is more sensitive to the green band as compare to the red and blue band; therefore, more samples of the green band can help in the better visual quality of the image. The sub-optimality of Bayer kind filter have been proved quantitatively in [61] where authors have proposed a panchromatic filter. Bayer CFA design may not be extended to capture multispectral images since it is specific to visible bands with fixed proportions for red, green, and blue bands. The panchromatic filter design is shown in Figure 3.1(b). This design is primarily based on the assumption that reconstruction quality can be enhanced by capturing some fraction of light from each of the visible band at every pixel, unlike Bayer pattern that completely discards information from other bands.

The paper [17] discusses the design requirements and evaluation criterion for generating a generic MSFA. Total three design requirements have been reported in this



(a) *Bayer filter* **(b)** *panchromatic filter*

Figure 3.1: *Bayer and panchromatic color filter array designs*

work. First, it requires prior knowledge about the importance of particular band to be captured by the MSFA since the filter array design is such that the band which has the high probability of appearance will be sampled more. The second design requirement discussed in the paper is about *spectral consistency* which says that pixels sensitive to certain spectral band should always have the same pattern of neighbors. This property is useful to avoid phenomenon such as *cross-talk*. Finally, *spatial uniformity* requires that MSFA should sample entire image as evenly as possible so that at every location in the image there are samples to be used during interpolation process. This MSFA design is generic enough that it can also generate Bayer filter and Sony-CFA as specific cases. This MSFA assumption is that we have prior knowledge of the application area where we want to use the camera. Therefore, the camera design will become application specific.

Based on above MSFA design, authors in [18] proposed multispectral demosaicing algorithm named BTES algorithm. They analyzed spectral correlation using two techniques namely constant color difference and constant edge information. It was found that the spectral correlation in multispectral images is not as high as that of color images. This is mainly due to the large inter-band gap as compared to color images. Since every band capture very different signature, the edge information of different spectral bands would not be the same. This statement is true only when there is large spectral gap however for multispectral images which have the low spectral gap we can have same edge information in different bands.

The work [1] proposes a multispectral demosaicing algorithm for five-band multispectral images. Authors purposed a multispectral color filter array (MCFA) to capture five bands namely red, green, blue, orange, and cyan. This MCFA has 50% samples of the green band so that this band can be interpolated with high accuracy as compared to other bands. This green band is then used as a guide image for the interpolation of other lower sampled bands. Figure 3.2 shows the block diagram of the proposed algorithm. Authors utilized adaptive kernel [62] during the upsampling process. This kernel is used in Gaussian upsampling and joint bilateral upsampling to obtain adaptive Gaussian upsampling and adaptive joint bilateral upsampling algorithms. First, the adaptive kernel is generated which is then applied on green band samples to generate a guide image. The guide image is then used to interpolate all the bands using an adaptive joint bilateral upsampling filter. A guided filter [63] based

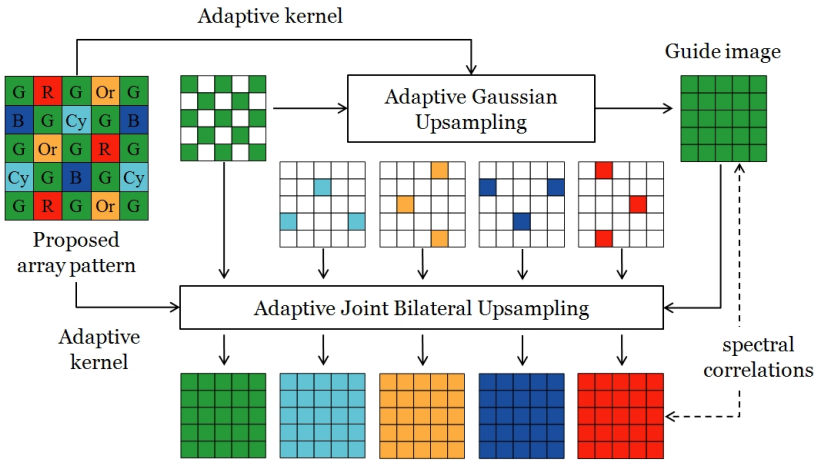


Figure 3.2: *Adaptive Kernel Upsampling Algorithm[1]*

multispectral demosaicing algorithm have been proposed in [2]. The guided filter generates the output by a linear transform of a given guide image. First, the adaptive Gaussian upsampling is applied on the green band samples to obtain a guide image which is then used in the guided filter to reconstruct full image.

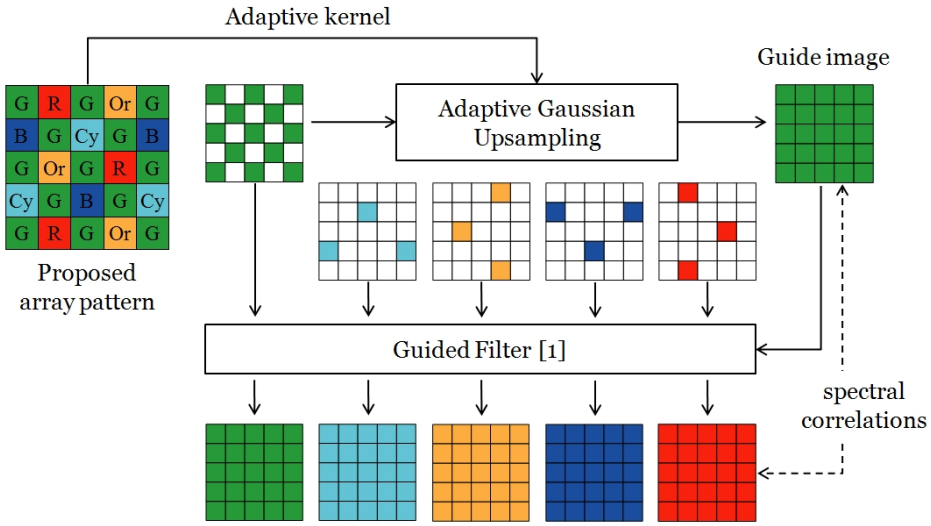


Figure 3.3: *Multispectral Demosaicing using Guided Filter [2]*

This algorithm has been purposed for five bands with the assumption that green band is the dominating band. This is not essentially true in general for multispectral imaging applications. Effect of increasing the number of bands have not been discussed. This algorithm also does not account for spectral gap between the bands.

An approach to capture near-infrared (NIR) band along with RGB bands has been discussed in [64] using single sensor architecture. Since silicon is sensitive to NIR wavelengths, the same Bayer filter can be modified to acquire NIR band along with RGB bands. Authors have proposed to remove interference filter also called hot-

mirror which is used to filter NIR wavelengths from reaching to the sensors inside digital cameras. When hot-mirror is removed then NIR information is also captured at each pixel of the mosaiced image. Authors have proposed to modify Bayer filter such that both the optical filters corresponding to green wavelength will have different transmittance as opposed to Bayer filter in which both the green filters have the same transmittance. The transmittance is randomly selected so as to have better reconstruction using compressed sensing since compressed sensing theory gives certain guarantees for reconstruction using random sampling matrices. The problem was formulated as sparse recovery problem using green and NIR band. Both green and NIR bands were sparsified using the combination of DCT and principle component analysis (PCA) matrices which were further updated using K-SVD dictionary learning approach. Red and blue bands were recovered using approach in [65] after subtracting the NIR information from corresponding pixels in the mosaiced image.

3.2 Uniform Multi-Spectral Filter Array

The CFA used in many digital cameras are based on Bayer pattern [14] which captures 50% samples of green band and 25% each of red and blue bands. The reason for higher sampling of green band is the claim that the human eye is more sensitive to green color as compared to red and blue. In case of multi-spectral images, there is no prior information about the preference of one band over the other and therefore we have designed the multispectral filter to collect equal number of samples for each band.

Algorithm 4 Generate Uniform Multi-Spectral Filter

1: input: K: number of bands e_j : K-element row vector having all elements as zero except j^{th} element which is one.
2: output: UMSF (Uniform Multi-Spectral Filter)
3: **Initialize** $P = \begin{pmatrix} e_K \\ e_1 \\ \vdots \\ e_{K-1} \end{pmatrix}$, $\text{UMSF} = \begin{pmatrix} row_1 \\ row_2 \\ \vdots \\ row_K \end{pmatrix}$
4: $row_1 = \begin{bmatrix} 1 & 2 & \dots & K \end{bmatrix}$
5: **for** $j = 2$ to K **do**
6: $\quad row_j = row_{j-1} \times P$
7: **end for**
8: return UMSF

Proposed Algorithm to generate the K-band multi-spectral filter is shown in Algorithm 4. There are two main multi-spectral filter design considerations namely *spectral consistency* and *spatial uniformity* which were introduced in [17]. The spectral consistency requirement is concerned with same reconstruction quality throughout the image. It requires that for each pixel, there should be same number of pixels of a certain spectral band at a certain distance. The proposed Uniform Multispectral Filter

B_1	B_2	...	B_{K-1}	B_K
B_2	B_3	...	B_K	B_1
⋮	⋮	...	⋮	⋮
B_{K-1}	B_K	...	B_{K-3}	B_{K-2}
B_K	B_1	...	B_{K-2}	B_{K-1}

Figure 3.4: *Uniform Multi-Spectral Filter*

Array (UMSFA) satisfies this requirement since neighborhood bands remain the same for each pixel in the image. Spatial uniformity requires that the number of samples should be collected from all parts in the image as evenly as possible. This requirement is also satisfied by the proposed UMSFA because samples corresponding to each band are uniformly captured in the whole image. The generated UMSFA design is shown in Fig. 3.4 to capture K-band multi-spectral image. Here $B_i \forall i \in [1, K]$ represents that only the intensity of i^{th} band will be captured by the image sensor at that pixel location. The captured under-sampled image is refereed as raw image which have only one sample at each pixel and remaining $K - 1$ samples have to be interpolated.

The advantage of UMSFA is that it can be easily extended to capture any number of bands with a repeatable deterministic architecture unlike random sampling patterns. Another advantage of having a uniform pattern is that, we can frame the demosaicing problem as a linear interpolation problem something which is not possible for random patterns.

B_1	B_2	B_3	B_4	B_1	B_2	B_3	B_4
B_2	B_3	B_4	B_1	B_2	B_3	B_4	B_1
B_3	B_4	B_1	B_2	B_3	B_4	B_1	B_2
B_4	B_1	B_2	B_3	B_4	B_1	B_2	B_3
B_1	B_2	B_3	B_4	B_1	B_2	B_3	B_4
B_2	B_3	B_4	B_1	B_2	B_3	B_4	B_1
B_3	B_4	B_1	B_2	B_3	B_4	B_1	B_2
B_4	B_1	B_2	B_3	B_4	B_1	B_2	B_3
B_1	B_2	B_3	B_4	B_1	B_2	B_3	B_4
B_2	B_3	B_4	B_1	B_2	B_3	B_4	B_1

(a) *Four band filter array*

B_3	B_4	B_1
B_4	B_1	B_2
B_1	B_2	B_3

(b) *Pattern1*

B_4	B_1	B_2
B_1	B_2	B_3
B_2	B_3	B_4

(c) *Pattern2*

B_1	B_2	B_3
B_2	B_3	B_4
B_3	B_4	B_1

(d) *Pattern3*

B_2	B_3	B_4
B_3	B_4	B_1
B_4	B_1	B_2

(e) *Pattern4*

Figure 3.5: *Example of multi-spectral filter and four* 3×3 *repeating patterns within this filter array to capture four bands using single-sensor.*

In general if we are capturing K bands of a multi-spectral image using single-sensor then there will be K different repeating patterns occurring in the raw image and for each of these K patterns we need to interpolate $(K - 1)$ intensity values at each pixel. To illustrate this point, Fig. 3.5 shows a concrete example of UMSFA to capture four bands using single-sensor. Figures 3.5(b), 3.5(c), 3.5(d), and 3.5(e) shows four example patterns with 3×3 window size that will occur in case of four band imagery. This four band UMSFA has four different repeating patterns and it is required to interpolate

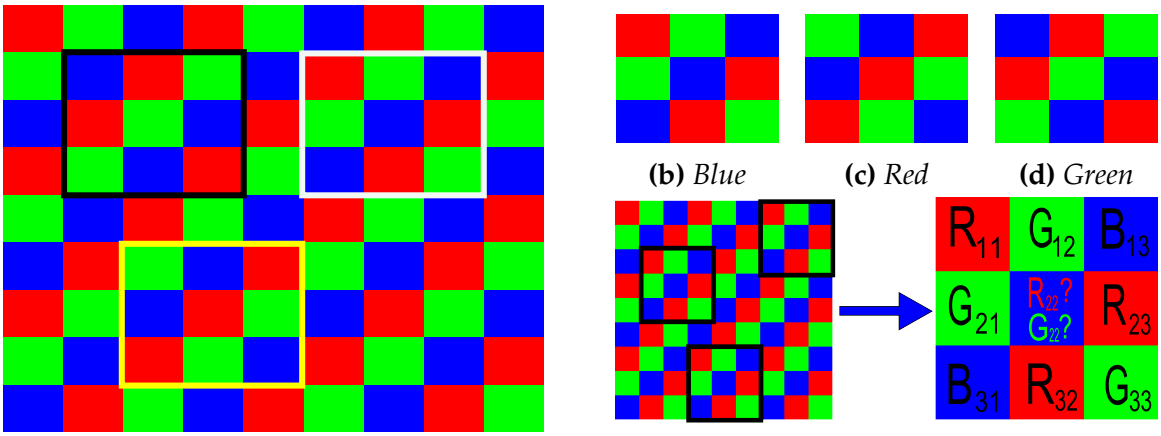
three different intensity values at each pixel.

3.3 Proposed Technique

Basic linear interpolation techniques such as bi-linear interpolation uses neighboring values of the same band to interpolate unknown value at a pixel. This strategy has scope for improvement because it does not consider inter-band correlation. Our proposed technique is inspired by linear time commercial techniques for color imaging such as [16] which is gradient corrected bi-linear interpolation that incorporates local information from same band as well as nearby bands. We have used same philosophy to propose linear time algorithm for interpolation of missing band values for multi-spectral imaging.

3.3.1 Proposed Demosaicing Algorithm

The interpolation of missing pixel values is based on exploring the dependence of a pixel on the neighborhood pixels of the same band as well as other bands. This dependence have been explored by representing intensity value at each pixel as linear combination of intensity values of neighborhood pixels in a window of size $w \times w$. Figure 3.6 shows an example of applying the proposed technique on a three band color



(a) *3 bands raw image from UMSFA*

(e) *Interpolation process*

Figure 3.6: *Example of proposed technique on RGB image with* 3×3 *window size. (a) The raw image have three different repeating patterns as shown in (b), (c), and (d). The central pixel is either blue, red, or green. For each of the repeating pattern, we have to estimate two values at the central pixel as shown in (e).*

image captured with proposed UMSFA. Figure 3.6(e) shows the interpolation process at blue pixel as central pixel. We propose to express the red and green pixel value at a

blue pixel as follows:

$$R_{22} = \alpha_1 R_{11} + \alpha_2 G_{12} + \cdots + \alpha_8 B_{32} + \alpha_9 G_{33} = [R_{11} \ldots G_{33}][\alpha_1 \ldots \alpha_9]^T$$
$$G_{22} = \beta_1 R_{11} + \beta_2 G_{12} + \cdots + \beta_8 B_{32} + \beta_9 G_{33} = [R_{11} \ldots G_{33}][\beta_1 \ldots \beta_9]^T,$$

where α_i and β_i represents the interpolation weights that are assumed to remain constant. These two equations are obtained corresponding to one 3×3 window but there will be as many equations as the number of blue pixels. Compactly, this linear combination can be expressed as:

$$y = \mathbf{u}^T \mathbf{x}$$

where $\mathbf{u}, \mathbf{x} \in \mathbb{R}^{w^2 \times 1}$, vector $\mathbf{u}$ is representing neighboring intensity values and vector $\mathbf{x}$ is representing contribution of corresponding neighboring values to interpolate one of the missing intensity value represented by y. More generally we can express the interpolation problem for K-band multi-spectral image with K repeating patterns and $(K-1)$ missing values at each pixel as follows:

$$Y = AX \tag{3.1}$$

where $Y \in \mathbb{R}^{M \times (K-1)}$, $A \in \mathbb{R}^{M \times w^2}$, and $X \in \mathbb{R}^{w^2 \times (K-1)}$. Each row of matrix A is made up of w^2 elements from $w \times w$ window. There will be total M such rows in A having $M = \left\lfloor \frac{m}{k} \right\rfloor \times \left\lfloor \frac{n}{k} \right\rfloor$ with raw image being of size $m \times n$. Each column of X represents coefficients by which neighboring pixels in $w \times w$ window contributed in the interpolation of $(K-1)$ missing band values. Since $M >> w^2$, therefore matrix A will be a tall matrix and the system of equations will become over-determined that can be solved using least square method. Since there are total K patterns in a K band imagery therefore system of equations (3.1) was solved $K \times (K-1)$ times to get filters for interpolating missing band values at each pixel.

Continuing with specific example of four-band UMSFA, suppose that an image has been captured by utilizing the UMSFA shown in Fig. 3.5. Each non-border pixel will be central pixel in one of the four repeating patterns. Consider the pattern 1 in Fig. 3.5(b) that capture first band value at central pixel. Thus, remaining three band values namely B_2, B_3, B_4 need to be estimated at the central pixel. These three band values can be estimated by exploiting their relationship with the known neighboring pixels of same band as well as other bands. Particularly, this relationship can be represented by expressing the central pixel value as the linear combination of neighboring pixel values. The unknown intensity values at B_2, B_3, B_4 for pattern 1 can be expressed as:

$$B_2 = \sum_{i=1}^{8} a_i x_{i1}, \quad B_3 = \sum_{i=1}^{8} a_i x_{i2}, \quad B_4 = \sum_{i=1}^{8} a_i x_{i3}$$

The x_{i1}, x_{i2}, and x_{i3} values represent the weights of neighboring pixel values a_i's for the interpolation of unknown B_2, B_3, B_4 respectively at the central pixel location. If the weights x_{i1}, x_{i2}, and x_{i3} are known then the unknown pixel values can be estimated. These weights remain constant for a particular pattern in the whole image. The above three equation are for one pixel. For a multispectral image of size $300 \times 300 \times 4$, the

raw image will be of size 300×300. The matrix A corresponding to each pattern will have $75 \times 75 = 5625$ rows and three unknown parameter vectors need to be estimated. In general, for each pattern in the K band raw image the interpolation process can be expressed as:

$$\begin{bmatrix} Y_1 & Y_2 & \dots & Y_{K-1} \end{bmatrix} = A \begin{bmatrix} X_1 & X_2 & \dots & X_{K-1} \end{bmatrix} \tag{3.2}$$

where $Y_i \ \forall i \in [1, K-1]$ represents unknown values, X_i represents interpolation weights, A represents known pixel values. This process is repeated for each pattern in the raw image to interpolate unknown pixel values at each central pixel location. The next section discusses how to estimate these weight vectors $X_i \ \forall i \in [1, K-1]$.

3.3.2 Learning Interpolation Parameters

When a camera is designed it is already known that corresponding to which wavelengths it will capture intensity values and for those particular wavelengths the weights can be learned in advance. A set of K band multi-spectral images can be used to learn the weights of neighboring pixels in the interpolation process. Since for full multispectral image the Y vector and the matrix A are known therefore weight vectors X_i in (3.2) can be learned by solving the linear regression problem:

$$\arg\min_{x} \|AX - Y\|_2. \tag{3.3}$$

Equation (3.3) can be solved $K-1$ times for each pattern and there are K different patterns in the K band imagery. Therefore total $K \times (K-1)$ weight vectors can be learned from full multispectral training images.

The interpolation process considers not only the same band values but also other band values in an interpolation window therefore these weight vectors X considers the spectral correlation of different bands and spatial correlation of the neighboring pixels.

A set of training images can be chosen such that they have various low and high frequency features so that the learned weight vectors are robust to different kinds of images. The interpolation window size should be such that there are samples from all the bands captured in the raw image e.g. a window size of 3×3 is not sufficient for 6 band multispectral images since in this window there will be too few samples to do effective interpolation. On the other hand a large window size will show image averaging effect. Experiments with different window sizes reveled that a window size of 5×5 is sufficient for four, five, and six band multispectral images.

The weight vector learning is a one-time process that can be done offline. These weights can be used in (3.2) to do the interpolation on any raw image captured over that particular wavelengths for which weights have been learned. There will be different weight vectors for different number of bands and different spectral gaps. This means that three band multi-spectral images may have different weights than three band RGB images due to different spectral gaps.

3.4 Experiments and Results

All the experiments in Matlab software on a Linux machine with core i7 and 8 GB RAM. The multi-spectral dataset used for experiments was of cooled CCD camera Apogee Alta U260 described in [3] where each image have 31 bands from $400nm$ to $700nm$ wavelength range. Five images from multi-spectral dataset [3] were used to estimate interpolation parameters for three to six band multi-spectral images. Experiments were also performed on color images and five images from color image dataset [66] were used to estimate parameters for color images. The parameters were learned by solving Eq. (3.3) which is a least square regression problem and hence our proposed algorithm is named Least-Square based Multispectral Demosaicing (LSMD) algorithm. Five images each of spatial dimensional 512×512 were found sufficient for learning parameters because parameter values were converging to fixed values and more training images had no effect on reconstruction quality. Since parameter learning is a convex problem for which efficient solvers exists therefore learning time was just three seconds. Parameters were learned separately for different number of bands.

After the weight vector estimation, experiments were performed with 20 multi-spectral images having three to six consecutive bands as well as 20 RGB color images. These images were chosen to be different from those that were used in learning parameter vector representing interpolation weights. Wiener filtering was applied on the reconstructed multi-spectral images to remove some patterned noise in the case of six band images where there was high undersampling. Table 3.1 summarizes comparative results of LSMD and BTES for multi-spectral and color images. Table 3.2 shows comparative results on RGB images of Kodak dataset. First five images were utilized for learning the parameters whereas other 19 images were used for testing the proposed algorithm. The average PSNR values shows that proposed algorithm is comparable to the non-adaptive method.

Average Peak Signal to Noise Ratio (PSNR) values show that LSMD outperforms BTES in all five cases. The differences in average PSNR values are 2.29 dB for three bands, 2.96 for four bands, 1.87 dB for five bands, and 1.27 dB for six bands whereas for color images it is 4.82 dB.

Proposed UMSFA was evaluated using static coefficient (SC) and consistency coefficient (CC) defined in [17]. The SC measures the spatial uniformity of each spectral plane that forms the MSFA. It is defined as:

$$SC = 1 - \frac{1}{K}\sum_{j=1}^{K}\frac{1}{1+\mu_j}, \quad \mu_j = \frac{1}{N_j}\sum_{i=1}^{N_j}\|F_i\|_2,$$

$$F_i = \sum_{k\in N_i} F_{ki}, \quad F_{ki} = \frac{1}{(x_k - x_i)^2 + (y_k - y_i)^2},$$

where N_j is the total number of active pixels in the band j. The CC is a measure for

Table 3.1: *Comparison of PSNR(dB) values for BTES and proposed LMSD algorithms on multispectral and RGB image datasets.*

	BTES					LMSD				
Img	3band	4band	5band	6band	RGB	3band	4band	5band	6band	RGB
1	48.83	46.34	45.3	44.26	26.31	50.67	49.29	46.05	43.71	32.93
2	36.63	34.08	32.56	30.91	33.32	38.81	37.26	35.24	33.56	37.99
3	47.33	44.99	43.34	41.64	34.6	46.38	42.24	37.91	34.47	39.57
4	42.36	37.97	36	34.48	33.71	44.63	41.25	38.32	34.95	36.76
5	55.28	52.74	51.09	49.45	26.66	56.51	55.08	52.35	49.26	31.08
6	38.58	35.88	34.47	32.94	27.79	40.07	38.13	36.73	35.55	33.39
7	45.25	42.81	41.77	40.91	33.49	47.8	46.14	44.96	44.04	37.52
8	50.38	46.81	44.96	43.4	23.63	53.26	50.95	48.41	45.93	30.83
9	50.87	48.65	47.23	45.51	32.43	50.34	49.68	47.8	45.24	38.11
10	51.92	48.93	46.82	44.61	32.42	52.67	51.34	48.99	46.34	38.2
11	44.84	42.21	40.48	38.56	29.19	45.92	43.95	42.29	40.73	34.17
12	51.59	48.5	46.64	44.55	33.46	53.72	52.03	48.91	46.17	39.31
13	50.79	48.48	46.62	44.56	23.89	52.39	51.77	48.99	46.37	27.94
14	45.87	43.17	41.9	40.35	29.25	49.54	47.38	45.04	43.15	32.95
15	48.23	45.57	43.49	41.67	33.24	48.27	46.4	44.63	43.09	36.32
16	43.92	40.97	39.3	37.66	31.28	45.23	43.4	42.14	40.95	37.37
17	41.38	38.09	36.62	35.02	32.1	44.12	40.31	36.81	33.88	35.43
18	47.18	45.23	44.08	42.84	28.17	49.38	47.52	45.32	43.07	32.02
19	36.25	32.88	31.65	30.54	28.15	40.22	38.55	36.94	35.64	35.15
20	46.7	43.9	42.62	41.51	31.81	50.51	48.4	46.48	44.62	36.18
Avg.	**46.21**	**43.41**	**41.85**	**40.27**	**30.25**	**48.02**	**46.05**	**43.72**	**41.54**	**35.16**

spectral consistency and it is defined as:

$$H_j(consistency) = -\sum_{i=1}^{m} p(T_{ji} \log p(T_{ji}) \tag{3.4}$$

$$CC_j = \frac{1}{1 + pRPD_j + pSBD_j} \times \frac{1}{1 + H_j(consistency)} \tag{3.5}$$

$$CC = 1 - \frac{1}{K} \sum_{j=1}^{K} CC_j \tag{3.6}$$

Table 3.3 shows comparative values of SC and CC for the proposed UMSF and other existing MSF. In the column headings *B* stands for *bands* e.g. 3B means 3 bands. The values in Table 3.3 corresponding to MSF are taken from Miao paper [17]. For both SC and CC, the UMSF got zero value which indicates that the proposed filter array design is better as compare to existing MSF.

Figure 3.7 shows comparative reconstruction results on the Lighthouse image from the Kodak dataset. The zoomed portions clearly shows the effectiveness of proposed LMSD algorithm.

