

HYBRID HYPERSPECTRAL IMAGE COMPRESSION USING ONLINE  
DICTIONARY LEARNING AND SPARSE CODING

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
ÇANKAYA UNIVERSITY

BY  
İREM ÜLKÜ

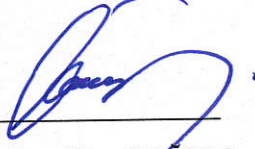
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN  
ELECTRONICS AND COMMUNICATION ENGINEERING

OCTOBER 2017


**Title of the Thesis: Hybrid Hyperspectral Image Compression Using Online Dictionary Learning and Sparse Coding**

Submitted by **İrem Ülkü**


Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

  
Prof. Dr. Can ÇOGUN  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

  
Prof. Dr. Sıtkı Kemal İDER  
Head of Department


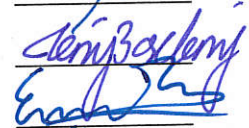



This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

  
Prof. Dr. Halil Tanyer EYYUBOĞLU  
Supervisor

**Examination Date: 24.10.2017**

**Examining Committee Members**

Prof. Dr. Halil Tanyer EYYUBOĞLU (Çankaya Univ.)  
Assoc. Prof. Dr. Hüsnü Deniz BAŞDEMİR (Çankaya Univ.)  
Asst. Prof. Dr. Emre SERMUTLU (Çankaya Univ.)  
Asst. Prof. Dr. Sevinç Figen ÖKTEM (METU)  
Asst. Prof. Dr. Elif VURAL (METU)

## STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: İrem ÜLKÜ

Signature: 

Date: 24.10.2017

## ABSTRACT

### HYBRID HYPERSPECTRAL IMAGE COMPRESSION METHOD BY USING ONLINE DICTIONARY LEARNING BASED ON SPARSE CODING

ÜLKÜ, İrem

PhD, Electronic and Communication Engineering Department

Supervisor: Prof. Dr. Halil Tanyer EYYUBOĞLU

October 2017, 100 pages

In this thesis a hybrid method is proposed, where an online dictionary learning approach based on the sparse coding scheme is adapted to compress hyperspectral images for the first time in the literature. In this method, various sparse representation algorithms are used to solve the sparse coding problem. Rate-distortion performances of different sparse representation algorithms are compared to those of the other compression algorithms. The information preservation performances are also evaluated by the anomaly detection application. The experimental results verify that compression performances of proximity based optimization algorithms and blind compressive sensing algorithms are superior to those of other algorithms as the bit rate increases.

Keywords: Sparse Coding, Compressive Sensing, Online Dictionary Learning, Hyperspectral Image Compression

## ÖZ

### SEYREK KODLAMA VE ÇEVİRİMİÇİ SÖZLÜK ÖĞRENME KULLANILARAK HİBRİT HİPERSPEKTRAL GÖRÜNTÜ SIKIŞTIRMASI

ÜLKÜ, İrem

Doktora, Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Tez Yöneticisi: Prof. Dr. Halil Tanyer EYYUBOĞLU

Ekim 2017, 100 sayfa

Bu tezde, seyrek kodlama tabanlı çevrimiçi sözlük öğrenme yaklaşımını literatürde ilk kez hiperspektral görüntülerin sıkıştırılması için adapte eden hibrit bir yöntem önerilmiştir. Seyrek kodlama tabanlı çevrimiçi sözlük öğrenme kullanılarak hiperspektral görüntülerin sıkıştırılması için hibrit bir yöntem olarak isimlendirilmiştir. Bu yöntemde, çeşitli seyrek temsil algoritmaları seyrek kodlama problemini çözmek amacıyla kullanılmıştır. Farklı seyrek temsil algoritmaları ile diğer sıkıştırma algoritmaları oran-bozulma performansları açısından karşılaştırılmışlardır. Bilgi koruma performansları da anomali sezimi uygulaması ile ayrıca ölçülmüştür. Deneysel sonuçlar kanıtıyor ki bit hızı arttıkça yakınlık bazlı eniyileme ve kör sıkıştırılmalı örnekleme algoritmalarına ait sıkıştırma performansları diğer algoritmalarından üstün olmaktadır.

Anahtar Kelimeler: Seyrek Kodlama, Sıkıştırılmalı Örnekleme, Çevrimiçi Sözlük Öğrenmesi, Hiperspektral görüntü sıkıştırılması

## ACKNOWLEDGMENTS

Foremost, I would like to thank Assoc. Prof. Dr. Behçet Uğur Töreyn for his contributions in the selection of the subject, construction of the method, conduction of the experiments and publication of the main journals throughout this study.

My sincere thanks also goes to Prof. Dr. Halil Tanyer Eyyubođlu for his orientation and support. I would also like to thank the rest of my thesis monitoring committee members: Assoc. Prof. Dr. Hüsnu Deniz Bařdemir and Asst. Prof. Dr. Emre Sermutlu for their valuable comments and supports.

I would like to thank Dr. Ersin Kızgut who was later involved in this thesis study. We made a lot of progress together. As a result of harmonious chemistry, new articles were published.

I would like to express my gratitude to Asst. Prof. Dr. Sevinç Figen Öktem and Asst. Prof. Dr. Elif Vural for their valuable comments. As a result of these comments, a clearer and more developed work has emerged.

I would like to express my appreciation to my mother and father for their endless support. It would be impossible to manage this compelling process without their support and encouragement.

Last but not the least, I would like to thank my best friend Sezin řentürk for supporting me spiritually throughout my life.

This thesis is supported in part by the Scientific and Technical Research Council of Turkey under National Young Researchers Career Development Program (3501 TUBITAK CAREER) grant with agreement number 114E200.

## TABLE OF CONTENTS

STATEMENT OF NON PLAGIARISM.....	iii
ABSTRACT .....	iv
ÖZ.....	v
ACKNOWLEDGMENTS.....	vi
TABLE OF CONTENTS.....	vii
LIST OF TABLES.....	xi
LIST OF FIGURES.....	xii
CHAPTERS:	
1. INTRODUCTION.....	1
2. RELATED WORK AND CONTRIBUTIONS.....	8
2.1 Prediction-Based Methods.....	9
2.2 Transformation-Based Methods.....	10
2.2.1 Transform step.....	10
2.2.2 Coding step.....	11
2.3 VQ-Based Methods.....	12
2.4 Sparse Representation-Based Methods.....	12
2.4.1 Sparse coding.....	12
2.4.1.1 Greedy pursuit based sparse coding algorithms.....	13
2.4.1.2 $l_p$ -norm regularization based sparse coding algorithms.....	14
2.4.1.2.1 Constraint based optimization algorithms.....	14
2.4.1.2.1.1 TNIPM algorithm.....	14
2.4.1.2.1.2 ADMM algorithm for solving LASSO problem.....	15

2.4.1.2.1.3 Dual active-set algorithm for solving BP problem.....	17
2.4.1.2.2 Homotopy based optimization algorithms.....	17
2.4.1.2.2.1 BPDN homotopy algorithm.....	17
2.4.1.2.3 Proximity based optimization algorithms.....	18
2.4.1.2.3.1 FISTA algorithm.....	20
2.4.1.2.3.2 SpaRSA algorithm.....	21
2.4.1.2.3.3 PALM algorithm.....	21
2.4.1.2.3.4 TwIST algorithm.....	22
2.4.1.2.3.5 GIST algorithm.....	22
2.4.1.2.4 Coordinate descent based optimization algorithms.....	23
2.4.1.2.4.1 Shooting algorithm.....	23
2.4.1.2.4.2 BCS PL-3DBS + 3DWPT algorithm...	24
2.4.1.2.4.3 OBD-BCS algorithm.....	24
2.4.1.2.5 $l_p$ -norm sparse coding ( $0 \leq p \leq 1$ ) algorithms.....	26
2.4.1.2.5.1 GISA algorithm.....	26
2.4.1.2.5.2 FOCUSS algorithm.....	27
2.4.1.3 Bayesian sparse coding algorithms.....	28
2.4.1.3.1 SBL algorithm.....	28
2.4.1.3.2 RLPHCS_Cov algorithm.....	29
2.4.2 Dictionary learning.....	30
2.4.2.1 Online learning methods vs batch methods.....	30
2.4.2.1.1 K-SVD algorithm.....	31
2.4.2.1.2 Online dictionary learning algorithm.....	32
2.5 Contributions.....	35
3. HYBRID HYPERSPECTRAL IMAGE COMPRESSION METHOD BY USING ONLINE DICTIONARY LEARNING BASED ON SPARSE CODING.....	36



3.1 Problem Statement.....	36
3.2 Algorithm.....	38
3.2.1 Algorithm 1.....	39
3.2.2 Algorithm 2.....	40
3.3 Rate-Distortion Results.....	43
3.3.1 Datasets.....	44
3.3.2 The rate-distortion results of AVIRIS and ROSIS datasets.....	46
3.3.2.1 Experiment 1 - Comparison with transformation-based, prediction-based or batch mode learning based lossy compression methods.....	47
3.3.2.2 Experiment 2 - Comparison of different sparse representation algorithms exploited in the hybrid method and comparison with transformation-based or sparse representation-based methods.....	52
3.3.2.3 Experiment 3 - Comparison with a state-of-the-art method.....	56
3.3.3 The rate-distortion results of HYPERION datasets.....	58
3.3.3.1 Experiment 1 - Comparison of the basic sparse representation algorithms exploited in the hybrid method.....	58
3.3.3.2 Experiment 2 - Comparison of the sparse representation algorithms exploited in the hybrid method which are based on proximity based optimization method.....	60
3.3.3.3 Experiment 3 - Comparison of several sparse representation algorithms with OBD-BCS sparse representation algorithm which are all exploited in the hybrid method.....	63
4. ANOMALY DETECTION RESULTS.....	66
4.1 Anomaly Detection Experiment 1 - Comparison of sparse representation algorithms exploited in the hybrid method which have the best rate-distortion performances.....	67
4.2 Anomaly Detection Experiment 2 - Comparison of the sparse representation algorithms which are based on proximity based optimization method.....	77

4.3 Anomaly Detection Experiment 3 - The application of anomaly detection by using only TNIPM sparse representation algorithm exploited in the hybrid method.....	82
5. CONCLUSION.....	86
REFERENCES .....	88



## LIST OF TABLES

Table 1 Sparse representation algorithms (OBD-BCS, TwIST, FOCUSS, Shooting, PALM, SpaRSA, GIST and FISTA) with corresponding sparse coding equations (this table contains the most recent sparse representation algorithms in the literature).....	42
Table 2 Sparse representation algorithms (SBL, gOMP, LASSO (ADMM), BP (Dual active-set), BPDN (Homotopy), GISA and TNIPM) with corresponding sparse coding equations (this table contains the older versions of the sparse representation algorithms in the literature) .....	43
Table 3 Information about the hyperspectral datasets used in this study which are taken from AVIRIS, HYPERION and ROSIS sensors.....	45
Table 4 Compression performances of sparse representation algorithms and other compression algorithms (BCS PL_3DBS + 3DWPT and CPPCA) [54] (The highest three PSNR values (in dBs) are shown in boldface for each bps value) .....	55
Table 5 Compression performances of sparse representation algorithms (The highest two PSNR values (in dBs) are shown in boldface for each bps value).....	69
Table 6 Area under curve (AUC) values of Salinas-A and Low Altitude hyperspectral datasets.....	72
Table 7 Compression performances of sparse representation algorithms (The highest two PSNR values (in dBs) are shown in boldface for each bps value) – Pavia University Dataset.....	79
Table 8 Compression performances of sparse representation algorithms.....	82

## LIST OF FIGURES

Figure 1 Grayscale image illustrations of (a) Lunar Lake (b) Low Altitude (c) Jasper Ridge (d) Mt. St. Helens (e) Lake Monona (f) Erta Ale hyperspectral datasets.....	44
Figure 2 Rate-distortion curves of the proposed sparse coding with online learning algorithm, predictive lossy compression algorithm and JPEG2000 part 2 algorithm for Yellowstone (uncalibrated scene 0) dataset [63].....	48
Figure 3 Rate-distortion curves of the proposed sparse coding with online learning algorithm, PCA+JPEG2000 algorithm and DWT+JPEG2000 algorithm for the Cuprite dataset cropped as 512x512x224 [70].....	49
Figure 4 Rate-distortion curves of the proposed sparse coding with online learning algorithm, FJOMP, with $n = 256$ , JOMP, with $n = 1024$ and 3DSPIHT for Cuprite dataset cropped as 512x512x32 [78].....	50
Figure 5 Rate-distortion curves of the proposed sparse coding with online learning algorithm, AR+SubPCA+JP2K algorithm and AR+PCA+JP2K algorithm for Moffett Field dataset [69] .....	51
Figure 6 Rate-distortion curves of the proposed sparse coding with online learning algorithm, 3D SPECK algorithm and modified 3D SPECK algorithm for Lunar Lake dataset [74] .....	51
Figure 7 Rate distortion curves of several sparse representation algorithms (ODL) and several hyperspectral compressive sensing algorithms (HCS) for Pavia University dataset when SNR is 20dB.....	57
Figure 8 Rate distortion curves of several sparse representation algorithms (ODL) and several hyperspectral compressive sensing algorithms (HCS) for Indian Pines dataset when SNR is 20dB.....	57
Figure 9 Rate-distortion curves of several sparse representation algorithms for Erta Ale hyperspectral dataset (cf. Table 3).....	59
Figure 10 Rate-distortion curves of several sparse representation algorithms for Mt. St. Helens hyperspectral dataset (cf. Table 3).....	59

Figure 11 Rate-distortion curves of several sparse representation algorithms for Lake Monona hyperspectral dataset (cf. Table 3).....	60
Figure 12 Rate-distortion performances of several sparse representation algorithms for Erta Ale hyperspectral dataset (cf. Table 3).....	61
Figure 13 Rate-distortion performances of several sparse representation algorithms for Mt. St. Helens hyperspectral dataset (cf. Table 3).....	62
Figure 14 Rate-distortion performances of several sparse representation algorithms for Lake Monona hyperspectral dataset (cf. Table 3).....	62
Figure 15 Rate-distortion performances of several sparse representation algorithms for Erta Ale hyperspectral dataset (cf. Table 3).....	63
Figure 16 Rate-distortion performances of several sparse representation algorithms for Mt. St. Helens hyperspectral dataset (cf. Table 3).....	64
Figure 17 Rate-distortion performances of several sparse representation algorithms for Lake Monona hyperspectral dataset (cf. Table 3).....	65
Figure 18 RX anomaly detection results of the Salinas-A hyperspectral image: (a) original image (b) OBD-BCS with 0.5 bps (c) OBD-BCS with 0.3 bps (d) OBD-BCS with 0.1 bps (e) BP with 0.5 bps (f) BP with 0.3 bps (g) BP with 0.1 bps (h) SpaRSA with 0.5 bps (i) SpaRSA with 0.3 bps (j) SpaRSA with 0.1 bps (k) LASSO with 0.5 bps (l) LASSO with 0.3 bps (m) LASSO with 0.1 bps.....	68
Figure 19 ROC Semi-Log curves for Salinas-A dataset at 0.1, 0.3 and 0.5 bps by using OBD-BCS algorithm.....	70
Figure 20 ROC Semi-Log curves for Salinas-A dataset at 0.1, 0.3 and 0.5 bps by using BP by using dual active set algorithm.....	71
Figure 21 ROC Semi-Log curves for Salinas-A dataset at 0.1, 0.3 and 0.5 bps by using SpaRSA algorithm.....	71
Figure 22 ROC Semi-Log curves for Salinas-A dataset at 0.1, 0.3 and 0.5 bps by using LASSO by using ADMM algorithm.....	72
Figure 23 RX anomaly detection results of the Low Altitude hyperspectral image: (a) original image (b) OBD-BCS with 0.5 bps (c) OBD-BCS with 0.3 bps (d) OBD-BCS with 0.1 bps (e) BP with 0.5 bps (f) BP with 0.3 bps (g) BP with 0.1 bps (h) SpaRSA with 0.5 bps (i) SpaRSA with 0.3 bps (j) SpaRSA with 0.1 bps (k) LASSO with 0.5 bps (l) LASSO with 0.3 bps (m) LASSO with 0.1 bps.....	74
Figure 24 ROC Semi-Log curves for Low Altitude dataset at 0.1, 0.3 and 0.5 bps by using OBD-BCS algorithm.....	75

Figure 25 ROC Semi-Log curves for Low Altitude dataset at 0.1, 0.3 and 0.5 bps by using BP by using dual active set algorithm.....	75
Figure 26 ROC Semi-Log curves for Low Altitude dataset at 0.1, 0.3 and 0.5 bps by using SpaRSA algorithm.....	76
Figure 27 ROC Semi-Log curves for Low Altitude dataset at 0.1, 0.3 and 0.5 bps by using LASSO by using ADMM algorithm.....	76
Figure 28 RX anomaly detection results of the Pavia University hyperspectral image: (a) original image (b) SpaRSA with 0.5 bps (c) SpaRSA with 0.3 bps (d) SpaRSA with 0.1 bps (e) BP with 0.5 bps (f) BP with 0.3 bps (g) BP with 0.1 bps (h) GIST with 0.5 bps (i) GIST with 0.3 bps (j) GIST with 0.1 bps (k) LASSO with 0.5 bps (l) LASSO with 0.3 bps (m) LASSO with 0.1 bps.....	78
Figure 29 ROC Semi-Log curves for Pavia University dataset at 0.1, 0.3 and 0.5 bps by using BP by using dual active set algorithm.....	80
Figure 30 ROC Semi-Log curves for Pavia University dataset at 0.1, 0.3 and 0.5 bps by using GIST algorithm.....	80
Figure 31 ROC Semi-Log curves for Pavia University dataset at 0.1, 0.3 and 0.5 bps by using SpaRSA algorithm.....	81
Figure 32 ROC Semi-Log curves for Pavia University dataset at 0.1, 0.3 and 0.5 bps by using LASSO by using ADMM algorithm.....	81
Figure 33 RX anomaly detection results of the Moffett Field hyperspectral image: (a) original image (b) reconstructed image with 0.4 bps (c) reconstructed image with 0.2 bps.....	83
Figure 34 RX anomaly detection results of the Cuprite hyperspectral image: (a) reconstructed image with 0.5 bps (b) reconstructed image with 0.1 bps.....	84

## CHAPTER 1

### INTRODUCTION

Hyperspectral images are obtained from hyperspectral sensors. They are composed of hundreds of contiguous narrow ( $0.010\ \mu\text{m}$ ) spectral bands from the visible ( $0.4\text{-}0.7\ \mu\text{m}$ ) region through the near-infrared (about  $2.4\ \mu\text{m}$ ) region of the electromagnetic spectrum. Each pixel of the hyperspectral image has a corresponding spectral signature that enables objects on earth to be uniquely defined [1]. Thanks to this spectral signature, hyperspectral images provide effective usage in fields where remote sensing applications are implemented such as agriculture, military and astronomy.

Hyperspectral images have huge image sizes since they have the additional spectral dimension. Fortunately, hyperspectral images have a high spectral inter-band correlation that ensures high rates of compression without any significant information loss [2]. During the downlink operation from satellite to ground, the available transmission bandwidth should be captured by applying compression [3]. Therefore, hyperspectral images are often compressed before storing and transmitting. Hyperspectral image compression is composed of two types: Lossy and lossless.

Although the lossless compression techniques are able to ensure full image quality, they cannot achieve high compression ratios. On the other hand, their lossy compression counterparts can transmit huge amounts of hyperspectral data with fast data rates despite the information loss disadvantage [4]. Indeed, suitable performance measurement tools can capture the quality degradation in the compressed image.

Compression methods are classified as prediction-based methods, transformation-based methods and vector quantization (VQ)-based methods. Indeed, transformation-based methods are the most popular ones. Principal component analysis (PCA) is a

transformation-based method which is based on the usage of spectral correlation characteristics. PCA realizes lossy compression based on the decorrelation of spectral bands [5]. However, the computational burden of the PCA scheme is huge. Compressive-projection principal component analysis (CPPCA) is the new version of the PCA method. CPPCA is compatible particularly with satellite-borne remote sensing applications [6]. There are some other transformation-based methods that utilize the decorrelation of spectral bands, such as Karhunen-Loeve transform (KLT), discrete wavelet transform (DWT) and discrete cosine transform (DCT) [7-10].

Instead of utilizing fixed filter coefficients in the decorrelation process, data-specific methods can also be used. Data-dependent methods perform the compression by using sparse representation based compression approach. By using such methods, compression performance improvement is achieved by considering the inherent characteristics of the data.

Dictionary learning is a data-specific method which recently became popular for hyperspectral image compression. Rather than using a pre-defined one, dictionary learning methods learn the dictionary directly from the hyperspectral image [11]. Dictionary learning is based on the solution of sparse coding problem. Indeed, the sparse representation is achieved by the iterative solution of dictionary learning and sparse coding problems [12-14]. This sparse representation aims to find the sparsest solution that requires solving non-deterministic polynomial-time hard (NP-hard)  $l_0$ -norm minimization problem [15]. Although it is difficult to obtain the optimal solution in NP-hard problems, the problem can be solved by convex optimization methods when the non-convex  $l_0$ -norm function is replaced by the convex  $l_1$ -norm function [16].

According to the technique used in the process, dictionary learning is classified into two categories, namely the batch methods and the online learning methods. By exploiting the singular value decomposition (SVD), K-SVD algorithm is a typical example of batch methods.



Batch processing algorithms require to reach the entire dataset at each iteration to perform learning, which makes them unsuitable for large datasets [17]. In order to cope with such an issue, online dictionary learning approach is developed, which is based on stochastic approximations during the learning process [18].

Using sparse coding methods along with dictionary learning for the purpose of hyperspectral image compression has been reported in the literature [15]. In the reference [15], a lossy hyperspectral image compression scheme is presented, which imposes a non-negativity condition on both the dictionary elements and the sparse coefficients. A similar approach is followed in this study which imposes a non-negativity condition on the dictionary elements and coefficients since it is based on the hyperspectral image compression as well. Even though the work in the reference [15] uses batch-mode gradient-type method, this study realizes the learning process by applying stochastic gradient descent-based online learning scheme [19, 20]. Another method [20] presents the idea of utilizing stochastic gradient descent-based online learning model for the purpose of the spatial-spectral joint classification of hyperspectral images. A similar model is practiced in this study to compress the hyperspectral images for the first time [21, 22].

Sparse representation algorithms are exploited to solve the sparse coding problem. In the literature, various categorizations are presented for the classification of sparse representation algorithms [13, 16]. In this study, sparse representation algorithms are analyzed in three categories. These are; greedy pursuit algorithms,  $l_p$ -norm regularization based algorithms and Bayesian algorithms.

Greedy pursuit algorithms find the sparsest solution which can be obtained by the minimization of the cost function with  $l_0$ -norm regularization [13]. The first and most basic algorithm in this category is the matching pursuit (MP) algorithm [23, 24]. This algorithm finds a solution by selecting the best dictionary element at each iteration according to a specific criterion until a desired termination condition is reached. Orthogonal matching pursuit (OMP) algorithm is the enhanced version of the MP algorithm that is based on orthogonal projection [25, 26]. Moreover, OMP algorithm is also enhanced to reduce the complexity corresponding mostly to the identification step.

This improved algorithm is called the generalized OMP (gOMP) algorithm that is associated with the approach of selecting multiple indices at each iteration [27]. The OMP algorithm is a special case of the gOMP algorithm that produces the solution with fewer iterations by selecting multiple indices [27]. Stagewise orthogonal matching pursuit (StOMP), regularized OMP (ROMP) and compressive sampling matching pursuit (CoSaMP) algorithms also fall under this category [28-30].

$l_p$ -norm regularization algorithms are composed of two types, namely,  $p \geq 1$  and  $0 < p < 1$ . In the  $p \geq 1$  case, only  $l_1$ -norm minimization is sufficiently sparse [13]. The  $l_1$ -norm minimization algorithms are classified as constrained based optimization algorithms, proximity based optimization algorithms, coordinate descent based optimization algorithms and homotopy based optimization algorithms.

Constrained based optimization algorithms are based on transforming a non-differentiable unconstrained problem into a differentiable constrained problem [13]. One example of such algorithms is gradient projection sparse reconstruction (GPSR) algorithm [31]. Interior-point method is another example that implements Newton's method to solve problems with moderate size data effectively [32]. Since the interior-point algorithm is somewhat time consuming for problems with large-scale data, the more efficient truncated Newton based interior-point method (TNIPM) was developed [33]. The alternating direction method (ADM) algorithm also belongs to this category [13]. Alternating direction method of multipliers (ADMM) algorithm is an ADM algorithm that handles a large scale problem by dividing it into smaller pieces [34]. The final example is the active-set algorithm that can be divided into two parts, namely, primal and dual [35, 36]. In this study, ADMM algorithm is used to solve Least Absolute Shrinkage and Selection Operator (LASSO) sparse coding problem and a dual active-set algorithm is utilized to solve a basis pursuit (BP) sparse coding problem.

Proximity based optimization algorithms are primarily based on reformulating the original sparse optimization problem by using a proximal operator such as a soft thresholding or hard thresholding operator [37, 38]. They are suitable for non-smooth and constrained problems, especially in significantly large-scale [38]. Many proximity algorithms can be

analyzed in this category. One of them is the iterative shrinkage thresholding algorithm (ISTA) [39]. An improved algorithm is fast iterative shrinkage thresholding algorithm (FISTA) [40, 41]. According to the literature, the global rate of convergence corresponding to FISTA is shown to be better while keeping the computational simplicity of ISTA [40]. Another algorithm is known as the sparse reconstruction by separable approximation (SpaRSA) [42]. The SpaRSA algorithm uses the Barzilai-Borwein (BB) spectral method and worm-starting technique to optimize the problem [13]. Additionally, two-step IST (TwIST) algorithms are considered in this category [43, 44]. The General Iterative Shrinkage and Thresholding (GIST) algorithm is proposed to efficiently solve the non-convex optimization problems for large-scale data sets and also belongs to this category [45, 46]. This algorithm iteratively solves a proximal operator problem and uses BB rules to identify a step size [45]. Augmented Lagrangian method (ALM) algorithm uses Lagrange multiplier to turn an equality constrained problem into an unconstrained problem [41]. The ALM algorithm is analyzed in two types as the primal augmented Lagrangian method (PALM) and the dual augmented Lagrangian method (DALM) according to whether a primal or dual  $l_1$ -minimization problem is solved, respectively [47].

Coordinate descent based optimization algorithms perform the optimization with respect to only one variable by fixing all other variables. All the variables are optimized iteratively in a cyclic manner until the convergence. The Shooting algorithm is classified in this category [48]. If the regularization term is generalized such that a partition is realized by groups, then the coordinate descent based optimization algorithms are generalized as block coordinate descent (BCD) based algorithms.

Non-convex  $l_p$ -norm ( $0 < p < 1$ ) regularized problems can be solved by algorithms such as the generalized iterated shrinkage algorithm (GISA) [49] and the focal underdetermined system solver (FOCUSS) algorithm [50, 51].

Homotopy based algorithms follow a procedure by tracing solutions along the continuous parameterized path while altering parameters to solve optimization problems [52]. The

LASSO homotopy algorithm is proposed to solve LASSO problems [53]. Moreover, the basis pursuit denoising (BPDN) homotopy algorithm also falls into this category. [54].

Bayesian compressive sensing algorithms category has one main example as the sparse Bayesian learning (SBL) algorithm [55].

In literature, variety of algorithms relying on the block compressed sensing (BCS) are proposed which are given as; smoothed projected Landweber (BCS-SPL) [56, 57], projected Landweber based on three-dimensional bivariate shrinkage (BCS PL-3DBS), 3D wavelet packet transform (BCS PL-3DBS + 3DWPT) [58].

The blind compressed sensing (BCS) algorithm aims to solve the sparse coding problem without prior knowledge of the sparsity basis. In this case, to guarantee the unique solution, some constraints are considered on the sparsity basis [59]. The algorithm used in the process is called an orthogonal block diagonal BCS (OBD-BCS) algorithm. Each iteration consists of an OMP algorithm and SVD algorithm. The OBD-BCS algorithm is indeed a compressive sensing framework. In addition to the sparse coding step, it also includes the dictionary learning step by utilizing a batch method. Here, the sparse coefficients are obtained by finding the corresponding basis with the help of this learning process.

Anomaly detection can measure the information preservation capability of different sparse representation algorithms; a popular method in the literature is to use Reed-Xiaoli (RX) anomaly detection algorithm [60]. RX Anomaly detection used in this study is to strengthen the obtained rate-distortion performance results. The detection performance of the original hyperspectral image is compared to that of the reconstructed hyperspectral image.

This study makes the following contributions:

- 1) This is the first study in the literature which adapts the online dictionary learning method proposed by Mairal et. al. [19] to the hyperspectral image compression.

- 2) A hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding is proposed by adapting different sparse representation algorithms to the solution of the sparse coding problem.

The sparse representation algorithms which are utilized in this study are as follows; gOMP algorithm, LASSO by using ADMM algorithm, BP by using dual active-set algorithm, SpaRSA algorithm, FISTA algorithm, TwIST algorithm, GIST algorithm, PALM algorithm, BPDN homotopy algorithm, TNIPM algorithm, GISA algorithm, OBD-BCS algorithm, SBL algorithm, FOCUSS algorithm and Shooting algorithm. This is the first hybrid study in the literature which adapts these different sparse representation algorithms to the case of online dictionary learning method proposed by Mairal et. al. [19].

As comparison of the proposed hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding to the other compression algorithms, BCS PL-3DBS + 3DWPT algorithm and CPPCA algorithm are used. Additionally, as a state-of-the-art compression scheme, RLPHCS\_Cov algorithm is exploited in the rate-distortion comparison.

- 3) Particularly, the most recent sparse representation algorithms in the literature such as those classified as proximity based optimization algorithms (SpaRSA algorithm, FISTA algorithm, TwIST algorithm, GIST algorithm and PALM algorithm) and blind compressive sensing algorithms (OBD-BCS algorithm), are analyzed further. The experimental results reveal that these algorithms offer better rate-distortion performance over the other sparse representation algorithms when they are adapted to solve the sparse coding problem in the first step of the algorithm.
- 4) In addition to the rate-distortion performance analysis, the quality of the reconstructed hyperspectral images is further evaluated by the anomaly detection application. Anomaly detection essentially measures the information preservation capability of the algorithms.

## CHAPTER 2

### RELATED WORK AND CONTRIBUTIONS

In this section, image compression techniques are introduced. Additionally, dictionary learning and sparse coding terms are explained with corresponding examples.

Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) is developed by NASA's Jet Propulsion Laboratory. AVIRIS sensor collects hyperspectral data that can be used for earth remote sensing. The data size is roughly higher than 500 Megabytes per flight which exceeds the available transmission bandwidth [61].

Efficient hyperspectral data compression methods are necessary to reduce the storage at the ground and match the available transmission bandwidth. Hyperspectral image compression techniques are divided into two categories, namely lossless and lossy compression.

In lossless compression, original image can be reconstructed from the compressed image. Since the compression ratio is low in lossless compression, it is generally not suitable for real-time transmission of the hyperspectral data [62].

Although much higher compression ratios can be achieved in lossy compression, some information loss is observed. Therefore, the distortion in the hyperspectral data should be measured to obtain an acceptable image quality. The basic quality evaluation tools are the statistical distortion measures such as signal-to-noise ratio (SNR) or peak signal-to-noise ratio (PSNR). The best way to evaluate the quality is to perform a real application both on the original and the compressed data [9].

In this study, lossy compression is considered. PSNR is used as the main quality measurement tool. As a real application, anomaly detection is applied. Consequently, a comprehensive benchmarking is performed to evaluate the image quality.

Hyperspectral image acquisition has a number of limitations such as the storage capacity of the satellite and the bandwidth capacity of the transmission line. Lossless compression techniques are not sufficient to allow real-time acquisition. These techniques are more suitable for the distribution of the data to the end users [9].

In real-time transmission of the hyperspectral data, lossy image compression techniques should be utilized to cope with the restrictions [62]. Most recent satellites, namely SPOT 4 and IKONOS both utilize the on-board lossy compression of hyperspectral data before the down-linking from satellite to ground [3].

