Structural Dictionary Learning and Sparse Representation with Signal and Image Processing Applications

Mahmoud Nazzal

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Doctor of Philosophy
in
Electrical and Electronic Engineering

Eastern Mediterranean University
August 2015
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Serhan Çiftçioğlu
Acting Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Doctor of Philosophy of in Electrical and Electronic Engineering.

Prof. Dr. Hasan Demirel
Chair, Electrical and Electronic Engineering Department

We 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 in Electrical and Electronic Engineering.

Prof. Dr. Hüseyin Özkaramanlı
Supervisor

Examining Committee

1. Prof. Dr. Enis Çetin ________________________________

2. Prof. Dr. Hasan Demirel ________________________________

3. Prof. Dr. Hüseyin Özkaramanlı ________________________________

4. Prof. Dr. Osman Kükrer ________________________________

5. Prof. Dr. Bülent Sankur ________________________________

ABSTRACT

The success of sparse representation as a signal representation mechanism has been well-acknowledged in various signal and image processing applications, leading to the state-of-the-art performances. Flexibility and local adaptivity form the main advantage of this representation. It has been widely acknowledged that dictionary design (number of dictionaries, and the number of atoms in a dictionary) has strong implications on the whole representation process. This thesis addresses sparse representation over multiple learned dictionaries aiming at enhancing the representation quality and reducing the computational complexity.

The first contribution in this work is performing dictionary learning and sparse representation in the wavelet domain, merging the desirable attributes of wavelet transform with the representation power of learned dictionaries. Simulations conducted over the problem of single-image super-resolution show that this representation framework is able to improve the representation quality while reducing the computational cost.

Our second contribution is a variable patch size sparse representation paradigm. In this setting, the size of the patch is adaptively determined to enhance the quality of sparse representation.

The third contribution is a strategy for designing directionally-structured dictionaries via subspace projections. Experimental results show that this strategy improves the quality of sparse representation at reduced computational complexity.

The fourth and major contribution is a strategy for residual component-based multiple structured dictionary learning. In this work, we show that a signal and its residual components subject to a sparse coding algorithm do not necessarily follow the same model, as commonly assumed in the multiple dictionary approaches in the literature so far. Accordingly, we propose a mechanism whereby training signal can potentially contribute to the learning of several dictionaries, based on the structure of each of its residual components. This strategy is shown to significantly improve the representation quality while using compact dictionaries.

The final contribution in this thesis aims at improving the representation quality of a learned dictionary by performing a second dictionary learning pass over the residual components of the training set. Simulations show that this learning strategy improves the quality of sparse representation.

Keywords: Sparse representation, dictionary learning, multiple dictionaries, residual components, structured dictionaries.

ÖZ

Seyrek sinyal temsiliyetinin farklı sinyal ve görüntü işleme uygulamalarındaki başarımı kabul görmektedir. Bu yötemin temel özelliği bir sözlükten seçilen birkaç prototip sinyal (atom) ile sinyallerin temsil edilmesidir. Bu yöntemin temel avantajı sinyallere uygulanabilir bir yapıya sahip olmasıdır. Sözlük tasarımı (sözlük ve atom sayıları) ve kullanımı sinyal temsiliyetinde büyük önem arzetmektedir. Bu tezdeki çalışmalar çoklu sözlükler kullanarak hem sinyal temsiliyetinde kaliteyi artırmayı ve hasaplama karmaşıklığını azaltmayı hedeflemektedir.

Birinci katkı sözlük öğrenme yönteminin dalgacık dönüşümü ile yapılmasıdır. Dalgacık dönüşümünün birçok özelliğinden faydalanarak sözlükler dalgacık alanında öğrenilmiştir. Tek görüntünün çözünürlüğünün artırılması konusunda elde edilen sonuçlarla hem kalitenin arttığı hem hasaplama karmaşıklığının azaldığı gösterilmiştir.