Table 3.2: *Comparative results on RGB images*

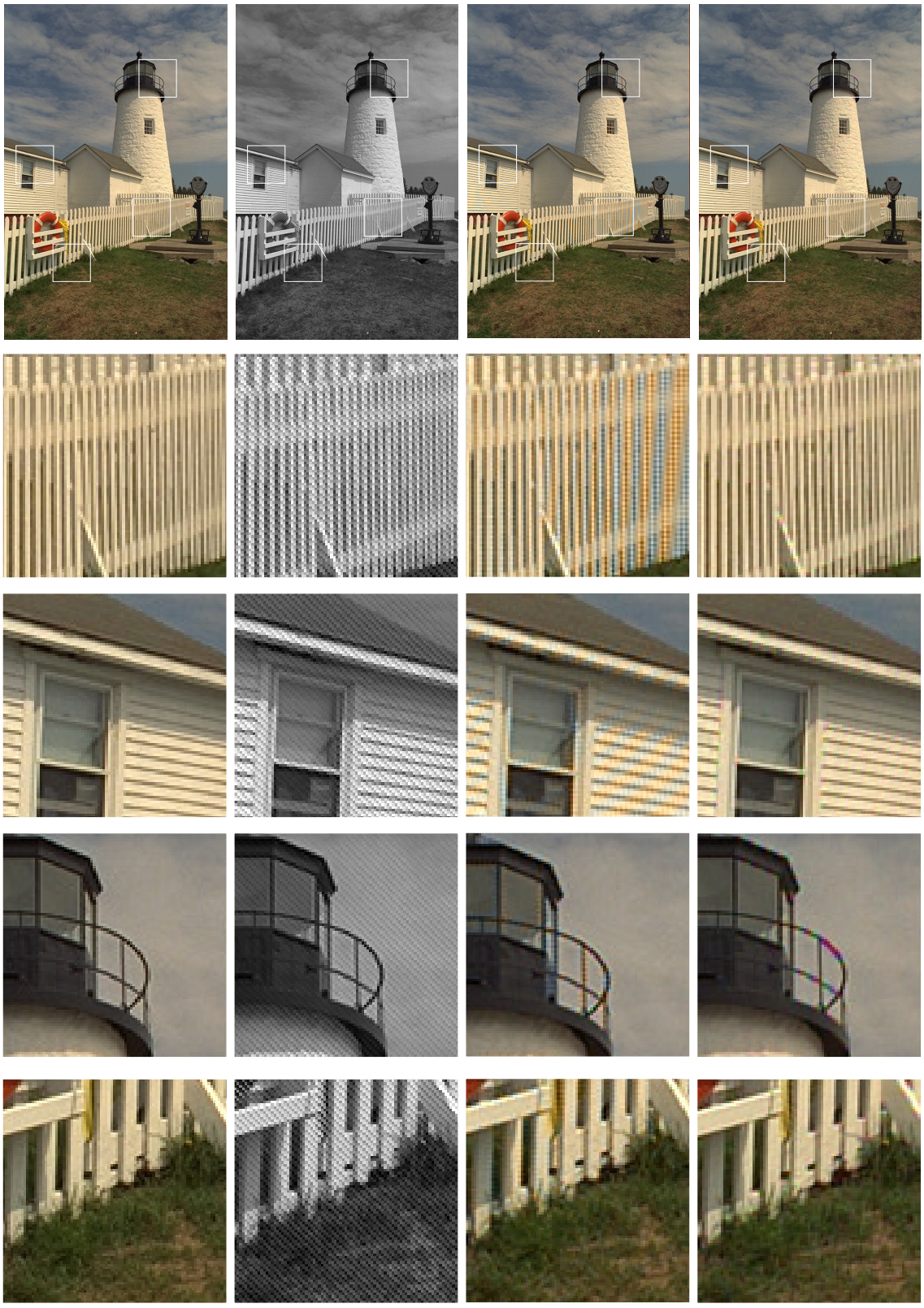
ImgNo.	Malver	LMSD
6	33.4	34.67
7	38.82	37.98
8	29.66	32.05
9	38.1	38.73
10	38.61	38.84
11	35.03	35.12
12	38.24	40.08
13	29.95	28.74
14	34.55	33.38
15	36.18	36.68
16	36.19	38.22
17	37.18	36.22
18	33.91	32.27
19	33.87	36.25
20	36.65	36.6
21	34.47	34.13
22	35.8	35.24
23	40.51	38.35
24	32.62	31.45
Avg:	35.57	35.68

Table 3.3: *Comparison of Multispectral Filter Arrays*

	Miao [17]				UMSF			
Metric	3B	4B	5B	6B	3B	4B	5B	6B
SC	0	0	0	0	0	0	0	0
CC	0.29	0	0.49	0.29	0	0	0	0

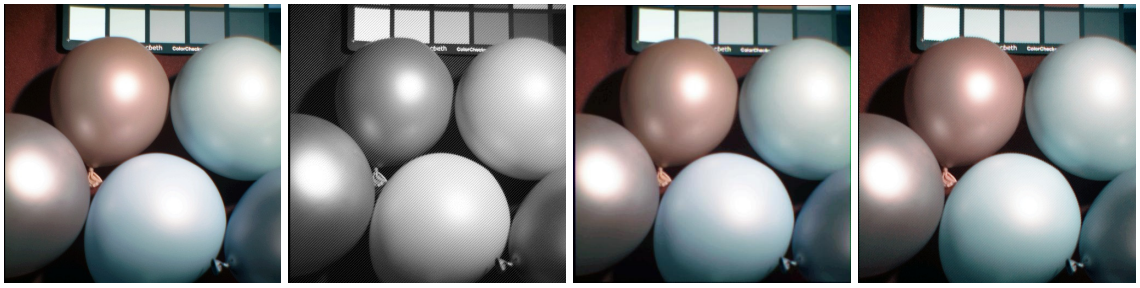
Figure 3.8 shows reconstruction results for a five band multispectral image using LMSD and BTES algorithms. Multispectral images are shown in false color composite after doing histogram equalization for better visual contrast. The reconstruction time of a five band multispectral image of size $512 \times 512 \times 5$ was around 80.91 seconds for BTES and 0.29 seconds for proposed LMSD technique on the same machine which indicates that proposed demosaicing technique is significantly faster. However these values are implementation dependent and shall vary on different machines.

The experiments were also conducted with different window-size of 5×5 and different number of bands. Table 3.5 shows the peak signal to noise ratio (PSNR) values for five multi-spectral images from the dataset [3]. Column heading B_jW_k represents that corresponding experiments have been done with j bands of the multi-spectral image in a window size of $k \times k$. For example third column heading B_4W_5 means that experiments have been done with 4 bands and 5×5 window size. Minimum average



(a) *Original* **(b)** *Raw* **(c)** *Malver [16]* **(d)** *LSMD*

Figure 3.7: *Demosaicing results on lighthouse image*



(a) *Original* **(b)** *Raw* **(c)** *BTES* **(d)** *LMSD*

Figure 3.8: *Reconstruction results for a portion of 5-band multispectral image with BTES and proposed LMSD algorithms. Images are shown in false color composite.*

b

Table 3.4: *Table representing band interpolation values for 4 band image with window size* 3×3

Pattern 1			Pattern 2			Pattern 3			Pattern 4		
Band 2			Band 1			Band 1			Band 1		
0.174	0.017	-0.057	-0.014	0.361	-0.038	0.269	0.067	-0.029	0.081	0.153	0.002
0.015	0.427	0.245	0.207	0.410	0.044	0.047	0.392	-0.025	0.098	-0.042	0.284
-0.083	0.326	-0.053	-0.004	0.054	-0.034	-0.024	-0.026	0.317	0.003	0.438	-0.025
Band 3			Band 3			Band 2			Band 2		
0.283	0.008	-0.117	-0.017	0.004	-0.046	-0.005	0.205	-0.059	0.031	0.264	0.004
0.007	0.401	0.212	0.008	0.690	0.184	0.120	0.634	0.004	0.236	0.018	0.128
-0.118	0.223	0.104	-0.056	0.212	0.021	-0.027	0.008	0.126	0.001	0.168	0.164
Band 4			Band 4			Band 4			Band 3		
0.060	0.498	0.013	0.482	-0.028	0.035	-0.055	0.021	0.034	-0.136	0.422	0.047
0.465	-0.080	0.035	-0.017	0.020	-0.008	0.033	-0.090	0.466	0.381	-0.080	0.065
0.012	0.038	-0.051	0.037	-0.015	0.479	0.034	0.499	0.047	0.042	0.101	0.165

Table 3.5: *PSNR values for reconstructions of multi-spectral images from dataset [3] using different number of bands and window sizes*

	Peak Signal to Noise Ratio (dB)					
Images	B_5W_5	B_5W_3	B_4W_5	B_4W_3	B_3W_3	B_2W_3
Balls	41.66	39.77	42.64	40.50	37.47	46.79
Paints	32.64	30.09	33.96	30.83	33.93	36.28
Lemons	36.92	34.88	37.89	34.50	43.53	41.28
Thread	37.46	35.73	38.78	37.13	40.92	45.09
Clay	40.91	38.29	42.63	38.88	42.42	44.52
Average	37.92	35.75	39.18	36.37	39.65	42.79

PSNR values in Table 3.5 is $35.75dB$ for the case of five band images which is also a good result consider low sampling rate for five band imagery.

Chapter 4

Hyperspectral Denoising

This chapter introduces hyperspectral denoising problem as an inverse problem. The section 4.1 considers impulse noise reduction problem followed by section 4.2 that generalizes it to mixed noise. A general noise model has been considered that accounts for Gaussian noise and sparse noise explicitly. Both the problems has been formulated as synthesis prior as well as total variation minimization problems.

4.1 Impulse Denoising

Generally a hyperspectral image is corrupted by several kinds of noise such as Gaussian noise, line strips, impulse noise, shot noise, etc.. Gaussian denoising is a widely addressed problem [67, 68] but, to the best of our knowledge, there are no prior studies on impulse denoising from hyperspectral images. Therefore, this problem of reducing impulse noise from hyperspectral images has been addressed in this section. There are two broad approaches to solve this problem. The first approach is to use median filtering and its variants such as [69–72]. The other approach is to exploit the sparsity of the image in some transform domain and formulate denoising as an optimization problem.

Let x be original gray-scale image and y be noise corrupted image then impulse denoising problem can be expressed as ℓ_1-norm minimizing sparse recovery problem:

$$\min_z \ \|y - D^T z\|_1 + \lambda \|z\|_1, \tag{4.1}$$

where z is the sparse representation of image x in orthogonal sparsifying transform domain D such that $z = Dx$. This formulation is called the synthesis prior (SP) problem; here the sparse transform coefficients are recovered from which the image is synthesized. Here, in Eq. (4.1) ℓ_1-norm of data fidelity term is minimized owing to the fact that impulse noise is sparse. Algorithms such as [73–75] have been proposed in literature to solve this problem.

Another widely used approach for image denoising is total variation (TV) regularization [43] that can be represented as follows:

$$\|y - x\|_1 + \|D_h x\|_1 + \|D_v x\|_1, \tag{4.2}$$

where D_h, D_v are horizontal and vertical finite difference operators. Several algorithms such as [44, 76–78] have been proposed in literature to solve TV based impulse denoising problem. This formulation is referred to as analysis prior (AP) problem.

All these studies are restricted to single channel denoising; for our problem these methods have to be applied separately on each band of a hyperspectral image which does not account for inter-band correlation present in hyperspectral images. Prior works in Gaussian noise removal from multi-band images have shown that exploiting the spectral correlation (along with spatial intra-band correlation) improves denoising performance. For example, there are TV based algorithms for color image denoising such as ColorTV[79], and MSBEL[80]. These algorithms are mainly designed for denoising of three channel color images. Algorithms have also been proposed for hyperspectral denoising based on spatial-spectral adaptive TV [67], spatial-spectral fusion technique [68] and spectral statistics [81] based technique but these algorithms are designed for Gaussian noise reduction.

It has been shown previously in [82, 83] that exploiting spatio-spectral correlation leads to better reconstruction in compressive hyperspectral imaging. Following these studies, we propose to exploit the inter-band spectral correlation and intra-band spatial correlation for denoising hyperspectral images corrupted by impulse noise. As the noise is sparse, we minimize ℓ_1-norm of data fidelity term. The spatio-spectral correlation leads to a sparse representation of the hyperspectral datacube in certain transform domains; the transform domain sparsity leads to ℓ_1-norm regularization of the ℓ_1-norm data fidelity. To the best of our knowledge there is no published work on denoising impulse noise corrupted hyperspectral images which utilizes both spatial redundancy and spectral correlation.

In this section, we have explored two different models for impulse denoising namely synthesis prior and analysis prior and experimentally found that synthesis prior model yields better results than analysis prior model. It is difficult to tell analytically why one method is better than the other. There are prior works related to Magnetic Resonance Imaging [84, 85] where it was found that analysis prior yields better results.

4.1.1 Problem Formulation

A hyperspectral data cube of dimension $m \times n \times d$ having d spectral bands in it can be represented as $X = \begin{bmatrix} x_1 & x_2 & \dots & x_d \end{bmatrix}$ where each $x_i \in \mathbb{R}^{mn \times 1}$ is a spectral band obtained by vertical concatenation. Define $y = vec(Y)$ as vector representation of any 2D matrix obtained by vertical stacking of columns of matrix Y, and $Y = mat(y)$ as its reverse operation. We use small letters for vectors and capital letters for matrices. Using these notations, image acquisition model can be expressed as:

$$Y = X + N,$$

where X is original image, Y is noise corrupted image, and N is the noise.

The columns of X are images and hence have a sparse representation in transform domains like wavelet or finite difference. Since the images are spectrally correlated,

the variation along the rows of X can be assumed to be smooth leading to a sparse representation in Fourier, wavelet or finite differencing. The spatio-spectral correlation can be jointly exploited by representing the datacube as a sparse signal: $Z = D_1 X D_2$. Here D_1, D_2 are sparsifying transforms along spatial and spectral dimensions respectively and Z denotes the sparse transform coefficients. Using this sparse signal representation, denoising problem can be framed as:

$$\min_Z \|Y - D_1^T Z D_2^T\|_1 + \lambda \|Z\|_1, \tag{4.3}$$

where λ is a regularization parameter. This is the general synthesis prior (GSP) formulation. For framing the analysis prior problem we model the images to be piecewise smooth and employ finite differencing to sparsify along the spatial dimension. Along the spectral dimension we employ an orthogonal sparsifying transform D. Using this, we get following general analysis prior (GAP) formulation:

$$\min_X \|Y - X\|_1 + \lambda \|D_h X D\|_1 + \lambda \|D_v X D\|_1, \tag{4.4}$$

where D_h, D_v are horizontal and vertical finite difference operators and λ is the regularization term.

For grayscale images having only one band the sparsifying transform along spectral dimension will not have any effect. In that case we get single band synthesis prior formulation of Eq. (4.1) as special case of (4.3) and TV denoising formulation in Eq. (4.2) as special case of (4.4). We are not aware of any efficient algorithm to solve such large scale problems. Therefore in the next section we derive algorithms for solving them.

4.1.2 Proposed Algorithms

This section briefly describes how to solve (4.3) and (4.4) using split-Bregman approach.

General Synthesis Prior (GSP) Algorithm

We repeat GSP problem for the sake of convenience.

$$\min_Z \|Y - D_1^T Z D_2^T\|_1 + \lambda \|Z\|_1, \tag{4.5}$$

Since the variable Z is not separable therefore we substitute: $P = Y - D_1^T Z D_2^T$ and $Q = Z$ then (4.5) can be rewritten as constrained optimization problem:

$$\begin{aligned} \underset{Z}{\text{minimize}} \quad & \|P\|_1 + \lambda \|Q\|_1 \\ \text{subject to} \quad & P = Y - D_1^T Z D_2^T \\ & Q = Z \end{aligned}$$

The above constrained optimization problem can be expressed as unconstrained optimization problem using weak penalty function as follows:

$$\underset{Z,P,Q}{\text{minimize}} \ \|P\|_1 + \lambda\|Q\|_1 + \mu_1\|P - Y + D_1^T Z D_2^T\|_F^2 + \mu_2\|Q - Z\|_F^2$$

where μ_1 and μ_2 are the regularization parameters and $\|\cdot\|_F$ represents Frobenius norm of a matrix. Since there are multiple regularization terms therefore split-Bregman [44] approach can be applied to solve this problem. Thus the above unconstrained problem can be expressed as:

$$\underset{Z,P,Q}{\text{minimize}} \ \|P\|_1 + \lambda\|Q\|_1 + \mu_1\|P - Y + D_1^T Z D_2^T - B_1^k\|_F^2 + \mu_2\|Q - Z - B_2^k\|_F^2$$

where Bregman variables B_1 and B_2 are updated iteratively as follows:

$$\begin{aligned} B_1^{k+1} &= B_1^k + Y - P - D_1^T Z D_2^T \\ B_2^{k+1} &= B_2^k + Z - Q \end{aligned}$$

The above problem has three separable variables (Z, P, Q) and therefore we can split the problem in three simple subproblems. Let $A \otimes B$ denote the Kronecker product between matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$ defined as:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B \dots a_{1n}B \\ a_{21}B & a_{22}B \dots a_{2n}B \\ \vdots & \vdots \quad \ddots \quad \vdots \\ a_{m1}B & a_{m2}B \dots a_{mn}B \end{bmatrix}_{mp\times nq}$$

Let $y_1 = vec(P - Y - B_1^k)$, $y_2 = vec(Q - B_2^k)$, $D = D_2 \otimes D_1^T$ then sub-problems can be expressed as follows:

$$\begin{aligned} \text{P1}: \quad & \min_z \mu_1\|y_1 + Dz\|_2^2 + \mu_2\|y_2 - z\|_2^2 \\ \text{P2}: \quad & \min_p \|p\|_1 + \mu_1\|p - y - b_1^k + Dz\|_2^2 \\ \text{P3}: \quad & \min_q \|q\|_1 + \mu_2\|q - z - b_2^k\|_2^2. \end{aligned}$$

Here we used the identity that $Y = AXB$ can be written as $y = (B^T \otimes A)x$. Above problem P1 is differentiable and convex whose solution we can obtain by solving

$$(\mu_1 D^T D + \mu_2 I)z = \mu_2 y_2 - \mu_1 D^T y_1,$$

iteratively using least square solvers. In case D_1 and D_2 are orthogonal transform we can simplify $D^T D$ as follows:

$$D^T D = (D_2^T \otimes D_1)(D_2 \otimes D_1^T) = D_2^T D_2 \otimes D_1 D_1^T = I,$$

which gives a closed form solution for problem P1 as:

$$z = \frac{\mu_2}{\mu_1 + \mu_2} y_2 - \frac{\mu_1}{\mu_1 + \mu_2} D^T y_1, \tag{4.6}$$

Problems P2 and P3 are ℓ_1-norm minimization problems in the following form:

$$\arg\min_x \|y - x\|_2^2 + \lambda\|x\|_1, \tag{4.7}$$

which can be solved by using soft-thresholding SoftTh(y, λ) operation: $\hat{x} = \text{sign}(y) \times \max\left\{0, |y| - \frac{\lambda}{2}\right\}$. Algorithm 5 summaries the steps of the proposed GSP algorithm.

Algorithm 5 General Synthesis Prior (GSP) Algorithm

input: $D_1, D_2, D = D_2 \otimes D_1^T$, Y, λ, μ_1, μ_2, MaxIter.
output: $\hat{X}$, denoised image.
for $k = 1$ to MaxIter **do**
    $Z^{k+1} = mat(z^k)$ from (4.6)
    $P^{k+1} = \text{SoftTh}\left(Y + B_1^k - \text{mat}(Dz), \frac{1}{\mu_1}\right)$
    $Q^{k+1} = \text{SoftTh}\left(Z + B_2^k, \frac{\lambda}{\mu_2}\right)$
    $B_1^{k+1} = B_1^k - P^{k+1} + Y - D_1 Z^{k+1} D_2$
    $B_2^{k+1} = B_2^k + Z^{k+1} - Q^{k+1}$
end for
$\hat{X} = D_1^T Z^{k+1} D_2^T$

General Analysis Prior (GAP) Algorithm

We rewrite GAP formulation from previous section:

$$\arg\min_X \|Y - X\|_1 + \lambda\|D_h X D\|_1 + \lambda\|D_v X D\|_1 \tag{4.8}$$

As before recast (4.8) as follows:

$$\begin{aligned} \underset{P,Q,R}{\text{minimize}} \quad & \|P\|_1 + \lambda\|Q\|_1 + \lambda\|R\|_1 \\ \text{subject to} \quad & P = Y - X \\ & Q = D_h X D \\ & R = D_v X D \end{aligned}$$

which, using split-Bregman approach can be expressed as:

$$\begin{aligned} \underset{X,P,Q,R,S}{\text{minimize}} \; \|P\|_1 + \lambda\|Q\|_1 + \lambda\|R\|_1 + \mu_1\|P - Y + X - B_1^k\|_F^2 + \mu_2\|Q - D_h X D - B_2^k\|_F^2 \\ + \mu_2\|R - D_v X D - B_3^k\|_F^2 \end{aligned}$$

where Bregman variables are updated as :

$$\begin{aligned} B_1^{k+1} &= B_1^k + Y - X - P \\ B_2^{k+1} &= B_2^k + D_h X D - Q \\ B_3^{k+1} &= B_3^k + D_v X D - R \end{aligned}$$

Let $y_1 = vec(P - Y - B_1^k)$, $y_2 = vec(Q - B_2^k)$, $y_3 = vec(R - B_3^k)$, $W_h = D^T \otimes D_h$, $W_v = D^T \otimes D_v$, then above problem can be split into four separable problems as follows:

$$\text{P4} : \min_x \mu_1 \|y_1 + x\|_2^2 + \mu_2 \|y_2 - W_h x\|_2^2 + \mu_2 \|y_3 - W_v x\|_2^2$$

$$\text{P5} : \min_P \|P\|_1 + \mu_1 \|Y - X - P + B_1^k\|_F^2$$

$$\text{P6} : \min_Q \|Q\|_1 + \mu_2 \|Q - D_h X D - B_2^k\|_F^2$$

$$\text{P7} : \min_R \|R\|_1 + \mu_2 \|R - D_v X D - B_3^k\|_F^2$$

Problem P4 is differentiable and after simplification we get :

$$\left[\mu_1 I + \mu_2 W\right] x = \mu_2 (W_h^T y_2 + W_v^T y_3) - \mu_1 y_1 \tag{4.9}$$

where $W = (W_h^T W_h + W_v^T W_v)$, though finite difference operator is not orthogonal but corresponding matrices in (4.9) is very sparse because

$$W_h^T W_h = (D \otimes D_h^T)(D^T \otimes D_h) = I_d \otimes D_h^T D_h$$

and similarly $W_v^T W_v = I_d \otimes D_v^T D_v$ makes the system of equations in (4.9) large and sparse system therefore few iterations of iterative solver such as LSQR[86] will suffice to approximate x. Problems (P5), (P6) and (P7) can be solved using soft-thresholding as described in previously. Algorithm 6 summarizes the GAP algorithm.

Algorithm 6 General Analysis Prior (GAP) Algorithm

1: input: $W_h, W_v, Y, \lambda, \mu_1, \mu_2$, MaxIter.
2: output: $\hat{X}$: denoised image.
3: **for** $k = 1$ to MaxIter **do**
4: $X^{k+1} = mat(\hat{x})$, solution from equation (4.9)
5: $P^{k+1} = \text{SoftTh}\left(Y - X + B_1^k, \frac{1}{\mu_1}\right)$
6: $Q^{k+1} = \text{SoftTh}\left(D_h X^{k+1} D + B_2^k, \frac{\lambda}{\mu_2}\right)$
7: $R^{k+1} = \text{SoftTh}\left(D_v X^{k+1} D - B_3^k, \frac{\lambda}{\mu_2}\right)$
8: $B_1^{k+1} = B_1^k + Y - P^{k+1} - X^{k+1}$
9: $B_2^{k+1} = B_2^k + D_h X^{k+1} D - Q^{k+1}$
10: $B_3^{k+1} = B_3^k + D_v X^{k+1} D - R^{k+1}$
11: **end for**
12: $\hat{X} = X^{k+1}$.

4.1.3 Experiments and Results

Two hyperspectral images were used for performing experiments. First image was of Reno city, NV, USA available from [87]. This image is from High Resolution Imager (HRI) sensor having $2m$ spatial resolution and 5 nm band spacing covering spectral range of 395 to 2450 nm. Second image was of Washington DC mall available

from [88]. This image is of Hyperspectral Digital Imagery Collection Experiment (HYDICE) sensor having $1m$ spatial resolution and 10-nm band spacing covering spectral range of 400 to 2500 nm. We used patchs of size $160 \times 160 \times 319$ from *Reno* image and $160 \times 160 \times 191$ from *WDC* image for experiments.

For GSP algorithm, orthogonal transform D_1 was selected as 2D Daubechies wavelet with filter length eight for spatial dimension. Along spectral dimension 1D Fourier transform was utilized as sparsifying basis in both GSP and GAP algorithms. We have empirically chosen the parameters required by our algorithm to yield good results for realistic noisy scenarios. Empirically best parameters for GSP algorithm were found to be $(\lambda, \mu_1, \mu_2, maxiter)$=$(2, 5, 5, 40)$ while for GAP best parameters were found to be $(\lambda, \mu_1, \mu_2, maxiter)$=$(2, 2, 2, 50)$. These parameters were tuned manually using cross validation. The tuning was performed so that the algorithms yield good results for a wide variety of noise levels. Our proposed technique is robust to large range of noise corruption (upto 50%). A window size of 5×5 was used with median filter. Since IRN algorithm converges in five iterations as mentioned in [89] therefore five iterations were used in all experiments.

When input image is of single band then GAP and GSP algorithms behave as AP and SP algorithms respectively. Experiments were performed with both salt and pepper noise as well as random valued impulse noise. Original hyperspectral images were corrupted by 10% to 70% noise to get noisy images which were then denoised using different algorithms. We selected median filtering based techniques as well as optimization based techniques to compare the proposed algorithms. In particular, median filter (MF), progressive switching median filter (PSMF) [90], and iterative re-weighted norm (IRN) [89] minimization algorithms were selected for comparison. We have also compared with SP and AP problems which are solved as special case of GSP and GAP algorithms. Since all these algorithms were developed for single band therefore each band was separately denoised by applying the algorithms iteratively.

Tables 4.1 shows experimental results on Reno image for both salt and pepper noise and random valued impulse noise while Table 4.2 shows experimental results for WDC image. PSNR(dB) was calculated for 10% to 70% noise levels on both Reno and WDC hyperspectral images. Column titled *Noisy* shows PSNR value between noisy and original image. Maximum PSNR value is bold faced. We notice that for WDC image at 10% noise level, PSMF algorithm gives best results for salt and pepper noise but PSNR decreases for all other noise levels. There is degradation in PSNR values for large noise levels (for 60% and more) for Reno image however this 60% and more impulse noise is an unlikely scenario to be encountered in practice. Notice that use of sparsifying basis for spectral dimension has increased PSNR values for both GSP and GAP algorithms compared to their sequential counterparts SP and AP algorithms.