Lossy and lossless compression methods are divided into three categories depending on the implementation. These are prediction-based methods, transformation-based methods and the VQ-based methods.

## **2.1 PREDICTION-BASED METHODS**

The main idea behind the prediction-based methods is the usage of the correlation among the image pixels [62]. At first, data is decorrelated through a predictor which seeks the correlation between spectral bands. Prediction error is coded with the help of an entropy coder afterwards [9].

One basic example is the differential pulse modulation (DPCM). Other examples are the lookup tables (LUT) and the usage of context-based, adaptive, lossless image coding (CALIC) [9]. In general, lossless compression techniques employ prediction-based methods [63].

Similar to the prediction-based methods for 2D images, some modifications are devised for hyperspectral image compression [64-66]. In order to match the low complexity requirement of the onboard compression, a lossy hyperspectral image compression

algorithm based on prediction is proposed [63]. It is called predictive lossy compression algorithm.

## **2.2 TRANSFORMATION-BASED METHODS**

Transform-based methods have two steps, namely the transform step and the coding step. In the transform step, data is transformed in a less correlated domain. In the coding step, data is encoded [9]. Transformation-based methods use a dictionary that can be either a DCT or a wavelet dictionary [67]. Most common examples are PCA which is also named as KLT, DCT and DWT. The joint photographic experts group (JPEG) compression standard is based on DCT and JPEG 2000 standard is based on DWT [68].

DCT mainly employs the linear combination of weighted basis functions by exploiting the frequency components in order to represent the input signal. DWT, on the other hand, realizes a decorrelation by splitting the data into two parts, namely a very significant part and a least significant part [3]. A generalization of DWT is called the discrete wavelet packet transform (DWPT).

In general, lossy compression techniques utilize transformation-based methods [63]. Since, each method has a different definition of the main steps, a detailed discussion is useful.

### **2.2.1 Transform step**

Methods rely on the utilization of spatial and spectral correlation prior to a quantization [3]. The most effective transform method is considered to be the KLT. However, the computational cost in this case is high. The modifications on the KLT method are studied to reduce this complexity. A new strategy, namely SubPCA, is also tested where only a subset of the principal components are used in compression rather than all components [69]. However, the PCA-based compression methods have the heavy encoder side burden. The bulk of the computation in the encoder side is transformed to the decoder side by employing the process based on projections-onto-convex-sets (POCS) optimization. This new method, namely CPPCA relies on conducting a data-specific PCA basis by using the random projections [6].



Wavelet family is accepted as another effective transform. The most popular ones are 9/7 and 5/3 wavelets. They are also adopted in the JPEG 2000 standard [9]. Hyperspectral image compression requires a 3D transform. Therefore, it is important to decide which combination of transforms would be better for hyperspectral data [63]. A typical 3D compression algorithm scheme follows the steps of coupling a spectral transform with a spatial wavelet transform in addition to applying a suitable coding algorithm [70]. Recently, a popular approach in the literature is to combine a 1D spectral de-correlator such as PCA, DWT or DCT with a spatial de-correlator such as JPEG 2000 standard [3]. Indeed, JPEG 2000 is well suited to the 3D hyperspectral image compression [70].

Using PCA together with JPEG 2000 in hyperspectral image compression provides a superior rate-distortion performance [69]. Indeed, the compression performance of PCA+JPEG2000 outperforms that of the DWT+JPEG2000 for hyperspectral data [70]. Therefore, in hyperspectral image compression, PCA provides a better compression performance than DWT according to the literature [69]. A new compression approach proposes the anomaly-adjusted (AA) algorithm to solve PCA+JPEG2000 which utilizes the anomalous pixels in the compression process [71]. An improved version of AA algorithm, which is called anomaly removal (AR)-based algorithm, is to compress the anomalous pixels by using a lossless scheme and to compress the remaining part of the image by using PCA+JPEG2000 or SubPCA+JPEG2000 [69].

### **2.2.2 Coding step**

The reduced number of coefficients are coded by several methods. It is aimed to obtain a long stream of zeros. The idea is to exploit the behavior of wavelength coefficients. Zero-tree algorithms such as the embedded zero-tree wavelet (EZW) algorithm, set partitioning in hierarchical trees (SPIHT) algorithm and set partitioned embedded block coder (SPECK) algorithm utilize this strategy which are used in hyperspectral data compression as well [72, 73]. SPIHT can be useful for onboard hyperspectral data compression.

Wavelet-based lossy compression methods defined for 2D images are also extended to 3D counterparts such as 3D SPIHT and 3D SPECK [70]. A simplified version of 3D SPECK

algorithm is proposed in the literature which is designed particularly for the lossy compression of hyperspectral images [74]. Instead of using one, generally compression techniques are realized by the combination of these steps. One example is using JPEG 2000 with 3D SPIHT [9].

### **2.3 VQ-BASED METHODS**

These methods directly quantize the data block. Decorrelation process is not performed [62]. A two-step process is carried out as a training step and a coding step. In the training step, the codebook is formed. In the coding step, an assignment between vectors and code-words is realized. One popular example is the generalized Lloyd algorithm (GLA) [9]. Due to the computational burden, this method is not effective for the hyperspectral data compression. In literature, several studies are conducted to exploit the vector quantization-based methods in hyperspectral image compression [75-77].

### **2.4 SPARSE REPRESENTATION-BASED METHODS**

Besides the mentioned compression techniques so far, another way to achieve hyperspectral image compression is by using sparse representation based compression approach [78]. Pre-designed dictionaries are often not suitable for the real natural data. Therefore, data-dependent dictionaries which are adapted to the observed data are becoming popular, especially for the hyperspectral data compression. In order to adapt the dictionary to specific data, dictionary learning should be conducted which is based on the sparse representation. Indeed, exploiting the learned dictionary to model the data is an important research direction in machine learning and signal processing, as well as in hyperspectral image compression recently. Sparse coding and dictionary learning concepts are discussed in the following sections.

#### **2.4.1 Sparse coding**

Sparse coding is the problem of finding a sparse coefficient vector  $\alpha$  when the signal measurement  $\mathbf{x}$  and the dictionary  $\mathbf{D}$  are given [78] and it can be formulated as;

$$\min_{\mathbf{a}} \|\mathbf{a}\|_0 \quad \text{s.t.} \quad \|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2^2 \leq k. \quad (1)$$

The sparse coding problem in (1) is so called the  $l_0$ -norm minimization problem. The sparsity of a vector comes from the  $l_0$ -norm which represents the number of nonzero elements. However, as the  $l_0$ -norm minimization is NP-hard, the  $l_1$ -norm relaxations are often considered. Indeed, the  $l_1$ -norm relaxation of the sparse coding problem (1) is convex such that it can be solved by linear programming technique [27].

Although both the  $l_0$ -norm and  $l_1$ -norm minimization problems are assumed to be sparse, that of the  $l_2$ -norm minimization cannot satisfy the condition of sparsity [13]. Indeed,  $l_2$ -norm measures the signal energy, it is not a measure of sparsity. The definitions of  $l_0$ -norm,  $l_1$ -norm and  $l_2$ -norm are as follows, where  $\mathbf{u}$  is a vector of length  $n$ .

$$l_0 - \text{norm} : \|\mathbf{u}\|_0 = \#\{i | \mathbf{u}_i \neq 0\}, \quad l_1 - \text{norm} : \|\mathbf{u}\|_1 = \sum_{i=1}^n |\mathbf{u}_i|, \quad l_2 - \text{norm} : \|\mathbf{u}\|_2 = \left( \sum_{i=1}^n \mathbf{u}_i^2 \right)^{1/2} \quad (2)$$

In this study, sparse coding scheme is generalized as sparse representation of the data. Here, the sparse representation methods are categorized into three groups, namely greedy pursuit based algorithms,  $l_p$ -norm regularization based algorithms and Bayesian algorithms. These algorithms have different strategies to solve the sparse coding equation in (1) and the ones which are used in this study will be explained.

#### 2.4.1.1 Greedy pursuit based sparse coding algorithms

The solution of the sparse coding equation (1) is considered as NP-hard, since the sparsest solution so-called  $l_0$ -norm turns it into a nonconvex combinatorial problem [79]. Greedy approaches aim to solve this original NP-hard sparse coding problem by approximate methods. Here, the exact  $l_0$ -norm problem is solved approximately.

As an example of greedy algorithms, gOMP is used in this study. It is the generalization of the OMP algorithm in such a way that multiple indices are chosen in each iteration [27]. The indices of the maximally correlated dictionary columns are included in a set. This set is composed of the estimated positions of the nonzero dictionary elements.

### 2.4.1.2 $l_p$ -norm regularization based sparse coding algorithms

The NP-hard sparse coding problem in (1) is replaced with the convex relaxation counterpart which is easier to solve. Here, the approximation of  $l_0$ -norm problem is solved exactly. Among the  $l_p$ -norm regularization based sparse coding equations, particularly the  $l_1$ -norm relaxation is considered in this study, since it is the only norm that induces sparsity as well as being convex [79].

The  $l_1$ -norm regularization problems are categorized as constraint based optimization algorithms, homotopy based optimization algorithms, coordinate descent based optimization algorithms and proximity based optimization algorithms. Each of these categories will be summarized with examples.

#### 2.4.1.2.1 Constraint based optimization algorithms

In constraint based optimization algorithms, some basic examples can be given as TNIPM algorithm, ADMM algorithm for solving LASSO problem and dual active-set algorithm for solving BP problem. The strategy is to transform the non-differentiable unconstrained sparse coding problem into a smooth differentiable constrained sparse coding problem.

The  $l_1$ -regularized least squares problem is defined as follows, where the first part represents the least squares problem and the second part is the  $l_1$ -norm regularization term;

$$\min \|\mathbf{D}\boldsymbol{\alpha} - \mathbf{x}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 \quad (3)$$

This is the  $l_1$ -norm relaxation of the sparse coding equation in (1).

##### 2.4.1.2.1.1 TNIPM algorithm

In order to solve the  $l_1$ -regularized least squares problem in (3), it is transformed into a convex quadratic problem with linear inequality constraints [33] such that it can be rewritten as;

$$\min \|\mathbf{D}\boldsymbol{\alpha} - \mathbf{x}\|_2^2 + \lambda \sum_{i=1}^k \mathbf{u}_i \quad \text{s.t.} \quad -\mathbf{u}_i \leq \boldsymbol{\alpha}_i \leq \mathbf{u}_i, \quad (4)$$

where  $i = 1, \dots, k$  with  $\alpha \in \mathbb{R}^k$  and  $u \in \mathbb{R}^k$ . Since the problem in (3) is converted to a convex quadratic counterpart as in (4), the resulting quadratic problem can be solved by using one of the convex optimization methods such as the interior-point method. In this study, the convex quadratic problem in (4) is solved via TNIPM algorithm, which a specialized interior-point method for solving relatively large scale problems [33]. TNIPM algorithm computes the search direction as an approximation to the Newton systems. This strategy is named as truncated Newton method in general. Here such a truncated Newton method is applied to the interior-point method [33].

#### 2.4.1.2.1.2 ADMM algorithm for solving LASSO problem

In order to solve the sparse coding equation in (1), ADMM can be used as well. Indeed, ADMM is very effective in solving the distributed convex optimization problems particularly the large scale ones [34]. The ADMM strategy is introduced briefly. The equality-constrained convex optimization problem is written as;

$$\min f(\alpha) \quad \text{s.t. } \mathbf{D}\alpha = \mathbf{x} \quad (5)$$

where  $\alpha \in \mathbb{R}^k$  and  $\mathbf{D} \in \mathbb{R}^{m \times k}$ . The Lagrangian of the problem in (5) can be defined as;

$$L(\alpha, \mathbf{y}) = f(\alpha) + \mathbf{y}^T (\mathbf{D}\alpha - \mathbf{x}) \quad (6)$$

and the corresponding augmented Lagrangian is as follows;

$$L_\rho(\alpha, \mathbf{y}) = f(\alpha) + \mathbf{y}^T (\mathbf{D}\alpha - \mathbf{x}) + (\rho/2) \|\mathbf{D}\alpha - \mathbf{x}\|_2^2 \quad (7)$$

where  $\rho > 0$  is the penalty parameter. The method of multipliers can be used to solve the problem in (7) and it has the following steps as;

$$\begin{aligned} \alpha^{v+1} &:= \arg \min_{\alpha} L_\rho(\alpha, \mathbf{y}^v) \\ \mathbf{y}^{v+1} &:= \mathbf{y}^v + \rho(\mathbf{D}\alpha^{v+1} - \mathbf{x}) \end{aligned} \quad (8)$$

where  $v$  represents the number of iterations [34]. Throughout the method, the optimality is achieved when the primal residual  $(\mathbf{D}\alpha^{v+1} - \mathbf{x})$  converges to zero. Along with the method

of multipliers, ADMM can also be defined which aims to solve the following type of problems;

$$\min f(\boldsymbol{\alpha}) + g(\mathbf{z}) \quad \text{s.t. } \mathbf{D}\boldsymbol{\alpha} + \mathbf{B}\mathbf{z} = \mathbf{x}. \quad (9)$$

The difference between problem (5) and problem (9) is that the variable  $\boldsymbol{\alpha}$  in (5) is now split into two parts, namely  $\boldsymbol{\alpha}$  and  $\mathbf{z}$  in the latter problem. The objective function is also separated into two [34]. Note that the functions  $f$  and  $g$  are assumed to be convex. The augmented Lagrangian of the problem in (9) is written in the form of;

$$L_\rho(\boldsymbol{\alpha}, \mathbf{z}, \mathbf{y}) = f(\boldsymbol{\alpha}) + g(\mathbf{z}) + \mathbf{y}^T (\mathbf{D}\boldsymbol{\alpha} + \mathbf{B}\mathbf{z} - \mathbf{x}) + (\rho / 2) \|\mathbf{D}\boldsymbol{\alpha} + \mathbf{B}\mathbf{z} - \mathbf{x}\|_2^2 \quad (10)$$

and the iterations of ADMM can be defined similar to those in (8) as;

$$\begin{aligned} \boldsymbol{\alpha}^{v+1} &:= \arg \min_{\boldsymbol{\alpha}} L_\rho(\boldsymbol{\alpha}, \mathbf{z}^v, \mathbf{y}^v) \\ \mathbf{z}^{v+1} &:= \arg \min_{\mathbf{z}} L_\rho(\boldsymbol{\alpha}^{v+1}, \mathbf{z}, \mathbf{y}^v) \\ \mathbf{y}^{v+1} &:= \mathbf{y}^v + \rho(\mathbf{D}\boldsymbol{\alpha}^{v+1} + \mathbf{B}\mathbf{z}^{v+1} - \mathbf{x}) \end{aligned} \quad (11)$$

which includes the minimization steps for  $\boldsymbol{\alpha}$  and  $\mathbf{z}$  variables together with a dual variable update step. The term alternating in ADMM comes from updating the variables  $\boldsymbol{\alpha}$  and  $\mathbf{z}$  in alternating fashion. The method is well suited to the problems that split into two distinct parts. Since the sparse coding problem is in this form of a loss function together with a regularization term, ADMM can be used to solve sparse coding equation in (1) when it is relaxed with  $l_1$ -norm. LASSO problem is the  $l_1$ -regularized sparse coding problem and defined as follows;

$$\min \frac{1}{2} \|\mathbf{D}\boldsymbol{\alpha} - \mathbf{x}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1. \quad (12)$$

LASSO problem can be re-written in ADMM form similar to (9) as;

$$\begin{aligned} \min f(\boldsymbol{\alpha}) + g(\mathbf{z}) \quad \text{s.t. } \boldsymbol{\alpha} - \mathbf{z} = 0 \\ \text{where } f(\boldsymbol{\alpha}) = (1/2) \|\mathbf{D}\boldsymbol{\alpha} - \mathbf{x}\|_2^2, g(\mathbf{z}) = \lambda \|\mathbf{z}\|_1 \end{aligned} \quad (13)$$

and the problem in (13) can be solved by using the ADMM iterations given in (11). This procedure is the solution of the LASSO sparse coding equation by using ADMM method.

#### 2.4.1.2.1.3 Dual active-set algorithm for solving BP problem

Another constraint based optimization method to solve the  $l_1$ -norm relaxation form of the sparse coding equation (1) is by using dual active-set algorithm [35]. First, a primal problem is considered as follows;

$$\min_{\alpha, \mathbf{z}} \|\alpha\|_2^2 + \frac{1}{2} \lambda \|\mathbf{z}\|_2^2 \quad \text{s.t. } \mathbf{D}\alpha + \lambda \mathbf{z} = \mathbf{x}. \quad (14)$$

When  $\lambda=0$ , the primal problem in (14) turns into a BP sparse coding problem. If  $\lambda>0$ , it becomes the form of BPDN sparse coding problem. The dual of the problem in (14) can be written as;

$$\max_{\mathbf{z}} \mathbf{x}^T \mathbf{z} - \frac{1}{2} \lambda \|\mathbf{z}\|_2^2 \quad \text{s.t. } -\mathbf{e} \leq \mathbf{D}^T \mathbf{z} \leq \mathbf{e} \quad (15)$$

where  $e$  corresponds to the vector of ones. The relation between primal and dual problems in (14) and (15) is such that the optimal  $\alpha$  values of the former are the optimal Lagrange multipliers for the inequality constraint of the latter [35]. Since the dual problem in (15) is a convex quadratic problem, it is favorable to exploit active-set methods to solve the dual problem. Active-set method updates the primal variables while maintaining the dual feasibility in (15) until an optimality condition is reached [35].

#### 2.4.1.2.2 Homotopy based optimization algorithms

The idea behind the homotopy based strategy is to change the parameters while tracing a parameterized path of solutions [13]. One main example of the homotopy based optimization algorithms is the BPDN homotopy algorithm.

##### 2.4.1.2.2.1 BPDN homotopy algorithm

The BPDN sparse coding problem is formulated as follows:

$$\min \frac{1}{2} \|\mathbf{D}\boldsymbol{\alpha} - \mathbf{x}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1. \quad (16)$$

The approach is to trace the solution path by changing the parameter  $\lambda$  while hopping from one optimization scheme to another [54]. When following the path, some optimality conditions should be maintained. If a vector  $\boldsymbol{\alpha}^*$  is the solution to the BPDN problem in (16), it must satisfy the following condition;

$$\left\| \mathbf{D}^T (\mathbf{D}\boldsymbol{\alpha}^* - \mathbf{x}) \right\|_\infty \leq \lambda. \quad (17)$$

The condition in (17) is indeed a set of different constraints for each  $\mathbf{D}^T (\mathbf{D}\boldsymbol{\alpha}^* - \mathbf{x})$ . In addition to the necessary condition in (17), a sufficient condition for having an optimal  $\boldsymbol{\alpha}^*$  is to equate the set of locations when the constraints in (17) are equal to  $\lambda$  and the set of locations when  $\boldsymbol{\alpha}^*$  is nonzero [54]. This set is denoted by  $\mathcal{S}$ . A vector  $\mathbf{z}$  is also defined which contains the signs of  $\boldsymbol{\alpha}^*$  on  $\mathcal{S}$ . By exploiting  $\mathcal{S}$  and  $\mathbf{z}$ , it is possible to find  $\boldsymbol{\alpha}^*$  as follows;

$$\boldsymbol{\alpha}^* = \begin{cases} (\mathbf{D}_{\mathcal{S}}^T \mathbf{D}_{\mathcal{S}})^{-1} (\mathbf{D}_{\mathcal{S}}^T \mathbf{x} - \lambda \mathbf{z}) & \text{on } \mathcal{S} \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Solution is achieved by changing the parameter  $\lambda$  while moving along the direction  $(\mathbf{D}_{\mathcal{S}}^T \mathbf{D}_{\mathcal{S}})^{-1}$  until any critical point is found [54]. At the critical points, both the set  $\mathcal{S}$  and the direction of the path are changed. Starting with a large value of  $\lambda$ , it is reduced to the desired level by changing from one critical point to the next. One element is added or removed from the set  $\mathcal{S}$  at each critical point along the solution path. This is a type of soft-thresholding approach which will be explained in detail later.

#### 2.4.1.2.3 Proximity based optimization algorithms

Proximity based optimization algorithms aim to solve non-smooth, constrained convex optimization problems by exploiting the proximity algorithm [13]. Since the sparse representation with  $l_1$ -norm regularization is a non-smooth convex optimization problem, this type of algorithms can be used to effectively solve such sparse coding problems.



Proximity algorithms rely on the proximal operator. This proximal operator is utilized to solve the computationally efficient sub-problem iteratively [13]. Before examining the example algorithms, a brief discussion about the proximal algorithms will be introduced.

A simple constrained optimization problem can be defined as follows;

$$\min \left\{ h(\mathbf{a}) \mid \mathbf{a} \in R^m \right\}. \quad (19)$$

By using the proximal algorithm, the constrained convex optimization problem in (19) can be re-written as;

$$\tilde{\mathbf{a}}^t = \arg \min \left\{ h(\mathbf{a}) + \frac{\tau}{2} \|\mathbf{a} - \mathbf{a}^t\|^2 \mid \mathbf{a} \in R^m \right\} \quad (20)$$

where  $\tau$  and  $\mathbf{a}^t$  are known. The simple problem in (19) can also be extended, and the linear constrained convex optimization problem is given as follows;

$$\min \left\{ F(\mathbf{a}) + G(\mathbf{a}) \mid \mathbf{a} \in R^m \right\}. \quad (21)$$

Again by using the proximity algorithm, the problem in (21) can be re-written as;

$$\mathbf{a}^{t+1} = \arg \min \left\{ F(\mathbf{a}) + \left\langle \nabla G(\mathbf{a}^t), \mathbf{a} - \mathbf{a}^t \right\rangle + \frac{1}{2\tau} \|\mathbf{a} - \mathbf{a}^t\|^2 \right\} \quad (22)$$

Particularly for the sparse representation problem with  $l_1$ -norm regularization case, the problem definition in (21) is re-arranged as follows;

$$\min P(\mathbf{a}) = \left\{ \lambda \|\mathbf{a}\|_1 + \|\mathbf{D}\mathbf{a} - \mathbf{x}\|_2^2 \mid \mathbf{a} \in R^m \right\}. \quad (23)$$

It is now possible to define the soft thresholding function and employ it in the solution. A general sparse coding problem in the form of;

$$\mathbf{a}^* = \min_{\mathbf{a}} h(\mathbf{a}) = \lambda \|\mathbf{a}\|_1 + \frac{1}{2} \|\mathbf{a} - \mathbf{s}\|_2^2 = \sum_{j=1}^m \lambda |\alpha_j| + \sum_{j=1}^m \frac{1}{2} (\alpha_j - s_j)^2 \quad (24)$$

and it has the solution that can be written by exploiting the soft thresholding function as follows [13];

$$\alpha_j^* = \begin{cases} s_j - \lambda, & \text{if } s_j > \lambda \\ s_j + \lambda, & \text{if } s_j < -\lambda \\ 0, & \text{otherwise} \end{cases} \quad (25)$$

The solution in (25) can also be expressed as  $\alpha^* = \text{shrink}(s, \lambda)$  such that the operator  $\text{shrink}(\cdot)$  is the proximal operator. After summarizing the proximal algorithm in a general sense, it is now time to introduce some example algorithms.

#### 2.4.1.2.3.1 FISTA algorithm

One important example of the proximity based optimization algorithm is FISTA which is the fastest version of ISTA. In this case, the following sparse coding problem is considered as;

$$\arg \min F(\boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{D}\boldsymbol{\alpha} - \mathbf{x}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 = f(\boldsymbol{\alpha}) + \lambda g(\boldsymbol{\alpha}) \quad (26)$$

such that it is difficult to find out a solution to (26). Therefore, the sparse coding problem in (26) is converted to a more easy problem by employing second order Taylor expansion to approximate  $f(\boldsymbol{\alpha})$  at a point of  $\boldsymbol{\alpha}^t$  and by approximating the corresponding Hessian matrix [13], as follows;

$$f(\boldsymbol{\alpha}) \approx \frac{1}{2} \|\mathbf{D}\boldsymbol{\alpha}^t - \mathbf{x}\|_2^2 + (\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^T \mathbf{D}^T (\mathbf{D}\boldsymbol{\alpha}^t - \mathbf{x}) + \frac{L}{2} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^T (\boldsymbol{\alpha} - \boldsymbol{\alpha}^t) = P_t(\boldsymbol{\alpha}, \boldsymbol{\alpha}^t) \quad (27)$$

where the solution of the problem in (27) is formulated by exploiting the proximal algorithm and it is in the form of;

$$\boldsymbol{\alpha}^{t+1} = \arg \min \frac{L}{2} \left\| \boldsymbol{\alpha} - \left( \boldsymbol{\alpha}^t - \frac{1}{L} \mathbf{D}^T (\mathbf{D}\boldsymbol{\alpha}^t - \mathbf{x}) \right) \right\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1. \quad (28)$$

In order to solve the problem in (28), the shrinkage operator in (25) is exploited in FISTA algorithm [13].

#### 2.4.1.2.3.2 SpaRSA algorithm

A proximity based optimization algorithm, namely SpaRSA, forms a general framework for solving the  $l_1$ -norm relaxation of the sparse coding equation in (1). The sparse coding problem in (26) is also considered in this case. First contribution of the SpaRSA algorithm is that it uses the worm-starting technique in order to optimize the parameter  $\lambda$  in problem (26) [13]. Additionally, the Hessian matrix is approximated by employing the BB spectral method which ensures a more reliable approximation. The Hessian matrix of  $f(\boldsymbol{\alpha})$  at  $\boldsymbol{\alpha}^t$  is as follows;

$$H_f(\boldsymbol{\alpha}) = \mathbf{D}^T \mathbf{D}. \quad (29)$$

#### 2.4.1.2.3.3 PALM algorithm

The Lagrange multiplier is utilized to transform an equality constrained problem into an unconstrained problem by producing an additional penalty function [13]. It is common to employ the ALM in the solution of sparse coding equations. Assume that the sparse coding problem of interest is in the following form;

$$\hat{\boldsymbol{\alpha}} = \arg \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{D}\boldsymbol{\alpha}. \quad (30)$$

The corresponding augmented Lagrangian function of the sparse coding problem in (30) can be written by adding an equality constrained function as follows;

$$L(\boldsymbol{\alpha}, \lambda) = \|\boldsymbol{\alpha}\|_1 + \frac{\lambda}{2} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 \quad \text{s.t.} \quad \mathbf{x} - \mathbf{D}\boldsymbol{\alpha} = 0. \quad (31)$$

The Lagrangian function form of the problem in (31) can be re-arranged in the form of;

$$\arg \min L_\lambda(\boldsymbol{\alpha}, \mathbf{z}) = \|\boldsymbol{\alpha}\|_1 + \frac{\lambda}{2} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \mathbf{z}^T (\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}) \quad (32)$$

where  $\mathbf{z} \in \mathbb{R}^n$  is the Lagrange multiplier vector [13]. The problem in (32) is indeed a joint optimization problem where it can be solved via the optimization of  $\boldsymbol{\alpha}$  and  $\mathbf{z}$  alternatively and it is written as;

$$\begin{aligned} \boldsymbol{\alpha}^{t+1} &= \arg \min L_{\lambda}(\boldsymbol{\alpha}, \mathbf{z}^t) = \arg \min (\|\boldsymbol{\alpha}\|_1 + \frac{\lambda}{2} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + (\mathbf{z}^t)^T \mathbf{D}\boldsymbol{\alpha}) \\ \text{where } \mathbf{z}^{t+1} &= \mathbf{z}^t + \lambda(\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}^{t+1}) \end{aligned} \quad (33)$$

The final form of the problem in (33) is solved by using FISTA algorithm [13].

At each iteration, the parameter  $\mathbf{z}$  is updated until the convergence. Here, the method, so-called PALM, exploits ALM to solve the problem in (32).

#### 2.4.1.2.3.4 TwIST algorithm

The TwIST algorithm is also a proximity based optimization algorithm. In order to solve the sparse coding problem, two-step TwIST algorithm is defined as follows [43], where  $a$  and  $b$  are convergence rate parameters;

$$\begin{aligned} \boldsymbol{\alpha}^1 &= \theta(\boldsymbol{\alpha}^0) \\ \boldsymbol{\alpha}^{t+1} &= (1 - \boldsymbol{\alpha})\boldsymbol{\alpha}^{t-1} + (a - b)\boldsymbol{\alpha}^t + b\theta(\boldsymbol{\alpha}^t) \\ \text{where } \theta(\boldsymbol{\alpha}^t) &= \boldsymbol{\alpha}^t - \tau \mathbf{D}^T (\mathbf{D}\boldsymbol{\alpha}^t - \mathbf{x}) \end{aligned} \quad (34)$$

and updating approach relies on two previous values, leading to a two-step algorithm.

#### 2.4.1.2.3.5 GIST algorithm

The sparse coding equations in the form of non-convex problems with large class of non-convex penalties can be solved by using a general framework so-called GIST algorithm [45]. GIST is itself a proximity based optimization algorithm that solves a proximal operator problem iteratively.

At each iteration, a line search is performed which is initiated by the BB rule. Therefore, this algorithm can efficiently obtain the step size. The solution is summarized as follows;

$$\boldsymbol{\alpha}^{t+1} = \arg \min \frac{1}{2} \left\| \boldsymbol{\alpha} - \left( \boldsymbol{\alpha}^t - \nabla l(\boldsymbol{\alpha}^t) \frac{1}{s^t} \right) \right\|^2 + \frac{1}{s^t} r(\boldsymbol{\alpha}) \quad (35)$$

where  $1/s^t$  is the step size of gradient descent,  $-\nabla l(\mathbf{a}^t)$  is the direction of gradient descent,  $r(\mathbf{a})$  is the regularizer and  $l(\mathbf{a})$  is the loss function. GIST performs a gradient descent followed by a proximal operator problem solution.

Since different sparse coding problems have different regularizers, GIST constitutes a general framework for solving various types of sparse coding problems either convex or non-convex [45].

#### 2.4.1.2.4 Coordinate descent based optimization algorithms

Another way to solve the sparse coding problem is by utilizing the coordinate descent algorithms. These algorithms follow the strategy that optimization is performed to only one variable while all other variables are fixed [79]. Until reaching the convergence, the procedure iterates over all variables in a cyclic sense.

Coordinate descent algorithms cannot generate the solution path itself, indeed the solution path can only be approximated by employing warm-restart scheme.

##### 2.4.1.2.4.1 Shooting algorithm

Shooting algorithm is a coordinate wise optimization approach and it is designed for the solution of  $l_1$ -regularized sparse coding problem. The LASSO sparse coding problem can be solved by employing this shooting algorithm which aims to minimize the term  $(\mathbf{x} - \mathbf{D}\mathbf{a})^T(\mathbf{x} - \mathbf{D}\mathbf{a})$  subject to the following constraint [48, 80];

$$\sum |\mathbf{a}_j| \leq t, \text{ where } t \geq 0 \quad (36)$$

At each iteration, the shooting algorithm uses the current value of  $\mathbf{a}^{(j)}$  to solve the following problem from 1 to  $k$ ;

$$\min_{\mathbf{a}_j} h'(\alpha_j) = \frac{1}{2} \|\mathbf{x}_j - \mathbf{d}_j \alpha_j\|_2^2 + \lambda |\alpha_j| + \lambda \|\mathbf{a}^{(-j)}\|_1, \quad (37)$$

where  $\mathbf{x}_j = \mathbf{x} - \mathbf{D}^{(-j)} \mathbf{a}^{(-j)}$ .

As the number of iterations goes to infinity, then algorithm converges to the co-ordinate wise minimum [80].

#### 2.4.1.2.4.2 BCS PL-3DBS + 3DWPT algorithm

When the regularization term is generalized as the  $l_1/l_q$ -norm with groups constituting a partition, then the coordinate descent based algorithms are generalized as BCD algorithms [81]. The  $l_1/l_q$ -norm penalty is defined as follows;

$$\sum_{j=1}^J c_j \|\mathbf{a}_{G_j}\|_q \quad (38)$$

where  $q > 1$ ,  $G$  represents the partition of variables into groups and  $c_j$  is for varying group sizes [79].