Tezdeki ikinci katki temsiliyet kalitesinin artırılması için sözlüklerin değişken yama boyutu kullanarak öğrenilmeşidir. Temisilyet kalitesi en iyi yama boyutu seçilerek iyileştirilmiştir. Üçüncü katkı yansıtma operatörleri kullanarak değişik yönlere sahip çoklu sözlük öğrenme yöntemi ve bu yönteme dayalı sinyal temsiliyet algoritması geliştirilmeşidir. Sinyal temsiliyetinde önerilen yöntemin hem kaliteyi hem de hasaplama karmaşıklığını iyileştirdiği gözlemlen miştir.

Tezde yapılan dördüncü katkı artık bileşenler tabanında çoklu ve yapısal özelliklere sahip sözlüklerin tasarlanmasdır. Öncelikle bir sinyalin ve onun artık bileşenlerinin farklı yapılarda olduğu gösterilmiştir. Bu gerçekten yola çıkarak artık bileşen sinyalleri kullanarak yeni bir çoklu yapısal sözlük öğrenme yöntemi önerilmiştir. Önerilen öğrenme yöntemine dayalı sinyal temsiliyet yöntemi de önerilmiştir. Önerilen yöntem ile bir sinyal birden fazla sözlüğün uyarlanmasına katkı yapabileceği gibi temsiliyet safhasında herhangi bir sinyal farklı yapısal özelliklere sahip sözlüklerden atomlar kullanılarak temsil edilebilmektedir. Önerilen yöntemin sinyal temsiliyetinde önemli iyileştirmeler sağladığı gösterilmiştir. Bu tezdeki beşinci ve son katkı ise öğrenilen sözlüğün temsiliyet kalitesini artırmak için hata sinyallerini kullanarak ikinci bir öğrenme safhası kullanmaktır. İkinci safhadaki öğrenmede problemi Lagrange en iyileme yöntemi ile çözülmüştür. İkinci safhadaki öğrenmede öğrenilen sözlüklerin ilk safhadaki kaliteyi düşüremeyecği sınırlaması getirilmiştir. Lagrange çarpanları yöntemi ve çizgi arama (line search) yöntemi kullanılmıştır. Yapılan simulasyonlar temsiliyet kalitesinin artıtrıla bilğini göstermiştir.

Anahtar Kelimeler:: Seyrek temsiliyet, sözlük öğrenme, çoklu sözlükler, artık bileşenler, yapısal sözlükler.

ACKNOWLEDGMENT

First of all, I would like to express my deepest gratitude to my supervisor, Prof. Dr. Hüseyin Özkaramanlı for his invaluable guidance, patience, support and dedication provided all-over this work. During all the stagers of this work, and still afterwards, Prof. Özkaramanlı has been a source of inspiration and a unique role model for me.

My sincere thanks are conveyed to the administrative body and the faculty members of the Electronic Engineering Department in EMU for providing me the great opportunity of being a research assistant during my studies. I deeply thank Prof. Dr. Aykut Hocanın and Prof. Dr. Hasan Demirel for their invaluable support and motivation. I acknowledge the commitment and devotion of my fellow colleague Miss Faezeh Yeganli shown throughout this work. My warmest regards go to all of my instructors throughout my undergraduate studies in Birzeit University in Palestine and my post-graduate studies at the Eastern Mediterranean University. I am always beholden to the enlightenment, inspiration, knowledge and scientific mentality they all gave me, limitlessly.

I also wish to thank all my friends and colleagues who made my stay at North Cyprus a joyful and wonderful stage of my life. My warm thanks are due to my parents, to Ammar, Abeer, Lana, Ibrahim and Rasha for their support and encouragement in every aspect, throughout my life. Everyday, I owe them more and more.

DEDICATION

Dedicated to Palestine

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

$\|\ \|_0$	Number of non-zero elements in a vector
$\|\ \|_2$	Euclidean vector norm
$\|\ \|_F$	Frobenius matrix norm
$\mathbf{w}$	Sparse approximation coefficient vector
$\mathbf{W}$	Matrix of sparse coding coefficient vectors
$\mathbf{X}$	Training set of vector signals
$\mathbf{r}$	Sparse approximation residual vector
Λ	Set of selected dictionary atoms for sparse coding
S	Sparsity
λ	Sparsity-representation fidelity balancing parameter
ϵ	Vector sparse approximation error tolerance
$\mathbf{I}$	The identity matrix
T	The transpose operator
tr	The trace operator
$\mathbf{p}$	Projection operator
$\mathbf{U}$	Signal space
K	Number of dictionary atoms
M	Number of structured dictionaries
n	Dimension of the signal space
$\mathbf{D}$	Dictionary
DL	Dictionary learning