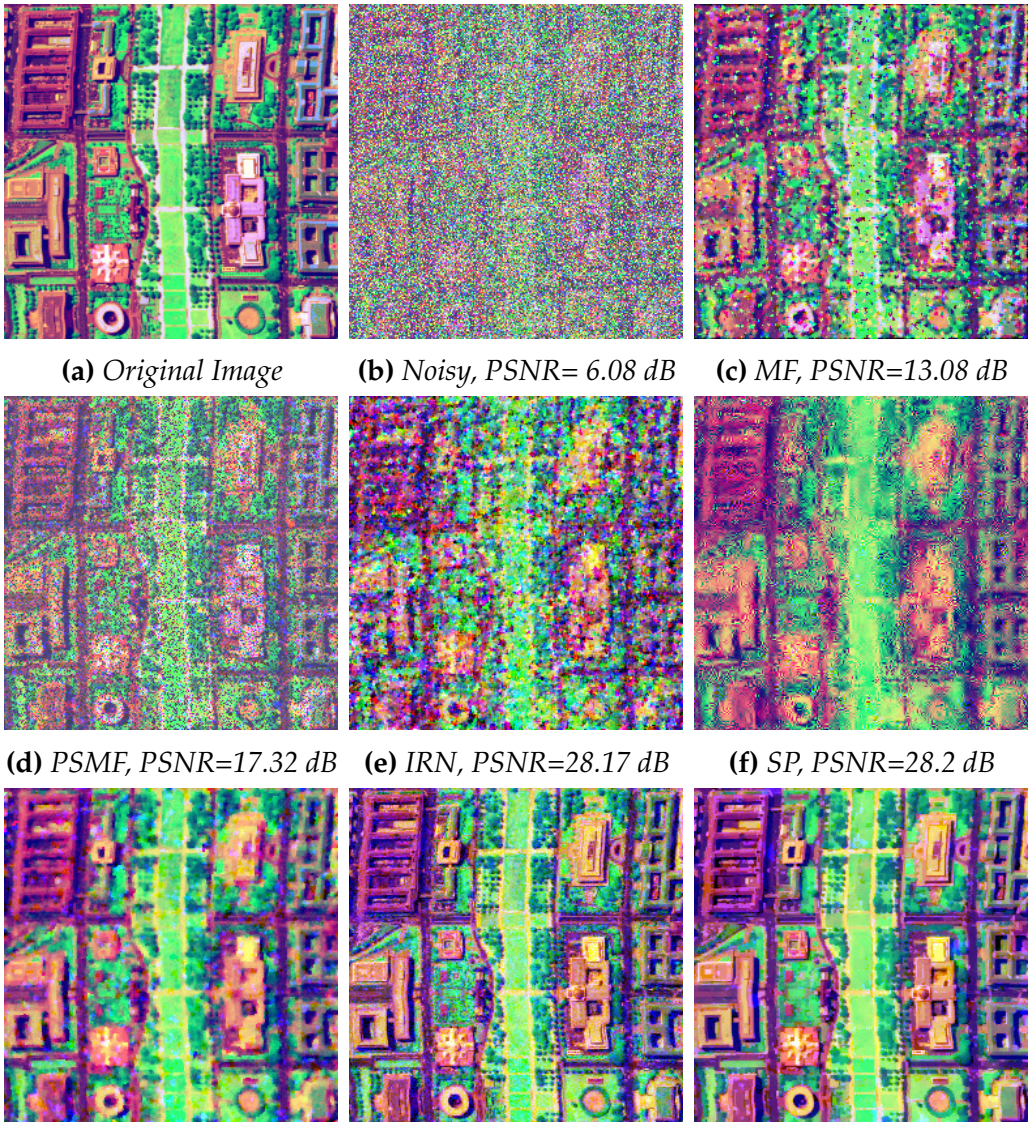
Figure 4.1 shows qualitative visual comparison of reconstruction quality for 50% salt and pepper noise on $256 \times 256 \times 191$ patch of WDC image. Three bands(20, 80, and 170) are shown in false color with histogram equalization for better visual quality only. Median filtering and PSMF are not able to remove high noise in Fig. 4.1(c) and Fig. 4.1(d). Median filtering is a very fast algorithm and gives comparable results on low noise levels. However, at 50% noise level, the median value in a 3×3 block may

Table 4.1: *PSNR (dB) values for Reno Image*

Salt and Pepper Noise								
Noise%	Noisy	MF	PSMF	IRN	SP	AP	GSP	GAP
10	13.82	34.28	38.18	36.26	31.95	33.73	**42.34**	38.54
20	11.41	40.07	31.73	35.43	31.42	33.29	**41.07**	37.69
30	9.21	39.05	26.45	34.15	30.93	32.82	**39.82**	36.84
40	8.19	33.40	22.00	31.86	30.62	32.34	**38.46**	35.97
50	7.94	24.80	19.58	32.01	29.18	31.79	**36.73**	34.98
60	7.03	17.86	16.11	27.50	28.29	31.16	**35.12**	33.92
70	6.16	12.48	11.37	21.48	27.42	30.24	**33.41**	32.54
Random Valued Impulse Noise								
10	16.98	34.51	33.85	35.96	32.28	34.02	**42.82**	39.06
20	14.37	33.86	28.63	33.61	32.03	33.74	**41.84**	38.45
30	13.47	30.70	22.97	27.47	31.52	33.16	**40.10**	37.46
40	12.90	23.14	18.56	18.68	30.39	31.57	**37.03**	35.42
50	11.28	16.57	14.93	12.75	26.56	23.91	**29.94**	28.37
60	10.73	12.74	12.04	9.48	**15.44**	10.71	11.18	11.23
70	9.74	9.38	9.91	6.64	**12.06**	5.85	9.80	7.21

Table 4.2: *PSNR(dB) values fro WDC image*

Salt and Pepper Noise								
Noise%	Noisy	MF	PSMF	IRN	SP	AP	GSP	GAP
10	14.61	38.30	**48.03**	40.32	36.68	38.21	44.69	41.24
20	11.60	37.95	36.57	39.32	36.05	37.80	**43.68**	40.50
30	9.84	38.78	28.05	38.15	35.78	37.38	**42.62**	39.74
40	8.59	35.02	22.12	36.56	36.32	36.91	**41.21**	38.97
50	7.62	27.42	16.62	34.71	34.52	36.39	**40.12**	38.08
60	6.83	20.45	13.02	31.24	31.25	35.83	**38.49**	36.97
70	6.16	15.13	10.62	26.83	27.10	35.19	**35.69**	35.58
Random Valued Impulse Noise								
10	17.91	38.24	31.51	40.11	36.61	38.17	**44.56**	41.16
20	14.89	37.84	29.37	39.01	35.90	37.69	**43.46**	40.35
30	13.14	37.33	27.52	37.90	35.17	37.20	**42.14**	39.48
40	11.89	36.56	25.90	36.59	34.28	36.62	**40.57**	38.48
50	10.92	35.00	24.23	34.74	33.84	35.98	**38.78**	37.26
60	10.13	32.01	22.48	32.37	32.85	35.16	**36.77**	35.82
70	9.46	27.88	21.10	29.44	31.08	34.00	33.95	**34.24**

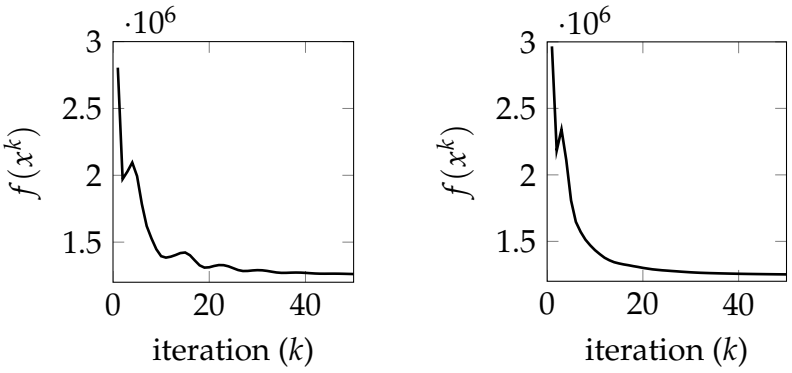


(a) *Original Image* **(b)** *Noisy, PSNR= 6.08 dB* **(c)** *MF, PSNR=13.08 dB*

(d) *PSMF, PSNR=17.32 dB* **(e)** *IRN, PSNR=28.17 dB* **(f)** *SP, PSNR=28.2 dB*

(g) *AP, PSNR=35 dB* **(h)** *GSP, PSNR=41.91 dB* **(i)** *GAP, PSNR=42.17 dB*

Figure 4.1: *Comparison of visual quality of denoising results for WDC image at 50% salt and pepper noise. Proposed GSP and GAP algorithms have better quality than other algorithms.*



(a) *GSP convergence* **(b)** *GAP convergence*

Figure 4.2: *Convergence graphs for GSP and GAP algorithm for 50% salt and pepper noise on WDC image.*

be the noisy value itself, therefore, at some pixels noisy salt and paper noise is visible. IRN reduces noise but image details are not well preserved as shown in Fig. 4.1(e). SP algorithm also reduces noise but over-smooths the image and AP algorithm introduces some blurry effect in the image. Both GSP and GAP were able to reduce noise as well as preserve edge information compared to other algorithms. We notice that PSNR need not always be consistent with visual quality. Visual quality is a subjective evaluation and depends on the application-whether one wants denoised but smooth images or sharp but noisy ones. PSNR value is high if the image is non-noisy even if it is overtly smoothed (as in our case with synthesis prior Fig. 4.1(f)); the PSNR values are low for sharp but noise images, e.g. MF (Fig. 4.1(c)) and IRN (Fig. 4.1(e)).

Figure 4.2 shows the convergence plot for our GSP and GAP algorithms. The objective function does not decrease monotonically. But this is expected as our algorithm is based upon the Split Bregman approach. After 30 iterations objective functions of both the algorithms do not decrease significantly hence we can choose number of iterations to be 30 for experimental purpose.

4.2 Mixed Denoising

This section extends the hyperspectral impulse denoising to mixed denoising.

4.2.1 Problem Formulation

Define $y = \text{vec}(Y)$ as vector representation of any 2D matrix obtained by vertical stacking of columns of matrix Y and $X = \text{mat}(x)$ as its reverse operation. We use small letters for vectors and capital letters for matrices. A hyperspectral data cube of dimension $m \times n \times d$ having d spectral bands can be represented as $X = \left[x_1 x_2 \dots x_d\right]$ where each $x_i \in \mathbb{R}^{mn \times 1}$ $(mn = m \times n)$ is a spectral band obtained by vertical concate-

nation. Using these notations, image acquisition model in the presence of Gaussian and sparse noise can be expressed as:

$$Y = X + S + G,$$

where $X \in \mathbb{R}^{mn \times d}$ is original image, Y is noise corrupted image, S is sparse noise, and G is Gaussian noise. This model was introduced in [41] for low-rank matrix recovery problem and later utilized for hyperspectral denoising in LRMR algorithm [20].

The joint spatio-spectral correlation can be exploited by using synthesis prior approach or analysis prior approach. Synthesis prior approach seeks for sparse coefficients in a sparsifying transform domain. Let Z be the sparse representation of image X such that $Z = D_1 X D_2$. Here D_1 is a 2D sparsifying transform applied along spatial dimension, and D_2 is 1D sparsifying transform applied on spectral dimension. Figure 4.3(a) shows the comparison of band-by-band DCT model with the proposed 3D-DCT model. It can be observed that 3D-DCT gives sparser representation of the image. Using these notations, general synthesis prior (GSP) formulation for mixed denoising problem can be expressed

$$\min_{Z,S} \|Z\|_1 + \|S\|_1 + \lambda \|Y - D_1^T Z D_2^T - S\|_F^2, \tag{4.10}$$

a solution to this problem can be derived as for general synthesis prior problem in previous section. The solution is also discussed in [91].

The pixels in most natural images are spatially correlated to its neighboring pixel values. The neighboring bands also exhibit high spectral correlation. This prior knowledge can be exploited by representing images as piece-wise smooth functions and can be modeled using total-variation regularization. The total variation for a gray scale image x can be expressed as

$$\mathrm{TV}(x) = \|D_h x\|_1 + \|D_v x\|_1$$

where D_h and D_v are horizontal and vertical 2D-finite differencing operators. This gray scale total variation model has been extended as Color Total Variation (CTV) model in [79] for color images. This CTV model can be extended for a hyperspectral image X with d bands as follows:

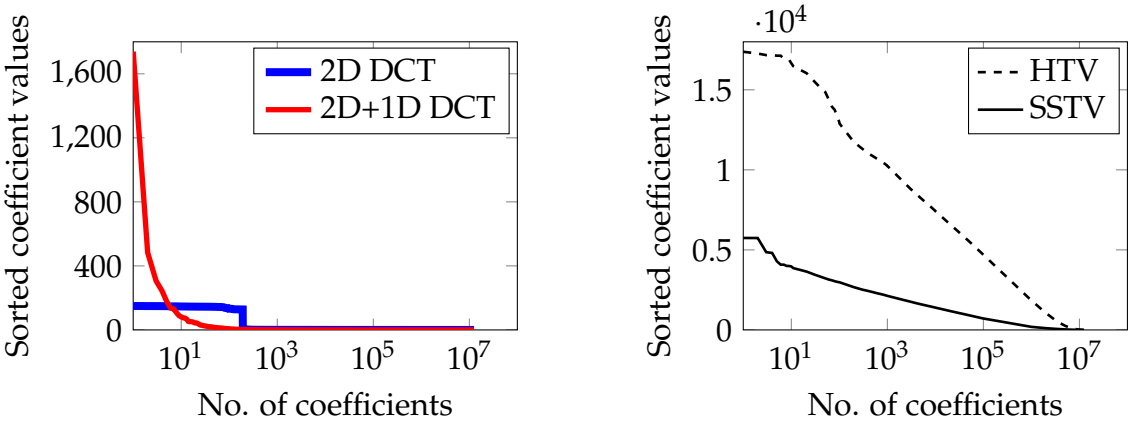
$$\mathrm{HTV}(X) = \sum_{i=1}^{d} TV(x_i). \tag{4.11}$$

The denoising problem using hyperspectral total variation (HTV) model can be expressed as:

$$\min_{X,S} \ \|Y - X - S\|_F^2 + \lambda \|S\|_1 + \mu \ \mathrm{HTV}(X) \tag{4.12}$$

This HTV model based hyperspectral denoising problem accounts for spatial correlation but does not consider spectral correlation present in neighboring bands of a hyperspectral image. We propose a spatio-spectral total variation (SSTV) model as follows:

$$\mathrm{SSTV}(X) = \|D_h X D\|_1 + \|D_v X D\|_1 \tag{4.13}$$



(a) *Comparison of sparsity by 3D-DCT and 2D-DCT.*

(b) *Comparison of HTV model and proposed SSTV model.*

Figure 4.3: *Sparsity comparison between DCT and total variation.*

where $D \in \mathbb{R}^{b \times b}$ is one-dimensional finite differencing operator applied on spectral signature of each pixel such that discrete gradient at i^{th} pixel is $\alpha_i = (D^T z)_i = z_{i+1} - z_i$, where z represents spectral signature of a pixel and $\alpha \in \mathbb{R}^{b \times 1}$ with $\alpha_b = 0$ at boundary. This model simultaneously explores correlation in both spatial and spectral dimension.

Figure 4.3(b) show the comparison of SSTV model with HTV model when applied on WDC hyperspectral image (data described in section 4). We observe that sorted discrete gradient coefficients for SSTV model give sparser representation compared to HTV model because SSTV model also exploits spectral correlation. The denoising problem using this SSTV model can be expressed as :

$$\min_{X,S} \ \|Y - X - S\|_F^2 + \lambda \|S\|_1 + \mu \ \mathrm{SSTV}(X) \tag{4.14}$$

where λ, μ are regularization parameters. Here ℓ_1-norm of S is minimized since S is representing sparse noise. Here $G = Y - X - G$ is representing Gaussian noise and therefore we are minimizing Frobenius norm of $Y - X - S$ to reduce variance of Gaussian noise.

Other studies in hyperspectral denoising has also used total variation based formulation such as [67]. However our study is markedly different from the aforesaid work in terms of noise model as well as total variation model.

Although researchers have proposed generic noise removal techniques [20, 23]; such an explicit formulation for jointly denoising Gaussian and sparse noise using spatio-spectral total variation has not been attempted before. We are not aware of any efficient algorithm that can solve the aforesaid problem. Therefore in the next section we describe how to solve this problem using split-Bregman [44] based approach.

4.2.2 Proposed Algorithm

General Synthesis Prior

This section discusses how to solve (4.10) using split-Bregman approach. This approach had been very successful in solving multiple penalty optimization problems [92, 93]. We repeat synthesis prior problem for the sake of convenience.

$$\min_{Z,S} \|Z\|_1 + \|S\|_1 + \lambda \|Y - D_1^T Z D_2^T - S\|_F^2,$$

Since the variable Z is not separable, we substitute $P = Z$ and $Q = S$ such that above problem can be rewritten as an unconstrained optimization problem :

$$\min_{Z,P,Q,S} \|P\|_1 + \|Q\|_1 + \frac{\lambda}{2}\|Y - D_1^T Z D_2^T - S\|_F^2 + \frac{\mu_1}{2}\|P - Z - B_1\|_F^2 + \frac{\mu_2}{2}\|Q - S - B_2\|_F^2$$

where λ, μ_1, μ_2 are regularization parameters and B_1, B_2 are Bregman variables. Since there are multiple regularization terms, we intent to follow the split-Bregman [44] approach which can be applied to solve this problem. Thus the above unconstrained problem can be split into following sub-problems as:

$$\begin{aligned}
\text{P1}: \quad & \min_{Z} \frac{\lambda}{2}\|Y - D_1^T Z D_2^T - S\|_F^2 + \frac{\mu_1}{2}\|P - Z - B_1\|_F^2 \\
\text{P2}: \quad & \min_{P} \|P\|_1 + \frac{\mu_1}{2}\|P - Z - B_1\|_F^2 \\
\text{P3}: \quad & \min_{Q} \|Q\|_1 + \frac{\mu_2}{2}\|Q - S - B_2\|_F^2 \\
\text{P4}: \quad & \min_{S} \frac{\lambda}{2}\|Y - D_1^T Z D_2^T - S\|_F^2 + \frac{\mu_2}{2}\|Q - S - B_2\|_F^2
\end{aligned}$$

Here subproblems P1 and P4 are least square problems with analytic solutions:

$$\begin{aligned}
Z &= \frac{1}{\lambda + \mu_1}\left(D^T(Y - S) + \mu_1(P - B_1)\right) \\
S &= \frac{1}{\lambda + \mu_2}\left(\lambda(Y - DZ) + \mu_2(Q - B_2)\right)
\end{aligned}$$

Subproblems P2 and P3 are of the form :

$$\arg\min_{x} \|y - x\|_2^2 + \lambda\|x\|_1$$

which can be solved by using soft-thresholding [94] $\text{SoftTh}(y, \lambda)$ operation:

$$\hat{x} = \text{sign}(y) \times \max\left\{0, |y| - \frac{\lambda}{2}\right\} \tag{4.15}$$

Algorithm 7 summaries the steps of proposed GSP algorithm for mixed denoising.

Algorithm 7 Generalized Synthesis Prior Algorithm for Mixed Noise Reduction

Input: $D_1, D_2, D = D_2 \otimes D_1^T$, Y, λ, μ_1, μ_2, Iter.
Output: $\hat{X}$, denoised image.
for $k = 1$ to Iter **do**
    $Z^{k+1} = \frac{1}{\lambda+\mu_1}\left(D^T(Y - S^k) + \mu_1(P^k - B_1^k)\right)$
    $P^{k+1} = \text{SoftTh}\left(Z^{k+1} + B_1^k, \frac{2}{\mu_1}\right)$
    $Q^{k+1} = \text{SoftTh}\left(S^{k+1} + B_2^k, \frac{2}{\mu_2}\right)$
    $S^{k+1} = \frac{\lambda}{\lambda+\mu_2}(Y - DZ^{k+1}) + \frac{\mu_2}{\lambda+\mu_2}(Q^{k+1} - B_2^k)$
    $B_1^{k+1} = B_1^k - P^{k+1} + Z^{k+1}$
    $B_2^{k+1} = B_2^k + S^{k+1} - Q^{k+1}$
end for
$\hat{X} = D_1^T Z^{k+1} D_2^T$

Spatio-Spectral Total Variation

The problem formulation (4.14) can be expressed as :

$$\min_{X,S} \|Y - X - S\|_F^2 + \lambda\|S\|_1 + \mu\|D_h X D\|_1 + \mu\|D_v X D\|_1$$

Let $\nabla_h = D^T \otimes D_h$, $\nabla_v = D^T \otimes D_v$ and using the property of Kronecker product that $A = BCD$ can be expressed as $a = (D^T \otimes B)c$. Further, let small letters represents vector form of corresponding matrix, we get

$$\min_{x,s} \|y - x - s\|_2^2 + \lambda\|s\|_1 + \mu\|\nabla_h x\|_1 + \mu\|\nabla_h x\|_1$$

above problem can be solved using alternating minimization as

$$x^{k+1} = \arg\min_x \|y - x - s^k\|_2^2 + \mu\|\nabla_h x\|_1 + \mu\|\nabla_v x\|_1$$
$$s^{k+1} = \arg\min_s \|y - x^{k+1} - s^k\|_2^2 + \lambda\|s\|_1$$

The s-problem can be solved using soft-thresholding whereas we solve the x-sub-problem using split-Bregman as follows,

$$\begin{aligned} \underset{x,d_h,d_v}{\text{minimize}} \quad & \|y - x - s^k\|_2^2 + \mu\|d_h\|_1 + \mu\|d_v\|_1 \\ \text{subject to} \quad & d_h = \nabla_h x \\ & d_v = \nabla_v x \end{aligned}$$

The above constrained optimization problem can be expressed as unconstrained optimization problem using quadratic penalty function as follows:

$$\underset{x,\, d_h,\, d_v}{\text{minimize}} \ \|y - x - s^k\|_2^2 + \mu\|d_h\|_1 + \mu\|d_v\|_1 + \frac{\nu}{2}\|d_h - \nabla_h x\|_2^2 + \frac{\nu}{2}\|d_v - \nabla_v x\|_2^2$$

where ν is regularization parameter. The same parameter ν is used for both horizontal and vertical total variation regularization terms for considering their equal contributions. Let $J(x, d_h, d_v) = \|y - x - s^k\|_2^2 + \mu\|d_h\|_1 + \mu\|d_v\|_1$, then the above objective function can be expressed as:

$$\underset{x,\, d_h,\, d_v}{\text{minimize}}\; J(x, d_h, d_v) + \frac{\nu}{2}\|d_h - \nabla_h x\|_2^2 + \frac{\nu}{2}\|d_v - \nabla_v x\|_2^2.$$

Bregman Iteration suggest to iteratively minimize the Bregman distance instead of minimizing J. Therefore, above problem can be written as

$$(x^{k+1}, d_h^{k+1}, d_v^{k+1}) = \underset{x,\, d_h,\, d_v}{\operatorname{argmin}}\; D_J^{p^k}(x, x^k, d_h, d_h^k, d_v, d_v^k) + \frac{\nu}{2}\|d_h - \nabla_h x\|_2^2 + \frac{\nu}{2}\|d_v - \nabla_v x\|_2^2$$

let $p_x^k \in \partial J(x^k)$, $p_{d_h}^k \in \partial J(d_h^k)$, and $p_{d_v}^k \in \partial J(d_v^k)$ with $p_x^0 = 0, p_{d_h}^0 = 0, p_{d_v}^0 = 0$ then

$$D_J^{p^k}(x, x^k, d_h, d_h^k, d_v, d_v^k) = J(x, d_h, d_v) - < p_x^k, x - x^k > - < p_{d_h}^k, d_h - d_h^k > - < p_{d_v}^k, d_v - d_v^k >$$

results in optimization problem

$$(x^{k+1}, d_h^{k+1}, d_v^{k+1}) = \underset{x,\, d_h,\, d_v}{\operatorname{argmin}}\; D_J^{p^k}(x, x^k, d_h, d_h^k, d_v, d_v^k) + \nu\|d_h - \nabla_h x\|_2^2 + \nu\|d_v - \nabla_v x\|_2^2$$

from the optimality condition at $(x^{k+1}, d_h^{k+1}, d_v^{k+1})$ we have

$$\begin{aligned}
0 &\in \partial J(x^{k+1}) - p_x^k - \nu\nabla_h^T(d_h^{k+1} - \nabla_h x^{k+1}) - \nu\nabla_v^T(d_v^{k+1} - \nabla_v x^{k+1}) \\
0 &\in \partial J(d_h^{k+1}) - p_{d_h}^k - \nu\nabla_h^T(d_h^{k+1} - \nabla_h x^{k+1}) \\
0 &\in \partial J(d_v^{k+1}) - p_{d_v}^k - \nu\nabla_h^T(d_v^{k+1} - \nabla_v x^{k+1})
\end{aligned}$$

with the initialization $p_x^0 = 0, p_{d_h}^0 = 0, p_{d_v}^0 = 0$

$$\begin{aligned}
p_x^{k+1} &= p_x^k + \nu\nabla_h^T(d_h^{k+1} - \nabla_h x^{k+1}) + \nu\nabla_v^T(d_v^{k+1} - \nabla_v x^{k+1}) \\
p_{d_h}^{k+1} &= p_{d_h}^k - \nu\nabla_h^T(d_h^{k+1} - \nabla_h x^{k+1}) = -\nu\nabla_h^T \sum_{i=1}^{k+1}(d_h^i - \nabla_h x^i) \\
p_{d_v}^{k+1} &= p_{d_v}^k - \nu\nabla_v^T(d_v^{k+1} - \nabla_v x^{k+1}) = -\nu\nabla_v^T \sum_{i=1}^{k+1}(d_v^i - \nabla_v x^i)
\end{aligned}$$

Define variables b_{d_h}, b_{d_v} with initialization $b_{d_h}^0 = 0, b_{d_v}^0 = 0$ such that

$$\begin{aligned}
b_{d_h}^{k+1} &= b_{d_h}^k + (d_h^{k+1} - \nabla_h x^{k+1}) = \sum_{i=1}^{k+1}(d_h^i - \nabla_h x^i), \text{ thus} \quad p_{d_h}^k = -\nu b_{d_h}^k \\
b_{d_v}^{k+1} &= b_{d_v}^k + (d_v^{k+1} - \nabla_v x^{k+1}) = \sum_{i=1}^{k+1}(d_v^i - \nabla_v x^i), \text{ thus} \quad p_{d_h}^k = -\nu b_{d_v}^k,
\end{aligned}$$

therefore we can express p_x^{k+1} as

$$p_x^{k+1} = \nu \nabla_h^T \sum_{i=1}^{k+1} (d_h^i - \nabla_h x^i) + \nu \nabla_v^T \sum_{i=1}^{k+1} (d_v^i - \nabla_v x^i) = \nu \nabla_h^T b_{d_h}^{k+1} + \nu \nabla_v^T b_{d_v}^{k+1}$$

Using these values, the objective function can be written as

$$\underset{x,\, d_h,\, d_v}{\operatorname{argmin}} J(x, d_h, d_v) - < p_x^k, x - x^k > - < p_{d_h}^k, d_h - d_h^k > - < p_{d_v}^k, d_v - d_v^k > \\ + \frac{\nu}{2} \|d_h - \nabla_h x\|_2^2 + \frac{\nu}{2} \|d_v - \nabla_v x\|_2^2$$

Taking the constant terms in the constant C_1, we get:

$$\underset{x,\, d_h,\, d_v}{\operatorname{argmin}} J(x, d_h, d_v) - < p_x^k, x > - < p_{d_h}^k, d_h > - < p_{d_v}^k, d_v > + \frac{\nu}{2} \|d_h - \nabla_h x\|_2^2 \\ + \frac{\nu}{2} \|d_v - \nabla_v x\|_2^2 + C_1$$

using the values of $p_x^k, p_{d_h}^k, p_{d_v}^k$ from above, we get

$$\underset{x,\, d_h,\, d_v}{\operatorname{argmin}} J(x, d_h, d_v) - \nu < \nabla_h^T b_{d_h}^k + \nabla_v^T b_{d_v}^k, x > + \nu < b_{d_h}^k, d_h > + \nu < b_{d_v}^k, d_v > \\ + \frac{\nu}{2} \|d_h - \nabla_h x\|_2^2 + \frac{\nu}{2} \|d_v - \nabla_v x\|_2^2 + C_1$$

rearranging the terms results in:

$$\underset{x,\, d_h,\, d_v}{\operatorname{argmin}} J(x, d_h, d_v) + \nu < b_{d_h}^k, d_h - \nabla_h x > + \nu < b_{d_v}^k, d_v - \nabla_v x > \\ + \frac{\nu}{2} \|d_h - \nabla_h x\|_2^2 + \frac{\nu}{2} \|d_v - \nabla_v x\|_2^2 + C_1,$$

by observing

$$\frac{\nu}{2} \left\| (a - b) + c \right\|^2 = \frac{\nu}{2} \|a - b\|_2^2 + \nu < c, a - b > + \frac{\nu}{2} \|c\|_2^2,$$

we obtain

$$\underset{x,\, d_h,\, d_v}{\operatorname{argmin}} J(x, d_h, d_v) + \frac{\nu}{2} \left\| d_h - \nabla_h x + b_{d_h}^k \right\|_2^2 + \frac{\nu}{2} \left\| d_v - \nabla_v x + b_{d_v}^k \right\|_2^2 + C_2$$

ignoring constant, the objective function becomes

$$\underset{x,\, d_h,\, d_v}{\operatorname{argmin}} \|y - x - s^k\|_2^2 + \mu \|d_h\|_1 + \mu \|d_v\|_1 + \frac{\nu}{2} \left\| d_h - \nabla_h x + b_{d_h}^k \right\|_2^2 + \frac{\nu}{2} \left\| d_v - \nabla_v x + b_{d_v}^k \right\|_2^2$$

The above problem is in three variables (x, d_h, d_v) but it is now separable and can be solved as:

$$\text{P1}: \quad \min_{d_h} \ \mu\|d_h\|_1 + \frac{\nu}{2}\left\|d_h - \nabla_h x + b_{d_h}^k\right\|_2^2$$
$$\text{P2}: \quad \min_{d_v} \ \mu\|d_v\|_1 + \frac{\nu}{2}\left\|d_v - \nabla_v x + b_{d_v}^k\right\|_2^2$$
$$\text{P3}: \quad \min_{x} \ \|y - x - s^k\|_2^2 + \frac{\nu}{2}\left\|d_h - \nabla_h x + b_{d_h}^k\right\|_2^2 + \frac{\nu}{2}\left\|d_v - \nabla_v x + b_{d_v}^k\right\|_2^2$$

The sub-problems P1, P2, P3, and the S-prroblem are of the form :

$$\arg\min_x \|y - x\|_2^2 + \alpha\|x\|_1$$

which can be solved by using soft-thresholding [94] operation:

$$\hat{x} = \text{SoftTh}(y, \alpha) = \text{sign}(y) \times \max\left\{0, |y| - \frac{\alpha}{2}\right\},$$

Sub-problem P3 is a least-square problem which after differentiating, yields

$$(I + \frac{\nu}{2}\nabla)x = 2(y - s^k) + \frac{\nu}{2}\nabla_h^T(d_h + b_{d_h}^k) + \nu\nabla_v^T(d_v + b_{d_v}^k) \tag{4.16}$$

with $\nabla = \nabla_h^T\nabla_h + \nabla_v^T\nabla_v$. This problem can be solved using iterative least square solvers such as LSQR [86]. Bregman variables b_{d_h}, b_{d_v} can be updated in each iterations (k) as follows:

$$b_{d_h}^{k+1} = b_{d_h}^k + (d_h^{k+1} - \nabla_h x^{k+1})$$
$$b_{d_v}^{k+1} = b_{d_v}^k + (d_v^{k+1} - \nabla_v x^{k+1})$$

Algorithm 8 summaries the steps of proposed spatio-spectral total variation (SSTV) algorithm.