The BCS-SPL algorithm is a very effective block based approach. Based on this algorithm, a similar BCS PL-3DBS algorithm is proposed where 3D WPT is also adapted into the procedure to obtain a better sparsity [82, 83].

The BCS PL-3DBS + 3DWPT algorithm is indeed a general compressive sensing framework rather than just solving the sparse coding problem. In this algorithm, a hyperspectral image of size  $n_l \times n_s \times n_b$  is partitioned into small non-overlapping 3D blocks of size  $B_{nl} \times B_{ns} \times B_{nb}$ . BCS PL-3DBS + 3DWPT algorithm employs biorthogonal 9/7 wavelet as the dictionary. Therefore, dictionary learning is not applied. Sparse coding is realized for each block by using projected Landweber onto the convex set and the soft thresholding algorithm [83]. The main concern here is to obtain a good reconstruction performance and the algorithm has a better performance in low bit rates.

#### 2.4.1.2.4.3 OBD-BCS algorithm

Classical compressive sensing approaches rely on the prior knowledge of the sparsity basis. On the other hand, a new compressive sensing scheme is introduced so-called blind compressive sensing. The term “blind” comes from the fact that this framework does not require the prior knowledge of the sparsity basis for sampling and reconstruction steps [59].

In order to guarantee the uniqueness of the solution in this case, some conditions are imposed on the sparsity basis. The unknown basis is called  $\mathbf{P}$  and it is assumed to be block diagonal such that the signals belonging to each channel are sparse under separate bases [59].

Here,  $\mathbf{A}$  is the measurement matrix that includes orthogonal bases. The number of blocks in  $\mathbf{P}$  is assumed to be an integer multiple of the number of orthogonal bases in  $\mathbf{A}$ . Another imposed condition is that  $\mathbf{P}$  should be orthogonal.

In the blind compressive sensing framework, which consists of both the sparse coding and the dictionary learning steps, a simple algorithm is proposed, namely OBD-BCS [59]. Algorithm has two alternating steps and it aims to find out the sparse coefficients as follows, without the knowledge of the sparsity basis;

$$\boldsymbol{\alpha} = \mathbf{P}\mathbf{S}. \quad (39)$$

It is assumed that  $\boldsymbol{\alpha}$  is sparse under a basis  $\mathbf{P}$  such that the sparse vector  $\mathbf{S}$  satisfies (39). In the first step, OBD-BCS algorithm computes  $\mathbf{S}$  by solving the corresponding sparse coding equation via OMP while  $\mathbf{P}$  is fixed.

In addition to the OMP, block coordinate relaxation (BCR) algorithm is also applied. It is implemented by dividing the elements of  $\mathbf{S}$  into blocks. Then, one block of  $\mathbf{S}$  is updated via soft thresholding at each iteration while the others are fixed. This sparse coding problem is given as follows and it is assumed that the maximum number of non-zero elements in  $\mathbf{S}$  is equal to  $k$ ;

$$\mathbf{S} = \arg \min \|\mathbf{x} - \mathbf{DPS}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{S}\|_0 \leq k \quad (40)$$

The main aim is not to find out the sparse basis  $\mathbf{P}$  and the sparse vector  $\mathbf{S}$  exactly, the algorithm is interested in finding the sparse coefficients  $\boldsymbol{\alpha}$  in (39). It is assumed that  $\mathbf{P}$  is an orthogonal 2-block diagonal matrix [59].

Second step follows the dictionary learning scheme by updating  $\mathbf{P}$  when  $\mathbf{S}$  is fixed by using SVD [59]. Here, the measurement vector  $\mathbf{x}$  is known and the structure of  $\mathbf{P}$  is imposed as block diagonal.

Any training set is not necessary in the OBD-BCS algorithm such that it is efficient to the cases when there is no access to the overall signals.

In this study, OBD-BCS algorithm is employed in order to find out the sparse coefficients  $\alpha$  which are the solutions of the sparse coding equation. According to (39), sparse coefficients are obtained from  $\mathbf{P}$  and  $\mathbf{S}$ . By using this vector  $\alpha$ , the online dictionary update step is performed afterwards.

#### 2.4.1.2.5 $l_p$ -norm sparse coding ( $0 \leq p \leq 1$ ) algorithms

The  $l_1$ -minimization problem is equivalent to that of  $l_0$ -minimization with high probability under certain conditions on the dictionary  $\mathbf{D}$  [49]. If the required conditions on  $\mathbf{D}$  are not satisfied, then  $l_1$ -minimization problem yields suboptimal solution.

Besides from the  $l_1$ -norm regularization type of sparse coding problems,  $l_p$ -norm sparse coding problems ( $0 \leq p \leq 1$ ) can also be used. Indeed,  $l_p$ -norm problems ( $0 \leq p \leq 1$ ) are sparser than those of  $l_1$ -norm and the corresponding sparse coding problem can be written as;

$$\min_{\alpha} \frac{1}{2} \|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda \|\alpha\|_p^p. \quad (41)$$

##### 2.4.1.2.5.1 GISA algorithm

The problem in (41) is an  $l_p$ -norm non-convex sparse coding problem. An algorithm so-called GISA can be exploited in the solution that uses soft thresholding [49].

First a simple type of  $l_p$ -minimization problem is taken into account as;

$$\min_{\alpha} \frac{1}{2} (\mathbf{x} - \alpha)^2 + \lambda |\alpha|^p. \quad (42)$$

In order to solve the  $l_p$ -norm sparse coding problem in (42), a generalized soft thresholding (GST) function is used such as;



$$T_p^{GST}(\boldsymbol{\alpha}; \lambda) = \begin{cases} 0, & \text{if } |\boldsymbol{\alpha}| \leq \tau_p^{GST}(\lambda) \\ \text{sgn}(\boldsymbol{\alpha}) S_p^{GST}(|\boldsymbol{\alpha}|; \lambda), & \text{if } |\boldsymbol{\alpha}| > \tau_p^{GST}(\lambda) \end{cases} \quad (43)$$

In (43), the following additional definitions are required as;

$$\tau_p^{GST}(\lambda) = (2\lambda(1-p))^{2-p} + \lambda p (2\lambda(1-p))^{2-p},$$

$$g_p(\theta; \lambda) = \theta + \lambda p \theta^{p-1}, \quad \theta_0 = (\lambda p (1-p))^{2-p}$$

when  $S_p^{GST}(\boldsymbol{\alpha}; \lambda)$  is the root of the equation  $g_p(\theta; \lambda) = |\boldsymbol{\alpha}|$ . GISA algorithm iteratively performs a gradient step and then a generalized thresholding step which can be defined as follows [49];

$$\boldsymbol{\alpha}^{(t+1)} = T_p^{GST}(\boldsymbol{\alpha}^{(t)} - \|\mathbf{D}\|^{-2} \mathbf{D}^T (\mathbf{D}\boldsymbol{\alpha} - \mathbf{x}); \|\mathbf{D}\|^{-2} \lambda). \quad (44)$$

#### 2.4.1.2.5.2 FOCUSS algorithm

Another way to solve  $l_p$ -norm problems ( $0 \leq p \leq 1$ ) is by using the FOCUSS algorithm which is itself an iterative reweighted least squares (IRLS) approximation [50, 84]. The loss function in FOCUSS is defined as;

$$l_{FOCUSS}(\boldsymbol{\alpha}) = \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_p^p. \quad (45)$$

The  $p$ -pseudo norm of  $\boldsymbol{\alpha}$  is expressed as such;

$$\|\boldsymbol{\alpha}\|_p^p = \boldsymbol{\alpha}^T W(\boldsymbol{\alpha}) \boldsymbol{\alpha} \quad (46)$$

If the current estimate of  $\boldsymbol{\alpha}$  is employed, then the  $p$ -pseudo norm is approximated by a weighted  $l_2$ -norm as follows [84];

$$W(\boldsymbol{\alpha}) = \text{diag}(|\boldsymbol{\alpha}_i|^{p-2}) \quad (47)$$

where  $i$  is from  $l$  to  $m$ . FOCUSS algorithm updates the diagonal matrix at each iteration based on the previous values.

### 2.4.1.3 Bayesian sparse coding algorithms

Bayesian approaches are used for solving sparse coding problem in (1) efficiently by assuming a prior distribution and maximizing the resulted joint probability to obtain the sparse coefficients [79].

#### 2.4.1.3.1 SBL algorithm

A typical example of the Bayesian sparse coding algorithm is SBL which seeks a probabilistic approach to solve a sparse coding problem [85]. It is aimed to find out the sparse coefficients  $\boldsymbol{\alpha} \in \mathbb{R}^m$  for the following general case which corresponds to a sparse coding problem;

$$\mathbf{x} = \mathbf{D}\boldsymbol{\alpha} + \boldsymbol{\varepsilon} \quad (48)$$

where  $\boldsymbol{\varepsilon}$  is an  $n$  dimensional vector. By utilizing from the Bayesian statistics, first the conditional probability  $P(\boldsymbol{\alpha}|\mathbf{x})$  is calculated. It is assumed that the noise  $\boldsymbol{\varepsilon}$  in the data is normally distributed with zero mean and variance  $\sigma^2$ . The likelihood of the data  $\mathbf{x}$  is then defined as;

$$P(\mathbf{x}|\boldsymbol{\alpha}, \sigma^2) = (2\pi)^{-n/2} |\mathbf{B}|^{1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{D}\boldsymbol{\alpha})^T \mathbf{B}(\mathbf{x} - \mathbf{D}\boldsymbol{\alpha})\right) \quad (49)$$

where  $\mathbf{B} = \sigma^{-2}\mathbf{I}$ . The Bayes' theorem is given as follows [85] such that  $h^1$  and  $h^2$  are the hyper-parameters of a Gamma distribution;

$$P(\boldsymbol{\alpha}|\mathbf{x}, h^1, h^2) = \frac{P(\mathbf{x}|\boldsymbol{\alpha}, h^2)P(\boldsymbol{\alpha}|h^1)}{P(\mathbf{x}|\boldsymbol{\alpha}, h^2)} \quad (50)$$

SBL algorithm employs an iterative approach to calculate the hyper-parameters, then sparse coefficients are estimated accordingly. Indeed, SBL algorithm is equivalent to the iterative weighted solution of the  $l_1$ -norm minimization sparse coding problem [68].

### 2.4.1.3.2 RLPHCS\_Cov algorithm

Recently, a novel reweighted Laplace prior based hyperspectral compressive sensing (RLPHCS) method, namely, RLPHCS\_Cov algorithm is proposed in the literature and it outperforms several state-of-the-art hyperspectral compressive sensing algorithms (OMP, StOMP, LASSO, Bayesian compressive sensing (BCS), reweighted  $l_1$ -norm based compressive sensing (RCS) and low rank/joint sparsity based hyperspectral compressive sensing (LRJS) ) in terms of the reconstruction performance [86]. Therefore, throughout this study RLPHCS\_Cov algorithm is assumed to be a state-of-the-art hyperspectral compression algorithm and it is used for comparison. RLPHCS\_Cov is indeed a compressive sensing framework which employs a hierarchical reweighted Laplace prior to model the distribution of sparsity together with a variable-based Bayesian model to obtain the optimal configuration of the prior to carry out the sparse coding step [86]. Haar wavelet based orthogonal dictionary is used. Therefore, dictionary learning is not performed.

In many compressive sensing algorithms in the literature, sparsity regularization terms such as  $l_0$ -norm or  $l_1$ -norm cannot take into account the structure information among the sparse coefficients and they cannot adapt themselves to the unknown noise. The motivation behind the state-of-the-art RLPHCS\_Cov algorithm is that the distribution of structured sparsity is modeled and adapted to the unknown noise [86]. The structured sparsity is modeled by utilizing a full covariance matrix. It is aimed to improve the reconstruction performance. The sparse signal  $\alpha$  is represented by a normal distribution for a hyperspectral image of size  $n_l \times n_s \times n_b$  as follows;

$$P(\alpha|\gamma) = \frac{\exp\left\{-\frac{1}{2}\|\alpha\|_{diag(\gamma)}^2\right\}}{(2\pi)^{(n_b n_l n_s)/2} |diag(\gamma)|^{(n_l n_s)/2}} \quad (51)$$

where  $diag(\gamma)$  is the  $\gamma$ -dependent diagonal matrix,  $\gamma=[\gamma_l, \dots, \gamma_{n_b}]^T$  is used to control the variation of each row in vector  $\alpha$ . Gamma distribution is assumed on the unknown  $\gamma$  and the proposed reweighted Laplace prior can capture the structure in the sparse signal [86].

Then, in order to obtain the optimal configuration of the reweighted Laplace prior together with the noise variance, a Bayesian model is exploited.

### 2.4.2 Dictionary learning

In classical dictionary learning problems, the aim is to minimize the following empirical cost function;

$$f_n(\mathbf{D}) = \frac{1}{n} \sum_{i=1}^n l(\mathbf{x}_i, \mathbf{D}), \quad (52)$$

where  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$  in  $R^{m \times n}$  is the finite training set of signals and  $\mathbf{D} \in R^{m \times k}$  is the dictionary. Here,  $n$  represents the number of samples,  $k$  refers to the number of columns in the dictionary,  $m$  is the signal dimension and  $l(\mathbf{x}, \mathbf{D})$  is the loss function. This loss function is defined as the optimal value of the  $l_1$ -regularized sparse coding problem in the literature [19] as;

$$l(\mathbf{x}, \mathbf{D}) = \min_{\boldsymbol{\alpha} \in R^k} \frac{1}{2} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1, \quad (53)$$

where  $\lambda$  is the regularization parameter. In this case, the sparse solution comes from the  $l_1$  penalty. The loss function in (53) can be different depending on the learning approach, as an example the one used in the reference study [19] is considered.

Dictionary learning is a typical example of the implementation of sparse coding. The first stage of the dictionary learning relies on the solution of sparse coding equation to obtain sparse coefficients.

#### 2.4.2.1 Online learning methods vs batch methods

It is important to point out the differences between online learning and batch methods.

Batch methods use the overall dataset at each iteration, yielding a huge amount of memory requirement [19]. Batch approaches learn a dictionary directly from the entire dataset [79]. If  $N$  training samples are used, then batch mode would perform the update only once [87]. In the case of sequentially arriving samples, such methods are not effective [88].

Online learning methods draw and process one sample from the entire dataset at a time. The learning process can continue by updating the current dictionary whenever a new data sample arrives. Therefore, these methods are suitable for large datasets. As an example, if  $N$  training samples are used, then online learning mode would perform the update  $N$  times [87]. It is assumed that the training set includes a set of *i.i.d* samples of a distribution  $p(\mathbf{x})$  and the method draws one element  $\mathbf{x}_n$  at a time [19]. Algorithm alternates between sparse coding and dictionary update steps afterwards. It is very common in online algorithms that *i.i.d* sampling of the unknown distribution  $p(\mathbf{x})$  is simulated by cycling over a randomly permuted training set, since the same data sample can be used several times [18].

The empirical loss function  $f_n(\mathbf{D})$  in (52) is indeed the batch version where it is obtained at once by solving the sparse coding problem using all the samples. On the other hand, the surrogate function  $\hat{f}_n(\mathbf{D})$ , which will be defined later in this section, is the online version where it uses the sparse coefficients from the earlier iterations [79]. This surrogate cost function upper-bounds the empirical cost function. The key contribution of the online dictionary learning method proposed by Mairal et. al. is to show that both the function in  $f_n(\mathbf{D})$  and in  $\hat{f}_n(\mathbf{D})$  converge almost surely to the same limit [19].

Employing dictionary learning with iterative batch processing scheme is proven to be inefficient for hyperspectral data [89]. In such large scale data, using dictionary learning based on online learning approach is more efficient.

In literature various examples can be found for online learning methods and batch methods. In this study, the online dictionary learning method proposed by Mairal et. al. and K-SVD are selected as the examples of online learning methods and batch methods, respectively. Now, these example methods will be explained in detail.

#### **2.4.2.1.1 K-SVD algorithm**

K-SVD is considered as an iterative algorithm. The dictionary learning problem in (52) is solved in two stages. In the first stage, the dictionary  $\mathbf{D}$  is fixed and  $\boldsymbol{\alpha}$  is minimized by solving the following sparse coding problem  $n$  times;

$$\min_{\mathbf{a}_i} \|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2^2 \quad \text{s.t.} \quad \|\mathbf{a}_i\|_0 \leq k. \quad (54)$$

Here,  $l_0$ -norm penalty is employed which yields the sparsest solution [13]. In the second stage, at each iteration  $\mathbf{a}$  is fixed and dictionary  $\mathbf{D}$  is optimized, where  $i = 1, \dots, n$ . K-SVD algorithm updates one column of the dictionary  $\mathbf{D}$  at a time while fixing the other columns [90]. It is a simple problem and based on SVD approach.

This learning approach is related to the minimization of the empirical cost function on the training data. K-SVD is considered as a batch method which corresponds to accessing the whole training set in the learning process.

As an example of the usage of batch method dictionary learning for hyperspectral image compression, an algorithm is proposed in the literature [78]. The method is the faster version of the joint orthogonal matching pursuit (JOMP) algorithm. According to the experimental results, this proposed method, namely fast joint orthogonal matching pursuit (FJOMP), outperforms the wavelet based compression algorithms at low bit rates in terms of the compression performance. FJOMP includes dictionary learning, sparse coding and entropy coding. Dictionary learning is employed by using K-SVD and sparse coding is realized by OMP algorithm [78].

#### 2.4.2.1.2 Online dictionary learning algorithm

In the work [19], an online dictionary learning algorithm is proposed which is suitable for large datasets. At each iteration, new sample  $\mathbf{x}_n$  is drawn from the data distribution and the dictionary is updated accordingly. The sample  $\mathbf{x}_n$  at iteration  $n$  is utilized to update previous dictionary  $\mathbf{D}_{n-1}$ . At the end of iteration  $n$ ,  $\mathbf{D}_n$  is obtained. This alternating process is repeated until a stopping criterion is realized.

Online dictionary learning is itself an iterative process which has two stages. In the first stage, the following sparse coding problem is solved;

$$\mathbf{a}_n = \min_{\mathbf{a}} \frac{1}{2} \|\mathbf{x}_n - \mathbf{D}_{n-1}\mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_1 \quad (55)$$

In the second stage, coefficients obtained from (55) are fixed and previous dictionary  $\mathbf{D}_{n-1}$  is updated to find the new dictionary  $\mathbf{D}_n$  at iteration  $n$ . Therefore, the following optimization problem should be solved;

$$\mathbf{D}_n = \min_{\mathbf{D}} \frac{1}{t} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|^2 + \lambda \|\mathbf{a}_i\|_1 \quad (56)$$

Since the coefficients are fixed, they can be removed and the objective function in (56) is adjusted as;

$$\mathbf{D}_n = \min_{\mathbf{D}} \frac{1}{t} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|^2. \quad (57)$$

If the matrix notation of the square-loss function in (57) is considered, then it can be rewritten as;

$$\begin{aligned} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|^2 &= \sum_{i=1}^n \text{trace}((\mathbf{x}_i - \mathbf{D}\mathbf{a}_i)(\mathbf{x}_i - \mathbf{D}\mathbf{a}_i)^T) \\ &= \sum_{i=1}^n \text{trace}((\mathbf{x}_i\mathbf{x}_i^T - \mathbf{x}_i\mathbf{a}_i^T\mathbf{D}^T - \mathbf{D}\mathbf{a}_i\mathbf{x}_i^T + \mathbf{D}\mathbf{a}_i\mathbf{a}_i^T\mathbf{D}^T)) \\ &= \sum_{i=1}^n (\text{trace}(\mathbf{x}_i\mathbf{x}_i^T) - 2\text{trace}(\mathbf{x}_i\mathbf{a}_i^T\mathbf{D}^T) + \text{trace}(\mathbf{D}^T\mathbf{D}\mathbf{a}_i\mathbf{a}_i^T)) \\ &= \text{trace}(\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T) - 2\text{trace}(\mathbf{D}^T \sum_{i=1}^n \mathbf{x}_i\mathbf{a}_i^T) + \text{trace}(\mathbf{D}^T\mathbf{D} \sum_{i=1}^n \mathbf{a}_i\mathbf{a}_i^T) \end{aligned}$$

If  $\mathbf{A}_n$  and  $\mathbf{B}_n$  are defined as;

$$\mathbf{A}_n = \sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T \in R^{k \times k}, \quad \mathbf{B}_n = \sum_{i=1}^n \mathbf{x}_i\mathbf{a}_i^T \in R^{m \times k} \quad (58)$$

and the first term is discarded since it does not depend on  $\mathbf{D}$ , then the final form is reorganized as follows;

$$\mathbf{D}_n = \min_{\mathbf{D}} \text{trace}(\mathbf{D}^T\mathbf{D}\mathbf{A}_n) - 2\text{trace}(\mathbf{D}^T\mathbf{B}_n). \quad (59)$$

The problem in (59) can be solved via the so-called block coordinate descent (BCD) algorithm by using  $\mathbf{D}_{n-1}$  as the warm restart for  $\mathbf{D}_n$ . The optimization problem (59) is only considered over the  $j$ -th column  $\mathbf{d}_j$  of  $\mathbf{D}_{n-1}$  while all other columns are fixed such that  $\mathbf{d}_j$  is updated by the following formula;

$$\frac{1}{(\mathbf{A}_n)_{jj}} (\mathbf{B}_n(:, j) - \mathbf{D}\mathbf{A}_n(:, j)) + \mathbf{d}_j. \quad (60)$$

If  $\|\mathbf{d}_j\|_2 > 1$  is obtained, then it is necessary to normalize the value to unit norm. Sequentially all the columns of  $\mathbf{D}$  are updated by changing  $j$ . The updated columns are employed to update the remaining ones [90]. The BCD algorithm used in the dictionary update stage solves (60) until convergence of  $\mathbf{D}$ , such as;

$$\mathbf{u}_j \leftarrow \frac{1}{\mathbf{A}_{jj}} (\mathbf{b}_i - \mathbf{D}\mathbf{a}_j) + \mathbf{d}_j, \quad \mathbf{d}_j \leftarrow \frac{1}{\max(\|\mathbf{u}_j\|_2, 1)} \mathbf{u}_j. \quad (61)$$

Here using (61), each column  $\mathbf{d}_j$  of the dictionary is updated. Each time when all the columns (blocks) are updated, the difference between  $\mathbf{D}$  and  $\mathbf{D}_{n-1}$  is calculated. If this difference is smaller than a specified threshold value, then the updated dictionary  $\mathbf{D}$  is utilized for the next iteration. Since  $\mathbf{D}_{n-1}$  is used as a warm restart for obtaining  $\mathbf{D}$  at the current iteration, a single iteration is empirically proven to be enough for updating the dictionary [19].

The study in the literature [19] follows the same strategy up to now in order to realize the online learning algorithm based on stochastic approximations. This study states that when the training set size goes to infinity in theory, then the expected cost  $f(\mathbf{D})$  is minimized instead of the empirical cost  $f_n(\mathbf{D})$ :

$$f(\mathbf{D}) \triangleq E_{\mathbf{x}}[l(\mathbf{x}, \mathbf{D})] = \lim_{n \rightarrow \infty} f_n(\mathbf{D}) \quad (62)$$

The objective function is defined as;

$$\hat{f}_n(\mathbf{D}) \triangleq \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2^2 + \lambda \|\mathbf{a}_i\|_1 \right). \quad (63)$$



The quadratic function  $\hat{f}_n(\mathbf{D})$  in (63) aggregates the previous  $\alpha_i$  values. Therefore, it is the surrogate of the empirical cost function  $f_n(\mathbf{D})$  in (52) [19]. Online dictionary learning approach is quite different than the batch methods such as K-SVD, since it utilizes one sample at a time without accessing the overall data set. This is why online dictionary learning approach is suitable for large scale datasets such as hyperspectral data. Dynamic datasets such as video sequences are also compatible with this approach.

## 2.5 CONTRIBUTIONS

The online dictionary learning method proposed by Mairal et.al. is chosen to be applied in this study due to its ability to efficiently solve large scale problems with millions of training samples. Hyperspectral images are inherently classified as large scale datasets and this method is well suited to the case of hyperspectral image compression task. The various solution strategies of sparse representation algorithms have different effects on the overall compression performance. This study mainly focuses on this aspect such that online dictionary learning method is adapted with many different sparse representation algorithms to solve the sparse coding equation in the first stage.

Up to now, the online dictionary learning method proposed by Mairal et.al. [19] is explained. Additionally, various dictionary learning and sparse coding algorithms are introduced briefly. It is now useful to summarize the main contributions as well as the novelties of this study to the literature.

The contribution of this study is twofold;

1-To the best of our knowledge, this is the first piece of work that adapts online dictionary learning method proposed by Mairal et. al. to the case of hyperspectral image compression.

2-Additionally, a hybrid hyperspectral compression method is proposed such that the sparse coding stage of the method is employed by many different sparse coding schemes. The effect of sparse coding problem on the general hyperspectral compression framework is investigated. This is the first time that online dictionary learning method proposed by Mairal et. al. is used with many different sparse coding algorithms.

## CHAPTER 3

### HYBRID HYPERSPECTRAL IMAGE COMPRESSION METHOD BY USING ONLINE DICTIONARY LEARNING BASED ON SPARSE CODING

In this section, hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding is explained. Here, the adaptation of online dictionary learning method proposed by Mairal et. al. to the case of hyperspectral image compression is introduced. The online dictionary learning method relies on stochastic approximations and it is suitable for large scale datasets such as hyperspectral images [18, 19]. In this study, the iterative online dictionary learning algorithm is used, which minimizes the surrogate function of the empirical cost under particular constraints at each iteration [18]. Then, various sparse coding algorithms are implemented to constitute a hybrid method.

#### 3.1 PROBLEM STATEMENT

The online dictionary learning method proposed by Mairal et. al. which is explained in Chapter 2 is adapted to the case of hyperspectral image compression. Here, the corresponding notation and adaptation will be presented.

The number of bands in the hyperspectral image is represented by  $n_b$ , the number of lines in the hyperspectral image is represented by  $n_l$ , the number of samples in the hyperspectral image is represented by  $n_s$  and the number of columns in the dictionary is defined as  $k$ .

The initial dictionary is expressed as  $\mathbf{D}_0 \in \mathbb{R}^{n_b \times k}$ . The auxiliary matrices for updating the dictionary are defined as  $\mathbf{A}_0 \in \mathbb{R}^{k \times k}$  and  $\mathbf{B}_0 \in \mathbb{R}^{n_b \times k}$ . The number of iterations is defined as  $T$ , the error is expressed as  $\mathbf{E} \in \mathbb{R}^{k \times l}$ , the regularization parameter is defined as  $\lambda \in \mathbb{R}$ , and the sparse coefficients are expressed as  $\boldsymbol{\alpha} \in \mathbb{R}^k$ .

In the dictionary learning process, optimization is performed on the empirical cost by considering a finite training set  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T]$  in  $R^{nbxT}$  containing all bands at some pixel [92]. The empirical cost is given as

$$f_T(\mathbf{D}) \triangleq \frac{1}{T} \sum_{i=1}^T l(\mathbf{x}_i, \mathbf{D}), \quad (64)$$

where  $\mathbf{D} \in R^{nbxk}$  represents the dictionary and  $l$  expresses the loss function. This loss function corresponds to the optimal value of  $l_1$ -norm sparse coding [19] given as

$$l(\mathbf{x}_t, \mathbf{D}) \triangleq \min_{\boldsymbol{\alpha} \in R^k} \frac{1}{2} \|\mathbf{x}_t - \mathbf{D}\boldsymbol{\alpha}_t\|_2^2 + \lambda \|\boldsymbol{\alpha}_t\|_1, \quad (65)$$

where  $\lambda$  represents the regularization parameter,  $\mathbf{x}_t$  expresses the training sample at iteration  $t$  and  $\boldsymbol{\alpha}_t$  defines the coefficient set at iteration  $t$ . In (65),  $l_1$  regularization ensures the sparsity.

A convex set of matrices  $\mathbf{C}$  is defined to avoid arbitrarily large elements in  $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_k]$  as well as arbitrarily small values of  $\boldsymbol{\alpha}_t$ . This convex set  $\mathbf{C}$  is given as

$$\mathbf{C} \triangleq \left\{ \mathbf{D} \in R^{nbxk} : \|\mathbf{d}_j\| \leq 1, \forall j = 1, \dots, k \right\}. \quad (66)$$

The minimization of the empirical cost  $f_T(\mathbf{D})$  with respect to  $\mathbf{D}$  is not convex. Therefore, it is modified as a joint optimization problem. The modified optimization problem is convex when the sparse coefficients  $\boldsymbol{\Gamma} = [\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_T] \in R^{kxT}$  are fixed, while the optimization is performed with respect to  $\mathbf{D}$ , and when  $\mathbf{D}$  is fixed while the optimization is performed with respect to sparse coefficients  $\boldsymbol{\Gamma}$ . This joint optimization problem is as follows:

$$\min_{\mathbf{D} \in \mathbf{C}, \boldsymbol{\Gamma} \in R^{kxT}} \frac{1}{T} \sum_{i=1}^T \left( \frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1 \right). \quad (67)$$

Equation (67) is solved as a convex optimization problem such that  $\mathbf{D}$  is minimized when  $\boldsymbol{\Gamma}$  is fixed, and  $\boldsymbol{\Gamma}$  is minimized when  $\mathbf{D}$  is fixed, respectively.

Instead of minimizing the empirical cost  $f_T(\mathbf{D})$ , minimizing the expected cost  $f(\mathbf{D})$  is much more computationally efficient [17]. This expected cost is given as

$$f(\mathbf{D}) \triangleq E_x[l(\mathbf{x}, \mathbf{D})] = \lim_{T \rightarrow \infty} f_T(\mathbf{D}), \quad (68)$$

where the unknown probability distribution of the data is utilized to find out the expectation. In the literature, it has been proved that the equality in (68) converges with the probability one [19].

For large scale data sets such as hyperspectral images, stochastic gradient algorithms provide a better rate of convergence [17, 19]. Therefore, in this study dictionary learning is realized by using projected first order stochastic gradient descent algorithm. According to this algorithm, dictionary  $\mathbf{D}$  is updated sequentially and is shown as [93];

$$\mathbf{D}_t = \Pi_C \left[ \mathbf{D}_{t-1} - \frac{\rho}{t} \nabla_D l(\mathbf{x}_t, \mathbf{D}_{t-1}) \right], \quad (69)$$

where  $\mathbf{D}_t$  represents the optimal dictionary at iteration  $t$ ,  $\rho$  presents the gradient step, and  $\Pi_C$  shows the orthogonal projector on  $\mathbf{C}$ . It is assumed that the training set  $\mathbf{X}$  has *i.i.d.* samples of the unknown distribution of the particular data [19].

### 3.2 ALGORITHM

In this study, an algorithm which consists of two parts is used. These two parts are solved alternately.

The sparse coding equation is solved by using  $\mathbf{x}_t$  from the current iteration, and  $\mathbf{D}_{t-1}$  from the previous iteration. When  $\boldsymbol{\alpha}_t$  is found, the following  $\hat{f}_t(\mathbf{D})$  function is minimized over set  $\mathbf{C}$  to obtain an updated dictionary  $\mathbf{D}_t$ :

$$\hat{f}_t(\mathbf{D}) \triangleq \frac{1}{t} \sum_{i=1}^t \frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1, \quad (70)$$

where  $\boldsymbol{\alpha}_i$  values are obtained.

In the literature, it has been proven that the empirical cost  $f_t(\mathbf{D})$  and the function  $\hat{f}_t(\mathbf{D})$ , which is quadratic in the  $\mathbf{D}$ , converge almost surely to the same limit [18]. Therefore, the function  $\hat{f}_t$  is the surrogate for the function  $f_t$ .

For the large values of  $t$ , the function  $\hat{f}_t$  is close to  $\hat{f}_{t-1}$  function. In these circumstances,  $\mathbf{D}_t$  is also close to  $\mathbf{D}_{t-1}$  such that it is effective to use  $\mathbf{D}_{t-1}$  as a warm restart for finding  $\mathbf{D}_t$ .