MOD	Method of optimized directions
SR	Super-resolution
SISR	Single-image super-resolution
LR	Low resolution
HR	High resolution
LF	Low frequency
HF	High frequency
MSE	Mean-squared error
DWT	Discrete wavelet transform
IDWT	Inverse discrete wavelet transform
PSNR	Peak signal-to-noise ratio
SSIM	Structural similarity index
BP	Basis pursuit
MP	Matching pursuit
OMP	Orthogonal matching pursuit
MAP	Maximum aposteriori
LASSO	Least absolute shrinkage and selection operator
LARS	Least angle regression
BPD	Basis pursuit denoising

Chapter 1

INTRODUCTION

1.1 Introduction

Digital signal processing is based on sampling continuous time or space signals with respect to time, space or frequency and then quantizing the acquired samples. The ability of such samples in extracting the intrinsic meaningful parts of a signal is of crucial importance. Therefore, signal representation (modeling) is an important area in the signal processing field. It has been widely acknowledged that better signal modeling is a reason for improving the performance in any of the signal processing application areas [1]. Intuitively, a model has to be carefully selected to be compatible with the problem in hand. One of the most successful and widely used signal representation techniques is sparse representation. This thesis addresses some of the open-ended problems concerning this representation and attempts at providing suitable solutions. The two purposes throughout this work can be summarized as enhancing the quality of the representation and lowering the levels of computational and storage costs, within the sparse signal model.

1.2 Background and Motivation

Sparse representation has been employed in various signal and image processing applications, leading to state-of-the-art performances. Examples include, and not limited to, classical audio, image and video processing applications such as denoising, deblurring, inpainting, compression, and super-resolution, as well as, speech and object recognition (source separation and classification), multimedia data mining, bioinformatic data decoding, applications also range from correcting error for corrupted data (face recognition despite occlusion) to detecting activities and events through a large network of sensors and computers [2].

The key property of sparse representation is its ability in capturing intrinsic signal features (information content) [3]. Such features depend on the problem of interest. It can be salient object features for recognition problems. In compression, such intrinsic features should be the most informative signal portions, allowing for an economical yet a meaningful description of data. In denoising, such features should be the attributes of the true signal buried in noise. In super-resolution, these features would be an invariant quantity from a low-resolution image that can be used to infer relative information about the unknown high-resolution image. Overall, sparse representation is shown able to capture the intended intrinsic information, regardless of the problem in hand [4].

As the name suggests, a sparse representation of a signal is the one that uses a few basis vectors to approximate the signal. Basis vectors are typically arranged as the columns of a dictionary matrix $\mathbf{D}$. A signal $\mathbf{x}$ can be sparsely represented as a linear combination of a few elements of a dictionary $\mathbf{D}$. This representation can be cast as $\mathbf{x} = \mathbf{D}\mathbf{w}$, where $\mathbf{w}$ is the sparse coding coefficient vector. Given $\mathbf{x}$ and $\mathbf{D}$, the determination of $\mathbf{w}$ is referred to as the problem of sparse coding. This problem aims at finding a loyal representation of $\mathbf{x}$ in terms of a few atoms in $\mathbf{D}$. Typically, the sparse coding problem can be viewed as an error minimization problem, where sparsity is fixed. Alternatively, it can be viewed as a sparsity minimization problem, with a specific representation error tolerance. The determination of $\mathbf{D}$ is of crucial importance to the sparse presentation model. In fact, there are two basic categories of dictionaries, these are:

I Mathematically-defined dictionaries: these are in fact pre-defined basis functions that give the support to a given signal. Examples include Fourier basis functions, wavelets, contourlets and several others. The main advantage of such dictionaries is the fact that sparse coding over them is carried out in two easy and fast steps. The first step is an inner product operation between the signal and the basis function to determine the representation coefficients. The second step is to threshold these coefficients leaving only a few non-zero ones, making the representation sparse. However, this sparsity enforcement paradigm is shown not to fit a wide set of signals. In other words, such a representation does not have the ability to fit a specific class of signals [5, 6].