Algorithm 8 Spatio-Spectral Total Variation Algorithm

input: Y, λ μ, ν, MaxIter
output: $\hat{X}$ (denoised image).
for $k = 1$ to MaxIter **do**
    $d_h^{k+1} = \text{SoftTh}(\nabla_h x^k - b_{d_h}^k, \frac{\nu}{2})$
    $d_v^{k+1} = \text{SoftTh}(\nabla_v x^k - b_{d_v}^k, \frac{\nu}{2})$
    $s^{k+1} = \text{SoftTh}(y - x^k, \lambda)$
    $c^{k+1} = x$ from (4.16)
    $b_{d_h}^{k+1} = b_{d_h}^k + (d_h^{k+1} - \nabla_h x^{k+1})$
    $b_{d_v}^{k+1} = b_{d_v}^k + (d_v^{k+1} - \nabla_v x^{k+1})$
end for
return $\hat{X} = X^{k+1}$

4.2.3 Experiments and Results

Experiments were performed with two hyperspectral image datasets. The first dataset is of Washington DC (WDC) mall [88] from Hyperspectral Digital Imagery Collection Experiment (HYDICE) sensor having $1m$ spatial resolution and 10-nm band spacing covering spectral range of 400-2500 nm. We considered a patch of size $256 \times 256 \times 190$ from *WDC* image for experiments. Second image was of Gulf of Mexico area [95] from SpecTIR having $2m$-spatial resolution and 5-nm band spacing covering spectral range of 395-2450-nm. A patch of size $256 \times 256 \times 160$ was considered for performing experiments.

The proposed algorithm SSTV was compared with four existing denoising algorithms LRMR [20], PCAW [23], GSP [91], and HTV. The LRMR [20] approach is based on a low rank matrix recovery algorithm [96]. It explores low-rank nature of a hyperspectral image for denoising. The PCAW [23] approach is a principle component analysis based (PCA) approach which keeps initial high energy PCA components unaltered and then perform wavelet-thresholding on low energy components along both spatial and spectral dimension. The GSP [91] algorithm is a general synthesis prior algorithm which also explicitly consider sparse noise in the problem formulation. It explores sparsity of the hyperspectral data cube along both spatial and spectral dimensions. HTV algorithm is based on band-by-band total variation model which also consider sparse noise. We can obtain HTV algorithm by setting 1D total variation operator D to identity operator in Algorithm 8.

All the unknown variables (X, S, P, Q, B_1, B_2) required by our algorithm were initialized to zero. All three parameter values $\lambda = 0.1,\ \mu = 0.2,\ \nu = 0.2$ were found empirically. Although our proposed algorithm (SSTV) depends on three regularization parameters (λ, μ, ν) but the algorithm is not very sensitive to the specific values of these parameters and allows a broad range of values. These parameters can be adjusted to get desired denoising strength however we had kept the parameter values constant for all the synthetic as well as real data experiments. The value of parameter λ adjusts the denoising strength corresponding to sparse noise whereas parameters μ and ν provides trade off between retaining original image and smoothness by total-variation regularization respectively.

Parameters for the LRMR algorithm (rank=4 and sparsity=4000) were set to yield the best results as described in the paper [20]. We utilized 3D-DCT as sparsifying transform in the general synthesis prior (GSP) algorithm i.e. $2D$-DCT to sparsify each spectral band image and 1D-DCT to sparsify across the spectral bands. The first five high energy PCA components were kept unaltered in the PCAW algorithm and wavelet shrinkage was applied with a window size of 1×3 as described in the paper [23]. The parameter value of $\lambda = 1,\ \mu = .5,\ \nu = 0.01$ were empirically found to give best results on both the hyperspectral datasets in HTV algorithm.

The first set of experiments were conducted to experimentally check the convergence of proposed algorithm. Figure 5.9 shows the variation of objective function (4.14) value with number of iterations. It can be observed that after 40 iterations objective function value converges to a locally optimal value. Therefore we can stop our algo-

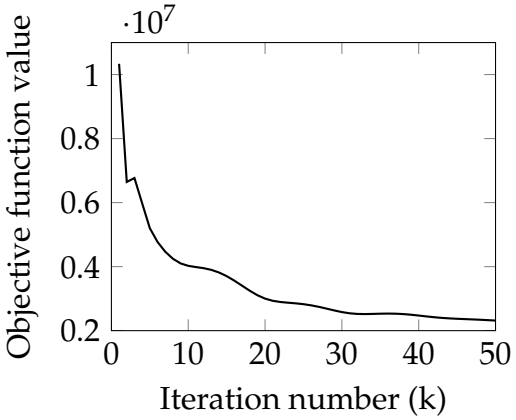


Figure 4.4: *Convergence of proposed SSTV algorithm*

rithm after 40 iterations.

The second set of experiments were conducted with synthetic noise so as to check the robustness of proposed method with respect to different noise levels. The denoising results are quantified using peak signal to noise ration (PSNR) and structural similarity index (SSIM). PSNR between original image X and reconstructed image Y was calculated as:

$$\text{PSNR} = \frac{1}{d}\sum_{i=1}^{d} 10\log_{10}\left(\frac{\max(x_i)^2}{\text{MSE}(\text{x}_\text{i}, \text{y}_\text{i})}\right)$$

where $\text{MSE}(x_i, y_i)$ represents mean square error between bands x_i and y_i. Structural similarity index takes into account hue, contrast and shape of restored image and gives a normalized score between zero and one where maximum value of one represents a perfect match. Table 4.3 summarizes the comparison of PSNR (dB) and SSIM values obtained from different experiments. Maximum value of PSNR and SSIM values for each experiment is bold faced. It can be observed from Table 4.3 that proposed SSTV algorithm have high PSNR and SSIM values when all four kinds of noise are present in the hyperspectral image. The HTV algorithm does not consider spectral information therefore PSNR and SSIM values are comparatively lower for this algorithm.

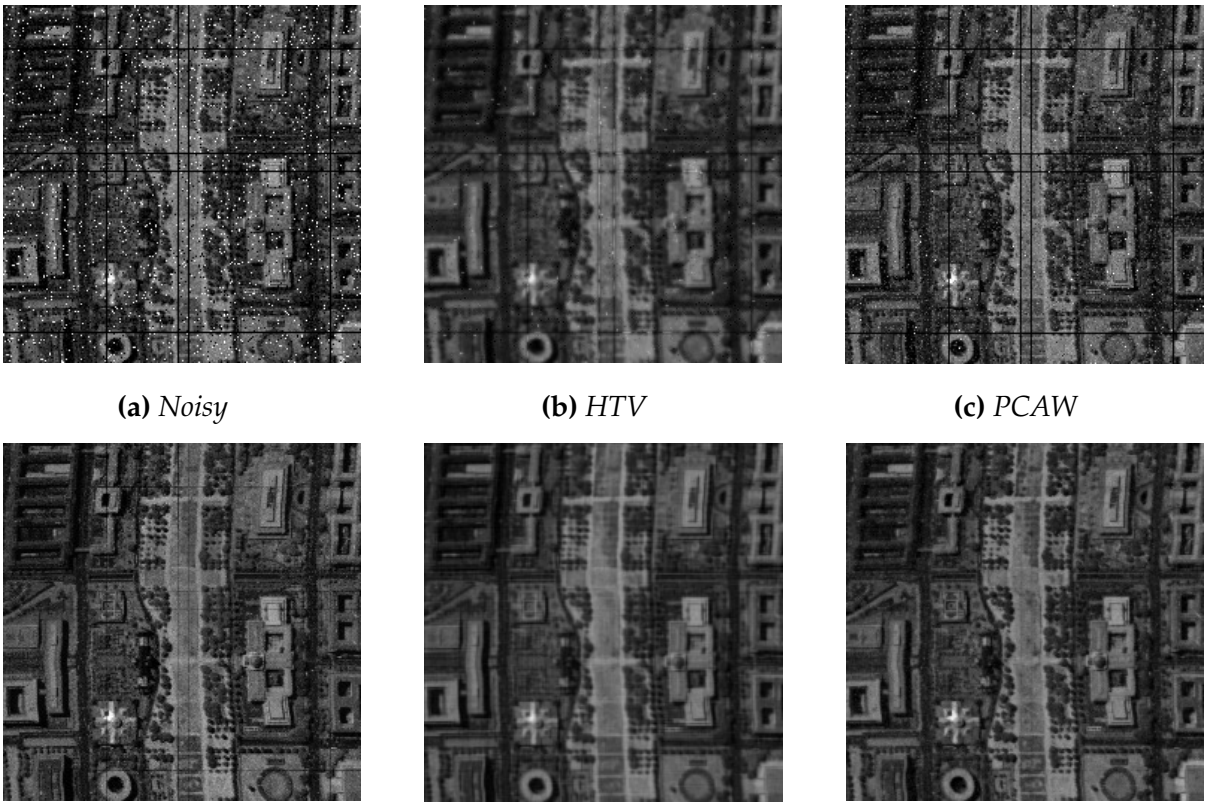
Figure 4.5 visually compares the denoising results on band 110 of WDC image when all four kinds of noise are present. All bands were corrupted by synthetically added Gaussian noise of SNR=20 and 5% impulse noise. Band number $60, 110, 111, 132$ were corrupted by four deadlines and line strips. Horizontal deadlines were located at $30, 100, 112, 220$ and vertical deadlines were located at $70, 118, 128, 220$. This mixed noise corrupted image is shown in Fig. 4.5(a). Both HTV and PCAW methods are able to reduce Gaussian noise and some impulse noise as observed in Fig. 4.5(b) and 4.5(c). Neither of these methods are able to reduce either deadlines or line strips. LRMR method is able to reduce all these four kinds of noise but still some noise can be visualized in reconstructed image 4.5(d). It can be observed from Fig. 4.5(e) and 4.5(f) that visual quality of denoised images by GSP method and proposed SSTV method is better than other methods compared against however GSP results look somewhat over-smooth.

Table 4.3: *Comparison of PSNR and SSIM values obtained by different algorithms for different kinds of mixed noise.*

Mixed noise	Peak Signal to Noise Ratio (dB)					
	Noisy	HTV	PCAW	LRMR	GSP	SSTV
Gaussian	30.29	35.18	40.44	39.98	36.86	**41.12**
Mix1	17.01	25.84	30.81	35.12	33.55	**40.38**
Mix2	14.17	21.14	25.99	27.59	32.96	**39.88**

Mixed noise	Structural Similarity Index					
	Noisy	HTV	PCAW	LRMR	GSP	SSTV
Gaussian	0.90	0.93	**0.99**	0.98	0.97	**0.99**
Mix1	0.52	0.78	0.90	0.97	0.95	**0.98**
Mix2	0.33	0.59	0.83	0.87	0.94	**0.98**

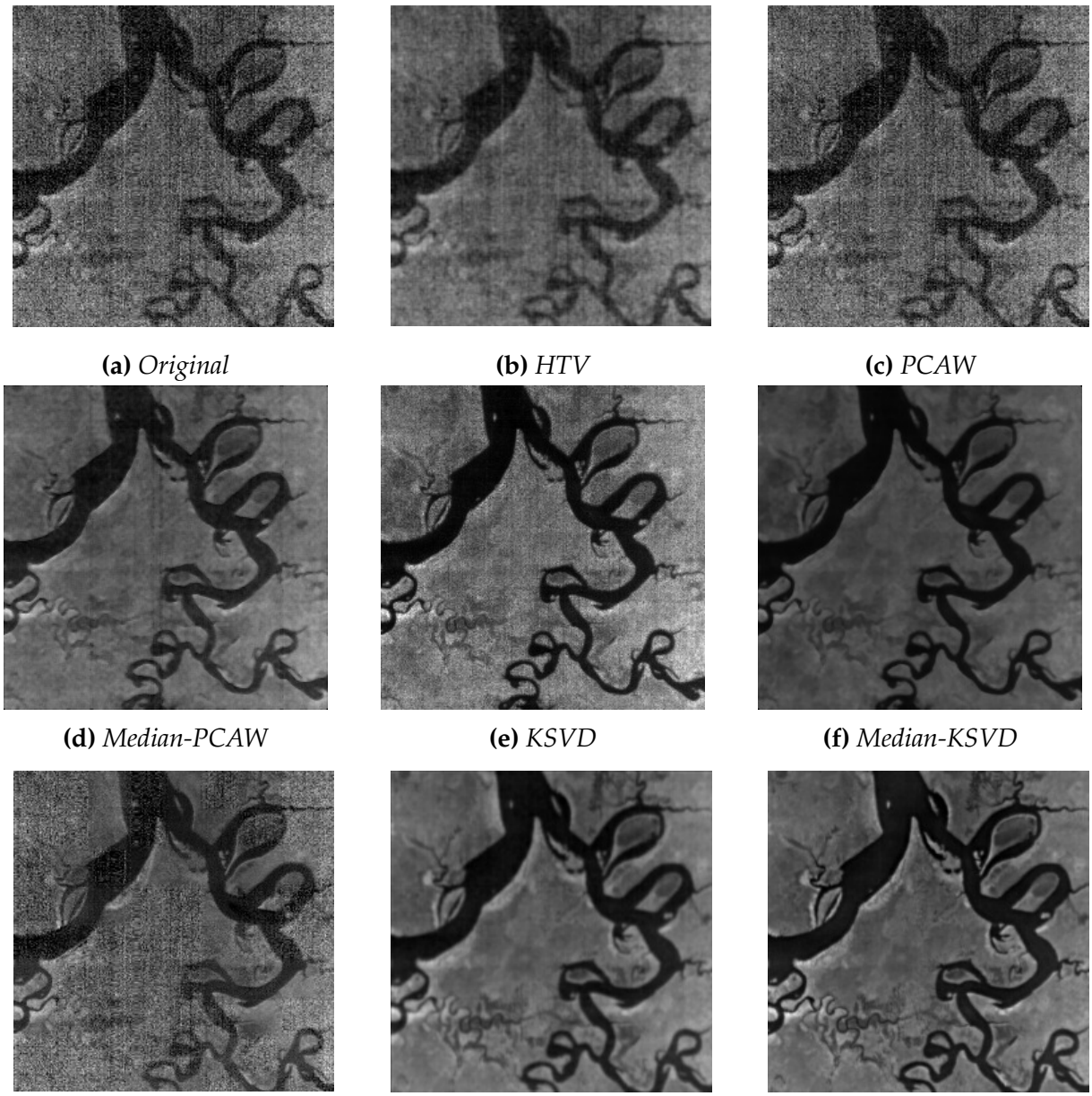
The third set of experiments were conducted with real noisy hyperspectral images. Denoising results obtained by different algorithms are shown in Fig. 4.6. Several bands of the Gulf image have noise. Figure 4.6(a) shows original band 120 of Gulf image which is a noisy band. HTV method (see Fig. 4.6(b)) and LRMR method (see Fig. 4.6(g)) have reduced noise but some noise is still visible. PCAW (see Fig. 4.6(c)) method is not able to reduce noise from this image. However, PCAW preceded by median filtering help in reducing high value noise. Figure 4.6(d) shows the improved results obtained from the median filtering followed by PCAW approach. Experiments were also performed with dictionary learning based KSVD algorithm and a variant in which firstly median filtering was performed followed by KSVD. Zooming images displayed in Fig. 4.6(h) and 4.6(i) clearly shows that different kinds of noise have been effectively reduced by both GSP and proposed SSTV methods.



(a) *Noisy* **(b)** *HTV* **(c)** *PCAW*

(d) *LRMR* **(e)** *GSP* **(f)** *Proposed*

Figure 4.5: *Comparison of visual quality of denoising results with synthetic noise in WDC hyperspectral image by different algorithms. Band 110 denoised by various algorithms.*



(a) *Original* **(b)** *HTV* **(c)** *PCAW*

(d) *Median-PCAW* **(e)** *KSVD* **(f)** *Median-KSVD*

(g) *LRMR* **(h)** *GSP* **(i)** *Proposed*

Figure 4.6: *Comparison of visual quality of denoising results on real noisy Gulf hyperspectral image by different algorithms. Band 120 denoised by various algorithms.*

Chapter 5

Hyperspectral Unmixing

Hyperspectral unmixing is the process of estimating constituent endmembers and their fractional abundances present at each pixel in a hyperspectral image. This chapter addresses the hyperspectral unmixing problem in a general scenario that consider the presence of mixed noise. The unmixing model explicitly takes into account both Gaussian noise and sparse noise. The section 5.1 describes how unmixing problem can be formulated as joint-sparse total variation regularized problem. The section 5.2 outlines an algorithm for solving the unmixing problem followed by section 5.3 on detailed experimental results on both synthetic and real hyperspectral images.

5.1 Problem Description and Formulation

This section describes how linear unmixing problem can be mathematically formulated as sparse recovery problem followed by our proposed problem formulation. Let I_n represents identity matrix of size $n \times n$. The operation $x = vec(X)$ represents vectorization operation on matrix X with columns appended whereas $X = mat(x)$ represents its inverse operation. A hyperspectral datacube of size $m \times n \times b$ can be represented as a matrix of size $b \times p$ where b is the total number of bands and $p = m \times n$ is the total number of pixels in the image. $M \in \mathbb{R}^{b \times e}$ represents mixing matrix also called endmember matrix in which each column represents spectral signature of an endmember. Let $\nabla = \begin{pmatrix} \nabla_h \\ \nabla_v \end{pmatrix}$ be total variation operator with ∇_h and ∇_v representing horizontal and vertical total variation operators respectively. The horizontal total variation is defined as the difference between pixel values along horizontal direction and can be expressed as $(\nabla_h X)_{i,j} = X_{i,j+1} - X_{i,j}$. Similarly, the vertical total variation is defined as the difference between pixel values along vertical direction and can be expressed as $(\nabla_v X)_{i,j} = X_{i+1,j} - X_{i,j}$. The $\ell_{2,1}$ norm of a matrix $A \in \mathbb{R}^{M \times N}$ is defined as

$$\|A\|_{2,1} = \sum_{i=1}^{M} \|\overrightarrow{a_i}\|_2 = \sum_{i=1}^{M} \sqrt{\sum_{j=1}^{N} a_{ij}^2}$$

whereas Frobenius norm $\|\cdot\|_F$ and ℓ_1-norm of a matrix are defined as follows:

$$\|A\|_F^2 = \sum_{i=1}^{M}\sum_{j=1}^{N} a_{ij}^2, \quad \|A\|_1 = \sum_{i=1}^{M}\sum_{j=1}^{N} |a_{i,j}|.$$

The Frobenius norm of a matrix is sum of squares of the entries whereas ℓ_1-norm of a matrix is sum of absolute values of its entries.

5.1.1 Problem Description

The linear unmixing problem for a pixel in the presence of Gaussian noise is represented as constrained linear regression model:

$$y = Ma + n, \quad \|a\|_1 = 1, \quad a_i \geq 0 \ \forall i \tag{5.1}$$

where $y \in \mathbb{R}^{b\times 1}$ is a pixel vector in b spectral bands, M is a mixing matrix with e number of endmembers as column vectors, $a \in \mathbb{R}^{e\times 1}$ is called abundance vector that represents the fraction of each endmember used in the formation of that pixel, and n represents Gaussian noise which accounts for various external environmental factors. The constraint $\|a\|_1 = 1$ represents abundance sum-to-one constraint to ensure that total contribution of each endmember in formation of a pixel is one. As it has been noticed in [36, 45, 97], all the endmembers present in a real hyperspectral image may not be available in the spectral library. Therefore, abundance sum may not be exactly equal to one. Also, if this constraint of $\|a\|_1 = 1$ is enforced then formulating the ℓ_1-norm minimization problem on a will be meaningless. Therefore, this work does not enforce this constraint in the problem formulation. The abundance non-negativity constraint represents that contribution can not be negative.

Since mixing matrix is known for hundreds of most commonly used materials. Therefore generally $e > b$ and (5.1) is an underdetermined system of linear equations. In general, an underdetermined system has infinite solutions therefore, we need additional constraints on the variable a to determine it uniquely. The observation that a pixel is mixture of very few endmembers as opposed to hundreds of available endmembers allow us to treat abundance vector a as sparse vector thus unmixing can be recast as compressed sensing[98, 99] problem :

$$\min_a \|y - Ma\|_2^2 \text{ subject to } \|a\|_0 \leq k \tag{5.2}$$

where k is the sparsity of a i.e. maximum number of non-zero elements of a. This is an NP-hard[100] problem whose solution can be approximated using greedy pursuit algorithms such as OMP [101], StOMP [50], CoSAMP [102], etc. It has been shown that under certain conditions solution of the NP-hard problem (5.2) can be approximated by solving its convex surrogate ℓ_1-norm minimization problem

$$\min_a \|y - Ma\|_2^2 + \lambda\|a\|_1. \tag{5.3}$$

This problem is a convex optimization problem, and various algorithms have been proposed in literature SPGL1 [49], FISTA [51], NESTA [103], Bregman Iteration [48],

etc., to solve this problem. The unmixing model in (5.1) can be extended for all the pixels as

$$Y = MA + N, \quad A \geq 0 \tag{5.4}$$

where $Y \in \mathbb{R}^{b \times p}$ is a matrix with p pixels as column vectors, $A \in \mathbb{R}^{e \times p}$ is sparse abundance matrix, N is Gaussian noise. This unmixing model can be though of as specialization of image denoising model :

$$Y = X + N$$

where $X \in \mathbb{R}^{b \times p}$ and $X = MA$ is clean hyperspectral image which imply that unmixing can lead to denoising provided that mixing matrix is known.

5.1.2 Proposed Formulation

A real hyperspectral image may contain a mixture of Gaussian and sparse noise therefore, we consider the mixed noise model for unmixing and account for both types of noise. The usual unmixing model in (5.4) can be extended as

$$Y = MA + S + G, \quad A \geq 0 \tag{5.5}$$

here S and G represents sparse and Gaussian noise respectively. The above noise model assumes both Gaussian and sparse noise to be additive noise. Sparse noise accounts for horizontal or vertical line strips, shot noise and any impulse noise present in a hyperspectral image. All these kinds of noise are termed as sparse noise since they corrupt few pixels in a hyperspectral image. By utilizing this model, we can formulate the unmixing problem as:

$$\min_{A,S} \|Y - MA - S\|_F^2 + \lambda_1 \|A\|_{2,1} + \lambda_2 \|S\|_1 \tag{5.6}$$

Here the first term is data fidelity term that is equivalent to minimizing the variance of Gaussian noise $G = Y - MA - S$. First regularization term is an $\ell_{2,1}$-norm minimization term on abundance matrix A which is also called joint-sparse regularization term. This term is based on the observation that in most hyperspectral images, a fewer endmembers are present compared to the available endmembers. This observation is mathematically modeled as joint-sparse regularization on matrix A with few non-zero rows, but each non-zero row is allowed to be dense. The second regularization term corresponds to minimizing ℓ_1-norm of sparse noise matrix S. Here ℓ_1-norm is minimized due to modeling assumption that sparse noise affects few pixels in the image.

As an alternative unmixing model, we can also exploit the fact that most natural images are spatially homogeneous i.e. it is highly likely that neighboring pixels are likely to belong to same class e.g. if there are some vegetation pixels in the image then the nearby pixels are also likely to be vegetation pixels. Therefore, the abundance maps can be considered as piece-wise smooth. The piece-wise smoothness can be modeled as total variation regularization [43].

$$\min_{A,S} ||Y - MA - S||_F^2 + \lambda_1 ||\nabla A^T||_1 + \lambda_2 \|S\|_1 \tag{5.7}$$

Here ∇ is two-dimensional total variation operator that applies total variation along both horizontal and vertical direction on a 2D image. The operator ∇ is applied on A^T because each abundance map is along rows of A.