Now, Algorithm 1 and Algorithm 2 will be discussed. These two algorithms are solved alternately.

### 3.2.1 Algorithm 1

In the first part of the algorithm, the sparse coding is performed. The minimization of (65) corresponds to the sparse coding operation of the  $l_1$ -regularized linear least squares problem.

TNIPM algorithm is utilized to solve this problem in the online dictionary learning method proposed by Mairal et. al. Since the function can be different than (65), in general, the optimization function is called *sparse coding equation* throughout this study. Various sparse coding equations belonging to different sparse representation algorithms are presented in Table 1 and Table 2.

Online learning takes place in this algorithm where at each iteration a random training sample is chosen from the hyperspectral data. At the end, only  $T$  number of samples are selected from the overall data to obtain the final learned dictionary. The corresponding error is calculated at each iteration when a new sample arrives. Therefore, if a new data sample arrives, the algorithm can update the error and the dictionary accordingly, without processing the whole dataset. The advantage of this online fashion is that large datasets as well as the dynamic ones can be processed effectively.

The initial value of the dictionary is computed randomly on set  $\mathbf{C}$ . At each iteration from  $1$  to  $T$ , sparse coding problem is solved via one of the different sparse coding algorithms. By using the obtained sparse coefficients, the parameters  $\mathbf{A}$  and  $\mathbf{B}$  are updated. Algorithm

2 is performed afterwards. After  $T$  number of iterations, the final learned dictionary is obtained. Algorithm details are depicted below.

### 3.2.2 Algorithm 2

In the second part of the algorithm dictionary is updated by utilizing the block-coordinate descent with the warm restart.  $\mathbf{D}_{t-1}$  is used as a warm restart. Each column of the dictionary  $\mathbf{D}_t$  is updated by employing step 4 in Algorithm 2 which is indeed equivalent to solving the problem in (70) [19]. The calculations from (57) to (61) from the previous chapter show the detailed description of this equality.

---

#### Algorithm 1

---

- 1: Construct initial dictionary  $\mathbf{D}_0$  randomly
  - 2: Initial values of matrices  $\mathbf{A}_0$  and  $\mathbf{B}_0$  are set to zero
  - 3: **for**  $t = 1$  to  $T$  **do**
  - 4:   Choose random  $\mathbf{x}_t \in R^{n_b}$  values from the hyperspectral image
  - 5:   Solve “*sparse coding equation*” (cf. Table 1 and Table 2)
  - 6:   Update  $\mathbf{A}_t = \mathbf{A}_{t-1} + \alpha_t \alpha_t^T$  and  $\mathbf{B}_t = \mathbf{B}_{t-1} + \mathbf{x}_t \alpha_t^T$
  - 7:   Using **Algorithm 2** find  $\mathbf{D}_t$
  - 8: **end for**
  - 9: Acquire the final learned dictionary  $\mathbf{D}_T$
- 

---

#### Algorithm 2

---

- 1: **repeat**
  - 2: **for**  $j = 1$  to  $k$  **do**
  - 3:   Update  $j^{\text{th}}$  column of  $\mathbf{D}_t$
  - 4:    $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_k] \in R^{n_b \times k}$ ,  $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_k] \in R^{k \times k}$  and  $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_k] \in R^{n_b \times k}$
  - 5:    $\mathbf{u}_j = \frac{1}{A(j, j)} (\mathbf{b}_j - \mathbf{D} \mathbf{a}_j) + \mathbf{d}_j$  and  $\mathbf{d}_j = \frac{1}{\max(\|\mathbf{u}_j\|_2, 1)} \mathbf{u}_j$  and
  - 6:    $E_j = \sqrt{\sum_{n_b} |\mathbf{d}_j^t - \mathbf{d}_j^{t-1}|^2}$
  - 7: **end for**
  - 8:  $E = \frac{1}{k} \sum_{j=1}^k E_j$
  - 9: **until**  $E < \text{Threshold}$
  - 10: Use updated dictionary  $\mathbf{D}$  in **Algorithm 1**
-

This is the algorithm where the dictionary updating part proposed by Mairal et.al. actually takes place. The steps from 2 to 6 should continue until the desired threshold value is greater than that of the error  $E$ . The learned dictionary is then used in Algorithm 1 as the dictionary at that particular iteration. The details of Algorithm 2 can be found below.

Here, compression is realized by finding a learned dictionary which is adapted particularly to the data. Linear combination of dictionary elements and sparse coefficients will represent the real data.

The algorithm which is explained up to now is indeed a framework and this framework is named as the *hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding* throughout this study. The term “hybrid” comes from the fact that various sparse coding problems are adapted to solve the sparse coding problem in Algorithm 1.

The following set of constraints are utilized in this study:

1 – To prevent from large dictionary elements, a convex set in equation (66) is applied.

2 –Non-negativity constraint is imposed on the sparse coefficients and dictionary elements [15]:  $\alpha_k \geq 0$ ,  $\mathbf{d}_{ij} \geq 0$ ,  $\forall i = 1, \dots, n_b$  and  $\forall j = 1, \dots, k$

Table 1 and Table 2 present sparse coding equations of various sparse representation algorithms, where  $t = 1, \dots, T$ ,  $\theta > 0$ ,  $\Psi \in R^{k \times l}$  and  $0 < p \leq 1$ . Indeed, sparse representation algorithms discussed in Chapter 2 in detail are summarized in Table 1 and Table 2.

Sparse representation algorithms are divided into two tables: Table 1 contains more recent algorithms such as those classified as blind compressive sensing and proximity based optimization algorithms. Table 2, contains the older versions of the sparse representation algorithms in the literature. It is important to note that, the online dictionary learning method proposed by Mairal et. al. employs only TNIPM algorithm as the sparse representation algorithm to solve the sparse coding problem. In this study, in addition to this TNIPM algorithm, various different sparse representation algorithms are exploited to

find out the most effective sparse representation algorithm for the case of hyperspectral image compression.

Table 1 Sparse representation algorithms (OBD-BCS, TwIST, FOCUSS, Shooting, PALM, SpaRSA, GIST and FISTA) with corresponding sparse coding equations (this table contains the most recent sparse representation algorithms in the literature)

ALGORITHM	SPARSE CODING EQUATION
OBD-BCS	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \frac{1}{2} \ \mathbf{x}_t - \mathbf{D}_{t-1} \mathbf{a}\ _2^2 \text{ s.t. } \ \mathbf{a}\ _0 \leq k$
TwIST	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \frac{1}{2} \ \mathbf{x}_t - \mathbf{D}_{t-1} \mathbf{a}\ _2^2 + \lambda \ \mathbf{a}\ _1$
FOCUSS	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \ \mathbf{a}\ _p^p \text{ s.t. } \mathbf{D}_{t-1} \mathbf{a} = \mathbf{x}_t$
Shooting	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \ \mathbf{x}_t - \mathbf{D}_{t-1} \mathbf{a}\ _2^2 + \lambda \ \mathbf{a}\ _1$
PALM	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \ \mathbf{a}\ _1 \text{ s.t. } \mathbf{D}_{t-1} \mathbf{a} = \mathbf{x}_t$
SpaRSA	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \frac{1}{2} \ \mathbf{x}_t - \mathbf{D}_{t-1} \mathbf{a}\ _2^2 + \lambda \ \mathbf{a}\ _1$
GIST	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \frac{1}{2n_b} \ \mathbf{D}_{t-1} \mathbf{a} - \mathbf{x}_t\ _2^2 + \lambda \sum_{j=1}^k \min( \mathbf{a}_j , \theta)$
FISTA	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \frac{1}{2} \ \mathbf{D}_{t-1} \mathbf{a} - \mathbf{x}_t\ _2^2 + \lambda \ \mathbf{a}\ _1$

The hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding which is described in this chapter is based on the online dictionary learning method proposed by Mairal et. al. It is indeed the adaptation of online dictionary learning method proposed by Mairal et. al. to the hyperspectral image compression case. Then, by adapting different sparse representation algorithms to solve the sparse coding



problem, a hybrid method is constructed. It is aimed to use different sparse representation algorithms from each category. The most basic and recent ones are selected.

Table 2 Sparse representation algorithms (SBL, gOMP, LASSO (ADMM), BP (Dual active-set), BPDN (Homotopy), GISA and TNIPM) with corresponding sparse coding equations (this table contains the older versions of the sparse representation algorithms in the literature)

ALGORITHM	SPARSE CODING EQUATION
SBL	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \frac{1}{2} \ \mathbf{x}_t - \mathbf{D}_{t-1} \mathbf{a}\ _2^2 + \frac{1}{2} \sum_{j=1}^k (\Psi'_j \mathbf{a}_j^2)$
gOMP	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \frac{1}{2} \ \mathbf{x}_t - \mathbf{D}_{t-1} \mathbf{a}\ _2$
LASSO (ADMM)	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \frac{1}{2} \ \mathbf{D}_{t-1} \mathbf{a} - \mathbf{x}_t\ _2^2 + \lambda \ \mathbf{a}\ _1$
BP (Dual active set)	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \ \mathbf{a}\ _1 \text{ s.t. } \mathbf{D}_{t-1} \mathbf{a} = \mathbf{x}_t$
BPDN (Homotopy)	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \frac{1}{2} \ \mathbf{x}_t - \mathbf{D}_{t-1} \mathbf{a}\ _2^2 + \lambda \ \mathbf{a}\ _1$
GISA	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \frac{1}{2} \ \mathbf{x}_t - \mathbf{D}_{t-1} \mathbf{a}\ _2^2 + \lambda \ \mathbf{a}\ _p^p$
TNIPM	$\mathbf{a}_t = \arg \min_{\mathbf{a} \in R^k} \frac{1}{2} \ \mathbf{x}_t - \mathbf{D}_{t-1} \mathbf{a}\ _2^2 + \lambda \ \mathbf{a}\ _1$

### 3.3 RATE-DISTORTION RESULTS

In this section, rate-distortion results of the hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding are provided [64]. Different sparse representation algorithms are adapted to solve the sparse coding problem in the first step of the method. The rate-distortion performances of those different hybrid cases are compared with each other and with that of the other compression algorithms such as BCS PL-3DBS + 3DWPT, and CPPCA. Additionally, the results are compared

with a state-of-the-art hyperspectral compression algorithm so-called RLPHCS\_Cov algorithm. The quality metric tool is the PSNR. The bit rate  $r$  is calculated in terms of the bits per sample (bps), and the formulation is as follows

$$r = \frac{z}{n_b} \cdot (b_d), \quad z < k, \quad (71)$$

where,  $z = \|\alpha\|_0$  represents the number of sparse coefficients,  $k$  defines the size of the dictionary  $D$ ,  $n_b$  is the number of bands in  $\mathbf{X}$ , and  $b_d$  represents the bit depth of  $\mathbf{X}$ .

### 3.3.1 Datasets

The information about hyperspectral datasets used in this study which are taken from the AVIRIS, HYPERION, and ROSIS sensors can be found in Table 3. The grayscale images of some of these datasets are illustrated in Figure 1.

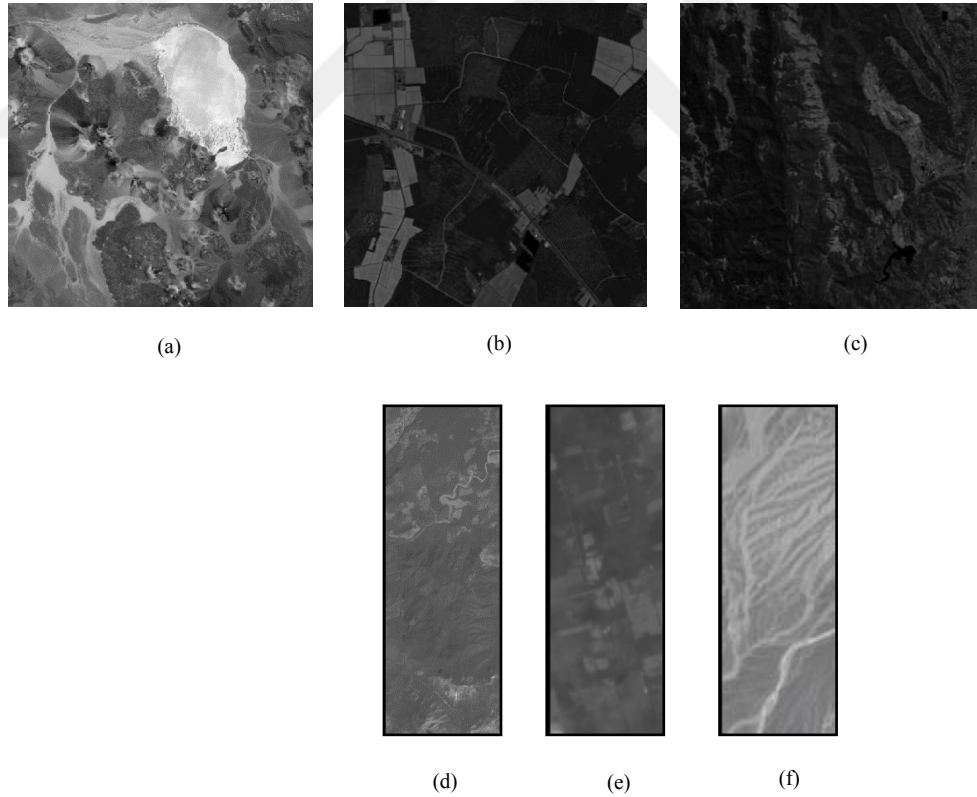


Figure 1 Grayscale image illustrations of (a) Lunar Lake (b) Low Altitude (c) Jasper Ridge (d) Mt. St. Helens (e) Lake Monona (f) Erta Ale hyperspectral datasets

AVIRIS is an airborne hyperspectral sensor. It aims to analyze the structure of the Earth surface and that of the atmosphere which has been flown by NASA/Jet Propulsion Laboratory (JPL) since 1984 [94]. Most of the AVIRIS hyperspectral images are archived by the NASA which are publicly available covering the years from 1999 to 2008 [95].

Table 3 Information about the hyperspectral datasets used in this study which are taken from AVIRIS, HYPERION and ROSIS sensors

<b>AVIRIS SENSOR - HYPERSPECTRAL DATA</b>						
<b>Name</b>	<b>No. Samples</b>	<b>No. Lines</b>	<b>No. Bands</b>	<b>Bit depth</b>	<b>Flight Number</b>	<b>Year</b>
Jasper Ridge	614	2587	224	16	f970403t01p02_r03	1997
Lunar Lake	614	1432	224	16	f970623t01p02_r07	1997
Low Altitude	614	3689	224	16	f960705t01p02_r05	1996
Salinas-A	83	86	204	12	f980717t01p02_r06	1998
Indian Pines	145	145	220	12	f920612t01p02_r03	1992
Yellowstone (uncalibrated scene 0)	680	512	224	16	f060925t01p00r12	2006
Cuprite	614	512	224	16	f970619t01p02r02	1997
Moffett Field	753	1924	224	16	f080611t01p00r07	2008
Lunar Lake	781	6955	224	16	f090819t01p00r06	2009
<b>HYPERION SENSOR - HYPERSPECTRAL DATA</b>						
<b>Name</b>	<b>No. Samples</b>	<b>No. Lines</b>	<b>No. Bands</b>	<b>Bit depth</b>	<b>Flight Number</b>	<b>Year</b>
Lake Monona	256	3176	242	12	EO1H0240302009166110PF	2009
Mt. St. Helens	256	3242	242	12	EO1H0460282009231110KF	2009
Erta Ale	256	3187	242	12	EO1H1680502010057110KF	2010
<b>RODIS SENSOR - HYPERSPECTRAL DATA</b>						
<b>Name</b>	<b>No. Samples</b>	<b>No. Lines</b>	<b>No. Bands</b>	<b>Bit depth</b>	<b>Flight Number</b>	<b>Year</b>
Pavia University	200	200	103	12	-	2002

The specifications concerning the AVIRIS hyperspectral datasets are as follows:

The AVIRIS datasets are stored in two bytes. The band interleaved by pixel, the (BIP) format is used. The AVIRIS datasets have a large endian byte order. They may have either 16 bits or 12 bits of bit depth. They can be expressed with the 10 nm spectral resolution and 0.4 - 2.5  $\mu\text{m}$  spectral range coverage [94]. In this study, the AVIRIS images are cropped as 512 lines by 512 samples by 224 bands.

Eventually, NASA launched an imaging spectrometer EO-1 HYPERION sensor in November 2000. HYPERION is a satellite hyperspectral sensor [94] and its specifications are listed as follows:

The HYPERION datasets are stored in two bytes. The BIP format is used. The byte order is the little endian. They have 12 bits of bit depth. The spectral resolution of the HYPERION datasets is 10 nm together with a spectral range coverage of 0.4 - 2.5  $\mu\text{m}$  [64]. In this study, the HYPERION images are cropped as 512 lines by 256 samples by 242 bands.

Reflective Optics Systems Imaging Spectrometer (ROSIS-03) optical sensor is used for the University of Pavia, Italy [96]. The spectral resolution is 4 nm, while the spectral range is from 0.43 to 0.86  $\mu\text{m}$  [97]. This Pavia University hyperspectral image has 1.3 m per pixel spatial resolution. It originally has 610 lines by 340 samples by 103 bands [96]; however the cropped version yielding 200 lines by 200 samples by 103 bands is utilized in this study.

### **3.3.2 The rate-distortion results of AVIRIS and ROSIS datasets**

In this section, three different experiments with different AVIRIS datasets are conducted to investigate the rate-distortion performance of the hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding.

The results of these experiments are examined under three different scenarios for the ease of presentation.

### **3.3.2.1 Experiment 1 - Comparison with transformation-based, prediction-based or batch mode learning based lossy compression methods**

Experiment 1 consists of the rate-distortion performance comparison between the hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding and the transformation-based, prediction-based or batch mode learning based lossy compression methods. [63, 69, 70, 74, 78, 86]. TNIPM algorithm is used to solve the sparse coding problem since this is used in the online dictionary learning method proposed by Mairal et. al. The hyperspectral datasets utilized in the experiments are Yellowstone (uncalibrated scene 0), Cuprite, Moffett Field and Lunar Lake.

It is important to note that, the rate-distortion performances of the lossy compression methods that are used in the comparison are gathered directly from the authors [21, 63, 69, 70, 74, 78, 86]. These methods are not re-evaluated in this study. Therefore, instead of using only one graph to compare all methods, separate graphs are introduced. At each graph, the same PSNR values stated in literature are utilized. Each graph consists of different lossy compression methods by employing different datasets. The rate-distortion results of the hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding are placed on these different graphs. Throughout the experiments, the compression methods exploit either a fixed dictionary or learn it particularly from the data. If dictionary learning is performed, it will be stated.

Figure 2 shows the rate-distortion performances, where the performance of the proposed sparse coding with online learning method is compared with those of the predictive lossy compression algorithm and JPEG2000 part 2 algorithm [63]. Yellowstone (uncalibrated scene 0) AVIRIS dataset is used in this experiment (cf. Table 3) and it is processed without cropping as stated in the literature [63]. The compression performance of the proposed sparse coding with online learning method, which is measured in terms of PSNR value is superior to those of the other two algorithms when the bps value is higher than 0.4 (cf. Figure 2). Here, compression methods used in the comparison are not depending on the dictionary learning. Therefore, our proposed hybrid method which depends on the online learning outperforms the others.

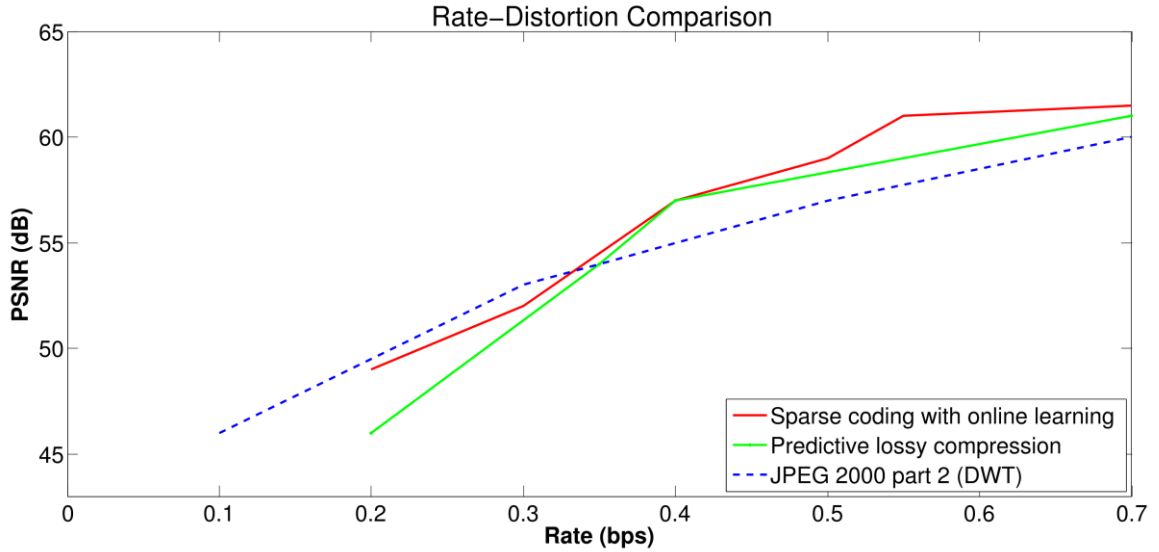


Figure 2 Rate-distortion curves of the proposed sparse coding with online learning algorithm, predictive lossy compression algorithm and JPEG2000 part 2 algorithm for Yellowstone (uncalibrated scene 0) dataset [63]

Figure 3 depicts the rate-distortion curve of the proposed sparse coding with online learning method as well as those of the PCA+JPEG2000 and DWT+JPEG2000 methods for the Cuprite AVIRIS dataset [70]. Dataset is cropped as the size of 512x512x224. These lossy compression approaches are comprised of PCA or DWT applied to the image, followed by the application of JPEG2000 based compression scheme. They are not depending on dictionary learning. The rate-distortion performance of the proposed sparse coding with online learning scheme is superior to those of the other state-of-the-art methods at all compression ratios as expected.

In Figure 4 rate-distortion performance of the proposed sparse coding with online learning algorithm is compared to those of the 3D-SPIHT algorithm, JOMP algorithm with  $n = 1024$  and FJOMP algorithm with  $n = 256$  [78]. Cuprite AVIRIS hyperspectral dataset is used. It is cropped as the size of 512x512x32. This comparison is particularly important and slightly different than others since some of the used compression algorithms are based on dictionary learning. Indeed, JOMP algorithm with  $n = 1024$  and FJOMP algorithm with  $n = 256$  are depending on dictionary learning, where the learning is performed by a

batch method, namely K-SVD. The proposed sparse coding with online learning algorithm yields rate-distortion performance that is superior to those of the other methods for all bps values from 0.1 through 0.5. Since online dictionary learning is more efficient in large datasets than batch counterparts, the proposed method which uses online dictionary learning is better than others.

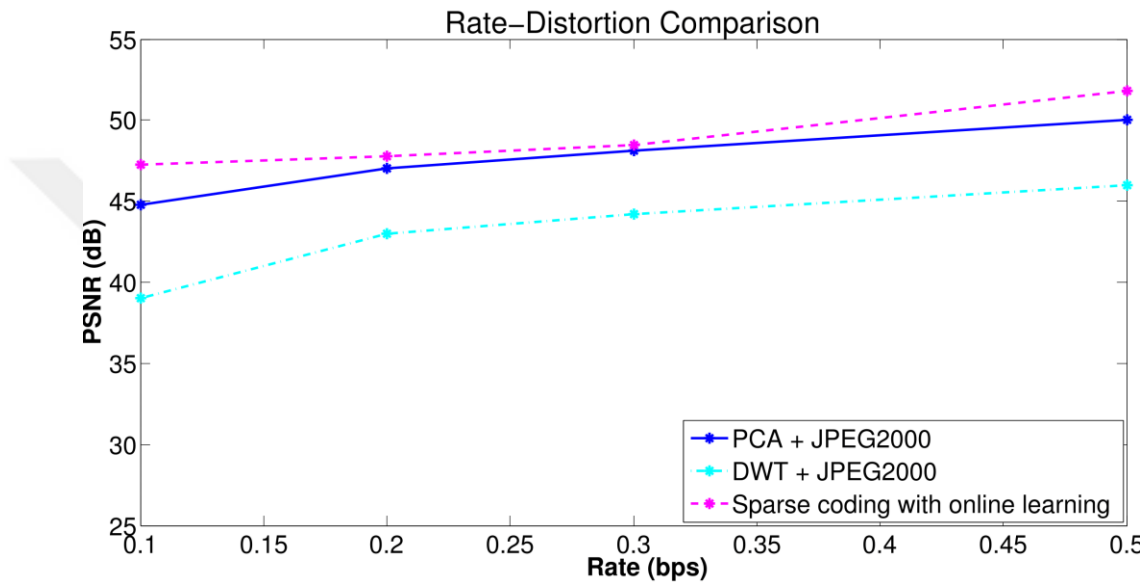


Figure 3 Rate-distortion curves of the proposed sparse coding with online learning algorithm, PCA+JPEG2000 algorithm and DWT+JPEG2000 algorithm for the Cuprite dataset cropped as 512x512x224 [70]

According to the results in Figure 4, the type of dictionary learning, such as the batch methods or the online learning methods, has a significant effect on the rate-distortion performance. Learning can increase the rate-distortion performance. Online learning can increase the performance even further.

Figure 5 presents the rate-distortion curves of the proposed sparse coding with online learning algorithm, AR-based algorithm in conjunction with PCA algorithm plus JPEG2000 (JP2K) algorithm and AR-based algorithm in conjunction with SubPCA algorithm plus JP2K algorithm [69]. Moffett Field hyperspectral dataset is cropped to the size of 512 lines, 512 samples and 224 bands. PSNR values of the proposed sparse coding

with online learning algorithm outperform those of the other state-of-the-art compression algorithms from 0.1 bps to 0.3 bps values.

The result is again reasonable due to the fact that the lossy compression methods used in the comparison are not depending on the dictionary learning. Since learning is not performed, these methods cannot be accurate enough in representing the dataset compared to the proposed sparse coding with online learning algorithm.

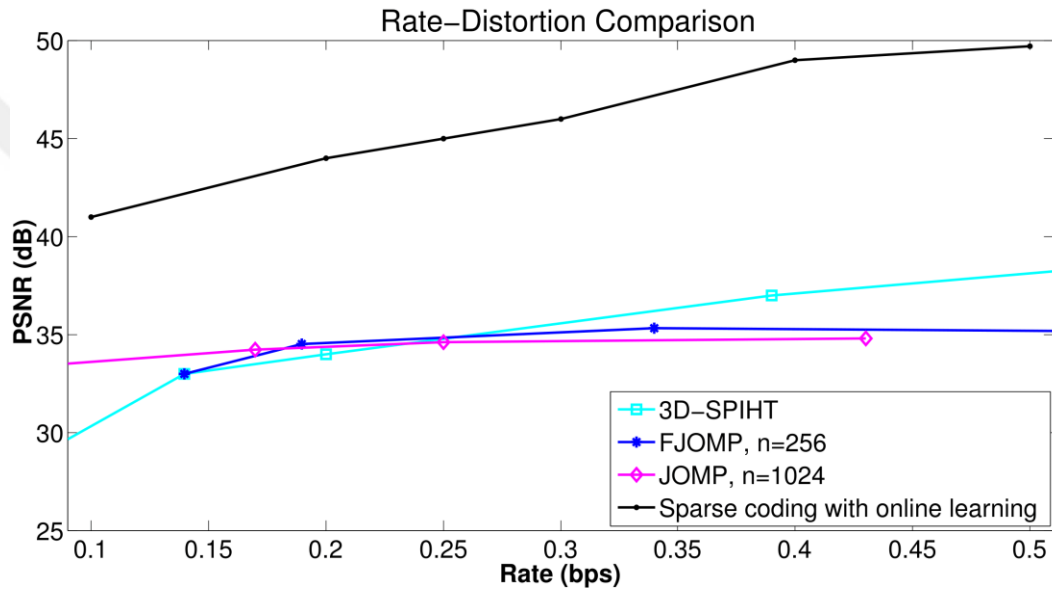


Figure 4 Rate-distortion curves of the proposed sparse coding with online learning algorithm, FJOMP, with  $n = 256$ , JOMP, with  $n = 1024$  and 3DSPIHT for Cuprite dataset cropped as  $512 \times 512 \times 32$  [78]

Rate-distortion performance comparison tests can be extended further by observing the performances of the proposed sparse coding with online learning method together with the state-of-the-art 3D SPECK algorithm and the modified 3D SPECK algorithm, as shown in Figure 6 [74]. AVIRIS hyperspectral scene from the 2009 acquisition, namely, Lunar Lake is cropped, yielding 256 lines, 256 samples and 32 band (cf. Table 3).



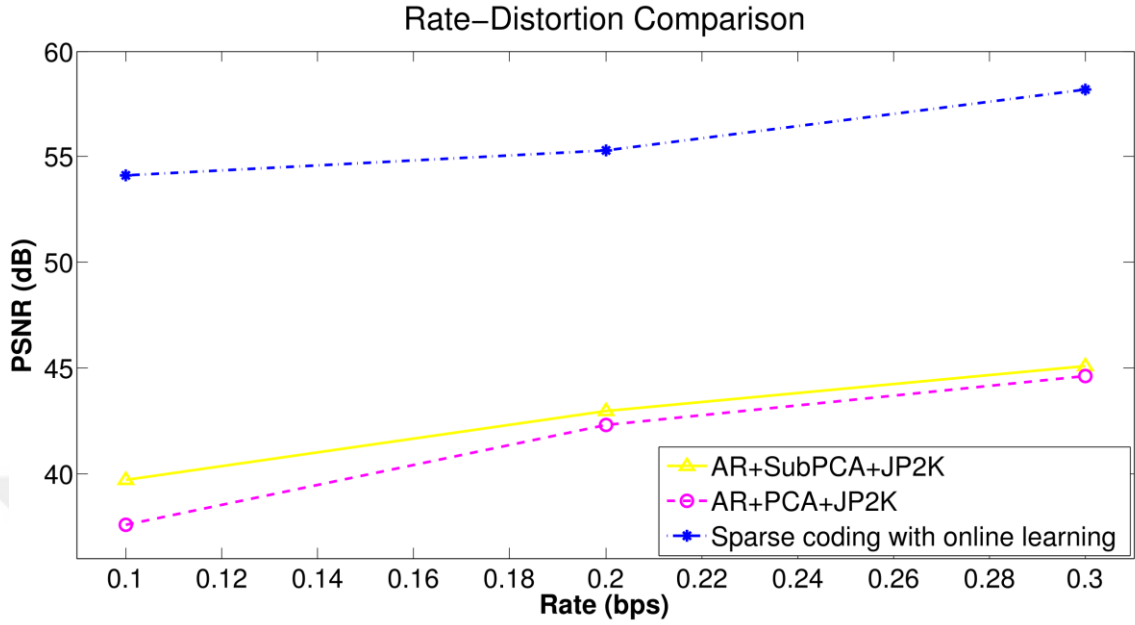


Figure 5 Rate-distortion curves of the proposed sparse coding with online learning algorithm, AR+SubPCA+JP2K algorithm and AR+PCA+JP2K algorithm for Moffett Field dataset [69]

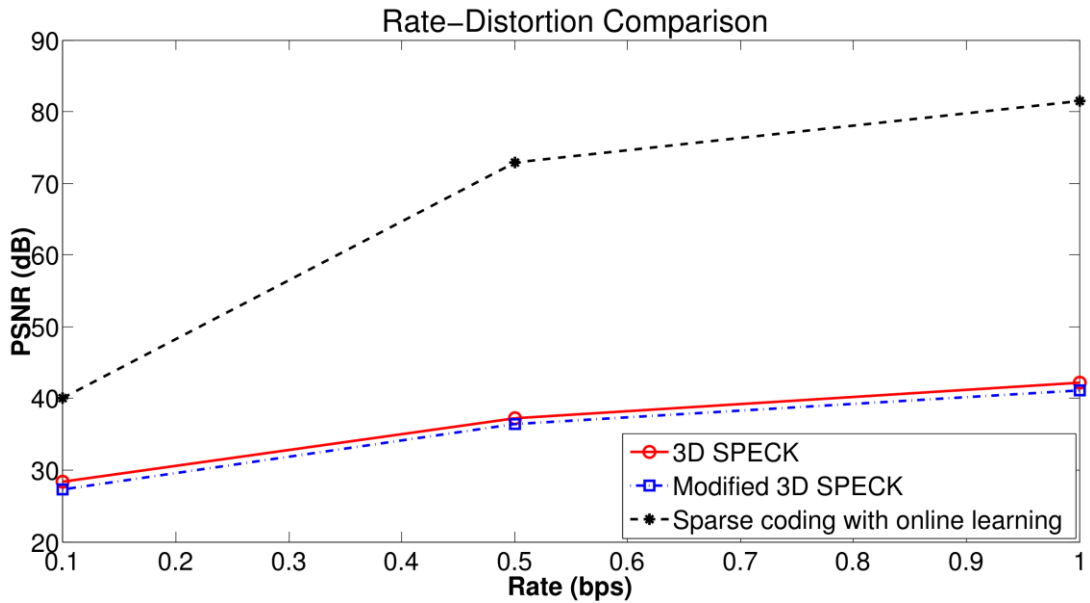


Figure 6 Rate-distortion curves of the proposed sparse coding with online learning algorithm, 3D SPECK algorithm and modified 3D SPECK algorithm for Lunar Lake dataset [74]

The superior rate-distortion performance of the proposed sparse coding with online learning method over the state-of-the-art compression algorithms from 0.1 bps through 1 bps can be seen from Figure 6. The result is expected due to the learning effect in the proposed method. All other methods exploited in the comparison are not depending on the dictionary learning.