II Learned dictionaries: a learned dictionary is a matrix whose columns are inferred from example signals. This matrix is initialized with a set of randomly selected signals, and then undergoes a training process over a set of training signals. During the training process, dictionary atoms are updated in such a way that serves for two purposes. The first one is to loyally represent the training data, and the second one is to keep the representation of the data sparse. Accordingly, the dictionary learning problem is a two-fold optimization problem aiming at these two purposes. Several dictionary learning algorithms exist in the literature that can give good solution to this problem.

The great advantage of a learned dictionary is its signal fitting capability. This is due to the fact that the atoms of a learned dictionary are prototype signal structures conveyed from natural signal examples. It has been shown that it is better to learn over-complete (redundant) dictionaries. Redundancy further improves the representation quality of a learned dictionary. Intuitively, a redundant dictionary contains more prototype signal structures, and is thus better able to approximate more signals.

Sparse coding over a learned dictionary is no more an inner product process. Atom selection is in fact a vector selection process. This process is shown to be non-deterministic polynomial-time (NP)-hard, i.e., computationally expensive. However, several vector selection algorithms exist in the literature and can give a good approximate solution to the sparse coding problem.

Despite the added benefit of redundancy, it significantly increases the computational complexity of sparse coding. Furthermore, high levels of redundancy tend to cause instabilities and degradation in the sparse coding solution. These concerns make an upper bound of feasible levels of redundancy.

Based on the above discussion, the objective of sparse coding is thus to learn a dictionary with an acceptable redundancy that can represent the signal space loyally and with a specific degree of sparsity. Given the fact that signal's variability in a class is less than the general signal variability, recent research has been directed towards dividing or classifying the signal space into a set of classes and learning a dictionary for each class. This leads to a set of class dictionaries. To perform sparse coding of a signal, the same classification criterion is applied in order to select the class this signal belongs to, and to eventually perform its sparse coding over the class dictionary. Many works came along this line of designing multiple dictionaries. The essential difference between them is the way they define classes, or more precisely, the classification criterion applied to the problem.

The work conducted in this thesis comes as an attempt to remedy, or partially solve the following open-ended problems in sparse representation over learned dictionaries

1. The need for an extensive training set for the training of a good representative dictionary.
2. Dictionary learning is a large-scale and highly non-convex problem. It requires high computational complexity, and its mathematical behavior is not yet well understood [7].
3. Non-linear sparse inverse problem estimation may be unstable and imprecise due to the coherence of the dictionary. [4]
4. Determining the optimal image patch size that governs the dictionary atom size. [8]

5. The need for a systematic approach for the design of directionally-structured dictionaries.

6. The need for an effective sparse coding paradigm to make the best use of multiple designed dictionaries [9, 10, 11, 12, 13, 14].

1.3 Thesis Contributions

In this thesis we have investigated the problem of learning multiple structured dictionaries. The first topic is learning wavelet-domain dictionaries with the corresponding wavelet-domain based sparse coding. In this work, we have shown that this idea naturally merges the desirable features of wavelet transform such as compactness, directionality and analysis in many levels, with the representative power of learned dictionaries. We have shown that this idea improves the quality of sparse representation at reduced computation complexity. This idea is studied over the problem of single-image super-resolution. The next contribution is to design projection operators for the purpose of dividing the signal space into more localized directional subspaces. The same operators are used to decompose an test signal into directional components. In this setting, a signal's sparse representation is the summation of the sparse codings of its subsapce components, each coded over its respective subsapce dictionary. We have shown that this dictionary learning and sparse representation paradigm enhances the representation quality at moderate levels of computational complexity. The third contribution is devising a strategy for adaptively selecting the patch size when performing sparse coding. This strategy is shown to improve the quality of sparse representation with a moderate increase in computational complexity. The next contribution is concerned with the learning of multiple structured dictionaries on a residual component level. In this work, we show that a signal and its residual components subject to a sparse coding technique are not necessarily consistent with the same model. Therefore, it is advantageous to perform the dictionary learning process on a residual component base. In this setting, each residual component contributes exclusively to the training of the dictionary most fitting its structure. In other words, this contribution forms a mechanism whereby the intended structure of the dictionaries is enhanced using residual components. Another contribution is a strategy for constrained residual-based dictionary re-training. In this setting, a first pass dictionary learning process is excused. Then, the residual components of the training set used in the first pass with respect to the obtained dictionary are calculated. These are then used to update that dictionary in such a way that the representation fidelity of the original training set is imposed. The work presented in this thesis formulates this learning paradigm as a constrained error minimization problem, which can be easily solved with Lagrange multipliers.