In this work, we propose to simultaneously exploit both the joint-sparsity as well as spatial smoothness of the abundance maps in the light of generic noise model. Thus the proposed hyperspectral unmixing problem formulation can be expressed as:

$$\min_{A,S} ||Y - MA - S||_F^2 + \lambda_1 ||\nabla A^T||_1 + \lambda_2 \|A\|_{2,1} + \lambda_3 \|S\|_1 \tag{5.8}$$

here λ_1, λ_2 and λ_3 are regularization parameters corresponding to total-variation term, joint-sparsity term, and sparse noise term respectively. These three models in (5.6), (5.7), and (5.8) estimates sparse noise S as a byproduct of the proposed formulations. Let $X = MA$ be the clean image then we can get denoised image $\hat{X} = M\hat{A}$ where $\hat{A}$ is the estimated abundance maps by solving (5.8). Along with generic noise model (5.5), we have exploited both joint-sparsity as well as piecewise-smoothness of abundance maps. We are not aware of any efficient algorithm to solve (5.8) therefore in the next section we briefly describe how to solve this problem using the split-Bregman [44] based technique.

5.2 Proposed Algorithm

This section describes how the split-Bregman [44] approach can be utilized to derive the algorithm for solving (5.8). The split-Bregman approach is suitable to solve (5.8) because it has been designed to handle multiple regularization terms.

The variable A is not separable in (5.8) therefore we utilize auxiliary variables P and Q to make the problem separable. Set $P = \nabla A^T$ and $Q = A$, then we get following constrained problem:

$$\begin{aligned} \underset{A,S,P,Q}{\text{minimize}} \quad & \|Y - MA - S\|_F^2 + \lambda_1 \|P\|_1 + \lambda_2 \|Q\|_{2,1} + \lambda_3 \|S\|_1 \\ \text{subject to} \quad & P = \nabla A^T \\ & Q = A \end{aligned}$$

This problem can be re-written into unconstrained form by using two Bregman relaxation variables B_1 and B_2 to get

$$\begin{aligned} \underset{A,S,P,Q}{\text{minimize}} \|Y - MA - S\|_F^2 + \lambda_1 \|P\|_1 + \lambda_2 \|Q\|_{2,1} + \lambda_3 \|S\|_1 + \mu_1 \|P - \nabla A^T - B_1\|_F^2 \\ + \mu_2 \|Q - A - B_2\|_F^2 \end{aligned}$$

where B_1 and B_2 are updated as:

$$\begin{aligned} B_1 &= B_1 + \nabla A^T - P \\ B_2 &= B_2 + A - Q \end{aligned}$$

Above problem is separable in each variable therefore can be written into following subproblems as

$$\begin{aligned}
\text{P1}: &\quad \min_{P} \ \mu_1\|P - \nabla A^T - B_1\|_F^2 + \lambda_1\|P\|_1 \\
\text{P2}: &\quad \min_{Q} \ \mu_2\|Q - A - B_2\|_F^2 + \lambda_2\|Q\|_{2,1} \\
\text{P3}: &\quad \min_{S} \ \|Y - MA - S\|_F^2 + \lambda_3\|S\|_1 \\
\text{P4}: &\quad \min_{A} \ \|Y - MA - S\|_F^2 + \mu_1\|P - \nabla A^T - B_1\|_F^2 + \mu_1\|Q - A - B_2\|_F^2
\end{aligned}$$

each of these problems can be solved iteratively by using Bregman iteration with Bregman variables updated in k^{th} iteration as

$$\begin{aligned}
B_1^{k+1} &= B_1^k + \nabla(A^k)^T - P^k \\
B_2^{k+1} &= B_2^k + A^k - Q^k
\end{aligned}$$

The problems $P1$ and $P3$ are of the form

$$\|y - x\|_2^2 + \lambda\|x\|_1$$

which can be solved by using soft-thresholding [94] operation:

$$\hat{x} = \text{SoftTh}(y, \lambda) = \text{sign}(y) \times \max\left\{0, |y| - \frac{\lambda}{2}\right\},$$

The problem $P2$ can be solved by using the procedure as described in section 3.3.3 of [104]. This is a ℓ_2-norm shrinkage operation on each row $q^{(i)} \ \forall i = 1, 2 \ldots e$, of matrix Q. This ℓ_2-norm shrinkage problem can be expressed as:

$$\min_x \|y - x\|_2^2 + \lambda\|y\|_2$$

whose solution is given by

$$\hat{x} = \text{Shrink}(y, \lambda) = \max\left\{\|y\|_2 - \frac{\lambda}{2}, 0\right\} \odot \frac{y}{\|y\|_2},$$

here $\odot$ represent element by element multiplication operation with the assumption that $0 \times \frac{0}{0} = 0$. The problem $P4$ is a differentiable convex optimization problem. After differentiating we get following linear system of equations with variable A :

$$M^T MA + \mu_1 A\nabla^T\nabla + \mu_2 A \quad = \quad M^T(Y - S) + \mu_1(P^T - B_1^T)\nabla + \mu_2(Q - B_2)$$

this equation can be re-written as

$$\begin{aligned}
\Psi a &= vec(M^T(Y - S)) + \alpha \quad \text{where} \\
\Psi &= (I_e \otimes M^T M)\mu_1(\nabla^T\nabla \otimes I_e) + \mu_1 I_{pe} \\
\alpha &= \mu_1(P^T - B_1^T)\nabla + \mu_2(Q - B_2))
\end{aligned} \tag{5.9}$$

The above system of linear equations is large and sparse whose solution can be approximated using algorithms such as LSQR [86]. Algorithm 9 outlines the steps of proposed Joint Sparse Total Variation (JSTV) algorithm. By setting $\lambda_1 = 0$, we can derive the solution of (5.6) which we refer as Split Bregman based Joint Sparse (SBJS) unmixing algorithm. Similarly $\lambda_2 = 0$ results in an algorithm that solves (5.7) which we refer as Split Bregman based Total Variation (SBTV) unmixing algorithm.

Algorithm 9 Proposed JSTV Algorithm for solving (5.8)

input: Y, λ_1, λ_2, μ_1, μ_2, innerIter, outerIter
output: $\hat{A}$ (Abundance maps).
for $j = 1$ to outerIter **do**
    for $k = 1$ to innerIter **do**
        $P^{k+1} = \text{SoftTh}(\nabla(A^k)^T + B_1^k, \frac{\lambda_1}{\mu_1})$
        $Q^{k+1} = \text{Shrink}(A^k + B_2^k, \frac{\lambda_2}{\mu_2})$
        $S^{k+1} = \text{SoftTh}(MA^K - Y, \lambda_3)$
        $A^{k+1} = \text{mat}(a)$ from (5.9)
        $B_1^{k+1} = B_1^k + D_h X^{k+1} D - P^{k+1}$
        $B_2^{k+1} = B_2^k + D_v X^{k+1} D - Q^{k+1}$
    end for
    $Y = Y - MA^k - S^k$
end for
return $\hat{A} = A^{j+1}$

5.3 Experiments and Results

This section describes the details of various experiments executed to validate the proposed method. First, datasets used in the experiments are described followed by evaluation metrics. After that, various synthetic data experiments and real data experiments are detailed with analysis of results.

5.3.1 Data Description

The existing USGS spectral library [105] was utilized in all the experiments. The library contains spectral signatures under six categories namely artificial, coatings, minerals, liquids, soil, and vegetation. We utilized endmembers from each of these categories in the experiments. We manually checked each endmember signature and removed some of the endmembers that had missing values for some wavelengths. Experiments were conducted with two synthetic and one real dataset. The first synthetic dataset has five abundance maps of 50×50 pixels with constant fraction value over a region. These abundance maps are shown in row one of Fig. 5.1. Each abundance map is composed of two or three endmembers as represented by the number

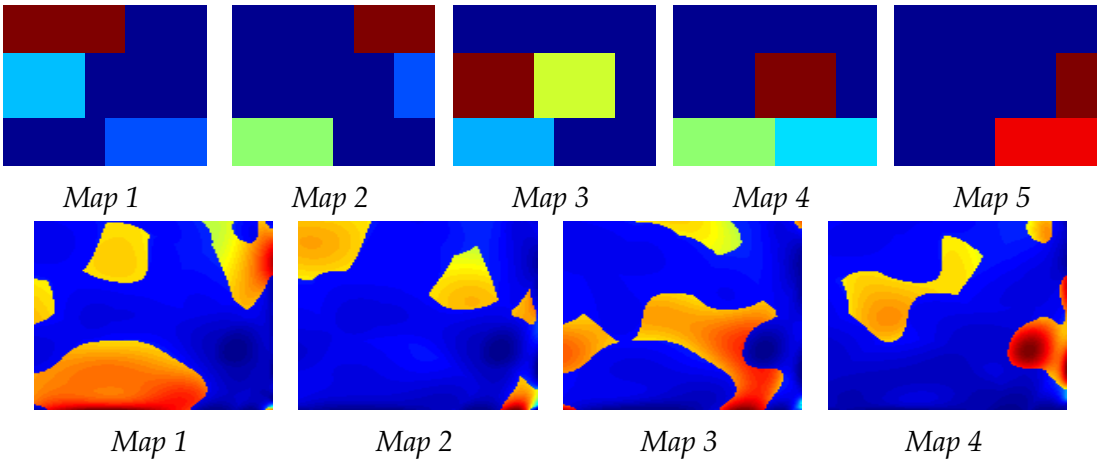


Figure 5.1: *Row one shows five synthetically generated abundance maps corresponding to first synthetic image whereas second row shows four abundance maps corresponding to second synthetic image.*

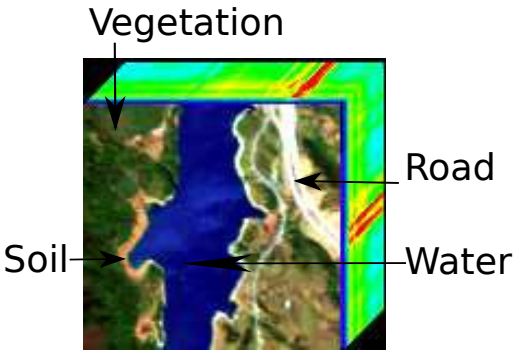


Figure 5.2: *Portion of Jasper image used in experiments.*

of rectangular boxes inside a map. Dark blue background color represents zero pixel value. Five endmembers were randomly selected to generate first synthetic image of dimension $50 \times 50 \times 224$. The second synthetic dataset was generated using HYDRA toolbox [106]. Four abundance maps of size 128×128 were generated using Legendre method and are shown in row two of Fig. 5.1. The second synthetic image of dimension $128 \times 128 \times 224$ was generated using four randomly selected endmembers from the spectral library. Both the datasets satisfy abundance sum to one constraint as well as abundance non-negativity constraint.

The experiments on the first synthetic image were conducted with the endmember matrix of dimension $\mathbb{R}^{224\times 269}$ such that angle between any two spectral signatures was at least 4 degree. The experiments with second synthetic image does not make any such assumption and utilized endmember matrix of dimensions $\mathbb{R}^{224\times 889}$. Real data experiments were done with a portion of Jasper Ridge image [107] of size $112 \times 118 \times 224$. A false color composite image is shown in Fig. 5.2. Several bands in this image are noisy bands. We had considered all 224 bands in real data experiments as opposed to many unmixing algorithms that remove noisy bands before doing unmixing. We did not have actual abundance maps for this real dataset however four major constituent endmembers can be easily recognized by visual interpretation. These

four endmembers are roads, vegetation, soil and water as indicated in the Fig. 5.2.

5.3.2 Synthetic Data Experiments

The synthetic data experiments were conducted to quantify the performance of proposed unmixing algorithm. Our work improves over the state of the art sparse regression based unmixing techniques sparse regression (SR) [45] and its variants Total Variation Spatial Regularization (SRTV) [46] and Collaborative Sparse Regression (CLSR) [47].

All three variables (B_1, B_2, S) required by our algorithm were initialized to zero. Each of the endmember in the matrix M was normalized by ℓ_2-norm. The abundance matrix A was initialized with random values such that they satisfy both the abundance sum constraint as well as the non-negativity constraint. The parameters λ_1, λ_2, λ_3, control the strength of smoothness, joint-sparsity and sparse noise term respectively. If the parameter values are zero then there is no denoising/unmixing and output is same as input. As we increase any parameter's value then that regularization term will be in effect. For example, a higher value of λ_1 will enforce resulting abundance maps to be highly joint sparse and it will trade-off with retaining original abundance maps. Similarly, a higher value of λ_2 will encourage results with low total variation of abundance maps. Increasing the value of λ_3 will promote results with low sparse noise. However, after a limit, high values of parameters tend to deviate resulting outputs image from original image.

The values of parameters for synthetic experiments were found experimentally using five-fold cross validation. The values for proposed JSTV algorithm used in experiments are $\lambda_1 = 2, \lambda_2 = 0.1, \lambda_3 = 1, \mu_1 = 0.04, \mu_2 = 1$. Parameter values of SBTV algorithm were found to be $\lambda_1 = 0.05, \lambda_2 = 0, \lambda_3 = 1, \mu_1 = 0.05, \mu_2 = 0$. Parameter values for SBJS algorithm were found to be $\lambda_1 = 0, \lambda_2 = 1, \lambda_3 = 1, \mu_1 = 0, \mu_2 = 2$. The parameters required by SR, SRTV, and CLSR algorithms were set as described in corresponding articles. In particular, parameter value of $\lambda = 5e^{-4}$ was used in SR algorithm. $\lambda = 3e^{-3}, \lambda_{TV} = 0.01, \mu = 0.01$ were used in SRTV algorithm and $\lambda = 1.3e^{-3}$ was used in CLSR algorithm. The two parameters of LRMR algorithm, rank and sparsity, were set to 8 and 15000 respectively. As the noise changes we need to adjust values of these parameters accordingly; however, the split-Bregman algorithm is quite robust to small changes in parameter values.

The first set of experiments was done to check the robustness of proposed unmixing method in the presence of different kinds of noise. Tables 5.1 and 5.2 quantifies the reconstruction quality of each method on two synthetic images using PSNR and SSIM values. The numerical values shown corresponding to each algorithm represents the average of PSNR values obtained for different abundance maps. The maximum value of PSNR and SSIM are boldfaced. Last two rows in both of these tables represent the case when mixed noise is present. This mixed noise causes a sharp decrease in performance of SR, SRTV, and CLSR algorithms because these three algorithms are not designed to handle impulse noise. The SBJS, SBTV, and proposed JSTV methods reduce that noise because of utilization of sparse noise concept in the unmixing framework.

Table 5.1: *PSNR and SSIM values for different Noise levels. The values shown are averaged for five abundance maps. Mix1 corresponds to Gaussian noise of SNR 30 dB and three vertical lines whereas mix2 is mix1 + 1% impulse noise.*

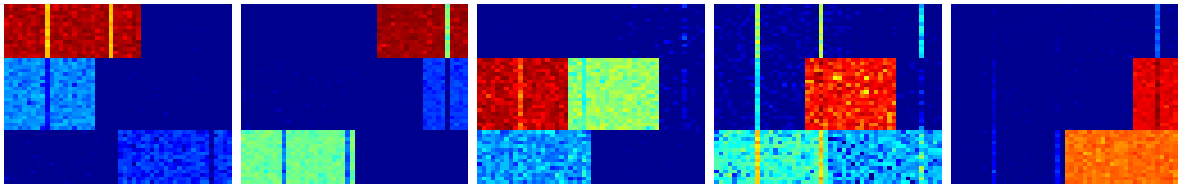
Noise	Peak Signal to Noise Ratio (dB)						Structural Similarity Index (SSIM)					
SNR(dB)	SR	SRTV	CLSR	SBJS	SBTV	JSTV	SR	SRTV	CLSR	SBJS	SBTV	JSTV
40	40.21	45.77	34.23	44.36	59.08	**61.94**	0.981	0.997	0.91	0.993	0.999	**0.999**
30	29.92	37.73	33.23	32.79	42.6	**46.82**	0.847	0.989	0.957	0.869	0.981	**0.99**
20	20.52	24.37	21.16	22.27	27.5	**29.87**	0.609	0.78	0.657	0.583	0.827	**0.84**
mix1	25.25	31.19	25.23	32.65	41.58	**46.84**	0.78	0.964	0.77	0.869	0.975	**0.994**
mix2	18.79	18.57	18.89	32.39	41.71	**46.23**	0.594	0.802	0.607	0.862	0.979	**0.992**

Table 5.2: *PSNR and SSIM values for different Noise levels. The values shown are averaged for five abundance maps. Mix1 corresponds to Gaussian noise of SNR 30 dB and three vertical lines whereas mix2 is mix1 + 1% impulse noise.*

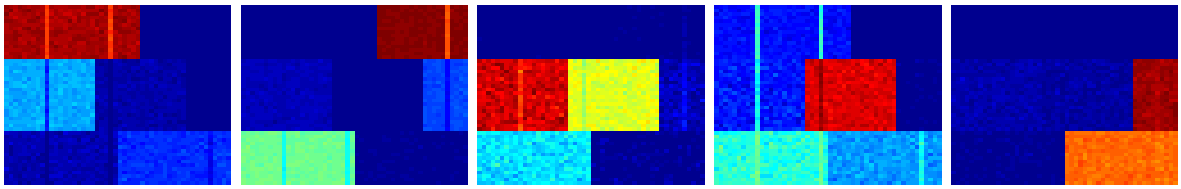
Noise	Peak Signal to Noise Ratio (dB)						Structural Similarity Index (SSIM)					
SNR(dB)	SR	SRTV	CLSR	SBJS	SBTV	JSTV	SR	SRTV	CLSR	SBJS	SBTV	JSTV
40	28.25	36.76	34.78	36.54	34.68	**43.12**	0.63	0.94	0.89	0.88	0.93	**0.98**
30	22.86	31.25	26.84	26.43	31.64	**39.21**	0.41	0.86	0.71	0.56	0.91	**0.97**
20	16.73	24.37	18.65	24.71	27.59	**25.32**	0.22	0.78	0.36	0.23	0.67	**0.74**
mix1	22.43	25.23	26.23	26.19	33.54	**38.98**	0.41	0.68	0.69	0.54	0.94	**0.97**
mix2	13.62	12.78	13.66	25.55	30.59	**38.56**	0.19	0.16	0.28	0.52	0.89	**0.97**

The results in Table 5.2 obtained on the second synthetic image are not quantitatively as good as the results in Table 5.1 for the first synthetic image. The main reason is the utilization of different unmixing matrices in both the cases. The endmembers matrix M in the second case was highly coherent compared to the first case.

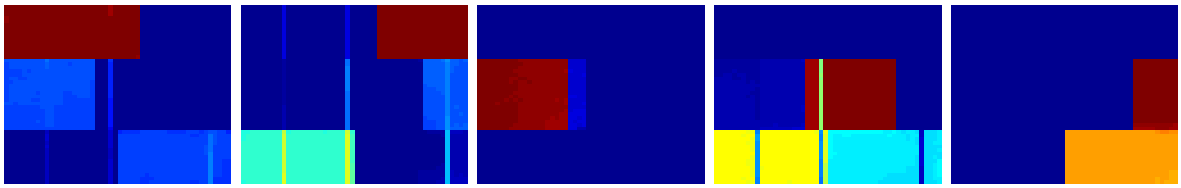
The second set of experiments were done on the synthetic image with mixed noise consisting of Gaussian noise and vertical line strips. Gaussian noise of signal to noise ratio (SNR) of 30 dB was added along with three vertical line strips. The noisy image is shown in Fig. 5.5(b). The unmixing algorithms were run on this noisy image. The resulting abundance maps are shown in Fig. 5.3. It can be observed from Fig. 5.3(a), 5.3(b), and 5.3(c) that SR, CLSR, and SRTV are not able to remove line strips from the abundance maps; however, the SRTV algorithm has reduced Gaussian noise. The SBJS algorithm is not able to reduce Gaussian noise but has reduced line strips as observed in Fig. 5.3(d). The proposed JSTV algorithm has reduced both the noise from all the abundance maps as can be seen by comparing estimated abundance maps in Fig. 5.3(f) with original abundance maps shown in Fig. 5.1. The unmixing results on the second synthetic image in the presence of only Gaussian noise of SNR 30 dB are shown in Fig. 5.4. Experimentally it was observed that all all abundance maps recovered by each algorithm should be shown because a particular unmixing algorithm may recover some abundance maps correctly while causing other abundance maps to be noisy.



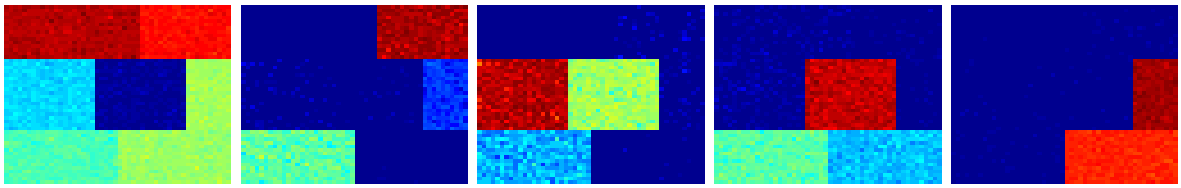
(a) *Abundance maps estimated by SR*



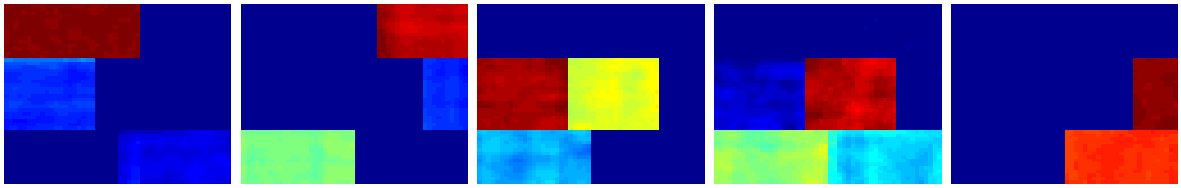
(b) *Abundance maps estimated by CLSR*



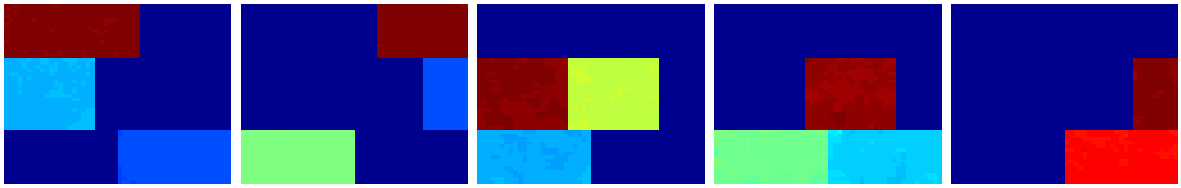
(c) *Abundance maps estimated by SRTV*



(d) *Abundance maps estimated by SBJS*

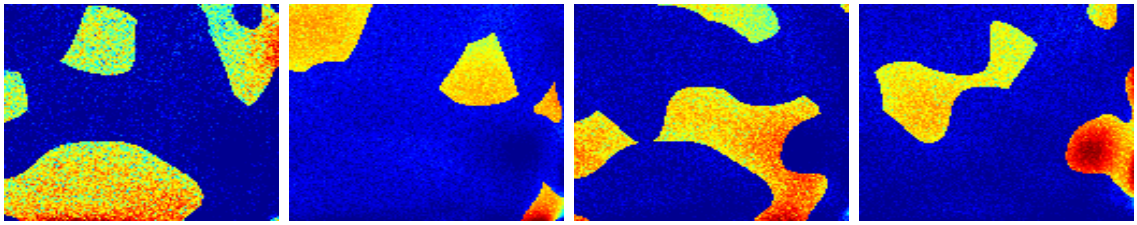


(e) *Abundance maps estimated by SBTV*

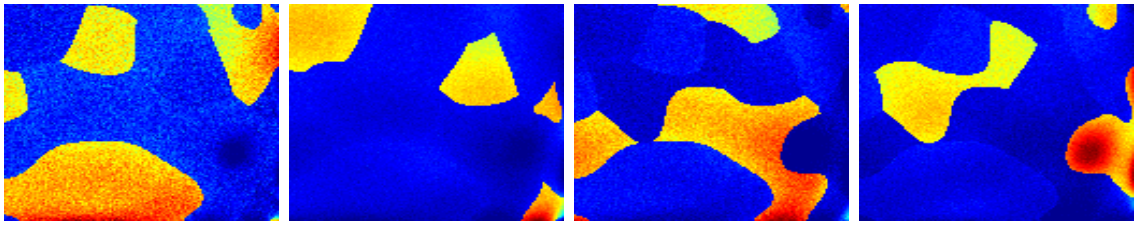


(f) *Abundance maps estimated by JSTV*

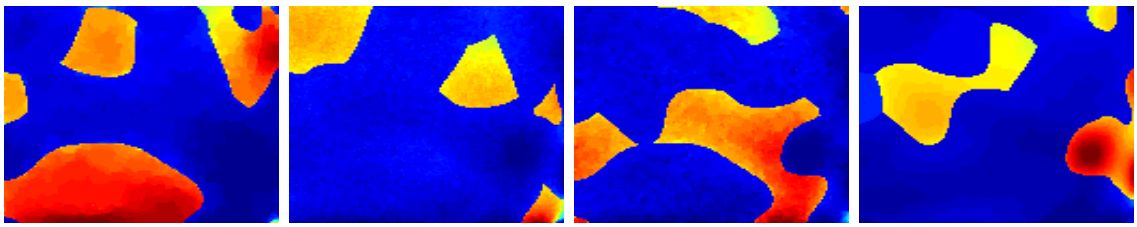
Figure 5.3: *Reconstructed Abundance maps by different algorithms*