### **3.3.2.2 Experiment 2 - Comparison of different sparse representation algorithms exploited in the hybrid method and comparison with transformation-based or sparse representation-based methods**

In experiment 2 section, the hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding is formulated with many different sparse representation algorithms to solve the sparse coding problem. Experiments are conducted to identify which sparse representation algorithm strategy has the best rate-distortion performance for hyperspectral image compression. To this end, AVIRIS and ROSIS hyperspectral datasets are used. The rate-distortion performances corresponding to these different sparse representation algorithms are compared with each other as well as with those of the other lossy compression algorithms. Rate-distortion performances of sparse representation algorithms, especially the ones that have recently been very popular in the literature such as blind compressive sensing algorithms and proximity based optimization algorithms, are analyzed further. Moreover, other compression algorithms in the literature such as CPPCA as the transformation-based method and BCS PL-3DBS + 3DWPT as the sparse representation-based method are utilized in the comparison [54].

In Table 4, the rate-distortion performances of different sparse representation algorithms adapted to the hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding are presented. Other lossy compression algorithms used in the comparison are also shown. Low Altitude, Lunar Lake, and Jasper Ridge are used as the AVIRIS datasets (cf. Table 3).

The quality metric tool, which reflects the rate-distortion performance, is PSNR in terms of dB. The PSNR values are calculated against the compression ratios in terms of the bps.

The lossy compression algorithms BCS PL-3DBS + 3DWPT and CPPCA, which are given in Table 4, are used for the comparison [54]. These compression algorithms are not depending on the dictionary learning. PCA compression algorithm is indeed the most basic and widely used one in the literature. Since, CPPCA algorithm is the improved version of PCA, the computationally efficient CPPCA is selected for comparison. Moreover, block based compression algorithms are also very popular and considered more efficient for large datasets. Since BCS PL-3DBS + 3DWPT is a block based compression algorithm, it is also chosen for comparison. The highest three PSNR values per each compression ratio are marked in boldface.

In Table 4, the best rate-distortion performance for Low Altitude image at 0.1 bps bit rate belongs to OBD-BCS algorithm. BCS PL-3DBS + 3DWPT algorithm has the highest PSNR value for the Lunar Lake image at 0.1 bps bit rate. At the same rate, SpaRSA algorithm shows the best performance in terms of PSNR value for the Jasper Ridge dataset.

Compression performances at 0.3 bps bit rate in Table 4 indicate that the gOMP algorithm is superior for the Low Altitude dataset. OBD-BCS algorithm has the highest PSNR value for the Lunar Lake image and the CPPCA algorithm outperforms for the Jasper Ridge image.

At 0.5 bps bit rate, the OBD-BCS algorithm is superior for the Low Altitude image as seen from Table 4. The CPPCA algorithm has the highest PSNR value for both the Lunar Lake and Jasper Ridge datasets.

According to Table 4, SpaRSA and GIST algorithms from the proximity based optimization algorithms category performed very well on all of the datasets such that they are ranked among the top five algorithms for compression ratio values of 0.3 bps and 0.5 bps. On the other hand, FISTA and TwIST algorithms which are also classified in the category of proximity based optimization approaches cannot be located among the best five algorithms.

If Table 4 is analyzed at the compression ratio of 0.1 bps, the GISA with  $p=0.5$  algorithm and OBD-BCS algorithms are involved among the top three algorithms with the highest PSNR values for at least two datasets. At the 0.3 bps case, the same situation is valid for BP by using dual active set and OBD-BCS algorithms. At the highest compression ratio of 0.5 bps, only the OBD-BCS algorithm involves among the algorithms with the best three PSNR values for all datasets.

According to Table 4, the OBD-BCS algorithm involves among the top three algorithms with the best compression performances at the highest compression ratio of 0.5 bps for all the datasets.

Since other sparse representation algorithms cannot show similar behavior at the 0.5 bps ratio, the OBD-BCS has a better performance than other algorithms at high compression ratios. The OBD-BCS algorithm belongs to the algorithms with the highest three PSNR values at all the compression ratios for the Low Altitude dataset. Moreover, it involves among the best three algorithms at 0.3 bps and 0.5 bps levels for the Lunar Lake image. For the Jasper Ridge dataset, the OBD-BCS algorithm is situated among the algorithms with the top three PSNR values at 0.1 and 0.5 bps ratios. Consequently, the OBD-BCS algorithm shows a better compression performance than all other sparse representation algorithms as well as the other lossy compression algorithms (BCS PL\_3DBS + 3DWPT and CPPCA) at high compression ratio values.

Although, the typical PSNR values for lossy compression are in 30dB-50dB range, higher values are better [98]. In this study, PSNR values observed in the rate-distortion performances are in the range of 40dB-80dB. Due to the learning step which fits the dictionary particularly to the data, such high PSNR values are obtained. Here, hyperspectral data is accurately represented by the learned dictionary.

PSNR is indeed a simple and widely used performance measurement tool, yet human visual system cannot perceive the difference between the original image and the reconstructed one with a high PSNR value. However, this does not mean that high PSNR values are meaningless.

Table 4 Compression performances of sparse representation algorithms and other compression algorithms (BCS PL\_3DBS + 3DWPT and CPPCA) [54] (The highest three PSNR values (in dBs) are shown in boldface for each bps value)

Low Altitude Image																		
Sparse Representation Algorithms																		
BPS	BCS PL_3DBS + 3DWPT	BP (Dual active set)	gOMP	LASSO (ADMM)	CPPCA	SpaRSA	GIST	BPDN (Homotopy)	GISA $p=0.4$	GISA $p=0.3$	GISA $p=0.5$	OBD-BCS	SBL	FOCUSS	Shooting	FISTA	TwIST	TNIPM
0.1	54.74	59.96	59.79	59.59	47.47	59.88	59.85	59.82	59.74	59.35	<b>59.98</b>	<b>60</b>	<b>59.97</b>	59.93	59.79	48.23	59.4	58.34
0.3	61.74	<b>70.16</b>	<b>70.28</b>	68.85	60.98	69.78	69.79	69.04	67.64	67.72	66.81	<b>69.99</b>	68.15	68.52	69.73	50.39	63.63	66.18
0.5	67.08	<b>73.24</b>	72.68	<b>73.52</b>	70.01	72.82	73.21	72	71.9	69.32	68.41	<b>73.56</b>	72.74	73.09	71.81	56.53	67.01	71.20
Lunar Lake Image																		
Sparse Representation Algorithms																		
BPS	BCS PL_3DBS + 3DWPT	BP (Dual active set)	gOMP	LASSO (ADMM)	CPPCA	SpaRSA	GIST	BPDN (Homotopy)	GISA $p=0.4$	GISA $p=0.3$	GISA $p=0.5$	OBD-BCS	SBL	FOCUSS	Shooting	FISTA	TwIST	TNIPM
0.1	<b>61.34</b>	59.55	58.37	59.54	48.43	59.51	<b>59.68</b>	59.57	59.58	59.43	<b>59.98</b>	59.22	59.6	59.54	59.57	43.23	59.17	58.86
0.3	69.38	<b>73.85</b>	73.84	73.34	72.19	<b>73.89</b>	73.62	68.58	69.41	69.11	66.81	<b>73.97</b>	72.07	73.16	69.87	47.44	65.67	70.71
0.5	72.62	<b>76.55</b>	74.92	75.2	<b>76.82</b>	75.07	75.37	71.81	71.57	70.88	68.41	<b>75.45</b>	75.01	74.19	74.47	53.85	67.46	72.39
Jasper Ridge Image																		
Sparse Representation Algorithms																		
BPS	BCS PL_3DBS + 3DWPT	BP (Dual active set)	gOMP	LASSO (ADMM)	CPPCA	SpaRSA	GIST	BPDN (Homotopy)	GISA $p=0.4$	GISA $p=0.3$	GISA $p=0.5$	OBD-BCS	SBL	FOCUSS	Shooting	FISTA	TwIST	TNIPM
0.1	56.78	59.41	59.4	59.3	30.2	<b>59.47</b>	58.83	59.32	59.28	59.39	59.38	<b>59.44</b>	58.6	59.37	<b>59.42</b>	47.58	59.03	58.06
0.3	64.21	69.23	70.01	70.67	<b>71.31</b>	70.69	70.15	69.56	68.45	66.94	68.66	70.57	<b>70.74</b>	<b>70.83</b>	70.73	54.54	64.77	66.74
0.5	69.95	71.71	71.14	<b>73.17</b>	<b>76.4</b>	72.49	72.24	72.44	70.46	69.58	69.86	<b>72.54</b>	72.46	72.17	72.27	55.55	67.15	71.34

Since PSNR is dimensionless, it has no sense to define an acceptable range for it. PSNR is employed only for comparing the performances of different lossy compression algorithms [99]. From this point of view, the PSNR values observed in this study are only for comparing the performances of various methods.

### **3.3.2.3 Experiment 3 - Comparison with a state-of-the-art method**

For further comparison, reconstruction performance of the state-of-the-art RLPHCS\_Cov algorithm is also compared to those of the different sparse representation algorithms based on online dictionary learning. According to the previous chapter, RLPHCS\_Cov algorithm is considered as the state-of-the-art compression algorithm, since it is already outperforms all the-state-of-the-art compression algorithms recently in the literature [86].

The signal to noise ratio (SNR) is fixed at 20dB. AVIRIS image of Pavia University and ROSIS image of Indian Pines datasets are used (cf. Table 3). Figure 7 and Figure 8 show PSNR values of different algorithms at various bps levels when Pavia University and Indian Pines datasets are used, respectively. Algorithms solved by the proposed sparse coding with online learning method are denoted by the abbreviation online dictionary learning (ODL). Algorithms solved by several hyperspectral compressive sensing methods given in the literature [86] are denoted by the abbreviation hyperspectral compressive sensing (HCS).

Figure 7 and Figure 8 indicate that the reconstruction performance of the OBD-BCS (ODL) algorithm is superior to those of the other algorithms at 0.5 bps level. Although for 0.5 bps compression level the OBD-BCS (ODL) algorithms is better for both datasets, setting the compression ratio to moderate levels such as 0.3 bps yields better state-of-the-art RLPHCS\_Cov performance for Pavia University dataset.

Even the most recent well-performed state-of-the-art hyperspectral compression algorithm, namely, RLPHCS\_Cov algorithm [86] does not achieve the performance of the proposed sparse coding with online learning when OBD-BCS algorithm is implemented.

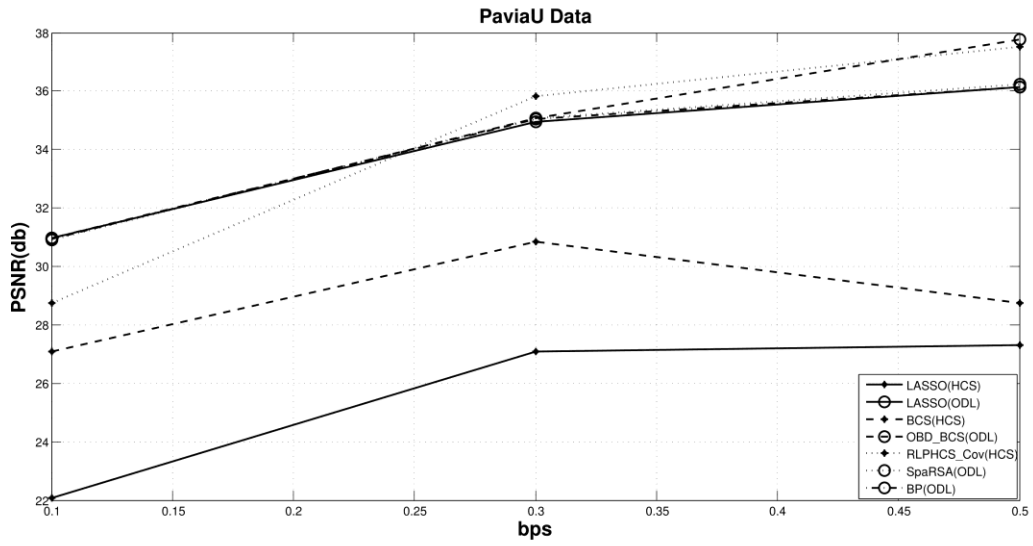


Figure 7 Rate distortion curves of several sparse representation algorithms (ODL) and several hyperspectral compressive sensing algorithms (HCS) for Pavia University dataset when SNR is 20dB

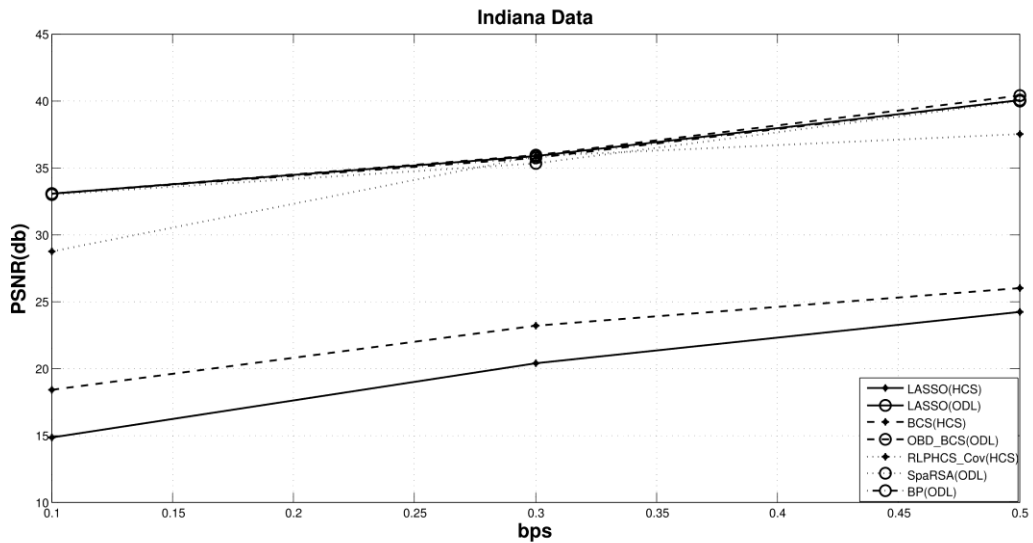


Figure 8 Rate distortion curves of several sparse representation algorithms (ODL) and several hyperspectral compressive sensing algorithms (HCS) for Indian Pines dataset when SNR is 20dB

The OBD-BCS algorithm is indeed a compressive sensing framework rather than just solving the sparse coding equation. An additional dictionary learning scheme is applied in order to find out the sparse coefficients in OBD-BCS algorithm which increases the accuracy. When these almost exact coefficients are used in the dictionary updating step in this study, it is expected to obtain much better results. According to the results, this expectation is realized, since OBD-BCS algorithm outperforms the others in terms of the rate-distortion performance.

### **3.3.3 The rate-distortion results of HYPERION datasets**

In this section, three different experiments are conducted using the HYPERION hyperspectral datasets. Because the approaches in the experiments are different, each one is presented in separate sections. In the first experiment only the basic sparse representation algorithms are compared, in the second one proximity based optimization algorithms are compared. For the third experiment, the comparison is performed by concerning the OBD-BCS algorithm which is a blind compressive sensing algorithm.

#### **3.3.3.1 Experiment 1 - Comparison of the basic sparse representation algorithms exploited in the hybrid method**

Experiment 1 comprises the comparison between the rate-distortion performances of basic sparse representation algorithms such as TNIPM algorithm, gOMP algorithm, LASSO by using ADMM algorithm and BP by using dual active-set algorithm [64]. Each of these sparse representation algorithms are adapted to the hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding.

Figure 9 shows the rate-distortion performances of TNIPM algorithm, gOMP algorithm, LASSO by using ADMM algorithm and BP by using dual active-set algorithm. Erta Ale hyperspectral dataset is used. It is cropped into a sub-image of size 512 lines by 256 samples by 242 bands (cf. Table 3).

For the bit rate values smaller than 0.45 bps, BP by using dual active-set algorithm shows the best performance among the sparse representation algorithms. For the bit rate values greater than 0.45 bps, the rate-distortion performance of TNIPM algorithm seem to be



better than the other sparse representation algorithms. The performance of gOMP algorithm is very poor beginning from roughly the 0.13 bps level.

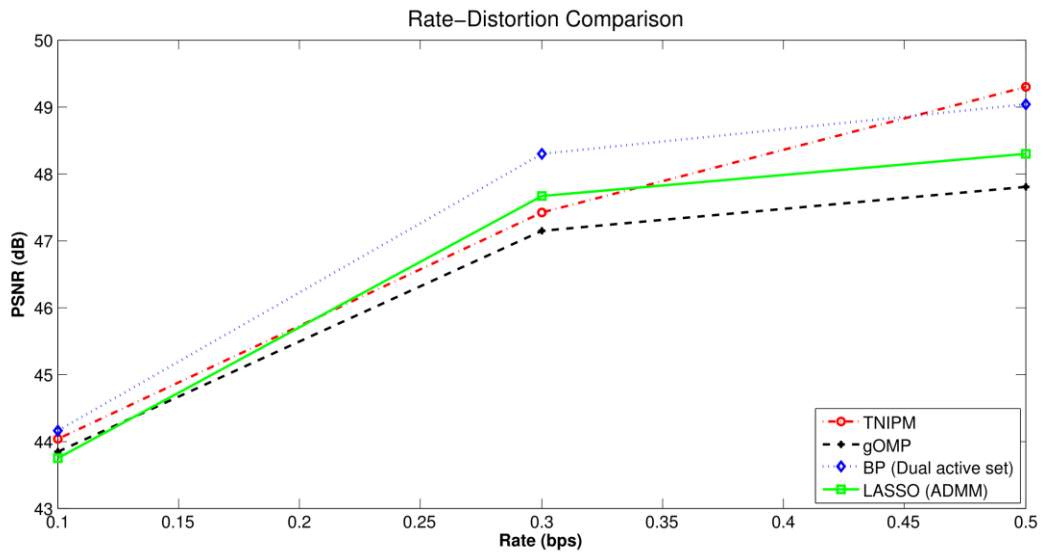


Figure 9 Rate-distortion curves of several sparse representation algorithms for Erta Ale hyperspectral dataset (cf. Table 3)

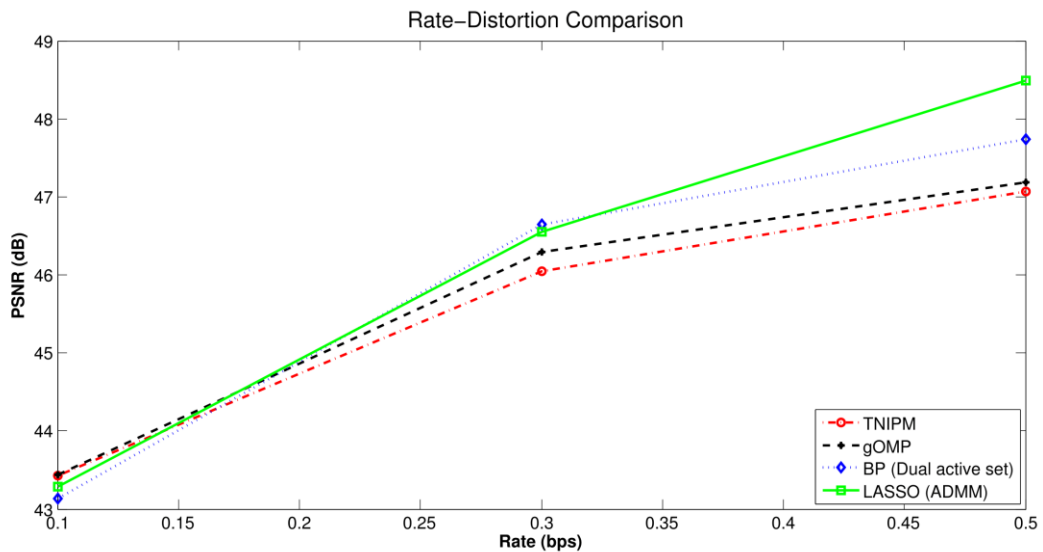


Figure 10 Rate-distortion curves of several sparse representation algorithms for Mt. St. Helens hyperspectral dataset (cf. Table 3)

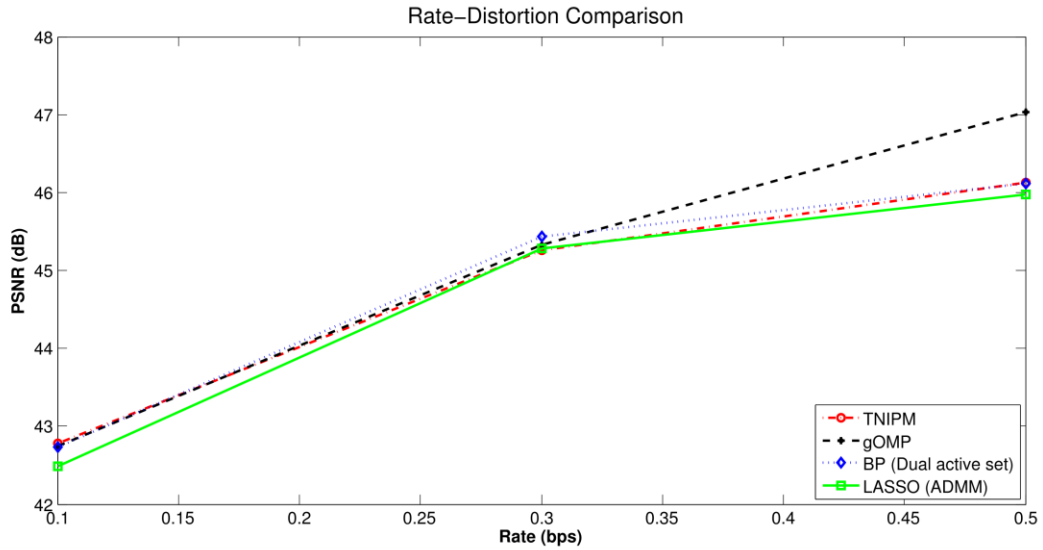


Figure 11 Rate-distortion curves of several sparse representation algorithms for Lake Monona hyperspectral dataset (cf. Table 3)

Figure 10 gives the rate-distortion curves of TNIPM algorithm, gOMP algorithm, LASSO by using ADMM algorithm and BP by using dual active-set algorithm for Mt. St. Helens dataset, while Figure 11 illustrates those for Lake Monona dataset. In Figure 10, LASSO by using ADMM algorithm has the best rate-distortion performance starting nearly from 0.3 bps bit rate; meanwhile in Figure 11 similar performance behavior is observed for gOMP algorithm.

### 3.3.3.2 Experiment 2 - Comparison of the sparse representation algorithms exploited in the hybrid method which are based on proximity based optimization method

Although experiment 2 introduces the rate-distortion curves of different sparse representation algorithms, the main goal here is to detect the performances of several proximity-based optimization algorithms such as SpaRSA, GIST and PALM in comparison to other existing algorithms. Rate-distortion comparisons are illustrated in Figure 12, Figure 13 and Figure 14. PSNR values in dBs are plotted against three different compression ratios in bps as bar graphs. The highest three PSNR values for each bit rate are marked with small black circles below the corresponding algorithms. Other sparse

representation algorithms used in this experiment include BP by using dual active-set algorithm, gOMP algorithm, LASSO by using ADMM algorithm, TNIPM algorithm and BPDN homotopy algorithm.

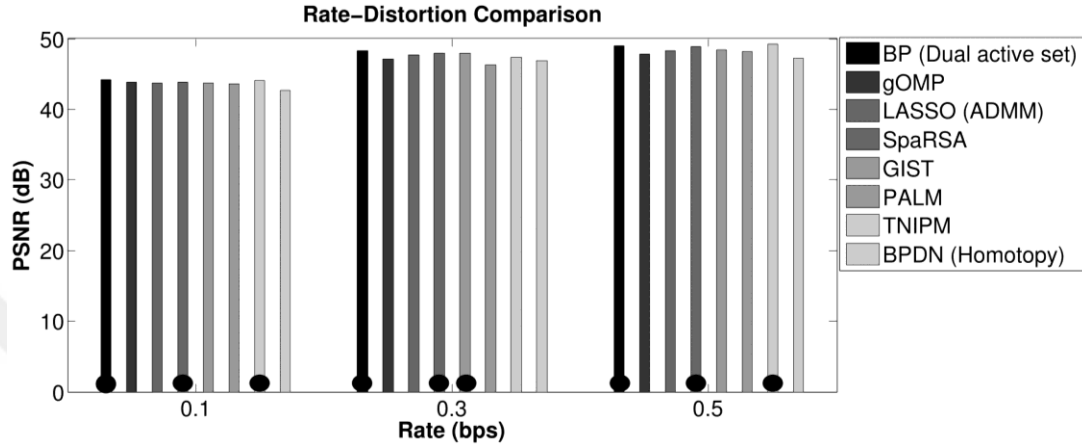


Figure 12 Rate-distortion performances of several sparse representation algorithms for Erta Ale hyperspectral dataset (cf. Table 3)

HYPERION datasets such as Erta Ale, Mt. St. Helens and Lake Monona are tested. All hyperspectral datasets that are used in the experiment are cropped into 512 lines by 256 samples by 242 bands (cf. Table 3).

TNIPM and SpaRSA algorithms are among the best three algorithms at 0.1 bps bit rate for Erta Ale and Lake Monona images as seen in Figure 12 and Figure 14. At 0.3 bps bit rate, BP by using dual active-set, SpaRSA and GIST algorithms are consistently ranked as the top three algorithms for all HYPERION datasets.

According to 0.5 bps bit rate results, SpaRSA algorithm is among one of the top three algorithms for all datasets. This algorithm is followed by GIST algorithm, which is among the best three algorithms for Mt. St. Helens and Lake Monona datasets as seen from Figure 13 and Figure 14.

As a result of this experiment, SpaRSA algorithm which is one of the key proximity based optimization scheme shows superior rate-distortion performance compared to those of the other sparse representation algorithms.

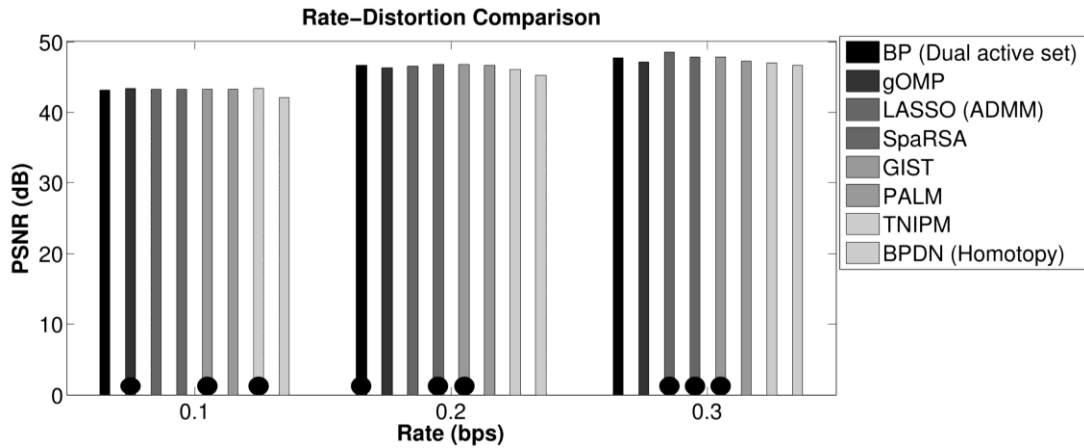


Figure 13 Rate-distortion performances of several sparse representation algorithms for Mt. St. Helens hyperspectral dataset (cf. Table 3)

In sparse coding, the final learned dictionary has highly correlated columns. Therefore, the sparse coding approaches that consider all possible values of the parameter  $\lambda$  such as proximity based optimization algorithms including soft thresholding strategy can provide more accurate solution than the others [19]. This is why the proximity based optimization algorithms such as SpaRSA and GIST that are used in the sparse coding stage provide better results.

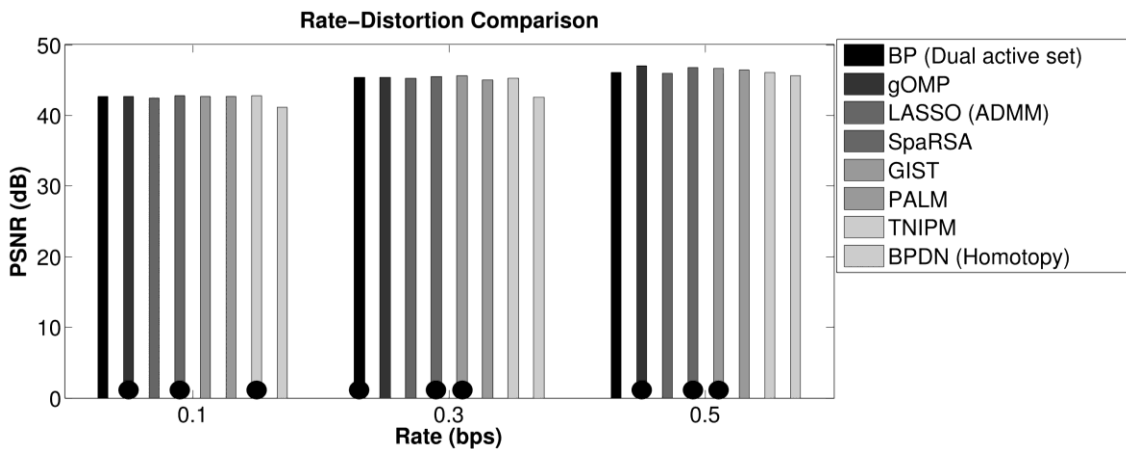


Figure 14 Rate-distortion performances of several sparse representation algorithms for Lake Monona hyperspectral dataset (cf. Table 3)

### 3.3.3.3 Experiment 3 - Comparison of several sparse representation algorithms with OBD-BCS sparse representation algorithm which are all exploited in the hybrid method

The aim of this experiment is to compare the rate-distortion performance of the OBD-BCS algorithm which is one of the blind compressive sensing approach to those of the other sparse representation algorithms. SpaRSA algorithm is also included in this section since it performed well in the previous experiment. The other sparse representation algorithms used in this experiment are BP by using dual active-set algorithm, PALM algorithm, LASSO by using ADMM algorithm, TNIPM algorithm, SBL algorithm, BPDN homotopy algorithm and FOCUSS algorithm.