1.4 Thesis Outline

This thesis is organized as follows: Chapter presents a basic introduction to dictionary learning and sparse coding, approaching the main contributions presented in this work. A concise literature review of dictionary learning, sparse coding, the existing methods of designing multiple dictionaries and their sparse coding along with the relevant classical image processing applications is outlined in Chapter 1.4. Chapter 2.6.2 introduces an approach for designing wavelet-domain dictionaries, along with experimental results testing its performance. Next, an approach for directionally structural dictionary learning and sparse coding based on subspace projections is detailed in Chapter 3.3.2, along with experimental validations. Chapter 5 presents an approach for sparse representation with a variable patch size, with the accompanying experiments. A strategy for residual component-based multiple structured dictionary learning is then outlined Chapter 5.3.2, with experiential tests. Chapter 6.3.2 presents a strategy for re-training a dictionary over residual components of a training set, along with numerical experiments. In Chapter 7.3.2, the conclusions of this work are made, and possible future works and extensions are stated.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

Sparse representation requires the availability of a dictionary and a vector selection technique to handle the representation process. In this chapter, a concise revision of sparse representation over learned dictionaries is presented. This is done by first presenting the problem statement of sparse coding. Then, the dictionary learning problem is formulated while revising some benchmark approaches to this problem. Afterwards, the shortcomings of the single dictionary usage are highlighted. Then, several approaches to sparse representation over multiple dictionaries are reviewed. Finally, image super-resolution and denoising are presented as two classical application areas of sparse representation.

2.2 Sparse Coding

Many signal and image processing operations are inherently ill-posed inverse problems. Regardless of the application encountered, the problem in hand can customarily be viewed as solving for the following (typically under-determined) system of equations.

$$\mathbf{D}\mathbf{w} = \mathbf{x}, \tag{2.1}$$

where one needs to effectively find a solution vector $\mathbf{w} \in \mathbf{R}^n$, given the matrix $\mathbf{D} \in \mathbf{R}^{n \times K}$ and the observation vector $\mathbf{x} \in \mathbf{R}^n$. Amongst the infinitely many possible vectors in the solution space, there exists a certain solution set which is the sparse solution set. The importance of the sparse solution has been well-established for many reasons depending on the application. In case of sampling where **D** is a sensing matrix, for example, the sparsest solution is the one that better describes the intrinsic information content in a signal. For the case of (image) compression, where **D** is dictionary representing the code book, the sparsest solution means a higher compression efficiency. For the case of image denoising, where **D** is a dictionary learned over image patches, such a sparse solution is more likely to be more noise-free as compared to the others.

Having a set of basis signals collected column-wise as a matrix **D**, representing a signal **x** as a linear combination of a few columns in **D** is referred to as sparse coding (representation). It is customary to learn **D** over a certain training set in such a way that $K > n$, i.e., **D** is said to be a redundant (an over-complete) basis. The sparse representation problem thus can be posed as

$$\underset{\mathbf{w}}{\operatorname{argmin}} \, ||\mathbf{w}||_0 \ \textit{subject to} \ \mathbf{D}\mathbf{w} = \mathbf{x}, \tag{2.2}$$

where $|| \ \ ||_0$ denotes the the number of non-zero elements in a vector.