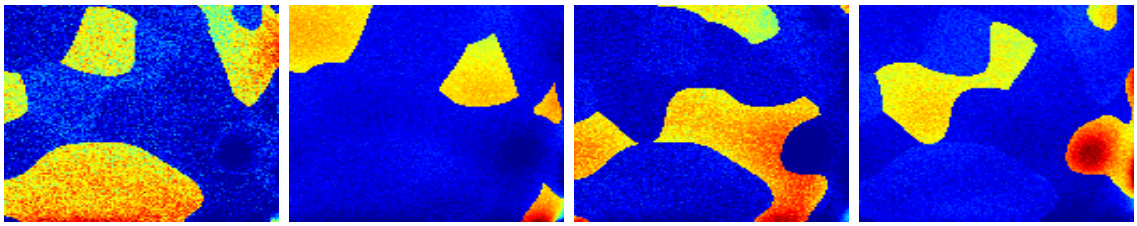
(a) *Abundance maps estimated by SR*



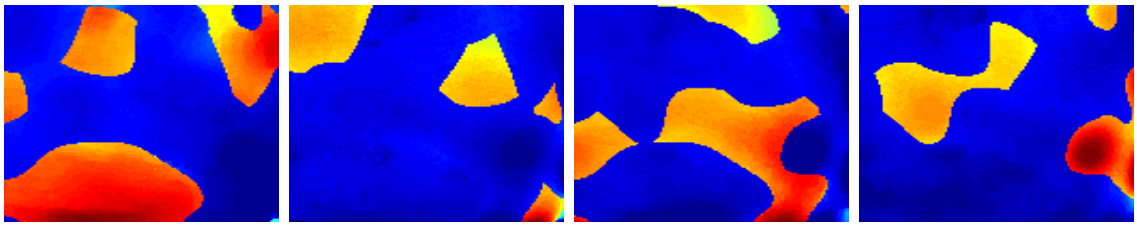
(b) *Abundance maps estimated by CLSR*



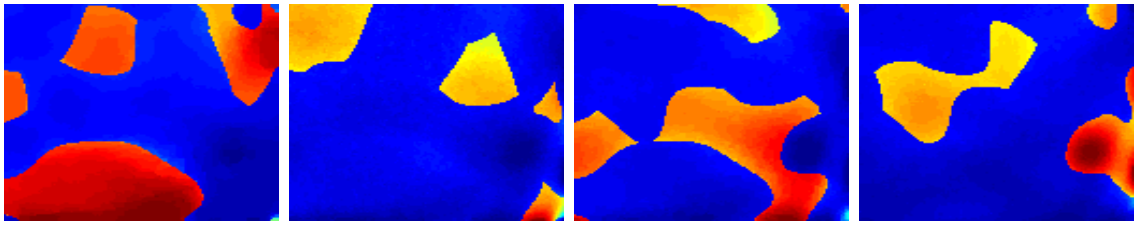
(c) *Abundance maps estimated by SRTV*



(d) *Abundance maps estimated by SBJS*

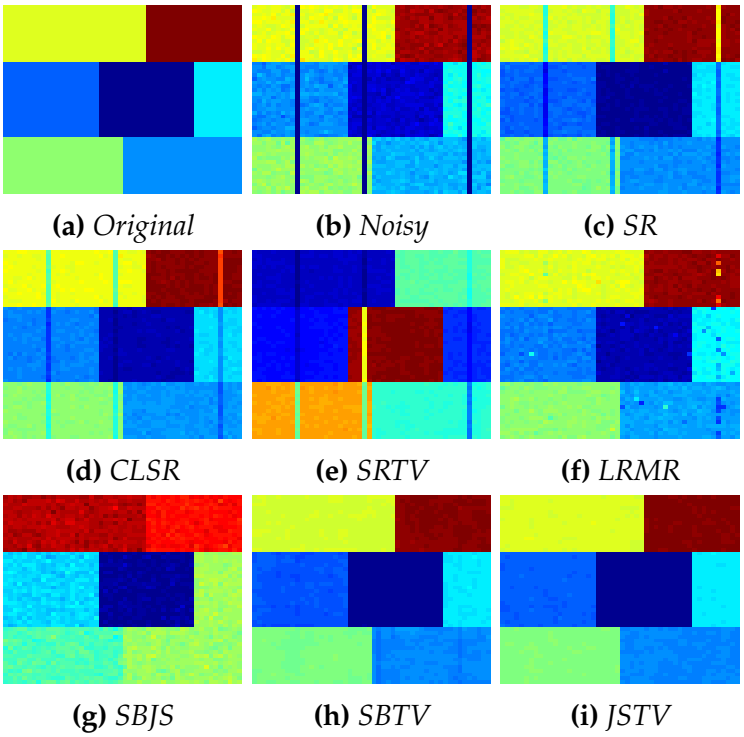


(e) *Abundance maps estimated by SBTV*



(f) *Abundance maps estimated by JSTV*

Figure 5.4: *Reconstructed Abundance maps by different algorithms*



(a) *Original* **(b)** *Noisy* **(c)** *SR*

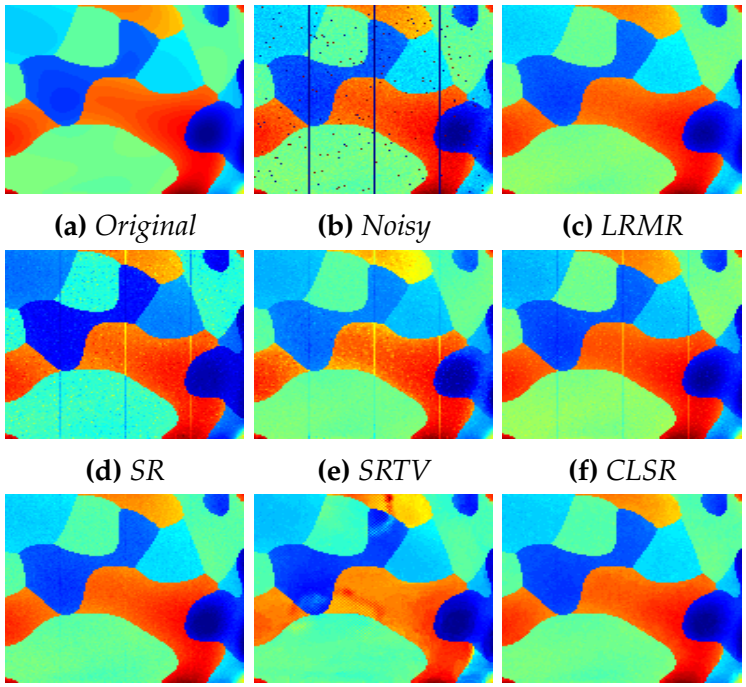
(d) *CLSR* **(e)** *SRTV* **(f)** *LRMR*

(g) *SBJS* **(h)** *SBTV* **(i)** *JSTV*

Figure 5.5: *Denoising results by different algorithms on synthetic image*

The third set of experiments was carried out to check denoising capability of proposed algorithm. The denoising results have been compared with existing denoising algorithm LRMR [20] that also takes mixed noise into account. The sparse unmixing models in (5.4) and (5.5) results not only in abundance maps but can also provide denoised hyperspectral image $\hat{X}$ when estimated abundance matrix $\hat{A}$ is multiplied by endmember matrix M i.e. $\hat{X} = M\hat{A}$ where $\hat{X}$ is estimated image. The denoising results are shown in Fig. 5.5. The synthetically generated abundance maps were multiplied with five randomly selected endmembers to generate $50 \times 50 \times 224$ synthetic hyperspectral image whose band 1 is shown in Fig. 5.5(a). Figures 5.5(g), 5.5(h), and 5.5(i) clearly shows the advantage of utilizing the concept of sparse noise in the unmixing framework. Since this image has a lot of smooth regions therefore Gaussian noise can be easily spotted in denoised images however compared to other algorithms the proposed JSTV algorithm has significantly reduced both kinds of noise. The denoising results with the second synthetic image are shown in Fig. 5.6. The noisy image of Fig. 5.6(b) was generated with mixed noise consisting of Gaussian noise of SNR 30 dB, 1% impulse noise, and three vertical lines. The denoising results are quantified using PSNR and SSIM values that are shown in Table 5.3. The denoised results produced by SBJS (see Fig. 5.6(g)), SBTV (see Fig. 5.6(h)), and JSTV (see Fig. 5.6(i)) algorithms in the presence of mixed noise, are visually more clear compared to results produced by of SR (see Fig. 5.6(d)), SRTV (see Fig. 5.6(e)) and CLSR (see Fig. 5.6(f)) since these three algorithms does not explicitly account for the presence of sparse noise in the problem formulations.

The fourth set of experiments was carried out to check the evolution of the abun-



(a) *Original* **(b)** *Noisy* **(c)** *LRMR*

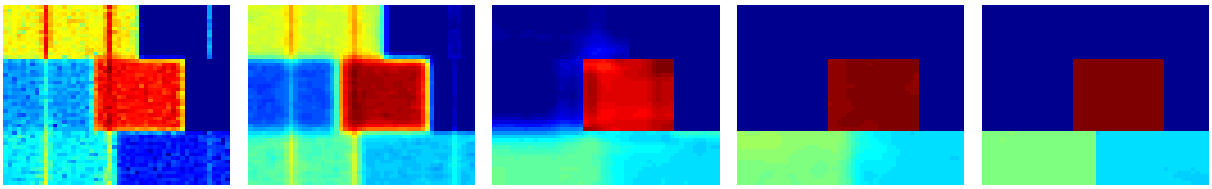
(d) *SR* **(e)** *SRTV* **(f)** *CLSR*

(g) *SBJS* **(h)** *SBTV* **(i)** *JSTV*

Figure 5.6: *Denoising results by different algorithms on synthetic image*

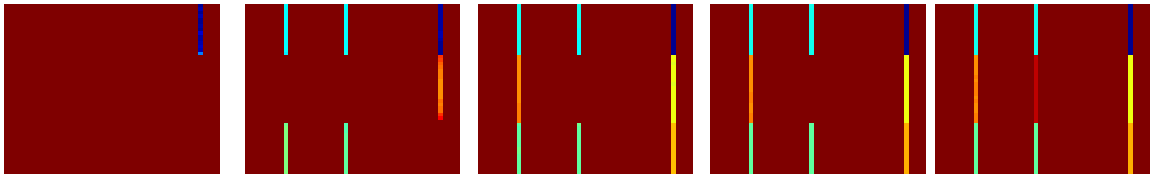
Table 5.3: *PSNR and SSIM values for denoised synthetic image 2*

	PSNR (dB)		SSIM	
Noise	LRMR	JSTV	LRMR	JSTV
40	51.12	49.39	0.99	0.99
30	42.01	43.38	0.99	0.99
20	31.57	32.36	0.97	0.98
Mix1	41.86	44.11	0.99	0.99
Mix2	41.41	40.90	0.99	0.99



2 iterations *5 iterations* *15 iterations* *30 iterations* *50 iterations*

Figure 5.7: *Effect of Iterations on Abundances*



5 iterations *10 iterations* *20 iterations* *30 iterations* *50 iterations*

Figure 5.8: *Noise estimation*

dance maps with iterations. Figure 5.7 shows one of the abundance maps after every few iterations. Since all abundance maps had the same behavior, therefore evolution of only one abundance map is shown. It can be observed that after two iterations, there are many endmembers visible in the image. After 15 iterations, relevant endmembers has been found and both kinds of noise have got reduced.

The fifth set of experiments shows the estimation of sparse noise. The proposed formulation in (5.8) estimates sparse noise term S because S is not known apriori. Figure 5.8 shows how line strips are estimated as part of sparse noise term as iterations increases. It shows that the inclusion of sparse noise term helps in handling unmixing problem in the presence of line strips.

The sixth set of experiments was carried out to check the convergence of proposed JSTV algorithm. The value of the objective function in (5.8) is plotted against the number of iterations. Figure 5.9 shows how objective function value decay to a locally optimal value. It can be deduced that fifty iterations are sufficient to reach the approximate solution; therefore, we can set maximum iterations to fifty.

The seventh set of experiments was carried out to compare time requirements of various algorithms. Public domain implementations of SR, CLSR, and SRTV available from [108] were utilized. Table 5.4 summarizes the time taken by various algorithms on the first synthetic image with Gaussian noise of 30 dB SNR and three vertical lines. Time was calculated on Intel Core i7 machine having 8 GB RAM with Linux operating system and Matlab2013a software. These time requirements shown in Table 5.4 are implementation dependent and may vary from machine to machine. Major time was consumed in approximating the least-square problem in each iteration and matrix-matrix multiplications. It can be noted that proposed JSTV can also be parallelized and applied on overlapping blocks simultaneously. Time reported is averaged over ten experiments.

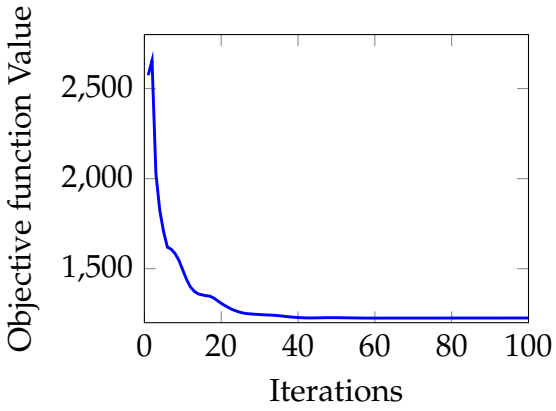


Figure 5.9: *Convergence of proposed JSTV algorithm*

Table 5.4: *Comparison of time requirements*

	SR	CLSR	SRTV	SBJS	SBTV	JSTV
Time (sec)	5.42	6.71	18.23	5.12	12.28	12.92

5.3.3 Evaluation Metric

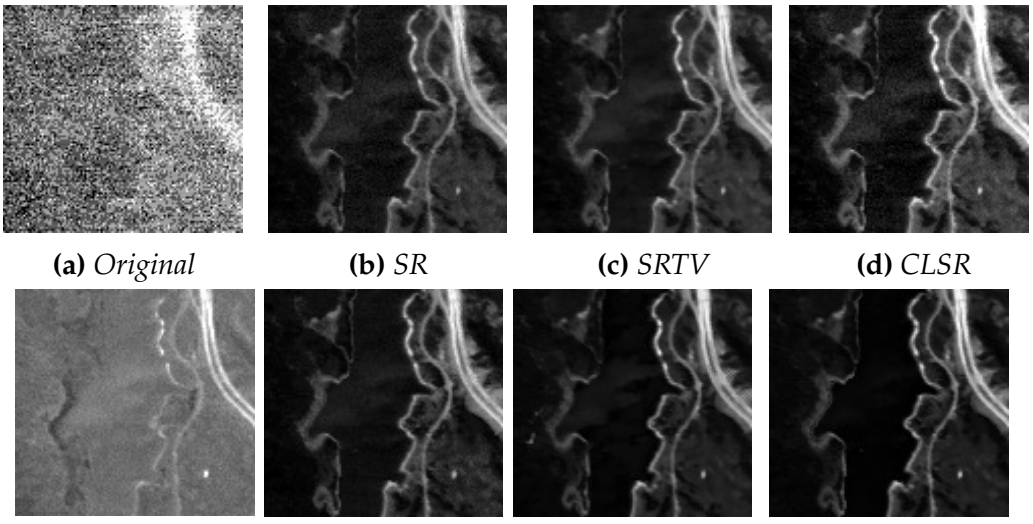
All the experimental results were quantified using two metrics namely PSNR and Structural Similarity Index (SSIM) [109]. PSNR between original image x and reconstructed image y was calculated as:

$$\text{PSNR} = 10\log_{10}\left(\frac{\max(x)^2}{\text{MSE}}\right).$$

The PSNR value for a hyperspectral image was calculated as the average of the sum of PSNR value for each band. The exact reconstruction will lead to the maximum value of PSNR as infinite. Higher the PSNR value better is the reconstruction quality. The SSIM [109] is based on luminance (l), contrast (c), and structure (s) terms, calculated as follows:

$$\begin{aligned}
\text{SSIM}(x,y) &= l(x,y)^{\alpha}\cdot c(x,y)^{\beta}\cdot s(x,y)^{\gamma}\\
l(x,y) &= \frac{2\mu_x\mu_y + c_1}{\mu_x^2+\mu_y^2+c_1}\\
c(x,y) &= \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2+\sigma_y^2+c_2}\\
s(x,y) &= \frac{2\sigma_{xy} + c_3}{\sigma_x\sigma_y+c_3}
\end{aligned}$$

where c_1, c_2, c_3 are constants and parameters α, β, γ were set to one. $\mu_x, \mu_y, \sigma_x, \sigma_y$ are mean and standard deviations for images x and y. σ_{xy} is cross-covariance between x, and y. SSIM value is normalized between zero and one where maximum value of one indicate exact reconstruction.



(a) *Original* **(b)** *SR* **(c)** *SRTV* **(d)** *CLSR*

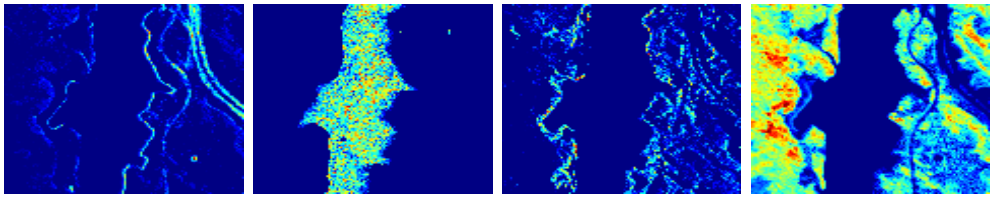
(e) *LRMR* **(f)** *SBJS* **(g)** *SBTV* **(h)** *JSTV*

Figure 5.10: *Denoising results on Jasper image obtained by different algorithms. Top row: Band 1 denoised by various algorithms. Bottom row: zoomed in portion of the same band marked with white square.*

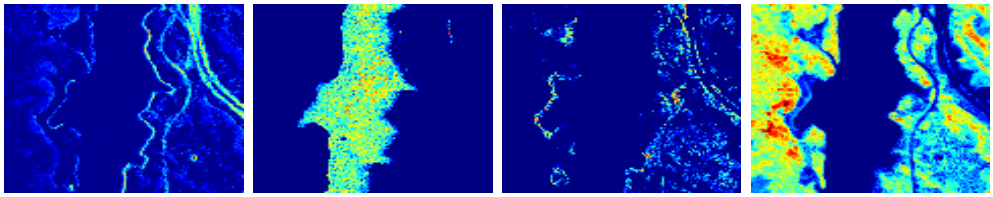
5.3.4 Real Data Experiments

The real data experiments were done on a portion of Jasper image as described in data description section. All 224 bands were considered during experiments. The four major abundances maps estimated by various algorithms are shown in Fig. 5.11. Visually we can compare the quality of estimating these abundances with the original image shown in Fig. 5.2. We can observe from Fig. 5.11(a) and 5.11(b) that SR and CLSR algorithms resulted in noisy abundance map corresponding to water endmember. Similarly, the abundance map for road related pixels also shows some other materials. The proposed JSTV method has resulted in comparatively cleaner abundance maps as shown in Fig 5.11(f).

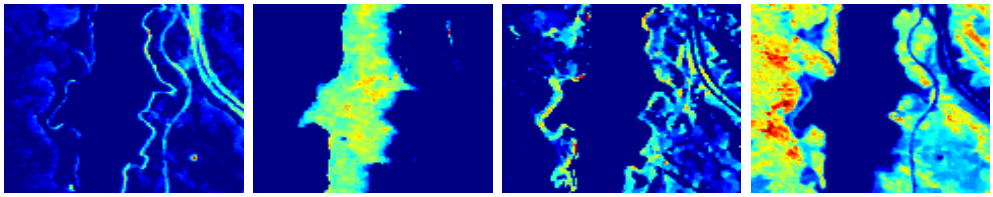
We have also compared denoising results by various algorithm since Jasper image have some noisy bands. Figure 5.10 shows bands 1 of Jasper image as well as its zoom portion over a small rectangular area. It can be observed from zoomed portions that proposed JSTV algorithm resulted in comparatively cleaner image.



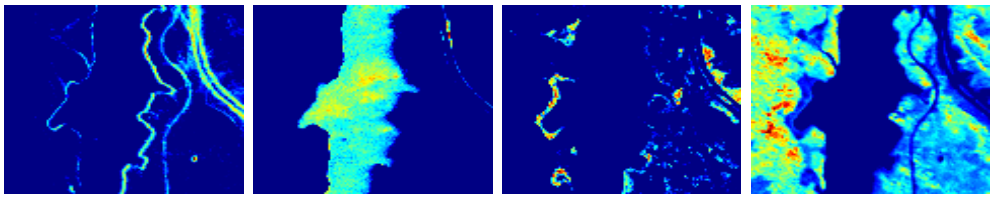
(a) *Abundance maps estimated by SR*



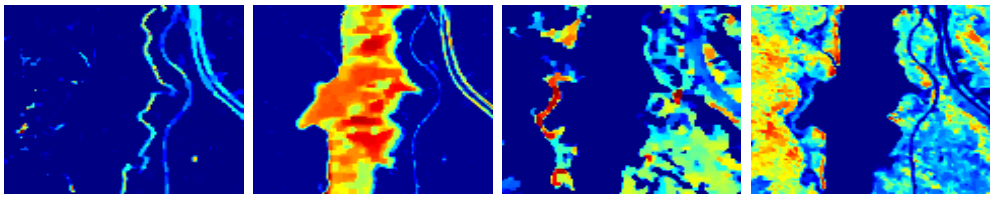
(b) *Abundance maps estimated by CLSR*



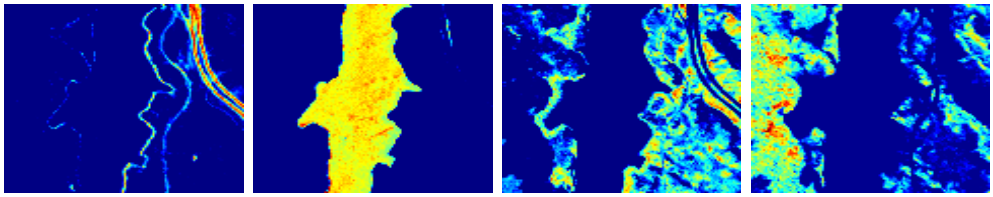
(c) *Abundance maps estimated by SRTV*



(d) *Abundance maps estimated by SBJS*



(e) *Abundance maps estimated by SBTV*



(f) *Abundance maps estimated by JSTV*

Figure 5.11: *Four major abundance maps estimated by different algorithms. Left to right : abundance maps corresponding to road, water, soil, and vegetation.*

Chapter 6

Hyperspectral Classification

This chapter discusses hyperspectral classification problem. First, a brief literature review is discussed followed by proposed greedy deep dictionary learning architecture and experimental results.

6.1 Literature Review

The problem of hyperspectral image classification is well known in the remote sensing research community. In the last half a decade, especially after the technology giants adopted deep learning, there have been a plethora of work on this topic - applications range from bio-metrics and computer vision to speech analysis, and from denoising to game learning, to say a few. However, the application of deep learning to the problem of hyperspectral image classification had been limited.

The concept of deep learning had been known for long. But it has only been in the last decade, after the advent of GPUs in everyday computing, deep learning became practical. Around the same time (2006) when deep learning started to gain momentum, another topic called dictionary learning also started gaining popularity. Like deep learning, the concept of dictionary learning was not new, but the seminal work on K-SVD popularized the technique. Ever since, plethora of dictionary learning papers have been published both for solving inverse problems and for prediction problems. Dictionary learning, although better understood conceptually (compared to deep learning), is mostly used in academic circles; it has not gained as much popularity in the industry so far.

Both deep learning and dictionary learning fall under the broader category of representation learning. Representation learning means automated feature extraction. Classical feature extraction techniques were either based on statistical models (Principal Component Analysis, Linear Discriminant Analysis etc.) or were hand-crafted (Haralick features, Scale Invariant Feature Transform, Local Binary Pattern etc.). Classical feature extraction techniques were 'designed' in the sense that it is based on some assumption / model made by the researcher / engineer regarding the nature of the data. On the other hand representation learning, 'learns' the model by itself, given the training data.

The concept of greedy deep dictionary learning was introduced in a recent unpub-

lished work [110]. Deep learning has been interpreted so far in terms of neural networks. Dictionary learning, on the other hand, is expressed in terms of basis learning for data representation. This work shows how dictionary learning can be interpreted as a neural network; and using the basic block of dictionary learning, we show how multiple layers can be concatenated to build a deep architecture.

Broadly speaking there are three pillars of deep learning -Convolutional Neural Net (CNN), Deep Belief Network (DBN) and Stacked Autoencoder (SAE); all of them have been used for classification problems in hyperspectral image classification. SAE was introduced for this problem in [111] by Chen et al; the same researchers applied DBN to the problem in the paper [112]. There are few papers that use CNN for the said problem [113, 114]. A limitation of CNN based methods is that they require large amount of data for training and yields poor results when the training data is limited.

Deep dictionary learning is a new concept. In this work, we compare our method with DBN and SAE and show that for limited amount of training data (practical scenario), our method can yields better results.

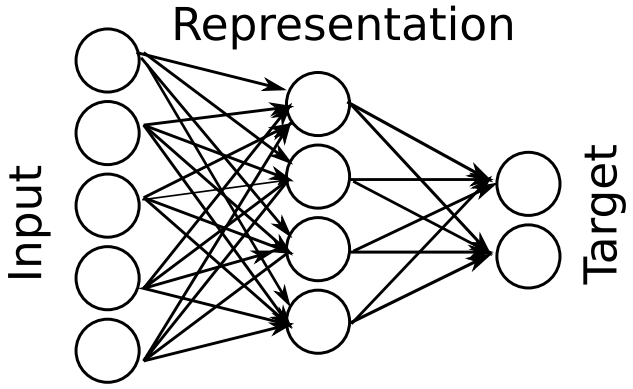


Figure 6.1: *Single layer NN*

Figure 6.1 shows the diagram of a simple neural network with one representation (hidden) layer. The problem is to learn the network weights between the input and the representation and between the representation and the target. This can be thought of as a segregated problem shown in Fig. 6.2.

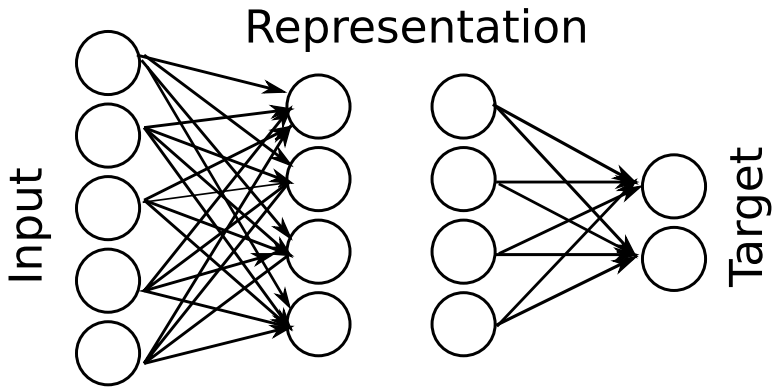


Figure 6.2: *Segregating the Neural Network*

Learning the mapping between the representation and the target is straightforward. The challenge is to learn the network weights (from input) and the representation. This is the study of representation learning.

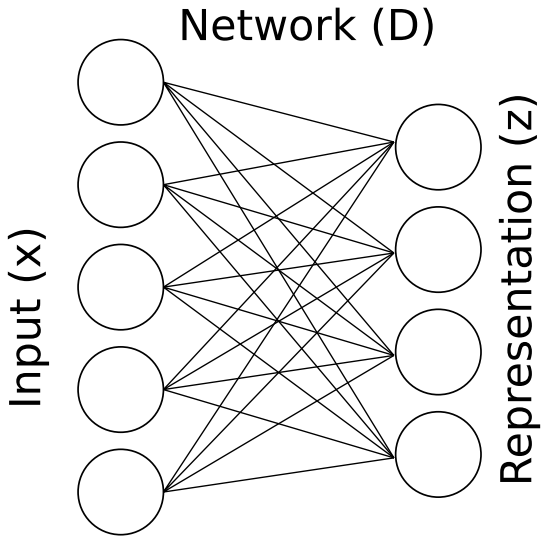


Figure 6.3: *Restricted Boltzmann Machine*

Restricted Boltzmann Machine (RBM) is a technique to learn the representation layer. The basic architecture is shown in Fig. 6.3. Broadly speaking, RBM learning is based upon maximizing the similarity between the projection of the data and the representation, subject to the usual constraints of probability. Loosely speaking, multiple layers of RBM are stacked to form a DBN. RBM has a probabilistic formulation. The other prevalent technique to train the representation layer of a neural network is by autoencoder. The architecture is described in Fig. 6.4.