The Erta Ale, Mt. St. Helens, and Lake Monona images, are used as HYPERION datasets (cf. Table 3). In Figure 15, Figure 16 and Figure 17, the PSNR values of these datasets against 0.1, 0.3, and 0.5 bps compression ratios for all sparse representation algorithms, are given. The compression performances of all the sparse representation algorithms are compared at different compression ratios.

The PSNR values are expressed in terms of dB, and they are plotted against the compression ratios in terms of bps. The corresponding compression ratios of the algorithms with highest three PSNR values are shown in circles.

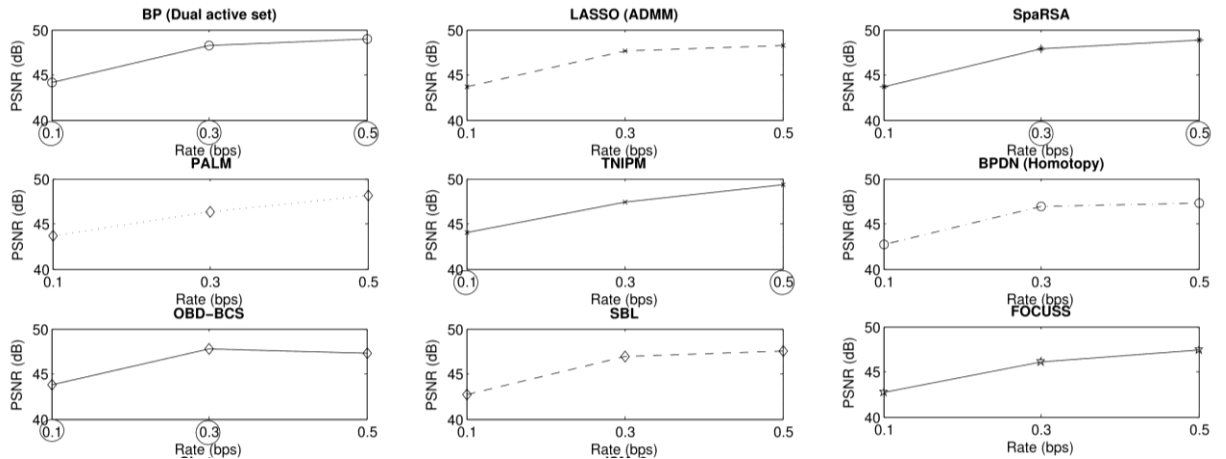


Figure 15 Rate-distortion performances of several sparse representation algorithms for Erta Ale hyperspectral dataset (cf. Table 3)

As seen from Figure 15, Figure 16 and Figure 17, BP by using the dual active set and TNIPM algorithms are among the best three algorithms for all the datasets at the 0.1 bps level. Similarly, the SpaRSA and OBD-BCS algorithms are involved among the algorithms with the highest three PSNR values for all the datasets at the 0.3 bps ratio.

At the highest compression ratio of 0.5 bps, the SpaRSA algorithm appears among the best three algorithms for all the datasets, while the OBD-BCS algorithm is situated among the top three algorithms for the Mt. St. Helens and Lake Monona datasets. Therefore, at high compression ratios, the SpaRSA and OBD-BCS algorithms show better compression performances.

In Figure 15, Figure 16, and Figure 17, the OBD-BCS algorithm belongs to the top three algorithms with the highest PSNR values at the compression ratio of 0.5 bps for the Mt. St. Helens and Lake Monona datasets.

At the moderate compression ratio of 0.3 bps, the OBD-BCS algorithm involves among the best three algorithms for all the datasets. At the lowest ratio of 0.1 bps, it is among the best three algorithms for the Erta Ale dataset only. The results indicate that the OBD-BCS algorithm shows a better compression performance when the compression ratio gets higher.

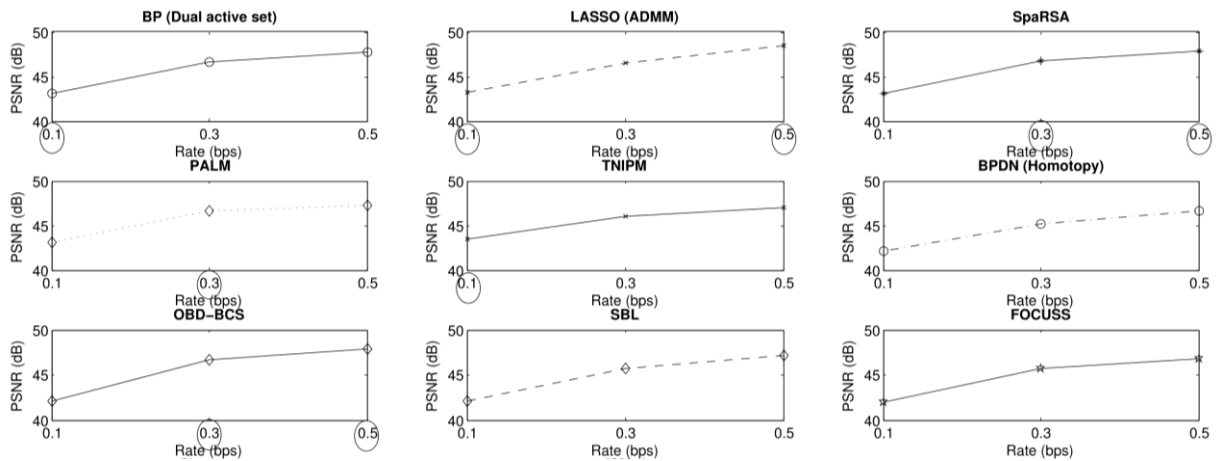


Figure 16 Rate-distortion performances of several sparse representation algorithms for Mt. St. Helens hyperspectral dataset (cf. Table 3)

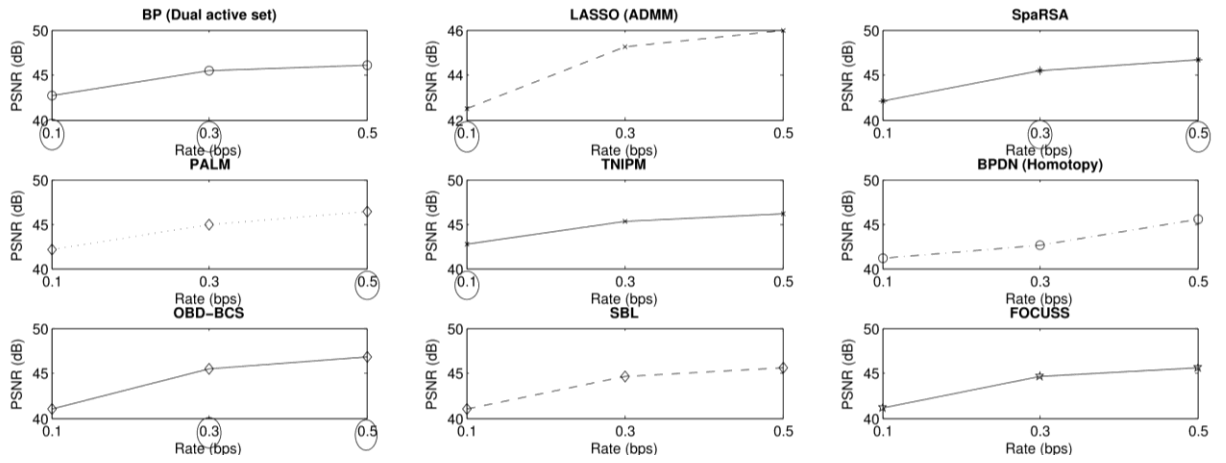


Figure 17 Rate-distortion performances of several sparse representation algorithms for Lake Monona hyperspectral dataset (cf. Table 3)

## CHAPTER 4

### ANOMALY DETECTION RESULTS

In lossy data compression task, it is important to concern the usefulness of the reconstructed image which corresponds to the information preservation performance apart from the rate-distortion performance [70].

One great strategy to evaluate the information preservation capability is to perform anomaly detection, since small yet important objects in the hyperspectral image may be disappeared in the compression. In addition to measure the distortion by employing PSNR in the previous chapter, the information preservation performance can also be evaluated by applying the anomaly detection in this chapter. It can be considered as a different tool for measuring the image quality.

Anomaly detection is exploited to make a further comparison between various sparse representation algorithms. Only the algorithms which performed well in the previous chapter are included in this anomaly detection analysis. Therefore, by applying a real application such as anomaly detection, more realistic comparison can be achieved.

Reed-Xiaoli (RX) anomaly detection algorithm is used for anomaly detection application [60]. Spectral signature which belongs to the input signal is compared with the mean values of each spectral band by using Mahalanobis distance,

$$\delta_{RX}(\mathbf{x}_i) = (\mathbf{x}_i - \mathbf{M})^T \mathbf{Cov}^{-1}(\mathbf{x}_i - \mathbf{M}) \quad (74)$$

where  $\mathbf{x}_i \in R^{nb}$ ,  $\mathbf{M}$  represents the mean of each spectral band and  $\mathbf{Cov}$  indicates the spectral covariance matrix.



Anomalous region is assumed to be present if  $\delta_{RX}(\mathbf{x}_i) \geq \eta$  condition is satisfied, where  $\eta$  represents the threshold value. The most appropriate threshold value is the one that is obtained from the desired false alarm probability. Covariance matrix  $\mathbf{Cov}$  is as follows:

$$\mathbf{Cov} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \mathbf{M})(\mathbf{x}_i - \mathbf{M})^T \quad (75)$$

where,  $N = nl \times ns$  and  $i = 1, \dots, N$ .

Anomaly detection is applied on Salinas-A, Pavia University and Low Altitude hyperspectral datasets (cf. Table 3). Sparse representation algorithms such as BP by using dual active set algorithm, LASSO by using ADMM algorithm, SpaRSA algorithm, GIST algorithm and OBD-BCS algorithm are utilized.

Anomaly detection experiments that are conducted in this study are examined under three different titles due to the differences in the used datasets and algorithms.

#### **4.1 ANOMALY DETECTION EXPERIMENT 1 - Comparison of sparse representation algorithms exploited in the hybrid method which have the best rate-distortion performances**

In this experiment, anomaly detection is applied on Salinas-A and Low Altitude hyperspectral datasets. Sparse representation algorithms adapted to the hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding; such as BP by using dual active set algorithm, LASSO by using ADMM algorithm, SpaRSA algorithm and OBD-BCS algorithm are utilized. These algorithms have the best rate-distortion performances according to the previous chapter.

Anomaly detection results are illustrated in Figure 18 for Salinas-A dataset. First, anomaly detection is applied on the original hyperspectral dataset whose results are presented in Figure 18(a).

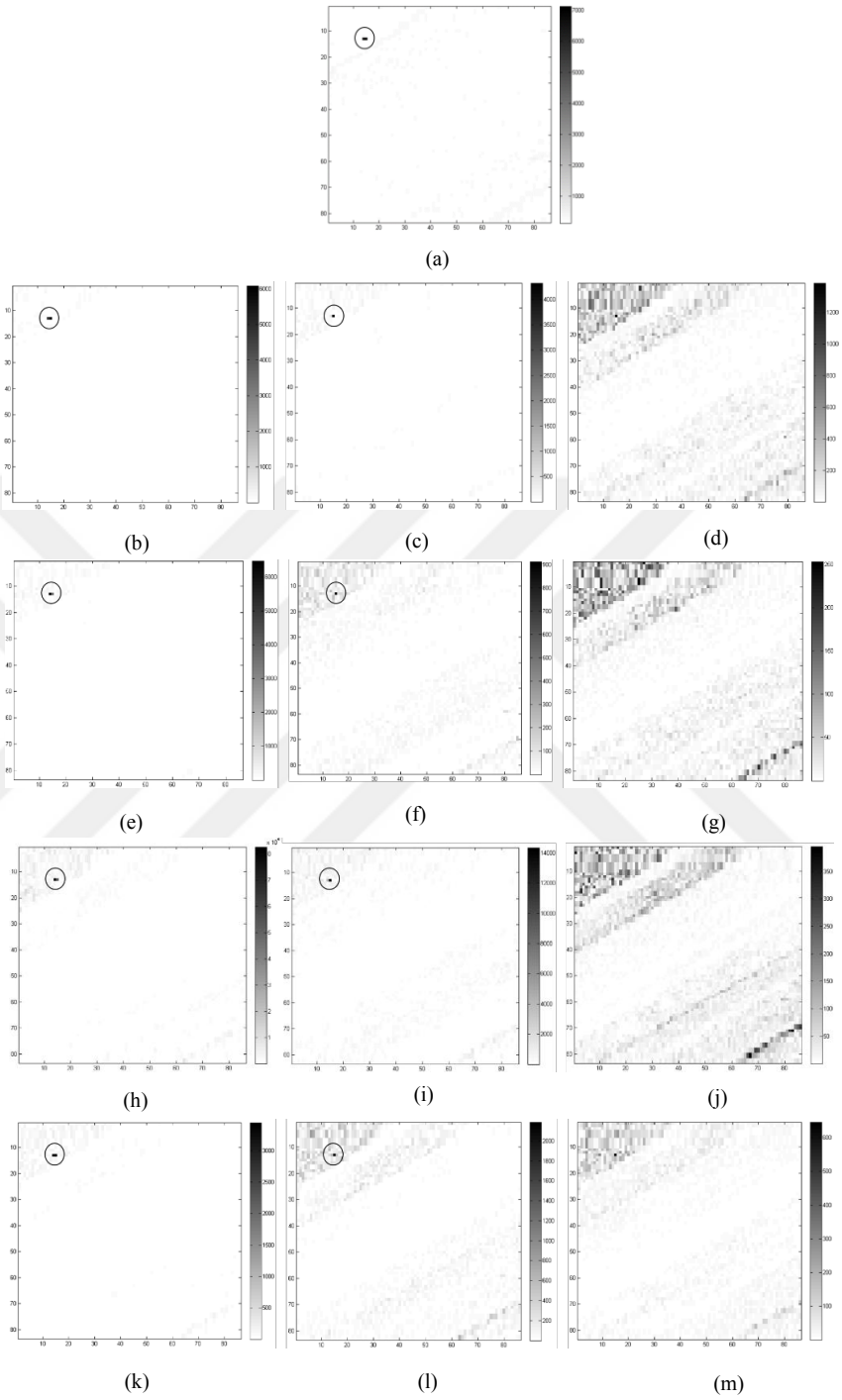


Figure 18 RX anomaly detection results of the Salinas-A hyp. image: (a) original image (b) OBD-BCS with 0.5 bps (c) OBD-BCS with 0.3 bps (d) OBD-BCS with 0.1 bps (e) BP with 0.5 bps (f) BP with 0.3 bps (g) BP with 0.1 bps (h) SpaRSA with 0.5 bps (i) SpaRSA with 0.3 bps (j) SpaRSA with 0.1 bps (k) LASSO with 0.5 bps (l) LASSO with 0.3 bps (m) LASSO with 0.1 bps

The desired anomaly is marked with a circle. Figure 18(b), Figure 18(c) and Figure 18(d) depict anomaly detection results when OBD-BCS algorithm is used for 0.5 bps, 0.3 bps and 0.1 bps bit rates, respectively. If BP by using dual active set algorithm is applied on the dataset, results in Figure 18(e), Figure 18(f) and Figure 18(g) are obtained for 0.5, 0.3 and 0.1 bps levels, respectively. Figure 18(h), Figure 18(i) and Figure 18(j) showing results of SpaRSA algorithm while Figure 18(k), Figure 18(l) and Figure 18(m) present results of LASSO by using ADMM algorithm.

None of the algorithms is able to detect the desired anomaly at 0.1 bps bit rate. Among the anomaly detection results at 0.5 bps bit rate, OBD-BCS algorithm seems to provide the best performance. At 0.3 bps level, the anomaly detection results for BP by using dual active set and LASSO by using ADMM algorithms perform worse than the ones for other algorithms.

The PSNR values of each sparse representation algorithms are also presented in Table 5 for 0.1, 0.3 and 0.5 bit rates in such a way to further strengthen the anomaly detection results obtained in Figure 18. The two highest PSNR values are marked in boldface.

Table 5 Compression performances of sparse representation algorithms (The highest two PSNR values (in dBs) are shown in boldface for each bps value)

BPS	Salinas-A Hyperspectral Dataset (cf. Table 3)			
	BP (Dual active set)	OBD-BCS	LASSO (ADMM)	SpaRSA
0.1	36.62	<b>36.67</b>	<b>36.65</b>	36.58
0.3	41.54	<b>41.89</b>	41.16	<b>42.61</b>
0.5	43.95	<b>43.98</b>	43.74	<b>43.96</b>
BPS	Low Altitude Hyperspectral Dataset (cf. Table 3)			
	BP (Dual active set)	OBD-BCS	LASSO (ADMM)	SpaRSA
0.1	<b>59.96</b>	<b>60</b>	59.59	59.88
0.3	<b>70.16</b>	<b>69.99</b>	68.85	69.78
0.5	73.24	<b>73.56</b>	<b>73.52</b>	72.82

Anomaly detection performances of different sparse representation algorithms can be assessed using receiver operating characteristic (ROC) curves. The ROC curves plot detection probability versus false alarm probability. ROC curves are plotted with a logarithmic x axis for better illustration.

Figure 19 shows the ROC Semi-Log curves of OBD-BCS algorithm at 0.1 bps, 0.3 bps and 0.5 bps rates when Salinas-A hyperspectral dataset is used. The probability of detection is represented by  $P_D$  and the probability of false alarm is represented by  $P_{FA}$ . Anomaly detection result at 0.5 bps rate is significantly better than those of the 0.3 bps and 0.1 bps levels.

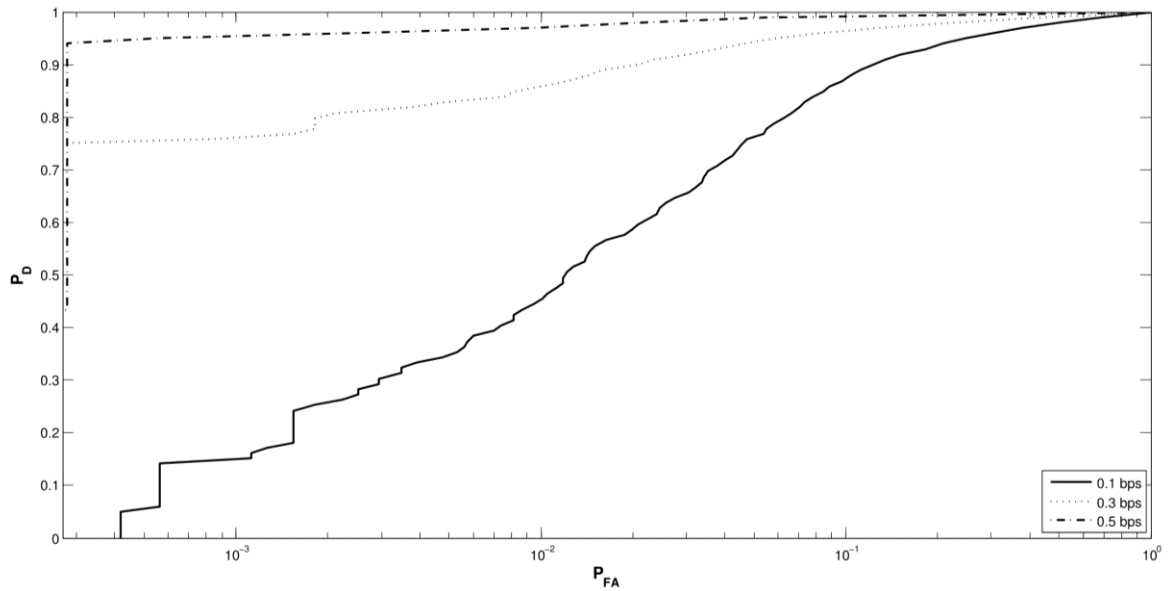


Figure 19 ROC Semi-Log curves for Salinas-A dataset at 0.1, 0.3 and 0.5 bps by using OBD-BCS algorithm

The ROC Semi-Log curves of BP by using dual active set algorithm at 0.1, 0.3 and 0.5 bps bit rates are depicted in Figure 20, where anomaly detection is conducted on the Salinas-A dataset. Here, the ROC performances at 0.5 bps and 0.3 bps rates are close to each other.

For the Salinas-A dataset, the ROC Semi-Log curves of SpaRSA algorithm and LASSO by using ADMM algorithm at various bit rates are illustrated in Figure 21 and Figure 22, respectively. Figure 21 and Figure 22 present anomaly detection performances that are compatible with the performances in Figure 18. In Figure 18(k) and Figure 18(l), there is a distinct difference between the performance of LASSO by using ADMM algorithm at 0.5 bps rate and that at 0.3 bps rate which can also be observed in Figure 22. In Figure

18(l), in addition to the desired anomaly part at 0.3 bps, some other parts are also observed which may lead to false detection. On the other hand, in Figure 18(h) and Figure 18(i) the performance of SpaRSA algorithm at 0.5 bps and 0.3 bps rates are positioned relatively closer to each other as also seen in the Figure 21.

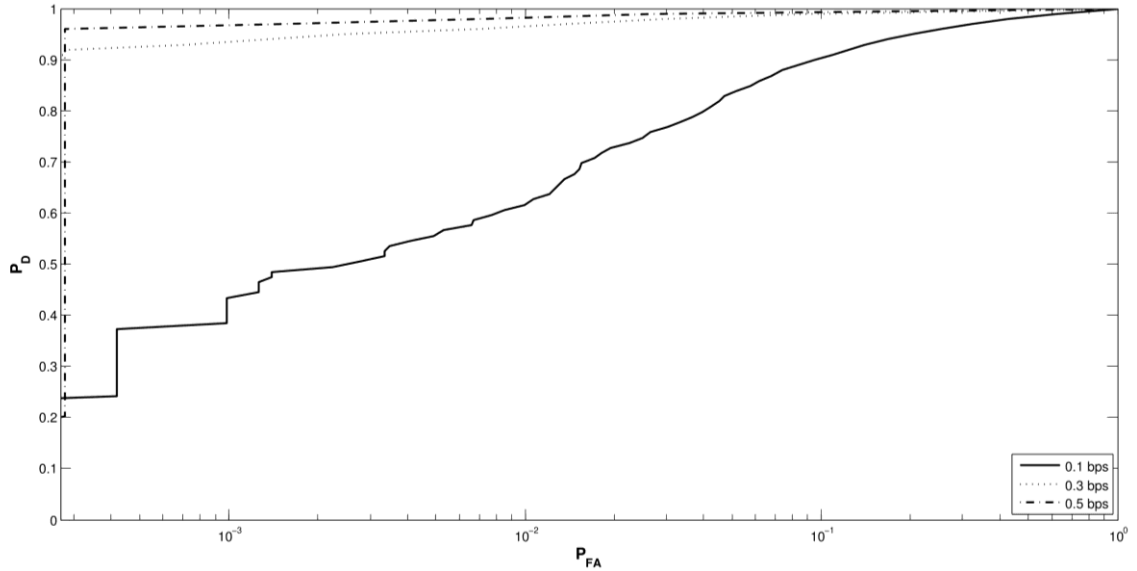


Figure 20 ROC Semi-Log curves for Salinas-A dataset at 0.1, 0.3 and 0.5 bps by using BP by using dual active set algorithm

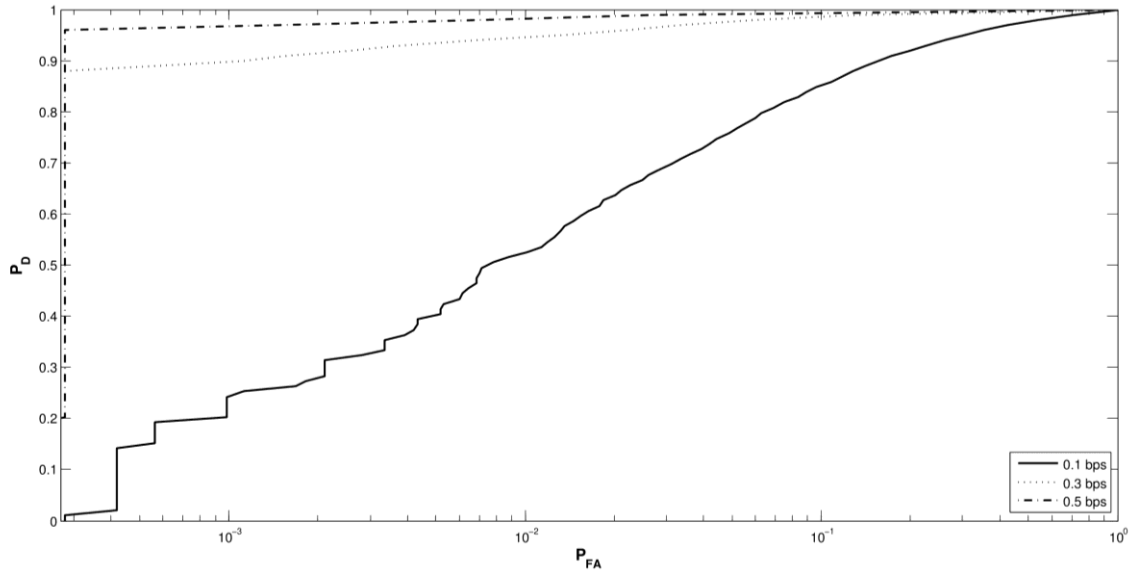


Figure 21 ROC Semi-Log curves for Salinas-A dataset at 0.1, 0.3 and 0.5 bps by using SpaRSA algorithm

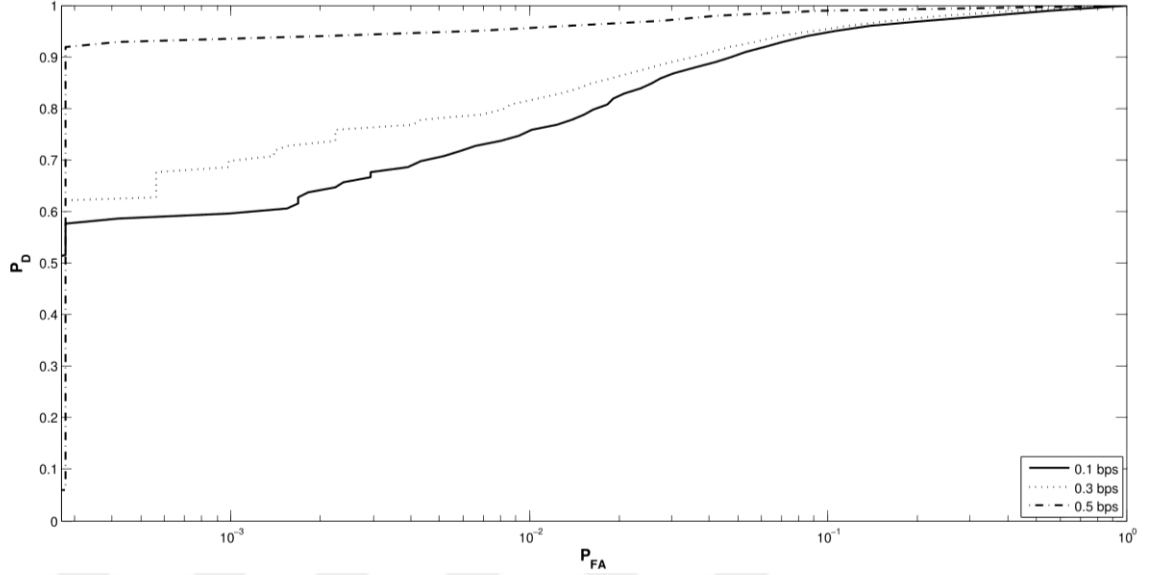


Figure 22 ROC Semi-Log curves for Salinas-A dataset at 0.1, 0.3 and 0.5 bps by using LASSO by using ADMM algorithm

In order to further evaluate the ROC curves, the area under curve (AUC) is employed as a performance metric that can be obtained by calculating the area under the ROC curve. Calculated AUC values are presented in Table 6.

In Table 6 for Salinas-A dataset, it can be seen that the best result is from OBD-BCS algorithm at 0.5 bps bit rate which is 0.9945. Similarly for the Low Altitude dataset, the highest value of the AUC corresponds to OBD-BCS algorithm at 0.5 bps.

Table 6 Area under curve (AUC) values of Salinas-A and Low Altitude hyperspectral datasets

Salinas-A Hyperspectral Dataset (cf. Table 3)			
	Area under ROC curve		
BPS	0.1	0.3	0.5
<b>BP (Dual active set)</b>	0.96	0.9934	0.9943
<b>OBD-BCS</b>	0.9464	0.9929	0.9945
<b>LASSO (ADMM)</b>	0.9741	0.9799	0.9928
<b>SpaRSA</b>	0.9424	0.992	0.9943
Low Altitude Hyperspectral Dataset (cf. Table 3)			
	Area under ROC curve		
BPS	0.1	0.3	0.5
<b>BP (Dual active set)</b>	0.9917	0.9932	0.9942
<b>OBD-BCS</b>	0.9887	0.9936	0.9943
<b>LASSO (ADMM)</b>	0.9906	0.9932	0.9931
<b>SpaRSA</b>	0.9896	0.9914	0.9937

Results in Table 5, Table 6 and Figure 19-Figure 22 demonstrate that the detection performance of OBD-BCS algorithm is better than those of the other algorithms for the case where the bit rate is high such as 0.5 bps. The illustrations in Figure 18 also suggest that the detection performance of OBD-BCS algorithm is the best of all at 0.5 bps bit rate. Anomaly detection is also applied on the Low Altitude hyperspectral dataset whose details can be found in Table 3. In Figure 23, anomaly detection results are illustrated. Figure 23(a) shows the desired anomaly when anomaly detection is applied on the original dataset. Figure 23(b), Figure 23(c) and Figure 23(d) indicate anomaly detection results of OBD-BCS algorithm at 0.5, 0.3 and 0.1 bps rates, respectively. Results of BP by using dual active set algorithm are given in Figure 23(e), Figure 23(f) and Figure 23(g). Results of the SpaRSA algorithm are shown in Figure 23(h)-Figure 23(j), while those of the LASSO by using ADMM algorithm are shown in Figure 23(k)-Figure 23(m).

At 0.5 bps bit rate, OBD-BCS algorithm detects the desired anomaly slightly better than the other algorithms, while the detection performances of SpaRSA algorithm and BP by using dual active set algorithm are almost identical. Despite the misleading anomaly parts observed at 0.3 bps rate, the desired anomaly can still be detected. Although all of the algorithms are able to detect the desired anomaly at all bit rates, detection performances are degraded severely at 0.1 bps level. According to the values in Table 5, OBD-BCS algorithm is among the best two algorithms in terms of PSNR values at 0.5, 0.3 and 0.1 bps rates. Particularly at 0.5 bps level, OBD-BCS algorithm has PSNR value of 73.56 which is the highest.

Figure 24 through Figure 27 depict the ROC Semi-Log curves of OBD-BCS, BP by using dual active set, SpaRSA and LASSO algorithms at 0.1 bps, 0.3 bps and 0.5 bps levels when Low Altitude hyperspectral dataset is employed. In fact, for both of the SpaRSA and BP by using dual active set algorithms, ROC Semi-Log curves of 0.5 bps and 0.3 bps are pretty close to each other.