If one allows a certain level of representation error tolerance ϵ, the sparse representation problem is said to be a sparse approximation problem. This can be expressed as solving for **w** in the following approximation.

$$\mathbf{x} \approx \mathbf{D}\mathbf{w}, \tag{2.3}$$

This problem can be formulated as the following optimization problem.

$$\underset{\mathbf{w}}{\operatorname{argmin}} \, ||\mathbf{w}||_0 \; subject \; to \; ||\mathbf{x} - \mathbf{D}\mathbf{w}||_2 \leq \epsilon, \tag{2.4}$$

where $|| \quad ||_2$ denotes the Euclidean vector norm. Clearly, the above formulation requires two objectives to be met. These are the representation sparsity and the representation fidelity. In the above formulation, the sparse approximation or sparse coding problem is posed as an error-constrained sparsity minimization problem. There is still another dual formulation that meets the aforementioned two requirements, but in the reverse direction. In this formulation, sparse coding is viewed as a sparsity-constrained error minimization problem, which can be posed as follows.

$$\underset{\mathbf{w}}{\operatorname{argmin}} \, ||\mathbf{x} - \mathbf{D}\mathbf{w}||_2 \; subject \; to \; ||\mathbf{w}||_0 < S, \tag{2.5}$$

where S denotes sparsity.

2.3 Sparse Approximation Approaches

The formulation in (2.5) is more commonly used for the purposes of representation. It is well-known that the calculation of the l_0 norm in (2.2) makes this formulation an NP-hard problem and therefore excessively computationally demanding. Therefore, research conducted so far in the field of sparse representation aims at merely approximating the sparse solutions with tractable complexity. There are basically two approaches to arrive at a suboptimal solution. These can be categorized into two main categories. The first category is greedy algorithms that approximate the l_0 norm solution. This is basically known as the matching pursuits (MP) methods. The second category is the convex-relaxation algorithms known as basis pursuit (BP) methods. These are based on replacing the l_0 minimization with l_1 minimization, giving effective solutions while significantly reducing the computational complexity of the problem.

2.3.1 Greedy Algorithms

This family of sparse coding algorithms try to effectively minimize the l_0 norm in an iterative manner. In each iteration, a certain signal portion is represented by picking a specific atom in $\mathbf{D}$. The process continues until a stopping criterion is met. The essence of these methods comes from the MP algorithm proposed by Mallat and Zhang [15]. Many other variants and extensions have also been proposed. One of the most successful extensions is the orthogonal matching pursuit (OMP) [16]. Herein, the MP and OMP methods are viewed.

2.3.1.1 Matching Pursuit. MP is the first attempt at solving the sparse coding problem in a greedy manner. It is a simple and effective approach to sparse coding. This method defines a residual vector $\mathbf{r}$ containing portions of $\mathbf{x}$ which have not yet been represented. This residual is initialized by the signal itself, and is iteratively represented in terms of atoms selected from $\mathbf{D}$, denoted by the set $\mathbf{d}_k$, at the k-th iteration. As MP iterates, $\mathbf{r}$ is minimized and updated after each iteration. The rationale of MP can be outlined as the following steps.

I. Initialize the coefficient vector as the zero vector $\mathbf{w} = \mathbf{0}$ and set the residual as $\mathbf{r} = \mathbf{x}$.

II. Compute the inner products between the residual and the atoms in the dictionary $c_k =< \mathbf{r}_r, \mathbf{d}_k >$.

III. Select the atom of the largest absolute inner product $k^* = \underset{1 \leq k \leq K}{\operatorname{argmax}} |c_k|$.

IV. Update the residual by subtracting the contribution of the optimal atom $\mathbf{r} \leftarrow \mathbf{r} - c_{k^*}\mathbf{d}_{k^*}$.

V. Repeat steps II to IV until a stopping criterion is met.

The main steps of MP are outlined in Algorithm 1.

Algorithm 1 Matching Pursuit (MP)

```
INPUT: $\mathbf{x}, \mathbf{D}, S$ or $\epsilon$.
OUTPUT: $\mathbf{w}$.
Initialization: $\mathbf{r} \leftarrow \mathbf{x}$, $i \leftarrow 1$.
while $i \leq S$ or $\|r\|_2 \leq \epsilon$ do
    $c = \mathbf{D}^T\mathbf{r}$                                              } Atom
    $k^* = \underset{1 \leq k \leq K}{\operatorname{argmax}} |c_k|$          } Selection
    $\mathbf{r} \leftarrow \mathbf{r} - c_{k^*}\mathbf{d}_{k^*}$             } Residual Update
    $i \leftarrow i + 1$
end while
```

2.3.1.2 Orthogonal Matching Pursuit. OMP [16] has been proposed as a better extension to the MP algorithm. In OMP, the same atom selection criterion is adopted. However, it differs in the way the residual is calculated. This is done by projecting the current residual on the complement of the subspace spanned by the atoms selected up to that point.