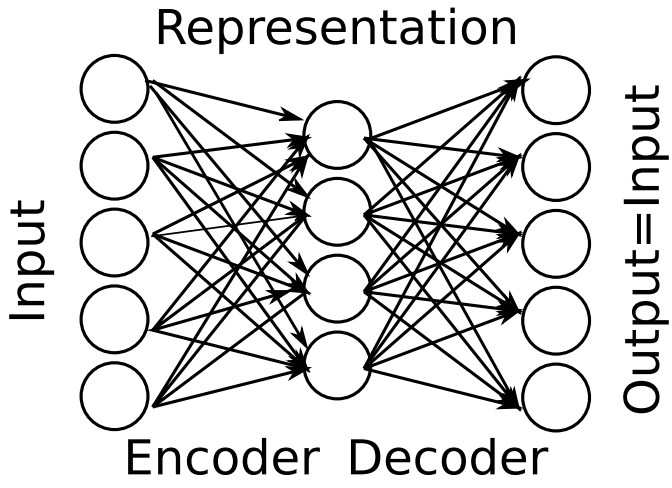


Figure 6.4: *Autoencoder*

$$\min_{W,W'} \left\| X - W'\phi(WX) \right\|_F^2 \tag{6.1}$$

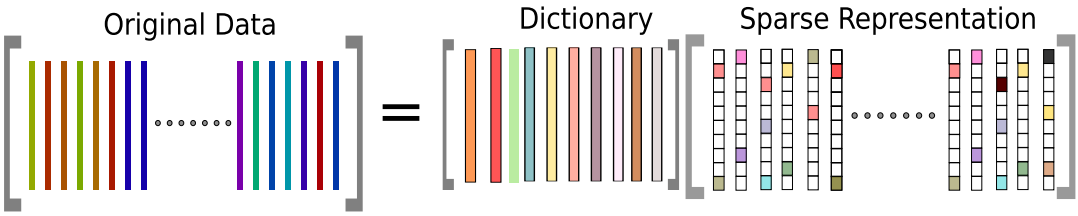


Figure 6.5: *Dictionary learning*

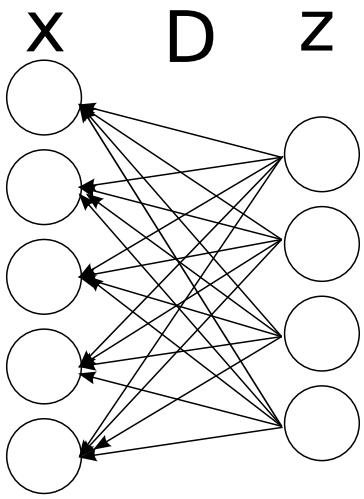


Figure 6.6: *Neural Network type interpretation*

The cost function for the autoencoder is expressed above. W is the encoder, and W' is the decoder. The autoencoder learns the encoder and decoder weights such that the reconstruction error is minimized. Essentially it learns the weights so that the representation retains all the information of the data, so that it can be reconstructed back. Once the autoencoder is learned, the decoder portion of the autoencoder is removed and the target is attached after the representation layer. Multiple units of such autoencoders are nested inside the other to form SAE.

The interpretation for dictionary learning is different. It learns a basis (D) for getting a representation (Z) of the data (X) as shown in Fig. 6.5. The columns of D are called 'atoms'. In this work, we look at dictionary learning in a different manner. Instead of interpreting the columns as atoms, we can think of them as connections between the input and the representation layer.

Unlike a neural network which is directed from the input to the representation, the dictionary learning kind of network points in the other direction – from representation to the input. Dictionary learning employs an Euclidean cost function Eq. (6.2), given by

$$\min_{D,Z} \|X - DZ\|_F^2 \,. \tag{6.2}$$

This is easily solved using alternating minimization. In every iteration, the first step is to update the coefficients assuming D is fixed and the next step is to update the dictionary assuming the coefficients are fixed. This alternating update of dictionary

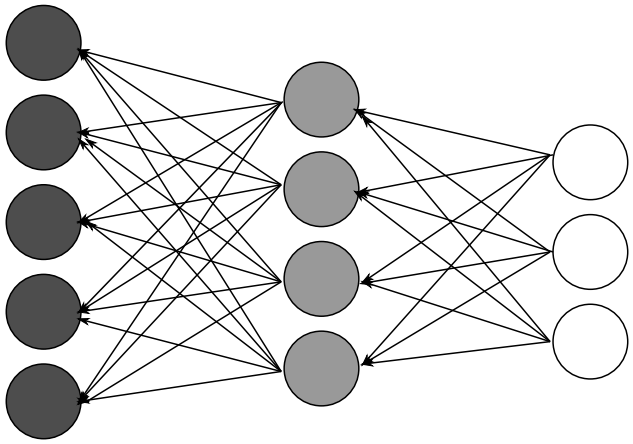


Figure 6.7: *Deep dictionary learning*

and coefficients continue till the algorithm converges to some local minima. Today most studies (following K-SVD [115]) impose an additional sparsity constraint on the representation (Z), but it is not mandatory.

6.2 Greedy Deep Dictionary Learning

We have established the connection between dictionary learning and neural network kind of representation learning. Building on that, we propose deeper architecture with dictionary learning. An example of two layer architecture is shown in Fig. 6.7. For the first layer, a dictionary is learned to represent the data. In the second layer, the representation from the first layer acts as input; it learns a second dictionary to represent the features from first level. This concept can be extended to deeper layers. In deep learning, instead of learning the full deep architecture in one go, the parts are usually learned in a greedy fashion [116]. There are two advantages of greedy learning. First, learning the basic blocks (RBM or autoencoder) are relatively simple and easy to implement. Second, deep architectures, having a large number of parameters are difficult to learn from limited training data. Breaking the problem into smaller unit erases the issue of over-fitting. In this work we follow the same greedy principle for our deep dictionary learning problem. The formulation for learning N levels of dictionaries is given by,

$$X = D_1(\phi(D_2\phi(...\phi(D_N Z)))) \tag{6.3}$$

Ideally, we would have to solve the following problem.

$$\min_{D_1,...,D_N,Z} \left\| X - D_1(\phi(D_2\phi(...\phi(D_N Z)))) \right\|_F^2 + \mu \|Z\|_1 \tag{6.4}$$

Following the standard assumption in dictionary learning, we incorporate a sparsity penalty on the coefficients. Solving Eq. (6.4) is very difficult. It is a highly non-convex problem with a large number of parameters to learn. Therefore, it suffers both from the problem of getting stuck in a local minima and from over-fitting.

As mentioned before, we solve it via the greedy approach, i.e. for the first layer, express $Z_1 = \phi(D_2\phi(...\phi(D_N X)))$ so that the problem in Eq. (6.3) can be recast as,

$$X = D_1 Z_1 \tag{6.5}$$

The coefficients in this layer are not supposed to be sparse, hence can be solved using Eq. (6.2). Once the coefficients for the first layer is learned, one can learn the second layer as a single layer of dictionary learning,

$$\phi^{-1}(Z_1) = D_2 Z_2 \tag{6.6}$$

where $Z_3 = \phi(D_3\phi(...\phi(D_N X)))$. This problem can also be solved using Eq. (6.2). Continuing in this fashion till the penultimate layer, we get,

$$Z_{N-1} = \phi(D_N Z) \Rightarrow \phi^{-1}(Z_{N-1}) = D_N Z \tag{6.7}$$

In the final layer, the coefficients are supposed to be sparse. Therefore the optimization problem is,

$$\min_{D_N, Z} \left\| \phi^{-1}(Z_{N-1} - D_N Z) \right\|_F^2 + \mu \|Z\|_1 \tag{6.8}$$

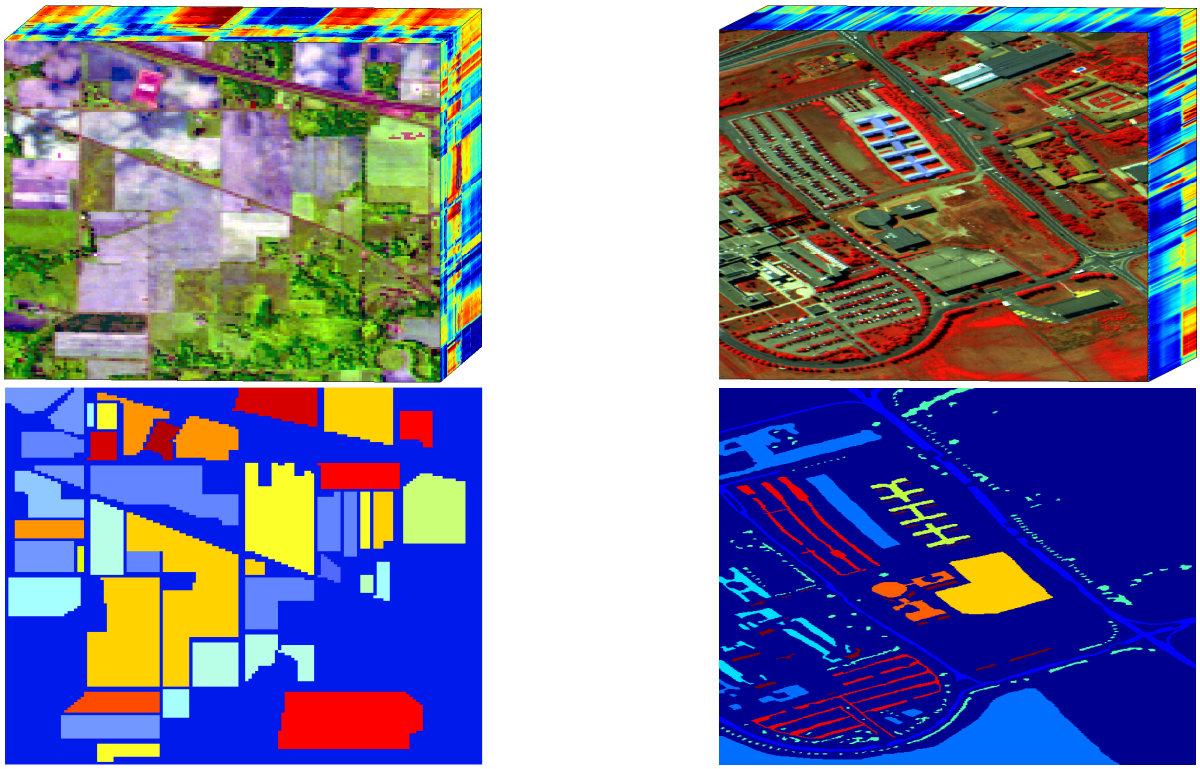
This is easily solved using simple alternating minimization [117].

The main advantage of our approach is that the problem of matrix factorization - the basic building block of dictionary learning is relatively well understood theoretically (compared to RBM or autoencoder). In fact there are mathematical convergence guarantees [118–121] on alternating minimization based dictionary learning (the technique used here to solve the sub-problems). Therefore our complete approach is bound to converge. Usually in deep learning there is a fine tuning stage where the greedily pre-trained layers are updated after concatenation. The objective of fine-tuning is to propagate the effects of subsequent layers to previous layers. In this work, we do not fine tune the deep dictionary learning architecture. There is evidence from practical applications in deep learning, such as FaceNet [122], that even without fine-tuning extremely good results can be obtained from greedily pre-trained models.

6.3 Experiments

Two hyperspectral image datasets were utilized in the experiments. The IndianPines dataset has 200 spectral reflectance bands after removing the bands covering the region of water absorption and 145×145 pixels of sixteen categories. The IndianPines dataset is shown in Fig. 6.8(a) along with the ground truth. The second dataset of Pavia University scene has 103 bands of with spatial resolution of 340×610 pixels of nine categories. This dataset is shown in Fig. 6.8(b). The background i.e. Class 0 from both the datasets was not considered in the experiments.

Prior studies on deep learning based classification assumed an overtly optimistic scenario [2, 3] - they assumed 80% (60% training + 20% validation) labelled data is



(a) *Indian Pines* **(b)** *Pavia University*

Figure 6.8: *Datasets used in Experiments along with the ground-truth*

available; and only 20% need to be predicted. In this work, we consider a more realistic scenario; we assume only 10% of the labelled data is available for IndianPines and only 2% for the PaviaU dataset. Tables 6.1 and 6.2 shows number of training samples used in training and testing respectively for IndianPines and PaviaU datasets. Input consists of raw data of all the spectral channels pixel-wise.

For each layer, we initialized the dictionary atoms by randomly choosing samples from the training set for dictionary learning. Also, we employ non-linear transformation in the form of the *tanh* function in each layer to generate robust features. We have a three layer deep architecture of 150-100-30 for Indian Pines and 3 layer deep architecture of 80-40-20 for Pavia University. The features learned from the last level were used to train a Neural Net Classifier.

The results were compared with Stacked Autoencoder (SAE) and Deep Belief Network(DBN). The implementation for the SAE has been obtained from [2] and for DBN from[3] with the DeeBNet toolbox V2.2 [14]. The architectures proposed in [2, 3] are used here. Tables 6.3 and 6.4 quantifies the classification results using class-wise accuracy, overall accuracy (OA), average accuracy (AA), and Kappa coefficient for both the datasets.

Figure 6.9 visually compares reconstruction quality of the proposed technique with SAE and DBN in the case of 10% training data of IndianPines image. Figure 6.10 shows results on PaviaU dataset with 2% training data. It can be observed that proposed

Class	Training Samples	Testing Samples	Total Samples
1	15	31	46
2	142	1286	1428
3	83	747	830
4	23	214	237
5	48	435	483
6	73	657	730
7	20	8	28
8	47	431	478
9	15	5	20
10	97	875	972
11	160	2295	2455
12	59	534	593
13	20	185	205
14	126	1139	1265
15	38	348	386
16	50	43	93

Table 6.1: *Number of training and test samples of IndianPines*

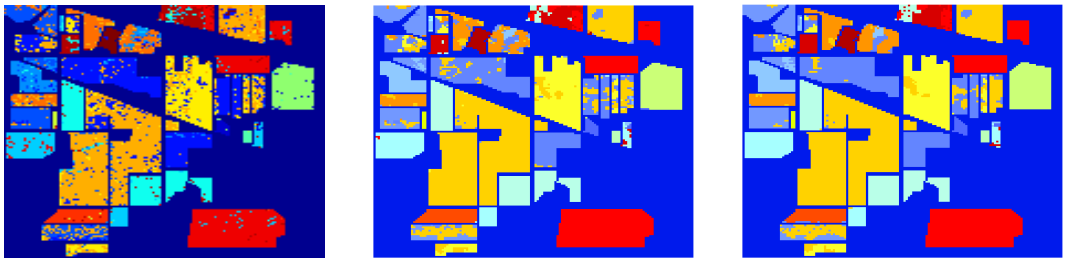
Class	Training Samples	Testing Samples	Total Samples
1	132	6499	6631
2	372	18277	18649
3	41	2058	2099
4	61	3003	3064
5	26	1319	1345
6	100	4929	5029
7	26	1304	1330
8	73	3609	3682
9	18	929	947

Table 6.2: *Number of training and test samples of PaviaU dataset*

technique both quantitatively and qualitatively produce better classification results.

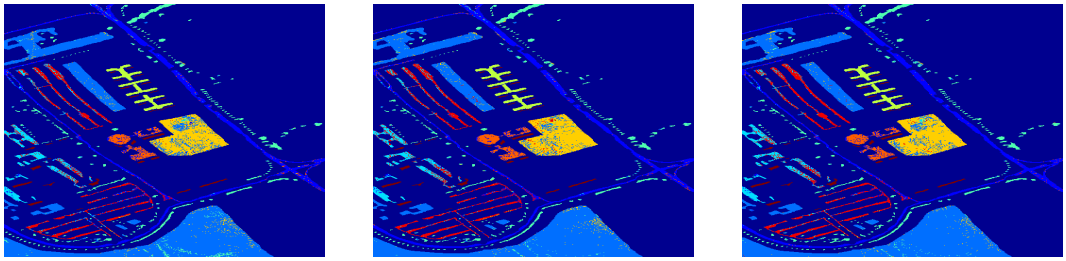
Class	SAE	DBN	Proposed
1	90.32	87.09	93.54
2	84.52	76.74	88.02
3	67.20	68.40	70.54
4	77.00	98.93	100.0
5	83.90	94.25	95.86
6	94.67	100.0	99.84
7	87.50	100.0	100.0
8	94.19	98.37	100.0
9	60.00	100.0	80.00
10	67.20	81.25	84.11
11	78.32	97.23	98.91
12	62.73	64.98	89.51
13	94.05	100.0	93.51
14	93.50	100.0	99.03
15	56.89	54.59	75.28
16	93.02	100.00	100.00
OA	79.98	87.47	92.06
AA	80.32	88.87	91.76
Kappa	0.771	0.855	0.909

Table 6.3: *Classification results on IndianPines dataset*



(a) *SAE* **(b)** *DBN* **(c)** *Proposed*

Figure 6.9: *Classification maps generated by different algorithms on IndianPines dataset*



(a) *SAE* **(b)** *DBN* **(c)** *Proposed*

Figure 6.10: *Classification maps generated by different algorithms on PaviaU dataset*

Class	SAE	DBN	Proposed
1	93.26	93.30	94.30
2	95.42	96.64	97.18
3	74.82	71.82	72.43
4	92.10	89.80	92.82
5	99.54	99.00	99.21
6	75.97	81.88	80.47
7	71.62	78.07	73.73
8	74.59	79.84	83.90
9	99.89	100	100
OA	89.25	90.81	91.5
AA	86.36	87.82	88.23
Kappa	0.857	0.878	0.887

Table 6.4: *Classification results on PaviaU dataset*

Chapter 7

Conclusions

This thesis focused on four related problems in multi-dimensional imaging namely multispectral demosaicing, hyperspectral denoising, hyperspectral unmixing, and hyperspectral classification. These are classical imaging problems that have been addressed in literature from different perspectives. This work has attempted to model them as compressed sensing and sparse recovery related problems.

The multispectral demosaicing problem is an extension of the color demosaicing problem that occurs in design of low cost and small size multi-spectral cameras. As there is no standard multispectral filter array to capture multiple bands, therefore, we proposed a uniform multispectral filter array such that minimum changes are required in the design of color imaging cameras to capture multispectral images. The focus of this work on multispectral imaging problem was on the reconstruction algorithm to create the full image from the raw image. When five bands are captured using a single-sensor then there are only 20% samples of each band in the raw image; therefore, reconstruction of the full image become very challenging. The single sensor based design is dependent on cheap silicon technology that can be used to capture images in the wavelength range of around 1100 nm.

Quantitative results proved that reconstruction quality of proposed multispectral demosaicing approach is better than the already existing algorithm used in the comparative study. Currently, experiments have been performed with three to six-band multispectral images and more experiments need to be carried out by increasing the number of bands and by varying spectral gap. Once offline training is completed, then the proposed algorithm is linear-time because every pixel is visited only once for doing interpolation. It is desirable to extend the current technique such that it does not depend on training data. This technique is limited to hand-held multispectral cameras and not readily extendable for push-broom or whisk-broom kind of multispectral imaging sensors. Further, there are pseudo-random and panchromatic filter array designs to capture multispectral image therefore, a comparative study with uniform filter array can be carried out.

The second problem discussed in the thesis is the hyperspectral denoising problem. Hyperspectral denoising is an important pre-processing step in many applications of these images such as classification, change detection, data fusion, etc.. Initially, we addressed the problem of reducing impulse noise followed by mixed noise reduction problem. There have been studies to remove impulse noise from grayscale images

and also Gaussian noise from hyperspectral images but little work on reducing impulse noise from hyperspectral images. The problem has been formulated both as a synthesis prior and an analysis prior form. Since impulse noise corrupt few pixels in the image, therefore, this noise has been modeled as sparse noise. Thus, our proposed formulation leads to ℓ_1-norm regularized ℓ_1-norm data fidelity minimization problem. We use two dictionaries to de-correlate hyperspectral datacube in both spatial and spectral dimensions that result in a very sparse representation.

The problem formulation considers a general noise model that explicitly accounts for not only Gaussian noise but also sparse noise. The inherent structure of hyperspectral images has been explored in the analysis prior case by utilizing $2D$ total-variation along spatial dimension and 1D total-variation along the spectral dimension. The synthesis prior approach utilized DCT along both spatial and spectral dimension. The denoising problem was formulated as an optimization problem whose solution has been derived using split-Bregman approach. Experiments were carried out using synthetic as well as real noise hyperspectral images. Synthetic noise experiments were performed to quantify the denoising strength of proposed technique using PSNR and SSIM. The quantitative and qualitative results demonstrate that proposed algorithm reduce a significant amount of noise from real noisy hyperspectral images compared to existing state of the art approaches.

The proposed denoising algorithm is applied to all the bands. This work did not look at the problem of identifying noise corrupted bands and applying denoising algorithm on the selected number of bands. This work assumed the impulse noise as additive noise and did not try to recognize the impulse noise corrupted pixels explicitly as done in some existing algorithms.

The third problem considered in this thesis is the unmixing problem, we have proposed a new approach for hyperspectral unmixing. This approach does not depend on the pure pixel assumption. We also relaxed the abundance sum-to-one constraint. This method exploits joint sparsity as well as the piecewise smoothness of abundance maps in the generic noise model which explicitly account for sparse noise. Experimental results suggest the advantage of proposed method over existing methods. Simultaneous utilization of both total-variation regularization and joint-sparse regularization is not redundant as both achieve different goals. Total variation regularization has explored smoothness of abundance maps whereas joint-sparsity exploit the fact that an endmember if present, shall be present at various locations in the same area.

This work utilized existing USGS spectral library for spectral signatures. The spectral signatures in the existing library can differ from the spectral signatures present in the image. Also, it is possible that real images have endmembers whose spectral signatures are not present in existing libraries; therefore, it is desirable to derive the endmember signatures directly from the hyperspectral image.

A new paradigm for deep learning called 'deep dictionary learning' was introduced for hyperspectral classification. The learning proceeds in a greedy fashion. In the first stage, a dictionary and the corresponding representation are learned from the input training samples. In the second layer, the representation from the first layer acts

as an input, and the second level of dictionary and representation are learned. The same process is continued for training deeper levels. The advantage of the greedy approach is twofold. First, it prevents over-fitting, since we do not need to train all the layers simultaneously. Second, the alternating minimization technique used for learning each layer of training enjoys certain convergence guarantees. In this preliminary work, the proposed technique was compared with two standard deep learning tools - deep belief network and stacked autoencoder. In the future, we will try to improve further by pre-processing the input. Our proposed method is unsupervised. There is a plethora of work in supervised dictionary learning for computer vision problems. In future, we would like to extend our work on robust dictionary learning to incorporate supervised dictionary learning penalties.

Source codes for multispectral demosaicing [123], hyperspectral denoising [124], and unmixing [125] are made available to promote reproducible research.

Publications

Journal Articles

1. **H.K. Aggarwal**, A. Majumdar. Hyperspectral Unmixing in the Presence of Mixed Noise using Joint-Sparsity and Total-Variation. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2016.

2. **H.K. Aggarwal**, A. Majumdar. Hyperspectral Image Denoising using Spatio-Spectral Total Variation. in IEEE Geoscience and Remote Sensing Letters, 13(3), 442-446,2016.

3. **H.K. Aggarwal**, A. Majumdar. Exploiting Spatio-Spectral Correlation for Impulse Denoising in Hyperspectral Images. in SPIE Journal of Electronic Imaging, 24(1), 013027, 2015.

4. S. Tariyal, **H.K. Aggarwal**, A. Majumdar. Removing Sparse Noise from Hyperspectral Images with Sparse and Low-rank Penalties. in SPIE Journal of Electronic Imaging Letters, 24(1), 013027, 2016.

5. A. Majumdar, N. Ansari, **H.K. Aggarwal**, P. Biyani. Impulse Denoising for Hyperspectral Images: A Blind Compressed Sensing Approach. Elsevier Signal Processing, 119, 136-141, 2016.

Conference Articles

1. **H.K. Aggarwal**, A. Majumdar. Sparse Filtering based Hyperspectral Unmixing, in IEEE Workshop on Hyperspectral Image and Signal Processing (**WHISPERS 2016**). Aug 21-24, 2016. Los Angeles, USA. (Accepted)

2. S. Tariyal, **H.K. Aggarwal**, A. Majumdar. Greedy Deep Dictionary Learning for Hyperspectral Image Classification, IEEE Workshop on Hyperspectral Image and Sig Proc (**WHISPERS 2016**). Aug 21-24, 2016. Los Angeles, USA. (Accepted)

3. **H.K. Aggarwal**, A. Majumdar. Compressive Hyper-Spectral Imaging in the Presence of Real Noise, IEEE International Geosci. and Remote Sensing Symposium (**IGARSS 2016**). July 10-15, 2016. Beijing, China.

4. **H.K. Aggarwal**, A. Majumdar. Robust Estimation for Subspace Based Classifiers, International Joint Conf. on Neural Network (**IJCNN 2016**). 24-29 July 2016. Vancouver, Canada.

5. **H.K. Aggarwal**, A. Majumdar. Blind Hyperspectral Denoising, National Conf. on Computer Vision, Pattern Recognition, Image Processing and Graphics (**NCVPRIPG 2015**). December 16-19, 2015. IIT-Patna, India.

6. **H.K. Aggarwal**, A. Majumdar. Mixed Gaussian and Impulse Denoising of Hyperspectral Images, in IEEE International Geoscience and Remote Sensing Symposium (**IGARSS 2015**). July 26-31, 2015. Milan, Italy.

7. **H.K. Aggarwal**, A. Majumdar. Blind Compressive Hyperspectral Imaging, in IEEE International Geoscience and Remote Sensing Symposium (**IGARSS 2015**). July 26-31, 2015. Milan, Italy.

8. A. Majumdar, N. Ansari, **H.K. Aggarwal**. Hyper-spectral Impulse Denoising: A row-sparse Blind Compressed Sensing Formulation, in IEEE International Conference on Acoustics, Speech, and Signal Processing (**ICASSP 2015**). April 19-24, 2015. Brisbane, Australia.

9. **H.K. Aggarwal**, A. Majumdar. Multi-spectral Demosaicing: A Joint-sparse Elastic-net Formulation, International Conf. on Advances in Pattern Recognition (**ICAPR 2015**). Jan 4-7, 2015. ISI Kolkata, India.

10. **H.K. Aggarwal**, A. Majumdar. Generalized Synthesis and Analysis Prior Algorithms with Application to Impulse Denoising, in Indian Conf. on Computer Vision, Graphics and Image Proc. (**ICVGIP 2014**). Dec. 14-17, 2014, IISc Bengaluru, India

11. **H.K. Aggarwal**, A. Majumdar. Single-Sensor Multi-Specral Image Demosaicing Algorithm Using Learned Interpolation Weights, in International Geoscience and Remote Sensing Symposium (**IGARSS 2014**). July 13-18, 2014, Quebec, Canada.

12. **H.K. Aggarwal**, A. Majumdar. Extension of Sparse Randomized Kaczmarz Method for Multiple Measurements, 22nd International Conf. on Pattern Recognition (**ICPR 2014**). Aug 24-28, 2014, Stockholm, Sweden.

13. **H.K. Aggarwal**, A. Majumdar. Compressive Sensing Multi-spectral Demosaicing from Single Sensor Architecture, in IEEE China Summit and Inter. Conf. on Signal and Info. Proces. (**ChinaSIP 2014**). July 9-13, 2014. Xi'an China.