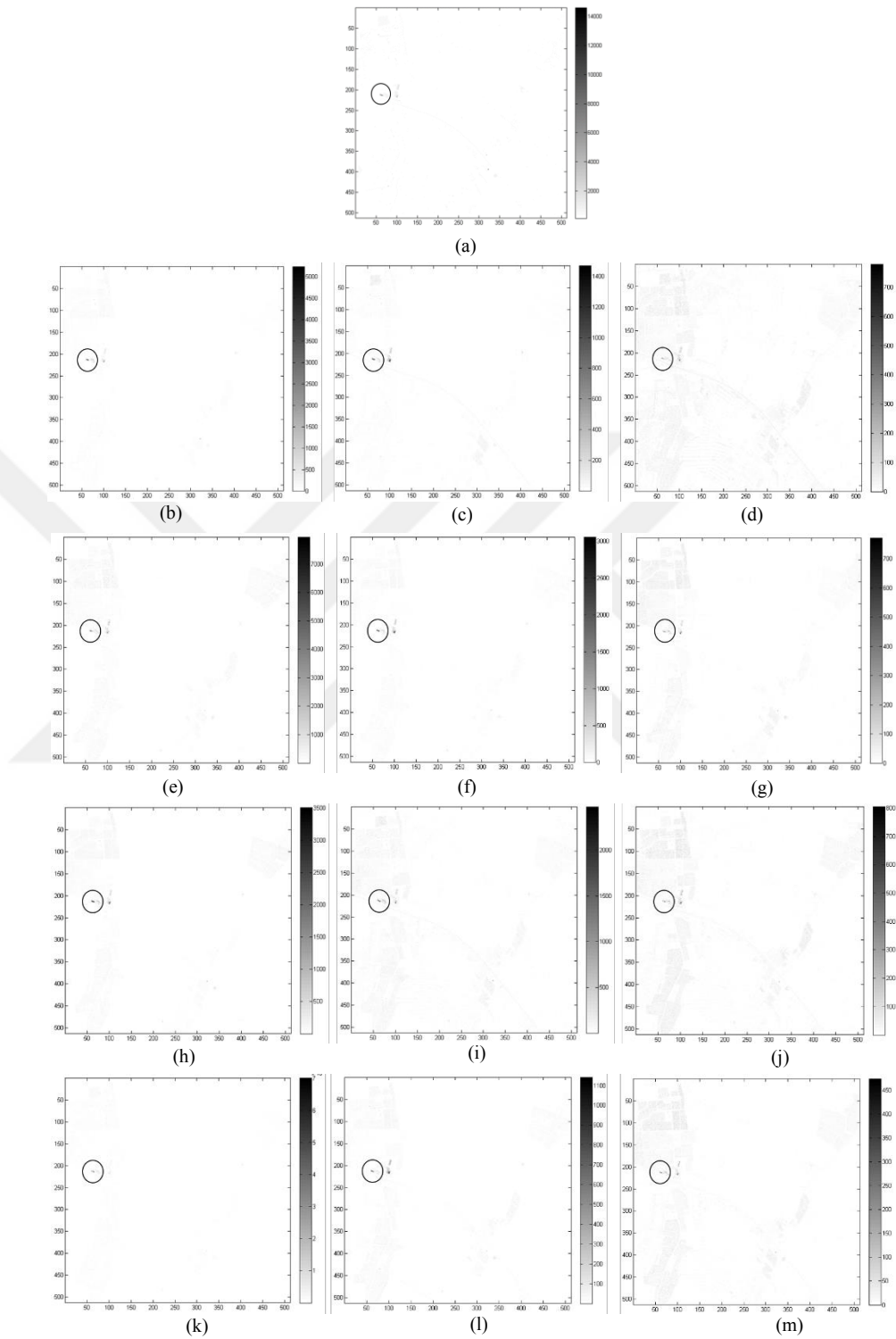


Figure 23 RX anomaly det. results of the Low Alt. hyp. image: (a) original (b) OBD-BCS 0.5 bps (c) 0.3 bps (d) 0.1 bps (e) BP 0.5 bps (f) 0.3 bps (g) 0.1 bps (h) SpARSA 0.5 bps (i) 0.3 bps (j) 0.1 bps (k) LASSO 0.5 bps (l) 0.3 bps (m) 0.1 bps



In this case, evaluating only the ROC Semi-Log curves is not enough to assess the information preservation performances of the algorithms. Evaluation of AUC values should also be taken into account. The superiority of OBD-BCS algorithm at 0.5 bps rate is supported by the results in Table 6. At 0.5 bps, OBD-BCS algorithm has the highest AUC value which is 0.9943.

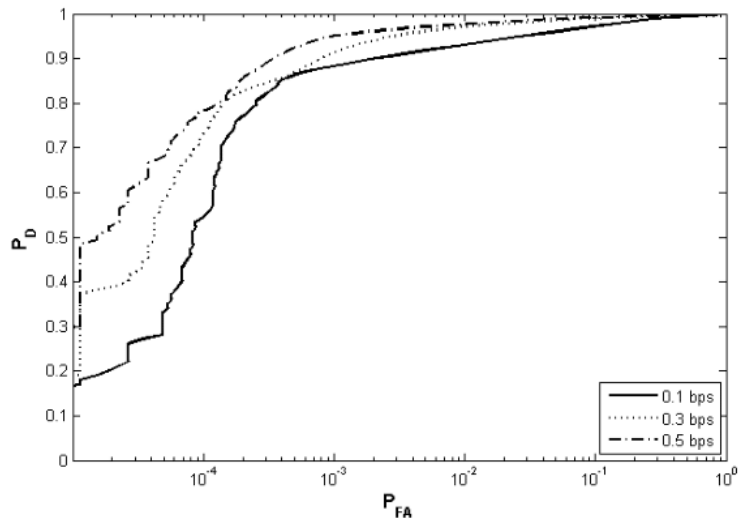


Figure 24 ROC Semi-Log curves for Low Altitude dataset at 0.1, 0.3 and 0.5 bps by using OBD-BCS algorithm

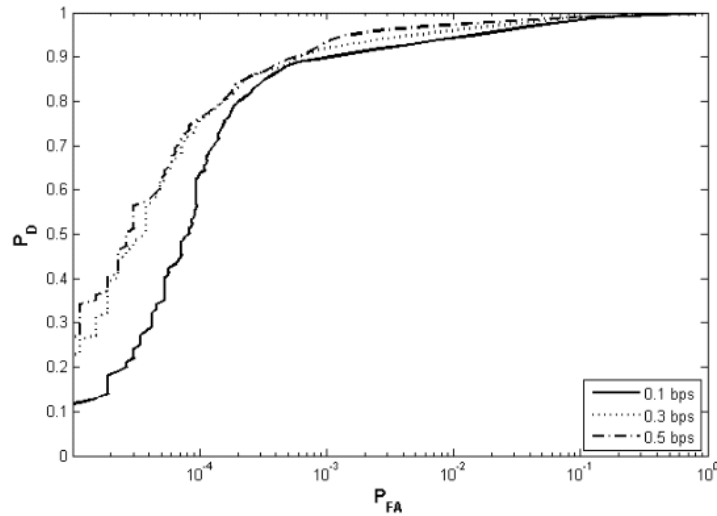


Figure 25 ROC Semi-Log curves for Low Altitude dataset at 0.1, 0.3 and 0.5 bps by using BP by using dual active set algorithm

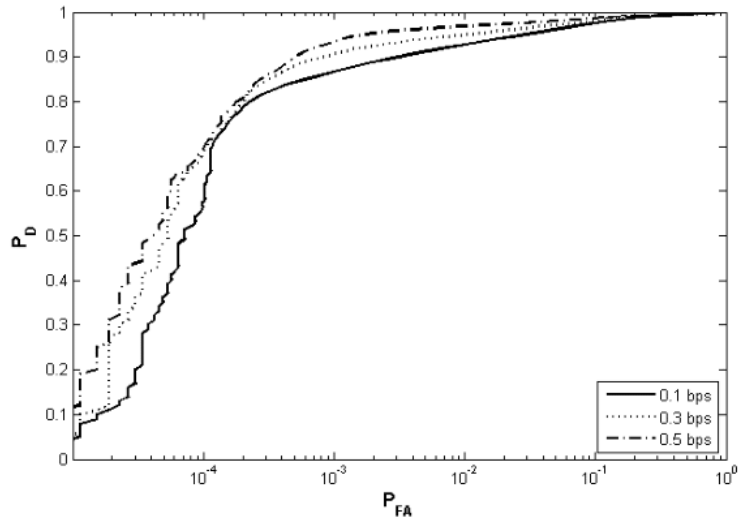


Figure 26 ROC Semi-Log curves for Low Altitude dataset at 0.1, 0.3 and 0.5 bps by using SpaRSA algorithm

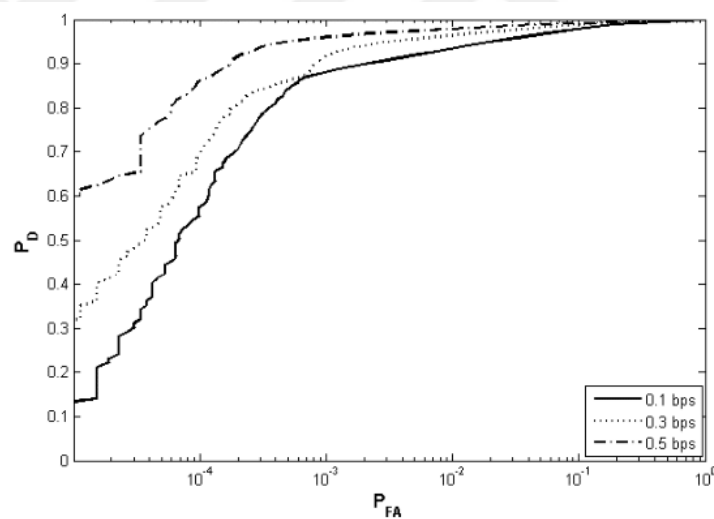


Figure 27 ROC Semi-Log curves for Low Altitude dataset at 0.1, 0.3 and 0.5 bps by using LASSO by using ADMM algorithm

Two major inferences can be observed from this anomaly detection application.

Firstly, the hyperspectral image which is compressed by using the hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding can estimate the original hyperspectral image. This is valid at 0.5 bps and 0.3 bps compression rates where considerable PSNR values are achieved. Instead of using the

original hyperspectral image to perform applications such as anomaly detection, it is effective to use the compressed version of it. This way the computational burden of the application is significantly reduced.

Secondly, as the bit rate increases, the compression performance of the OBD-BCS algorithm becomes more likely to outperform that of the other algorithms. Therefore, hyperspectral image compression using OBD-BCS algorithm based on online dictionary learning performs better particularly at 0.5 bps rate or higher.

#### **4.2 ANOMALY DETECTION EXPERIMENT 2 - Comparison of the sparse representation algorithms which are based on proximity based optimization method**

This section presents anomaly detection scheme which is applied on Pavia University hyperspectral dataset (cf. Table 3). BP by using dual active set algorithm, LASSO by using ADMM algorithm, SpaRSA algorithm and GIST algorithm are applied. The aim of this experiment is to test the information preservation performance of the main proximity based optimization algorithms such as SpaRSA algorithm and GIST algorithm.

Anomaly detection results for the Pavia University hyperspectral image are illustrated in Figure 28 in which two anomaly parts are marked with circles. Figure 28(a) displays the anomaly detection result of the original image. Figure 28(b), Figure 28(c) and Figure 28(d) illustrate the anomaly detection results of SpaRSA at 0.5, 0.3 and 0.1 bps bit rates, respectively. SpaRSA is able to detect the anomaly parts both at 0.5 and 0.3 bps levels.

Anomaly detection results belonging to BP by using dual active set algorithm, GIST algorithm and LASSO by using ADMM algorithm at 0.5, 0.3 and 0.1 bps rates are given in Figure 28(e)-Figure 28(g), Figure 28(h)-Figure 28(j) and Figure 28(k)-Figure 28(m), respectively. As a conclusion, the anomaly detection capabilities of SpaRSA are significantly superior to those obtained by other sparse representation algorithms.

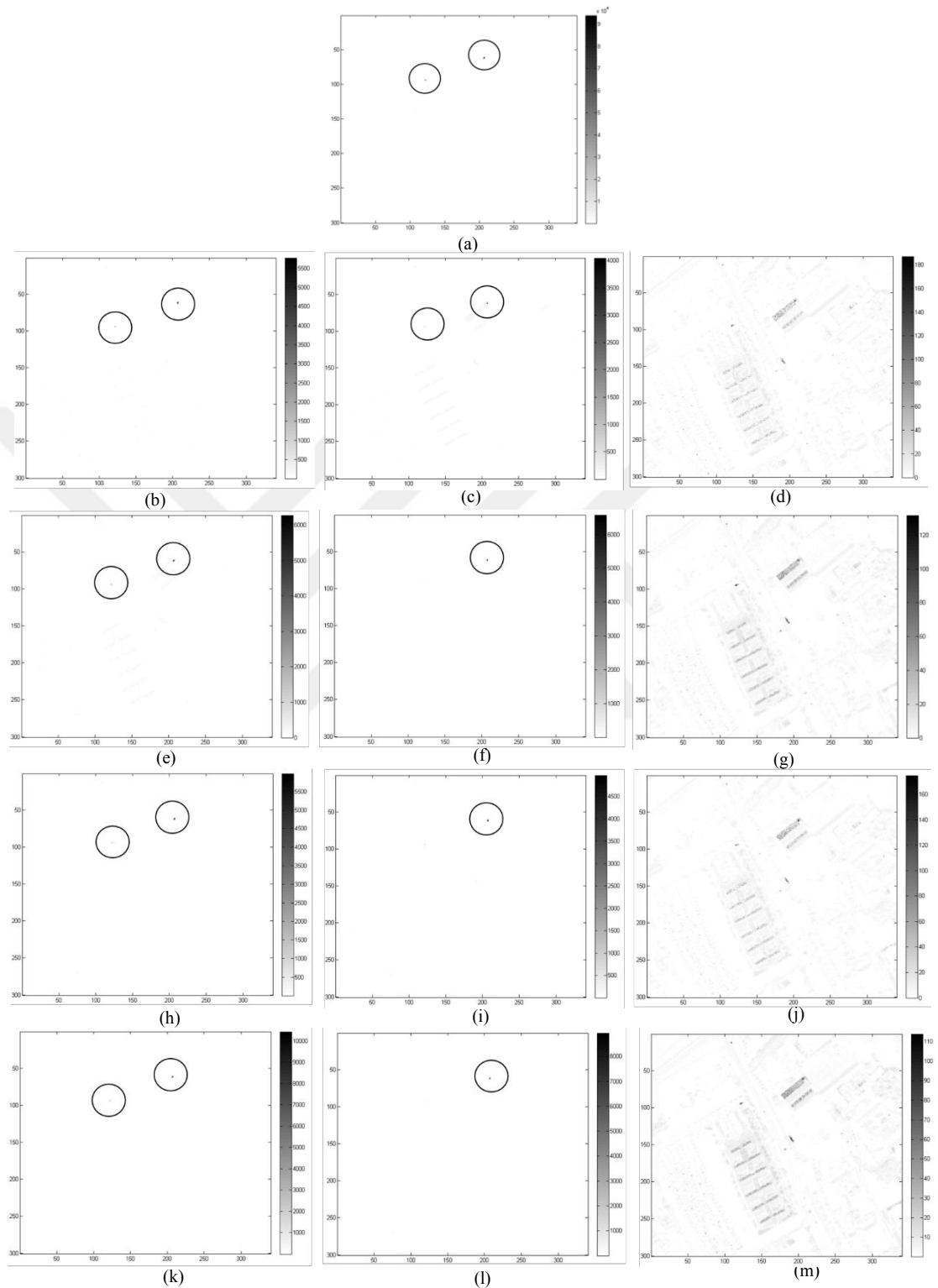


Figure 28 RX anomaly det. results of the Pavia University hyp. image: (a) original (b) SpaRSA 0.5 bps (c) 0.3 bps (d) 0.1 bps (e) BP 0.5 bps (f) 0.3 bps (g) 0.1 bps (h) GIST 0.5 bps (i) 0.3 bps (j) 0.1 bps (k) LASSO 0.5 bps (l) 0.3 bps (m) 0.1 bps

In order to assess the robustness of the anomaly detection results given in Figure 28, they should be based on the numerical PSNR values. Therefore, the corresponding PSNR values of each sparse representation at 0.1, 0.3 and 0.5 bps levels are given in Table 7 for Pavia University hyperspectral dataset. The highest two PSNR values are marked in boldface.

Table 7 Compression performances of sparse representation algorithms (The highest two PSNR values (in dBs) are shown in boldface for each bps value) – Pavia University Dataset

BPS	Pavia University Image			
	Sparse Representation Algorithms			
	LASSO (ADMM)	BP (Dual active set)	SpaRSA	GIST
0.1	<b>50.3037</b>	50.2574	<b>50.28</b>	50.0464
0.3	60.8013	60.6832	<b>60.9671</b>	<b>60.8486</b>
0.5	62.7433	62.3752	<b>62.8532</b>	<b>62.7913</b>

Concerning the PSNR values for Pavia University dataset in Table 7, at 0.5 and 0.3 bps rates SpaRSA algorithm is the best whereas GIST algorithm is ranked as the second. In this case, proximity based optimization algorithms seemed to have a better performance when compared to other algorithms.

Additional analyses are performed to further explore the information preservation ability of different sparse representation algorithms at various bit rates. ROC Semi-Log curves of BP by using dual active set algorithm, GIST algorithm, SpaRSA algorithm and LASSO by using ADMM algorithm are displayed in Figure 29-Figure 32 for the Pavia University hyperspectral dataset, respectively. In Figure 29 and Figure 31, ROC Semi-Log curves belonging to 0.5 bps and 0.3 bps bit rates are almost overlapping. In Figure 30 and Figure 32, even though a slight difference between the ROC Semi-Log performances of 0.5 bps and 0.3 bps, they are still relatively close to each other.

Anomaly detection results yield two major implications.

First, hyperspectral image which is compressed by using the hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding can effectively estimate the original hyperspectral dataset at 0.5 and 0.3 bps bit rates together

with a considerable PSNR values. Therefore, instead of using the original hyperspectral image in particular applications, one can use directly the compressed image which has a relatively smaller size.

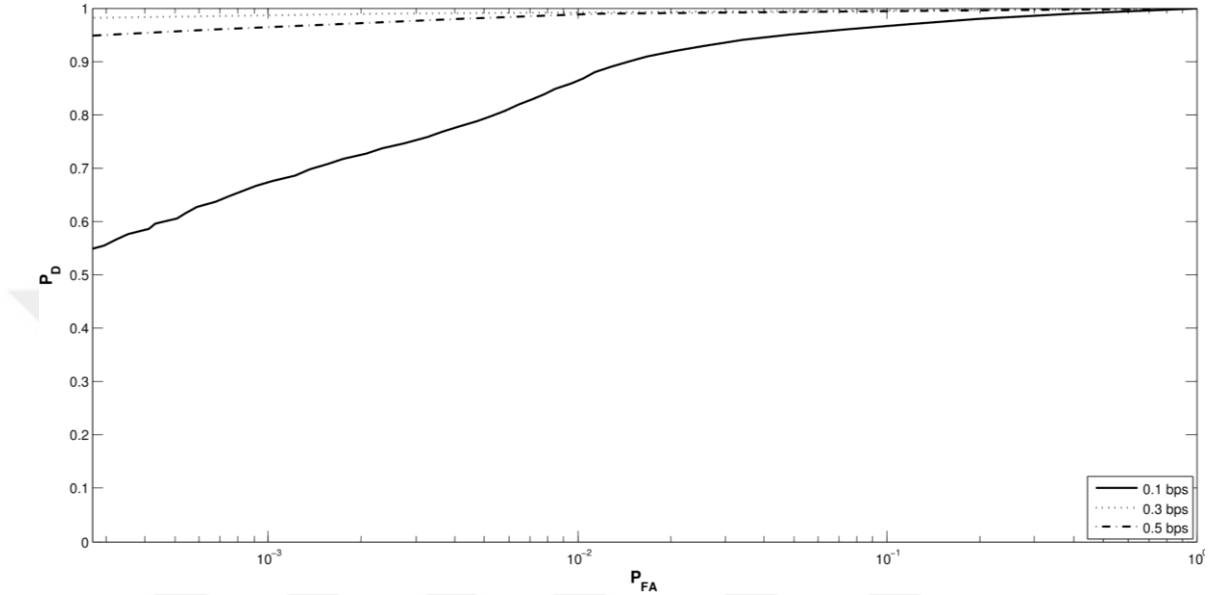


Figure 29 ROC Semi-Log curves for Pavia University dataset at 0.1, 0.3 and 0.5 bps by using BP by using dual active set algorithm

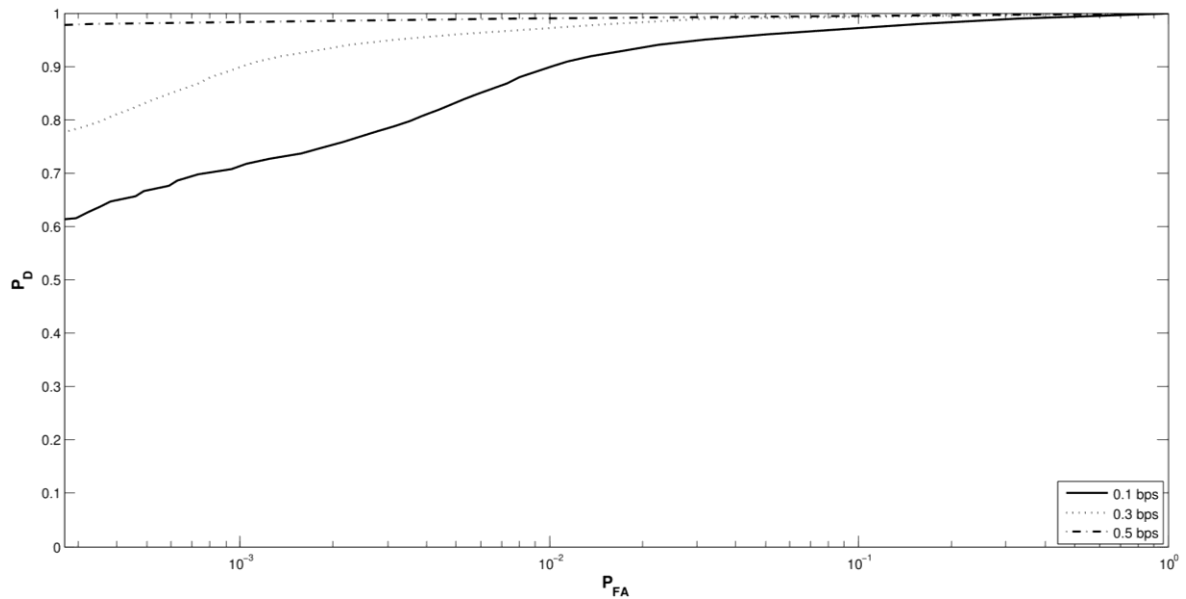


Figure 30 ROC Semi-Log curves for Pavia University dataset at 0.1, 0.3 and 0.5 bps by using GIST algorithm

Second, using SpaRSA algorithm in the hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding is slightly superior to the other sparse representation algorithms for each datasets.

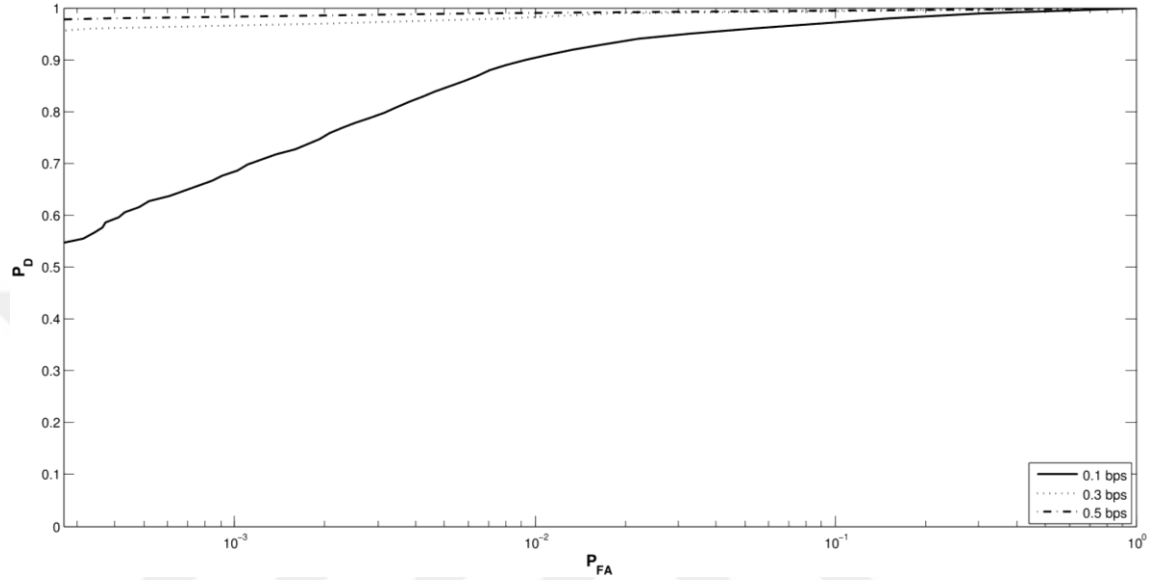


Figure 31 ROC Semi-Log curves for Pavia University dataset at 0.1, 0.3 and 0.5 bps by using SpaRSA algorithm

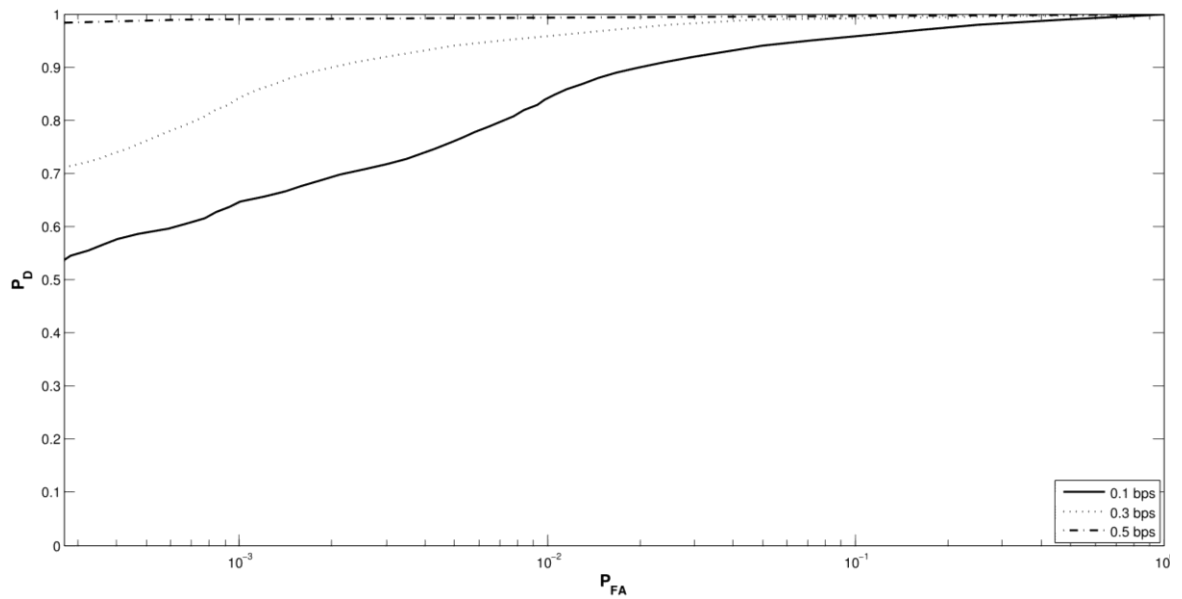


Figure 32 ROC Semi-Log curves for Pavia University dataset at 0.1, 0.3 and 0.5 bps by using LASSO by using ADMM algorithm

### 4.3 ANOMALY DETECTION EXPERIMENT 3 - The application of anomaly detection by using only TNIPM sparse representation algorithm exploited in the hybrid method

The three curves in Figure 33 are for three different RX anomaly detection results. Moffett Field with f970620t01p02r03sc02 flight number is used in the experiment which is an AVIRIS hyperspectral dataset. This test image was taken in the year 1997. It is a reflectance data with 10,000 reflectance factor. Dataset is stored as 16-bit signed integers. Although Moffett Field dataset in BIP notation has 512 lines by 614 samples by 224 bands, only the first 100 lines are utilized in this study. In Table 8, the corresponding PSNR values are presented for 0.2 bps and 0.4 bps bit rates. In this experiment only TNIPM algorithm is used as the sparse representation algorithm, since the reference method proposed by Mairal et. al. employs only TNIPM algorithm as the sparse representation algorithm [19].

Table 8 Compression performances of sparse representation algorithms

<b>Moffett Field Hyperspectral Dataset (cf. Table 3)</b>	
<b>BPS</b>	TNIPM algorithm
0.2	39.94
0.4	41.35
<b>Cuprite Hyperspectral Dataset (cf. Table 3)</b>	
<b>BPS</b>	TNIPM algorithm
0.1	41.45
0.5	49.75

Figure 33(a) introduces the results when RX anomaly detection is applied on the original Moffett Field dataset. The desired anomaly parts are marked with dashed circles. Figure 33(b) shows the anomaly detection results at 0.4 bps rate, while Figure 33(c) shows those at 0.2 bps rate. Even the hyperspectral dataset compressed at 0.2 bps rate can still be able to detect the desired anomalies. Further anomaly detection tests are performed by using Cuprite hyperspectral dataset (cf. Table 3). A sub-scene of 101 lines by 101 samples by 170 bands is cropped from the Cuprite hyperspectral dataset. PSNR values of this dataset



is summarized in Table 8 for 0.1 bps and 0.5 bps bit rates. The results of the RX anomaly detection tests which are applied on Cuprite dataset are summarized in Figure 34.

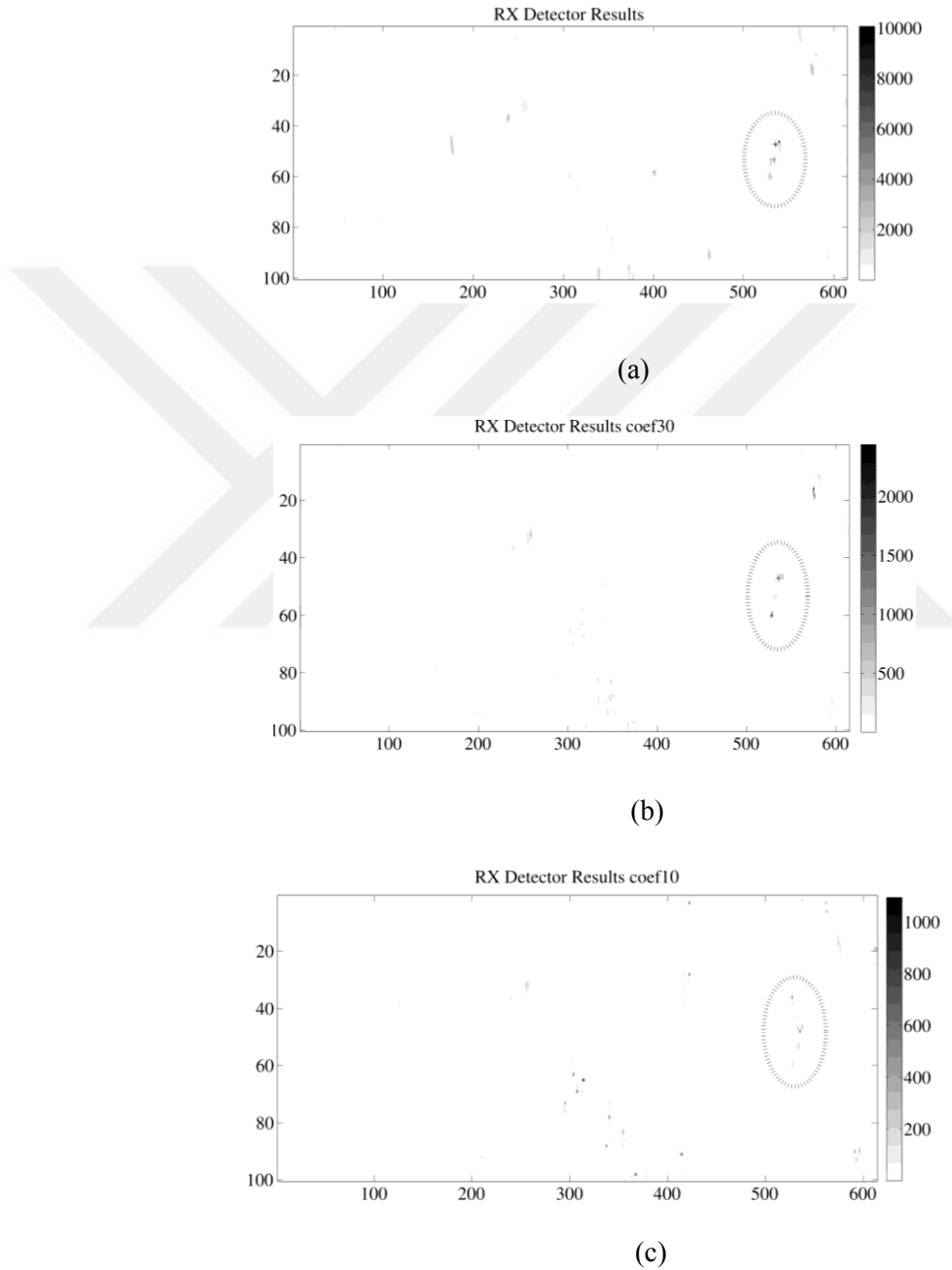


Figure 33 RX anomaly detection results of the Moffett Field hyperspectral image: (a) original image (b) reconstructed image with 0.4 bps (c) reconstructed image with 0.2 bps

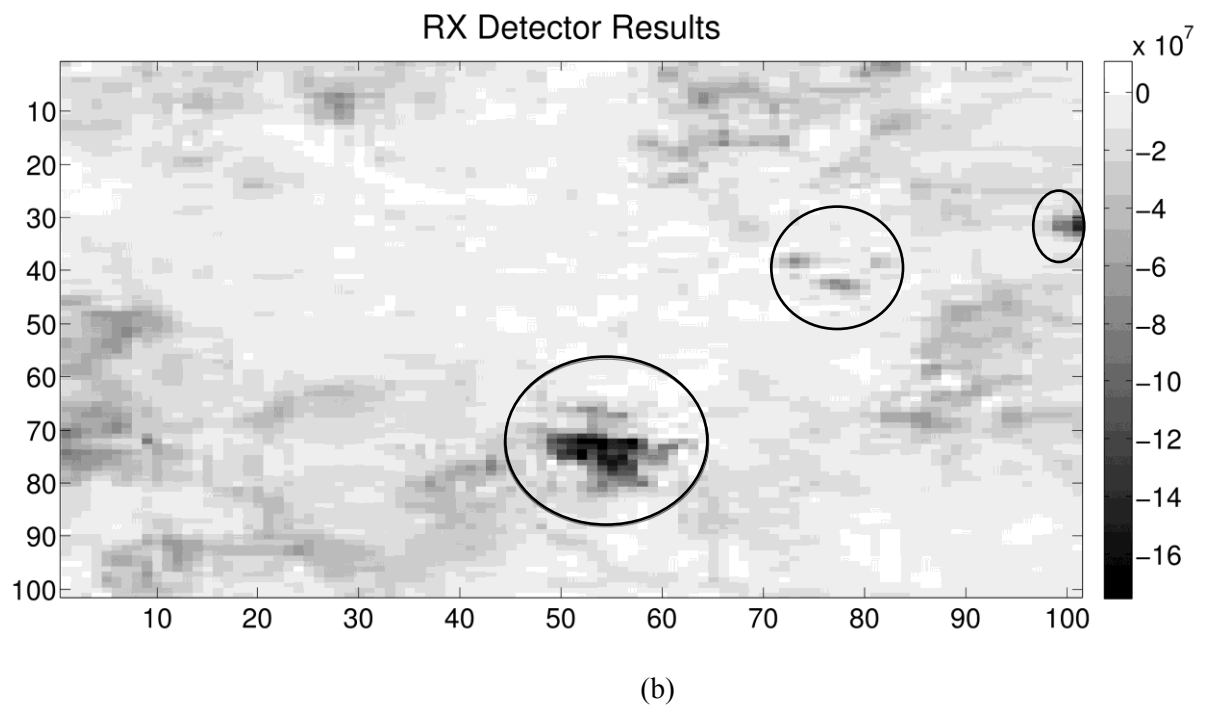
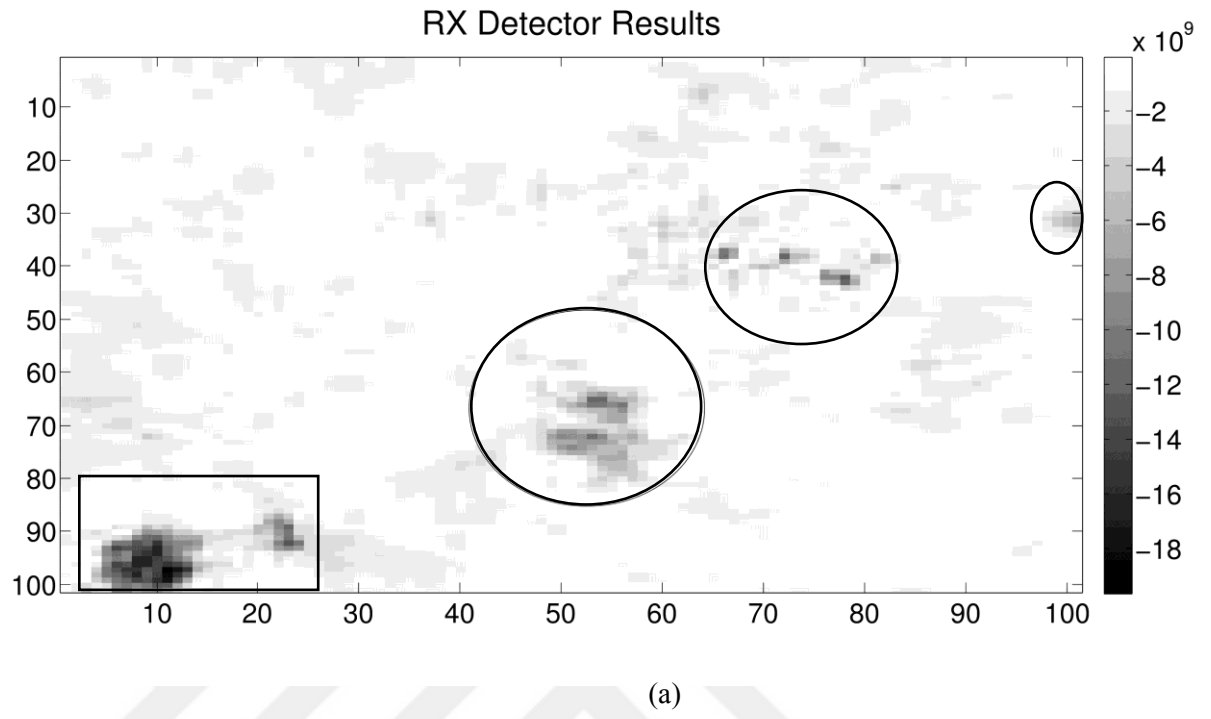


Figure 34 RX anomaly detection results of the Cuprite hyperspectral image: (a) reconstructed image with 0.5 bps (b) reconstructed image with 0.1 bps

Figure 34(a) shows the result in which dataset is compressed with 0.5 bps rate, while Figure 34(b) shows that of the 0.1 bps rate. Desired anomaly detection parts are marked with solid lines.

Although the anomaly parts marked with circles in Figure 34(a) can be detected in Figure 34(b) as well, the anomaly part marked with rectangle in Figure 34(a) cannot be detected in Figure 34(b).

The results of this experiment indicate that even the compressed hyperspectral image at a rate as low as 0.1 bps is able to detect the desired anomaly parts substantially. Since the aim of this experiment is to understand the detection performance of the proposed sparse coding with online learning approach rather than comparing the detection performances of different sparse representation algorithms, only TNIPM algorithm is used as a sparse representation algorithm.

## CHAPTER 5

### CONCLUSION

In this study, the online dictionary learning method proposed by Mairal et. al. [19] is adapted to the compression of hyperspectral images for the first time in the literature. Additionally, various sparse representation algorithms are utilized for the solution of the sparse coding problem in the algorithm which forms a hybrid approach. According to the rate-distortion performance results, this hybrid hyperspectral image compression method by using online dictionary learning based on sparse coding outperforms other compression methods in the literature including the state-of-the-art one.

Among the sparse representation algorithms used in this hybrid method, the proximity based optimization algorithms and blind compressive sensing algorithms which are the most recent and popular ones in the literature are analyzed further. The rate-distortion performances of these algorithms are superior to those based on other sparse representation algorithms. In addition to the rate-distortion measure, anomaly detection is also applied to test the information preservation performance of these algorithms.

Further analysis revealed that the rate-distortion performance of the OBD-BCS algorithm which is a blind compressive sensing scheme is superior to those of the other sparse representation algorithms when the bps value is greater than or equal to 0.5. Following the OBD-BCS, the second best performing algorithm is the SpaRSA algorithm at compression rates higher than 0.5 bps which belongs to the category of proximity based optimization approaches.

Since OBD-BCS exploits the learning process, the corresponding sparse coefficients are more accurate. These sparse coefficients are also subjected to an online learning process

in this hybrid method. Therefore, it is reasonable to observe that OBD-BCS algorithm has the best performance. On the other hand, the proximity based methods consider all possible parameter values by performing soft thresholding such that it is expected to obtain accurate results. Since SpaRSA is a computationally efficient proximity based algorithm, the observed results are reasonable.

The rate-distortion performances in terms of PSNR of both OBD-BCS algorithm and SpaRSA algorithm become higher as the bit rate gets higher. High bit rate lossy compression is valid for many earth observation missions. For instance, earth remote sensing which is performed by Pleiades (Optical Imaging Constellation of CNES) is realized by lossy compression at high bit rates up to 3 bps [63]. Therefore, the approach proposed in this study is promising to be employed in real hyperspectral data compression applications, since it performs well at high compression rates.

## REFERENCES

- [1] **PLAZA, A., PLAZA, J.** et. al. (2009) Lossy hyperspectral image compression tuned for spectral mixture analysis applications on nvidia graphics processing units, in *SPIE Optical Engineering+ Applications* (2-6 August 2009), San Diego 74550F-74550F.
- [2] **CHANG, C. I.** (2013) *Hyperspectral Data Processing: Algorithm Design and Analysis*, Hoboken, John Wiley & Sons, New Jersey.
- [3] **PENNA, B., TILLO, T.** et. al. (2007) Transform coding techniques for lossy hyperspectral data compression, *IEEE Transactions on Geoscience and Remote Sensing*, 1408-1421. vol. 45.
- [4] **CHRISTOPHE, E., LÉGER D.** et. al. (2005) Quality criteria benchmark for hyperspectral imagery, *IEEE Transactions on Geoscience and Remote Sensing*, 2103-2114, vol. 43.
- [5] **FOWLER, J. E.** (2008) Compressive-projection principal component analysis for the compression of hyperspectral signatures, in *IEEE Data Compression Conference (DCC)* (25-27 March 2008), Snowbird 83-92.
- [6] **FOWLER, J. E.** (2009) Compressive-projection principal component analysis, *IEEE Transactions on Image Processing*, 2230-2242. vol. 18.
- [7] **MAGLI, E., GABRIELLA, O.** et. al. (2004) Optimized onboard lossless and near-lossless compression of hyperspectral data using CALIC, *IEEE Geosc. Remote Sens. Lett.*, 21–25. vol. 1.
- [8] **BILGIN, A., ZWEIG, G.** et. al. (2000) Three-dimensional image compression with integer wavelet transforms, *Applied Optics*, 1799–1814. vol. 39.

- [9] **CHRISTOPHE, E.** (2011) *Hyperspectral data compression tradeoff*. In: *Optical Remote Sensing*, Springer, Berlin Heidelberg.
- [10] **CHRISTOPHE, E., CORINNE, M.** et. al. (2008) Hyperspectral image compression: adapting SPIHT and EZW to anisotropic 3-D wavelet coding, *IEEE Trans. Image Process.*, 23346–23348. vol. 17.
- [11] **SOLTANI-FARANI, A., RABIEE H. R.** et. al. (2015), Spatial-aware dictionary learning for hyperspectral image classification, *IEEE Transactions on Geoscience and Remote Sensing*, 527-541. vol. 53.
- [12] **CHARLES, A. S., OLSHAUSEN B. A.** et. al. (2011) Learning sparse codes for hyperspectral imagery, *IEEE Journal of Selected Topics in Signal Processing*, 963-978. vol. 5.
- [13] **ZHANG, Z., XU, Y.** et. al. (2015) A survey of sparse representation: algorithms and applications, *IEEE Access*, 490-530. vol. 3.
- [14] **WRIGHT, J., YANG, A. Y.** et. al. (2009) Robust face recognition via sparse representation, *IEEE transactions on pattern analysis and machine intelligence*, 210-227. vol. 31.
- [15] **ELAD, M.** (2010) *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, New York.
- [16] **YANG, J., PENG, Y.** et. al. (2009) Ways to sparse representation: an overview, *Science in China series F: information sciences*, 695-703. vol. 52.
- [17] **BOTTOU. L., BOUSQUET, O.** (2007) The tradeoffs of large scale learning. In: *Advances in Neural Information Processing Systems, NIPS*, 161–168. vol. 20.
- [18] **MAIRAL, J., BACH, F.** et. al. (2010) Online learning for matrix factorization and sparse coding, *Journal of Machine Learning Research*, 19-60. vol. 11.
- [19] **MAIRAL, J., BACH, F.** et. al. (2009) Online dictionary learning for sparse coding, in *Proceedings of the 26th annual international conference on machine learning (ICML)* (14-18 June 2009), Montreal, QC. 689-696.

- [20] **WANG, Z., NASRABADI, N.M.** et. al. (2014) Spatial-spectral classification of hyperspectral images using discriminative dictionary designed by learning vector quantization, *IEEE Trans. Geosci. Remote Sens.*, 4808–4822. vol. 52.
- [21] **ÜLKÜ I., TÖREYİN, B. U.** (2015) Sparse coding of hyperspectral imagery using online learning, *Signal, Image Video Process.*, 959–966. vol. 9.
- [22] **ÜLKÜ I., TÖREYİN, B. U.** (2015) Sparse representations for online-learning-based hyperspectral image compression, *Applied optics*, 8625-8631. vol. 54.
- [23] **MALLAT S. G., ZHANG, Z.** (1993) Matching pursuits with time-frequency dictionaries, *IEEE Transactions on signal processing*, 3397-3415. vol. 41.
- [24] **DAVIS, G. M., MALLAT S. G.** et. al. (1994) Adaptive time-frequency decompositions, *Optical Engineering*, 2183-2191. vol. 33.
- [25] **TROPP J. A., GILBERT, A. C.** (2007) Signal recovery from random measurements via orthogonal matching pursuit, *IEEE Transactions on information theory*, 4655-4666. vol. 53.
- [26] **PATI, Y. C., REZAIIFAR R.** et. al. (1993) Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition, in *Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers* (1-3 November 1993), Pacific Grove, California, 40-44.
- [27] **WANG, J., KWON S.** et. al. (2012) Generalized orthogonal matching pursuit, *IEEE Transactions on signal processing*, 6202-6216. vol. 60.
- [28] **DONOHO, D. L., TSAIG, Y.** et. al. (2012) Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit, *IEEE Transactions on Information Theory*, 1094-1121. vol. 58.
- [29] **NEEDELL D., R. VERSHYNIN, R.** (2009) Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit, *Foundations of computational mathematics*, 317-334. vol. 9.



- [30] **GUI, G., WAN, Q.** et. al. (2010) Sparse multipath channel estimation using compressive sampling matching pursuit algorithm, *arXiv preprint arXiv: 1005.2270*, [Online]. Available: [arxiv.org/pdf/1005.2270](http://arxiv.org/pdf/1005.2270)
- [31] **NOWAK R. D., WRIGHT, S. J.** (2007) Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, *IEEE Journal of selected topics in signal processing*, 586-597. vol. 1.
- [32] **BOYD, S., VANDENBERGHE, L.** (2004) *Convex optimization.*, Cambridge university press, Cambridge.
- [33] **KIM, S. J., KOH, K.** et. al. (2007) An interior-point method for large-scale-regularized least squares, *IEEE journal of selected topics in signal processing*, 606-617. vol. 1.
- [34] **BOYD, S., PARIKH, N.** et. al. (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends® in Machine Learning*, 1-122. vol. 3.
- [35] **FRIEDLANDER M., SAUNDERS, M.** (2012) A dual active-set quadratic programming method for finding sparse least-squares solutions, *University of British Columbia*, [Online]. Available at: <https://web.stanford.edu/group/SOL/software/asp/bpdual.pdf>.
- [36] **CHEN, S. S., DONOHO D. L.** et. al. (2001) Atomic decomposition by basis pursuit, *SIAM review*, 129-159. vol. 43.
- [37] **CONDAT, L.** (2014) A generic proximal algorithm for convex optimization—application to total variation minimization, *IEEE Signal Processing Letters*, 985-989. vol. 21.
- [38] **PARIKH N., BOYD, S. P.** (2014) Proximal Algorithms, *Foundations and Trends in optimization*, 127-239. vol. 1.

- [39] **FIGUEIREDO M. A., NOWAK, R. D.** (2005) A bound optimization approach to wavelet-based image deconvolution, in *IEEE International Conference on Image Processing (ICIP)* (11-14 September 2005), Genova II-782.
- [40] **BECK A., TEBoulLE, M.** (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM journal on imaging sciences*, 183-202. vol. 2.
- [41] **YANG, A. Y., SASTRY, S. S. et. al. A.** (2010) Fast l1-minimization algorithms and an application in robust face recognition: A review, in *IEEE International Conference on Image Processing (ICIP)* (26-29 September 2010), Hong Kong 1849-1852.
- [42] **WRIGHT, S. J., NOWAK R. D. et. al.** (2009) Sparse reconstruction by separable approximation, *IEEE Transactions on Signal Processing*, 2479-2493. vol. 57.
- [43] **BIOUCAS-DIAS J. M., FIGUEIREDO, M. A.** (2007) A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration, *IEEE Transactions on Image processing*, 2992-3004. vol. 16.
- [44] **BIOUCAS-DIAS J. M., FIGUEIREDO, M. A.** (2007) Two-step algorithms for linear inverse problems with non-quadratic regularization, in *IEEE International Conference on Image Processing (ICIP)* (16-19 September 2007), San Antonio, TX I-105.
- [45] **GONG, P., ZHANG, C. et. al.** (2013) A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems, in *30th International Conference on Machine Learning (ICML)* (16-21 June 2013), Atlanta 37-45.
- [46] **GONG, P., ZHANG, C. et. al.** (2013) GIST: General Iterative Shrinkage and Thresholding for Non-convex Sparse Learning, *Tsinghua University*, [Online]. Available at: [http://www.public.asu.edu/~pgong5/papers/manual\\_GIST.pdf](http://www.public.asu.edu/~pgong5/papers/manual_GIST.pdf).
- [47] **YANG, A. Y., ZHOU, Z. et. al.** (2013) Fast-minimization algorithms for robust face recognition, *IEEE Transactions on Image Processing*, 3234-3246. vol. 22.

- [48] **FU, W.J.** (1998) Penalized regressions: the bridge versus the lasso, *Journal of computational and graphical statistics*, 397-416. vol. 7.
- [49] **ZUO, W., MENG, D.** et. al. (2013) A generalized iterated shrinkage algorithm for non-convex sparse coding, in *Proceedings of the IEEE international conference on computer vision (ICCV)* (1-8 December 2013), Sydney 217-224.
- [50] **GORODNITSKY I.F., RAO, B.D.** (1997) Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm”, *IEEE Transactions on signal processing*, 600-616. vol. 45.
- [51] **COTTER, S. F., RAO, B. D.** et. al. (2006) Sparse solutions to linear inverse problems with multiple measurement vectors, *IEEE Transactions on Signal Processing*, 2477-2488. vol. 53.
- [52] **OSBORNE, M. R., PRESNELL B.** et. al. (2000) A new approach to variable selection in least squares problems, *IMA journal of numerical analysis*, 389-403. vol. 22.
- [53] **DONOHO, D., TSAIG, Y.** (2006). Fast solution of  $l_1$ -norm minimization problems when the solution may be sparse, *Stanford University, Stanford* [Online]. Available at: <http://dsp.rice.edu/sites/dsp.rice.edu/files/cs/FastL1.pdf>.
- [54] **ASIF, M. S., ROMBERG, J.** (2009) Dynamic Updating for  $l_1$  Minimization, *arXiv preprint arXiv: 0903.1443*, [Online]. Available at: <http://arxiv.org/pdf/0903.1443v2.pdf>.
- [55] **JI, S., XUE Y.** et. al. (2008), Bayesian compressive sensing, *IEEE Transactions on Signal Processing*, 2346-2356. vol. 56.
- [56] **TIPPING, M. E.** (2001) Sparse Bayesian learning and the relevance vector machine, *Journal of machine learning research*, 211-244. vol. 1.
- [57] **BARANIUK, R.G., CEVHER, V.** et. al. (2010) Model-based compressive sensing”, *IEEE Transactions on Information Theory*, 1982-2001. vol. 56.

- [58] **NEEDELL D., TROPP, J. A.** (2009) CoSaMP: Iterative signal recovery from incomplete and inaccurate samples”, *Applied and Computational Harmonic Analysis*, 301-321. vol. 26.
- [59] **GLEICHMAN S., ELDAR, Y. C.** (2011) Blind compressed sensing, *IEEE Transactions on Information Theory*, 6958-6975. vol. 57.
- [60] **REED, S.I., YU, X.** (1990) Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution, *IEEE Trans. Acoust. Speech Signal Process.*, 1760–1770. vol. 38.
- [61] **MOTTA, G., RIZZO, F., STORER, J. A.** (2006) *Hyperspectral Data Compression*, Germany: Springer-Verlag, Berlin.
- [62] **WANG, L., ZHAO C.** (2015) *Hyperspectral Image Processing.*, Germany: Springer, Berlin.
- [63] **RICCI, M., MAGLI, E.** (2013) Predictor analysis for onboard lossy predictive compression of multispectral and hyperspectral images, *J. Appl. Remote Sens.*, 074591–074591. vol. 7.
- [64] **KIELY, A. B., KLIMESH, M. A.** (2009) Exploiting calibration-induced artifacts in lossless compression of hyperspectral imagery, *IEEE Transactions on Geoscience and Remote Sensing*, 2672-2678. vol. 47.
- [65] **MAGLI, E.** (2009) Multiband lossless compression of hyperspectral images, *IEEE Trans. Geosci. Remote Sens.*, 1168–1178, vol. 47.
- [66] **MIELIKAINEN, J., TOIVANEN, P.** (2008) Lossless compression of hyperspectral images using a quantized index to lookup tables, *IEEE Geosci. Remote Sens. Lett.*, 474–478, vol. 5.
- [67] **HOREV, I., BRYT, O., RUBINSTEIN, R.** (2012) Adaptive image compression using sparse dictionaries, *in: Proc. of IWSSIP* (11-13 April 2012) Vienna, Austria. 592–595.

- [68] **CHENG, H.** (2015) *Sparse representation, modeling and learning in visual recognition*, Springer, London.
- [69] **QIAN, D., WEI, Z.** et. al. (2008) Anomaly-based hyperspectral image compression, in *2008 IGARSS, IEEE International Geoscience and Remote Sensing Symposium* (611 July 2008), Boston, MA 974–977.
- [70] **QIAN, D., FOWLER, J.E.** (2007) Hyperspectral Image Compression Using JPEG2000 and Principal Component Analysis, *IEEE Trans. Geosci. Remote Sens.*, 201–205. vol. 4.
- [71] **PENNA, B., TILLO, T.** et. al. (2007) Hyperspectral image compression employing a model of anomalous pixels, *IEEE Geosci. Remote Sens. Lett.*, 664–668, vol. 4.
- [72] **DRAGOTTI, P. L., POGGI, G., RAGOZINI, A. R. P.** (2000) Compression of multispectral images by three-dimensional SPIHT algorithm, *IEEE Trans. Geosci. Remote Sens.*, 416–428, vol. 38.
- [73] **FRY, T. W., HAUCK, S.** (2002) Hyperspectral image compression on reconfigurable platforms, in *IEEE Symposium on Field-Programmable Custom Computing Machines* (24 April 2002) Napa, CA, USA. 251–260.
- [74] **NGADIRAN, R., BOUSSAKTA, S.** et. al. (2010) Hyperspectral image compression with modified 3D SPECK, in *7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP)* (21-23 July 2010), Newcastle upon Tyne 806–810.
- [75] **MOTTA, G., RIZZO, F., STORER, J. A.** (2003) Compression of hyperspectral imagery, in *Proc. Data Compression Conf.* (25-27 March 2003), Snowbird, UT, USA. 333–342.
- [76] **PICKERING, M., RYAN, M.** (2001) Efficient spatial-spectral compression of hyperspectral data, *IEEE Trans. Geosci. Remote Sensing*, 1536–1539, vol. 39.

- [77] **RYAN, M. J., ARNOLD, J. F.** (1997) The lossless compression of AVIRIS images by vector quantization, *IEEE Trans. Geosci. Remote Sensing*, 546–550, vol. 35.
- [78] **HUO, C., ZHANG, R.** et. al. (2012) Hyperspectral data compression using sparse representation, in *2012 4th Workshop on Hyperspectral Image and Signal Processing (WHISPERS)* (4-7 June 2012), Shanghai 1–4.
- [79] **RISH, I., GRABARNIK, G.** (2014) *Sparse Modeling: Theory, Algorithms, and Applications*, USA: CRC Press. Boca Raton, FL.
- [80] **PENDSE, G. V.** (2011) A tutorial on the lasso and the shooting algorithm, Technical report, P.A.I.N Group, *Imaging and Analysis Group - McLean Hospital*, Harvard Medical School.
- [81] **BACH, F., JENATTON, R.** et. al. (2012) Optimization with sparsity-inducing penalties, *Foundations Trends Mach. Learning*, 1–106, vol. 4.
- [82] **ESLAHI, N., AGHAGOLZADEH A.** et. al. (2014), Block compressed sensing images using accelerated iterative shrinkage thresholding, in *22nd Iranian Conference on Electrical Engineering (ICEE)* (20-22 May 2014), Tehran 1569-1574.
- [83] **HOU Y., ZHANG, Y.** (2014) Effective hyperspectral image block compressed sensing using thress-dimensional wavelet transform, in *IEEE Geoscience and Remote Sensing Symposium (IGARSS)* (13-18 July 2014), Quebec City 2973-2976.
- [84] **BURDGE, B., KREUTZ-DELGADO, K., MURRAY, J.** (2010) A unified FOCUSS framework for learning sparse dictionaries and non-squared error, in *Signals, Systems and Computers (ASILOMAR)* (7-10 Nov. 2010), Pacific Grove 2037 –2041.
- [85] **NYEO S.L., ANSARI, R.R.** (2011) Sparse bayesian learning for the laplace transform inversion in dynamic light scattering, *Journal of Computational and Applied Mathematics*, 2861–2872, vol. 235.

- [86] **ZHANG, L., WEI W.** (2016) Exploring Structured Sparsity by a Reweighted Laplace Prior for Hyperspectral Compressive Sensing, *IEEE Transactions on Image Processing*, 4974-4988. vol.
- [87] **BEGG, R., LAI, D., PALANISWAMI, M.** (2008) *Computational Intelligence in Biomedical Engineering*, CRC Press. Boca Raton, FL.
- [88] **SAAD, D.** (1999) *Online Learning in Neural Networks*, Cambridge Univ. Press. London, U.K.
- [89] **JIFARA, W., FENG, J., ZHANG. B.** et al. (2017) Hyperspectral image compression based on online learning spectral features dictionary, *Multi. Tools and Appl.*, 1573-7721.
- [90] **G. CHEN, G., NEEDELL, D.** (2015) Compressed Sensing and Dictionary Learning, Preprint, [Online]. Available at: <http://www.ams.org/meetings/short-courses>
- [91] **MAIRAL, J., BACH, F.** et. al. (2009) Online dictionary learning for sparse coding, in *Proceedings of the 26th annual international conference on machine learning (ICML)* (14-18 June 2009), Montreal, QC. 689-696.
- [92] **OLSHAUSEN, B. A., FIELD, D. J.** (1997) Sparse coding with an overcomplete basis set: A strategy employed by V1?, *Vision research*, 3311-3325. vol. 37.
- [93] **AHARON, M., ELAD, M.** (2008) Sparse and redundant modeling of image content using an image-signature-dictionary, *SIAM Journal on Imaging Sciences*, 228-247. vol. 1.
- [94] **KRUSE, F. A.** (2002) Comparison of AVIRIS and Hyperion for hyperspectral mineral mapping, in *11th JPL Airborne Geoscience Workshop* (4-8 March 2002), Pasadena vol. 4.
- [95] AVIRIS Latitude/Longitude Quicklooks. (2011, April 19). [Online]. Retrieved from <http://aviris.jpl.nasa.gov/html/aviris.quicklooks.html>

- [96] **TARABALKA, Y., BENEDIKTSSON, J. A.** et. al. (2009) Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques, *IEEE Transactions on Geoscience and Remote Sensing*, 2973-2987. vol. 47.
- [97] **DALLA MURA, M., VILLA, A.** et. al. (2011) Classification of hyperspectral images by using extended morphological attribute profiles and independent component analysis, *IEEE Geoscience and Remote Sensing Letters*, 542-546. vol. 8.
- [98] **MASTRIANI, M., GAMBINI, J.** (2010) Fast Cosine Transform to Increase Speed-up and Efficiency of Karhunen-Loève Transform for Lossy Image Compression, *International Journal of Engineering and Mathematical Sciences*, 82-92, vol. 6.
- [99] **SALOMON, D.** (2004) *Data Compression: The Complete Reference*, Springer. New York.



## CURRICULUM VITAE

### PERSONAL INFORMATION

Surname, Name: Ülkü, İrem

Nationality: Turkish (TC)

Date and Place of Birth: 1 January 1988, Ankara

Marital Status: Single

Phone: +90 312 233 10 00/1508

e-mail: [iremulku@cankaya.edu.tr](mailto:iremulku@cankaya.edu.tr)

### EDUCATION

Degree	Institution	Year of Graduation
MS	METU Department of Electrical and Electronics Engineering	2013
BS	Çankaya Univ. Department of Electronic and Communication Engineering	2010
BS	Çankaya Univ. Department of Industrial Engineering	2009/Valedictorian

### WORK EXPERIENCE

Year	Place	Enrollment
2014-Present	Çankaya Univ. Department of Electrical and Electronics Engineering	Research/Teaching Assistant
2010-2014	Çankaya Univ. Department of Electronic and Communication Engineering	Research/Teaching Assistant
2009-2010	Çankaya Univ. Department of Industrial Engineering	Research/Teaching Assistant

## **FOREIGN LANGUAGES**

Advanced English (TOEFL, 96)

## **PUBLICATIONS**

### Journal Papers (SCI):

\* İ. Ülkü and B. Uğur Töreyn, "Sparse coding of hyperspectral imagery using online learning," *Signal, Image and Video Processing*, 9(4), 959-966 (2015).

\* İ. Ülkü and B. Töreyn, "Sparse representations for online-learning-based hyperspectral image compression," *Appl. Opt.*, 54(15), 8625-8631 (2015).

\* İ. Ülkü, E. Kızgut, "Large-scale Hyperspectral Image Compression via Sparse Representations based on Online Learning," *International Journal of Applied Mathematics and Computer Science*, 28(1), (2018)

### Conference and Workshop Papers (International):

\* İ.Ülkü and B.U.Töreyn "Hyperspectral Image Compression Using an Online Learning Method," in *Proc. of SPIE Defense, Security and Sensing (DSS) Satellite Data Compression, Communications, and Processing Conference*, (Internal Society for Optics and Photonics, 2015), Baltimore, Maryland, USA.

\* İ.Ülkü and B.U.Töreyn, "Lossy Compression of Hyperspectral Images using Online Learning Based Sparse Coding," in *Proc. of International Workshop On Computational Intelligence for Multimedia Understanding (IWCIM)*, (IEEE, 2014), Paris, France, pp. 1-5.

\* İ. Ülkü and E. Kızgut, "Hyperspectral Compressive Sensing based on Online Dictionary Learning," in *Imaging and Applied Optics 2017 (3D, AIO, COSI, IS, MATH, pcAOP)*, OSA Technical Digest (online) (Optical Society of America, 2017), paper ITh4E.1.

## **HOBBIES**

Fashion, Pilates, Fitness, Movies, Tennis.