14. A. Gogna, A. Shukla, **H.K. Aggarwal** and A. Majumdar, Split Bregman Algorithms for Sparse / Joint-sparse and Low-rank Signal Recovery: Application in Compressive Hyperspectral Imaging, in IEEE International Conference on Image Processing (**ICIP 2014**), October 27-30, 2014, Paris, France.

15. **H.K. Aggarwal**, A. Majumdar. Multi-Spectral Demosaicing Technique for Single-Sensor Imaging, in National Conf. on Comp. Vision, Pattern Recog., Image Proc. and Graphics (**NCVPRIPG 2013**). December 18-21, 2013. IIT-Jodhpur, India.

16. **H.K. Aggarwal**, A. Majumdar, R. Ward. Reconstruction Algorithm for Multi-Spectral Image Demosaicing, in 15th IASTED international conference on signal and image processing (**SIP 2013**). July 17-19, 2013. Banff, Canada.

References

[1] Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Multispectral Demosaicking Using Adaptive Kernel Upsampling. In *IEEE Int. Conf. Image Process.*, pages 3157–3160, 2011. ISBN 9781457713033.

[2] Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Multispectral Demosaicking Using Guided Filter. In *IS&T/SPIE Electron. Imaging*, volume 8299, pages 82990o—-82990o, jan 2012.

[3] Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K Nayar. Generalized Assorted Pixel Camera: Postcapture Control of Resolution, Dynamic Range, and Spectrum. *IEEE Trans. Image Process.*, 19(9):2241–53, sep 2010. ISSN 1941-0042. doi: 10.1109/TIP.2010.2046811.

[4] Miroslav Kubat, Robert C Holte, and Stan Matwin. Machine learning for the detection of oil spills in satellite radar images. *Mach. Learn.*, 30(2-3):195–215, 1998.

[5] Ira Leifer, William J Lehr, Debra Simecek-Beatty, Eliza Bradley, Roger Clark, and Others. State of the art satellite and airborne marine oil spill remote sensing: Application to the BP Deepwater Horizon oil spill. *Remote Sens. Environ.*, 124:185–209, 2012.

[6] Sajad Farokhi, Usman Ullah Sheikh, Jan Flusser, and Bo Yang. Near infrared face recognition using Zernike moments and Hermite kernels. *Inf. Sci.*, 316:234—-245, 2015.

[7] Jiayi Ma, Ji Zhao, Yong Ma, and Jinwen Tian. Non-rigid visible and infrared face registration via regularized Gaussian fields criterion. *Pattern Recognit.*, 48(3):772–784, 2015.

[8] Haitao Zhao and Shaoyuan Sun. Sparse tensor embedding based multispectral face recognition. *Pattern Recognit.*, 133:427–436, 2014.

[9] Juliane Bendig, Kang Yu, Helge Aasen, Andreas Bolten, Simon Bennertz, Janin Broscheit, Martin L Gnyp, and Georg Bareth. Combining UAV-based plant height from crop surface models, visible, and near infrared vegetation indices for biomass monitoring in barley. *Int. J. Appl. Earth Obs. Geoinf.*, 39:79–87, 2015.

[10] Bingfang Wu, Jihua Meng, Qiangzi Li, Nana Yan, Xin Du, and Miao Zhang. Remote sensing-based global crop monitoring: experiences with China's CropWatch system. *Int. J. Digit. Earth*, 7(2):113—-137, 2014.

[11] Mahdieh Hosseinjani Zadeh, Majid H Tangestani, Francisco Velasco Roldan, and Inaki Yusta. Mineral exploration and alteration zone mapping using mixture tuned matched filtering approach on ASTER data at the central part of Dehaj-Sarduiyeh Copper Belt, SE Iran. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 7(1):284—-289, 2014.

[12] Calina Mihaela Antonina, Parascab Sorin Viorel, Dan Savastrua, and Dragos Maneaa. Hyperspectral Imaging in the Medical Field: Present and Future. *Appl. Spectrosc. Rev.*, 49 (6):435–447, 2014.

[13] Harini Nagendra, Richard Lucasb, João Pradinho Honradoc, Rob H.G. Jongmand, Cristina Tarantinoe, Maria Adamoe, and Paola Mairotaf. Remote sensing for conservation monitoring: Assessing protected areas, habitat extent, habitat condition, species diversity, and threats. *Ecol. Indic.*, 33:45—-59, 2013.

[14] Bryce E Bayer. Color imaging array, July 20 1976. US Patent 3,971,065.

[15] Bahadir K Gunturk, John Glotzbach, Yucel Altunbasak, Ronald W Schafer, and Russel M Mersereau. Demosaicking : Color Filter Array Interpolation. *IEEE Signal Process. Mag.*, 22(1):44–54, 2005.

[16] H.S. Malvar and R. Cutler. High-quality linear interpolation for demosaicing of Bayer-patterned color images. In *IEEE Int. Conf. Acoust. Speech, Signal Process.*, volume 3, pages 485–488. Ieee, 2004. ISBN 0-7803-8484-9.

[17] Lidan Miao and Hairong Qi. The design and evaluation of a generic method for generating mosaicked multispectral filter arrays. *IEEE Trans. Image Process.*, 15(9):2780–91, sep 2006. ISSN 1057-7149.

[18] Lidan Miao, Hairong Qi, Rajeev Ramanath, and Wesley E Snyder. Binary tree-based generic demosaicking algorithm for multispectral filter arrays. *IEEE Trans. Image Process.*, 15(11):3550–8, nov 2006. ISSN 1057-7149.

[19] Johannes Brauers and Til Aach. A Color Filter Array Based Multispectral Camera. In *Work. Farbbildverarbeitung*, 2006.

[20] Hongyan Zhang, Wei He, Liangpei Zhang, Huanfeng Shen, and Qiangqiang Yuan. Hyperspectral Image Restoration Using Low-Rank Matrix Recovery. *IEEE Trans. Geosci. Remote Sens.*, 52(8):4729—-4743, 2014.

[21] Peng Liu, Fang Huang, Guoqing Li, and Zhiwen Liu. Remote-Sensing Image Denoising Using Partial Differential Equations and Auxiliary Images as Priors. *IEEE Geosci. Remote Sens. Lett.*, 9(3):358–362, may 2012. ISSN 1545-598X. doi: 10.1109/LGRS.2011.2168598.

[22] Xuefeng Liu, Salah Bourennane, and Caroline Fossati. Nonwhite Noise Reduction in Hyperspectral Images. *IEEE Geosci. Remote Sens. Lett.*, 9(3):368–372, may 2012.

[23] Guangyi Chen and Shen-En Qian. Denoising of Hyperspectral Imagery Using Principal Component Analysis and Wavelet Shrinkage. *IEEE Trans. Geosci. Remote Sens.*, 49(3): 973–980, 2011.

[24] Xuefeng Liu, Salah Bourennane, and Caroline Fossati. Denoising of Hyperspectral Images Using the PARAFAC Model and Statistical Performance Analysis. *IEEE Trans. Geosci. Remote Sens.*, 50(10):3717–3724, oct 2012. ISSN 0196-2892. doi: 10.1109/TGRS.2012.2187063.

[25] Jin Xu, Wei Wang, Jinghuai Gao, and Wenchao Chen. Monochromatic Noise Removal via Sparsity-Enabled Signal Decomposition Method. *IEEE Geosci. Remote Sens. Lett.*, 10 (3):533–537, may 2013. ISSN 1545-598X. doi: 10.1109/LGRS.2012.2212271.

[26] Daniele Cerra, M Rupert, and Peter Reinartz. Noise Reduction in Hyperspectral Images Through Spectral Unmixing. *IEEE Geosci. Remote Sens. Lett.*, 11(1):109–113, 2014.

[27] Ming Yan. Restoration of Images Corrupted by Impulse Noise and Mixed Gaussian Impulse Noise Using Blind Inpainting. *SIAM J. Imaging Sci.*, 6(3):1227–1245, 2013.

[28] Yu Xiao, Tieyong Zeng, Jian Yu, and Michael K. Ng. Restoration of images corrupted by mixed Gaussian-impulse noise via l1âĂŞl0 minimization. *Pattern Recognit.*, 44(8):1708–1720, aug 2011. doi: 10.1016/j.patcog.2011.02.002.

[29] Jun Liu, Xue-Cheng Tai, Haiyang Huang, and Zhongdan Huan. A weighted dictionary learning model for denoising images corrupted by mixed noise. *IEEE Trans. Image Process.*, 22(3):1108–20, mar 2013. ISSN 1941-0042. doi: 10.1109/TIP.2012.2227766.

[30] Junhwa Chi and M. M. Crawford. Spectral Unmixing-Based Crop Residue Estimation Using Hyperspectral Remote Sensing Data: A Case Study at Purdue University. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, PP(99):1–10, 2014.

[31] Martin Hedegaard, Christian Matthäus, Søren Hassing, Christoph Krafft, Max Diem, and Jürgen Popp. Spectral unmixing and clustering algorithms for assessment of single cells by Raman microscopic imaging. *Theor. Chem. Acc.*, 130(4-6):1249–1260, 2011.

[32] Inmaculada Dopido, Alberto Villa, Antonio Plaza, and Paolo Gamba. A Quantitative and Comparative Assessment of Unmixing-Based Feature Extraction Techniques for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 5(2):421–435, 2012. ISSN 1939-1404. doi: 10.1109/JSTARS.2011.2176721.

[33] Alp Ertürk. Enhanced Unmixing-Based Hyperspectral Image Denoising Using Spatial Preprocessing. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 8(6):2720–2727, 2015.

[34] Stephanie Delalieux, Pablo J. Zarco-Tejada, Laurent Tits, Miguel Angel Jimenez Bello, Diego S. Intrigliolo, and Ben Somers. Unmixing-based fusion of hyperspatial and hyperspectral airborne imagery for early detection of vegetation stress. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 7(6):2571–2582, 2014.

[35] Yanfeng Gu, Ye Zhang, and Junping Zhang. Integration of spatial-spectral information for resolution enhancement in hyperspectral images. *IEEE Trans. Geosci. Remote Sens.*, 46 (5):1347–1358, 2008. ISSN 01962892. doi: 10.1109/TGRS.2008.917270.

[36] 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. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 5(2):354–379, 2012.

[37] Rob Heylen, Mario Parente, and Paul Gader. A review of nonlinear hyperspectral unmixing methods. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 7(6):1844–1868, 2014. ISSN 21511535. doi: 10.1109/JSTARS.2014.2320576.

[38] Joseph W Boardman. Automating spectral unmixing of AVIRIS data using convex geometry concepts. In *Summ. 4th Annu. JPL Airborne Geosci. Work.*, pages 11–14, 1993.

[39] Michael E Winter. N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data. In *SPIE Int. Symp. Opt. Sci. Eng. Instrum.*, pages 266–275, 1999.

[40] Lidan Miao and Hairong Qi. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Trans. Geosci. Remote Sens.*, 45(3):765–777, 2007. ISSN 01962892. doi: 10.1109/TGRS.2006.888466.

[41] Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candès, and Yi Ma. Stable Principal Component Pursuit. In *IEEE Int. Symp. Inf. Theory*, pages 1518–1522, 2010.

[42] M. Mishali and Y.C. Eldar. Reduce and Boost: Recovering Arbitrary Sets of Jointly Sparse Vectors. *IEEE Trans. Signal Process.*, 56(10):4692–4702, oct 2008. ISSN 1053-587X. doi: 10.1109/TSP.2008.927802.

[43] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear Total Variation Based Noise Removal Algorithms. *Phys. D Nonlinear Phenom.*, 60(1):259–268, 1992.

[44] Tom Goldstein and Stanley Osher. The Split Bregman Method for L1-Regularized Problems. *SIAM J. Imaging Sci.*, 2(2):323–343, jan 2009. ISSN 1936-4954. doi: 10.1137/080725891.

[45] Marian-Daniel Iordache, José M. Bioucas Dias, and Antonio Plaza. Sparse Unmixing of Hyperspectral Data. *IEEE Trans. Geosci. Remote Sens.*, 49(6):2014–2039, 2011.

[46] Marian-daniel Iordache, José M Bioucas-dias, and Antonio Plaza. Total Variation Spatial Regularization for Sparse Hyperspectral Unmixing. *IEEE Trans. Geosci. Remote Sens.*, 50 (11):4484–4502, 2012.

[47] Marian-daniel Iordache, José M Bioucas-dias, and Antonio Plaza. Collaborative Sparse Regression for Hyperspectral Unmixing. *IEEE Trans. Geosci. Remote Sens.*, 52(1):341–354, 2014.

[48] Wotao Yin, Stanley Osher, Donald Goldfarb, and Jerome Darbon. Bregman Iterative Algorithms for L1-minimization with Applications to Compressed Sensing. *SIAM J. Imaging Sci.*, 1(1):143–168, 2008.

[49] Ewout Van Den Berg and Michael P Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912, 2008.

[50] David L. Donoho, Yaakov Tsaig, Iddo Drori, and Jean Luc Starck. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *IEEE Trans. Inf. Theory*, 58(2):1094–1121, 2012.

[51] Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sci.*, 2(1):183–202, jan 2009.

[52] L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comput. Math. Math. Phys.*, 7(3):200–217, jan 1967.

[53] Elaine T Hale, Wotao Yin, and Yin Zhang. A Fixed-Point Continuation Method for L1-Regularized Minimization with Applications to Compressed Sensing. Technical report, University of California, Los Angles, 2007.

[54] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online Dictionary Learning for Sparse Coding. In *Int. Conf. Mach. Learn.*, pages 689–696, 2009.

[55] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Trans. Image Process.*, 17(1):53–69, jan 2008. ISSN 1057-7149.

[56] Michael Elad. *Sparse and Redundant Representations From Theory to Applications in Signal and Image Processing*. Springer, 2010. ISBN 978-1-4419-7010-7.

[57] K. Engan, S.O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *IEEE Int. Conf. Acoust. Speech, Signal Process.*, volume 5, pages 2443–2446 vol.5. IEEE, 1999.

[58] Michal Aharon, Michael Elad, and Alfred Bruckstein. K -SVD : An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Trans. Signal Process.*, 54(11):4311–4322, 2006.

[59] Shane F Cotter, Bhaskar D Rao, Kjersti Engan, and Kenneth Kreutz-delgado. Sparse Solution to Linear Inverse Problems With Multiple Measurement Vectors. *IEEE Trans. Signal Process.*, 53(7):2477–2488, 2005.

[60] Ewout van den Berg and Michael P. Friedlander. Theoretical and Empirical Results for Recovery From Multiple Measurements. *IEEE Trans. Inf. Theory*, 56(5):2516–2527, may 2010. ISSN 0018-9448. doi: 10.1109/TIT.2010.2043876.

[61] Keigo Hirakawa, Patrick J Wolfe, and Senior Member. Spatio-Spectral Color Filter Array Design for Optimal Image Recovery. *IEEE Trans. Image Process.*, 17(10):1876–1890, 2008.

[62] Hiroyuki Takeda, Sina Farsiu, and Peyman Milanfar. Kernel regression for image processing and reconstruction. *IEEE Trans. Image Process.*, 16(2):349–66, feb 2007. ISSN 1057-7149.

[63] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(6):1397–409, jun 2013.

[64] Zahra Sadeghipoor, Yue M Lu, and Sabine Süsstrunk. A novel Compressive Sensing Approach to Simultaneously Acquire Color and Near-Infrared Images on a Single Sensor. In *IEEE Int. Conf. Acoust. Speech, Signal Process.*, pages 1646–1650, 2013.

[65] Keigo Hirakawa and Thomas W Parks. Adaptive homogeneity-directed demosaicing algorithm. *IEEE Trans. Image Process.*, 14(3):360–9, mar 2005.

[66] Lee Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color Demosaicking by Local Directional Interpolation and Nonlocal Adaptive Thresholding. *J. Electron. Imaging*, 20(2): 023016–023016–16, 2011. doi: 10.1117/1.3600632.

[67] Qiangqiang Yuan, Liangpei Zhang, and Huanfeng Shen. Hyperspectral Image Denoising Employing a Spectral Spatial Adaptive Total Variation Model. *IEEE Trans. Geosci. Remote Sens.*, 50(10):3660–3677, 2012.

[68] Qiangqiang Yuan, Liangpei Zhang, and Huanfeng Shen. Hyperspectral Image Denoising With a Spatial–Spectral View Fusion Strategy. *IEEE Trans. Geosci. Remote Sens.*, 52(5): 2314–2325, 2014.

[69] Fenge Chen, Guorui Ma, Liyu Lin, and Qianqing Qin. Impulsive noise removal via sparse representation. *J. Electron. Imaging*, 22(4):43014, 2013.

[70] Lianghai Jin, Hong Liu, Xiangyang Xu, and Enmin Song. Quaternion-based color image filtering for impulsive noise suppression. *J. Electron. Imaging*, 19(4):43003–43012, 2010.

[71] Djordje Baljozović, Branko Kovačević, and Aleksandra Baljozović. Novel method for removal of multichannel impulse noise based on half-space deepest location. *J. Electron. Imaging*, 21(1):13025–13027, 2012.

[72] Tom Mélange, Mike Nachtegael, and Etienne E Kerre. Random impulse noise removal from image sequences based on fuzzy logic. *J. Electron. Imaging*, 20(1):13016–13024, 2011.

[73] Youshen Xia and Mohamed S Kamel. Cooperative Recurrent Neural Networks for the Constrained L 1 Estimator. *IEEE Trans. Signal Process.*, 55(7):3192–3206, 2007.

[74] Jose L. Paredes and Gonzalo R. Arce. Compressive sensing signal reconstruction by weighted median regression estimates. *IEEE Trans. Signal Process.*, 59(6):2585–2601, 2011.

[75] Haoying Fu, Michael K. Ng, Mila Nikolova, and Jesse L. Barlow. Efficient Minimization Methods of Mixed l2-l1 and l1-l1 Norms for Image Restoration. *SIAM J. Sci. Comput.*, 27 (6):1881–1902, jan 2006.

[76] Chunlin Wu. Augmented Lagrangian Method for Total Variation Restoration with Non-Quadratic Fidelity. *Inverse Probl. Imaging*, 5(1):237–261, 2011.

[77] Yuying Shi and Qianshun Chang. Efficient Algorithm for Isotropic and Anisotropic Total Variation Deblurring and Denoising. *J. Appl. Math.*, 2013:1–14, 2013.

[78] Joan Duran, Bartomeu Coll, and Catalina Sbert. Chambolle's Projection Algorithm for Total Variation Denoising. *Impage Process. Line*, 3(2013):311–331, 2013.

[79] P Blomgren and T F Chan. Color TV: total variation methods for restoration of vector-valued images. *IEEE Trans. Image Process.*, 7(3):304–9, jan 1998. ISSN 1057-7149. doi: 10.1109/83.661180.

[80] Leah Bar, Alexander Brook, Nir Sochen, and Nahum Kiryati. Deblurring of color images corrupted by impulsive noise. *IEEE Trans. Image Process.*, 16(4):1101–11, apr 2007. ISSN 1057-7149.

[81] Antony Lam, Imari Sato, and Yoichi Sato. Denoising Hyperspectral Images Using Spectral Domain Statistics. In *Int. Conf. Pattern Recognit.*, pages 477–480, 2012. ISBN 9784990644109.

[82] Marco F Duarte and Richard G Baraniuk. Kronecker Compressive Sensing. *IEEE Trans. Image Process.*, 21(2):494–504, 2012.

[83] Marco F Duarte and Yonina C Eldar. Structured Compressed Sensing : From Theory to Applications. *IEEE Trans. Signal Process.*, 59(9):4053–4085, 2011.

[84] Michael Lustig, David Donoho, and John M Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magn. Reson. Med.*, 58(6):1182–95, dec 2007.

[85] Angshul Majumdar and Rabab K. Ward. On the Choice of Compressed Sensing Priors and Sparsifying Transforms for MR Image Reconstruction: An experimental study. *Signal Process. Image Commun.*, 27(9):1035–1048, oct 2012.

[86] Michael A Saunders. Solution of Sparse Rectangular Systems Using LSQR and CRAIG. *BIT Numer. Math.*, 35(4):588–604, 1995.

[87] Reno. Dataset, Last accessed: 10-June-2014. "`http://www.spectir.com/services/airborne-hyperspectral/`", 2013.

[88] WDC. Dataset, online last accessed: 14-Nov-2015. `https://engineering.purdue.edu/$\sim$biehl/MultiSpec/hyperspectral.html`, 2015.

[89] Paul Rodríguez and Brendt Wohlberg. Efficient minimization method for a generalized total variation functional. *IEEE Trans. Image Process.*, 18(2):322–32, mar 2009. ISSN 1057-7149. doi: 10.1109/TIP.2008.2008420.

[90] Zhou Wang and David Zhang. Progessive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images. *IEEE Trans. Circuits Syst. Analog Digit. Signal Process.*, 46(1):78–80, 1999.

[91] Hemant Kumar Aggarwal and Angshul Majumdar. Mixed Gaussina and Impulse Denoising of Hyperspectral Images. In *Int. Geosci. Remote Sens. Symp.*, pages 429–432, 2015.

[92] Brendt Wohlberg, Rick Chartrand, and James Theiler. Local Principal Component Pursuit for Nonlinear Datasets. In *IEEE Int. Conf. Acoust. Speech, Signal Process.*, pages 3925—-3928, 2012.

[93] Anupriya Gogna, Ankita Shukla, Hemant Kumar Aggarwal, and Angshul Majumdar. Split Bregman Algorithms for Sparse / Joint-sparse and Low-rank Signal Recovery : Application in Compressive Hyperspectral Imaging. In *IEEE Int. Conf. Image Process.*, pages 1302 – 1306, 2014.

[94] Mário A T Figueiredo and Robert D Nowak. An EM Algorithm for Wavelet-Based Image Restoration. *IEEE Trans. Image Process.*, 12(8):906–916, 2003.

[95] Suwanee. Dataset, last accessed : 14 Nov 2014. `http://www.spectir.com/services/airborne-hyperspectral/`, 2015.

[96] Tianyi Zhou and Decheng Tao. GoDec : Randomized Low-rank and Sparse Matrix Decomposition in Noisy Case. In *Int. Conf. Mach. Learn.*, pages 33–40, 2011.

[97] Qing Qu, Nasser M Nasrabadi, and Trac D Tran. Abundance Estimation for Bilinear Mixture Models via Joint Sparse and Low-Rank Representation. *IEEE Trans. Geosci. Remote Sens.*, 52(7):4404–4423, 2014.

[98] D.L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, apr 2006. ISSN 0018-9448.

[99] Emmanuel Candes, Justin Romberg, and Terence Tao. Stable Signal Recovery from Incomplete and Inaccurate Measurements. *Commun. Pure Appl. Math.*, 59(8):1207–1223, 2006.

[100] Balas Kausik Natarajan. Sparse approximate solutions to linear Systems. *SIAM J. Comput.*, 24(2):227–234, 1995.

[101] Joel A Tropp and Anna C Gilbert. Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit. *IEEE Trans. Inf. Theory*, 53(12):4655–4666, 2007.

[102] D. Needell and J. a. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmon. Anal.*, 26(3):301–321, 2009. ISSN 10635203. doi: 10.1016/j.acha.2008.07.002.

[103] Stephen Becker, Jerome Bobin, and Emmanuel J. Candes. NESTA : A Fast and Accurate First-Order Method for Sparse Recovery. *SIAM J. Imaging Sci.*, 4(1):1–39, 2011.

[104] Jianing V Shi, Aswin C Sankaranarayanan, Christoph Studer, and Richard G Baraniuk. Video compressive sensing for dynamic MRI. *BMC Neurosci.*, 13(Suppl 1):P183, 2012. ISSN 1471-2202. doi: 10.1186/1471-2202-13-S1-P183.

[105] USGS. Spectral library, 2015. URL `http://speclab.cr.usgs.gov/spectral-lib.html`.

[106] Grupo de Inteligencia Computacional and Spain Universidad del País Vasco / Euskal Herriko Unibertsitatea (UPV/EHU). Hyperspectral Imagery Synthesis (EIAs) toolbox., 2015. URL `http://www.ehu.es/ccwintco/index.php/Hyperspectral_Imagery_Synthesis_tools_for_MATLAB`.

[107] AVIRIS. Jasper Image. `http://www.eoc.csiro.au/hswww/Overview.htm`, 2015.

[108] J Bioucas Dias. source codes of SR, SRTV, CLSR. `http://www.lx.it.pt/%7ebioucas/code.htm`, 2015.

[109] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–12, apr 2004. ISSN 1057-7149.

[110] Snigdha Tariyal, Angshul Majumdar, Richa Singh, and Mayank Vatsa. Greedy Deep Dictionary Learning. Technical report, Indraprastha Institute of Information Technology-Delhi, 2016.

[111] Yushi Chen, Zhouhan Lin, Xing Zhao, Gang Wang, and Yanfeng Gu. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 7(6):2094–2107, 2014.

[112] Yushi Chen, Xing Zhao, and Xiuping Jia. SpectralâĂŞSpatial Classification of Hyperspectral Data Based on Deep Belief Network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, PP(99):1–12, 2015.

[113] Konstantinos Makantasis, Konstantinos Karantzalos, Anastasios Doulamis, and Nikolaos Doulamis. Deep Suptervised Learning for Hyperspectral Data Classification Through Convolutional Neural Networks. In *IEEE Geosci. Remote Sens. Symp.*, pages 4959–4962, 2015.

[114] Wei Hu, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li. Deep convolutional neural networks for hyperspectral image classification. *J. Sensors*, 2015:1–13, 2015.

[115] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse learned representations for image restoration. In *Proc. 4th World Conf. Int. Assoc. . . .*, pages 1–10, 2008.

[116] We Train, Deep Architectures, Intermediate Representations, and Sharing Features. *Learning Deep Architectures for AI By Yoshua Bengio*. now, 2009.

[117] A Rakotomamonjy. Applying alternating direction method of multipliers for constrained dictionary learning. *Neurocomputing*, 106:126—-136, 2013.

[118] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *ACM Symp. Theory Comput.*, pages 665—-674, 2013.

[119] Alekh Agarwal, Animashree Anandkumar, and Prateek Jain. Learning sparsely used overcomplete dictionaries via alternating minimization. In *Conf. Learn. Theory*, pages 123—-137, 2014.

[120] Daniel A Spielman, Huan Wang, and John Wright. Exact Recovery of Sparsely-Used Dictionaries. In *Conf. Learn. Theory*, pages 37–1, 2012.

[121] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. More algorithms for provable dictionary learning. Technical report, Princeton University, 2014.

[122] Schroff Florian and Philbin Dmitry, Kalenichenko James. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 815 – 823, 2015.

[123] Hemant Kumar Aggarwal. Matlab Implementation of Multispectral Demosaicing, 2014. URL `http://www.mathworks.com/matlabcentral/fileexchange/43425-multi-spectral-demosaicing-algorithms`.

[124] Hemant Kumar Aggarwal. Mixed Noise Reduction, 2015. URL `http://www.mathworks.com/matlabcentral/fileexchange/49145-mixed-noise-reduction`.

[125] Hemant Kumar Aggarwal. Matlab Implementation of Hyperspectral Unmixing Algorithm, 2016. URL `http://www.mathworks.com/matlabcentral/fileexchange/56831-hyperspectral-unmixing-and-denoising`.