OMP initializes the residual $\mathbf{r}$ with $\mathbf{x}$, and the set of selected atoms Λ as the empty set ϕ. The first step is to calculate the inner product between the current residual and the complement of the selected atoms denoted by Λ^c. Initially, Λ^c is the whole dictionary $\mathbf{D}$. In each iteration, the atom in $\mathbf{D}$ that has the maximal inner product is selected to be in the set Λ. After this selection, $\mathbf{r}$ is updated by calculating the vector of coefficients $\mathbf{x}^*_\Lambda$ derived from projecting the signal onto the subspace spanned by the selected atom(s). This is achieved by computing the Moore-Penrose pseudo-inverse of the sub-dictionary as ${\mathbf{D}_\Lambda}^\dagger = [\mathbf{D}^T_\Lambda \mathbf{D}_\Lambda]^{-1}{\mathbf{D}_\Lambda}^T$ that contains the selected (active) atoms, where $\dagger$ and T denote Moore-Penrose pseudo-inverse and the transpose operator, respectively . A summary of the main OMP steps is presented in Algorithm 2.

Algorithm 2 Orthogonal Matching Pursuit (OMP)

```
INPUT: $\mathbf{x}, \mathbf{D}, S$ or $\epsilon$.
OUTPUT: $\mathbf{w}$.
Initialization: $\mathbf{r} \leftarrow \mathbf{x}$, $i \leftarrow 1$ and $\Lambda \leftarrow \phi$
while $i \leq S$ or $\|\mathbf{r}\|_2 \leq \epsilon$ do
    $c = \mathbf{D}_{\Lambda^c}\mathbf{r}$                        }
    $k^* = \underset{1 \leq k \leq K}{\text{argmax}} |c_k|$       }  Atom Selection
    $\Lambda \leftarrow \Lambda \cup k^*$                          }
    $\mathbf{w}_{\Lambda^*} \leftarrow {\mathbf{D}_\Lambda}^\dagger \mathbf{x}$
    $\mathbf{r} \leftarrow \mathbf{x} - \mathbf{D}_\Lambda \mathbf{w}_{\Lambda^*}$        }  ResidualUpdate
    $i \leftarrow i + 1$
end while
```

The principal advantage of OMP over MP is the fact that the inner product is calculated between the residual and the atoms that are not yet in the selected set Λ. This is because the current residual is naturally orthogonal to the subspace spanned by the atoms in Λ. So, the inner product of the residual and any atom in Λ is zero ($< \mathbf{r}_k, \mathbf{D}_k >= 0, \forall k$), and there is no need to perform such a calculation. Technically, this difference means that an atom can not be selected more than once. In view of these ideas, it has been shown that OMP converges to a zero residual (error) within n iterations, where n is the dimension of the signal space. The orthogonality character of the residual in OMP requires solving the following least-squares problem $\mathbf{w}_\Lambda^* = \underset{\mathbf{w}_\Lambda}{\operatorname{argmin}} ||\mathbf{x} - \mathbf{D}\mathbf{w}_\Lambda||_2$ at each residual update. This means a pseudo-inverse calculation with each iteration. This forms a corresponding computational complexity overhead of OMP as compared to MP, despite the improved approximation quality of OMP over MP.

Several other greedy approaches to sparse coding are proposed in the literature as extensions which are based on MP and OMP. Examples include regularized orthogonal matching pursuit (ROMP) [17], the compressive sampling matching pursuit (CoSaMP) [18] and the subspace pursuit (SP) [19]. These methods essentially aim at offering convergence grantees that come in the context of the restricted isometry concept in compressive sampling. Interested reader is referred to [20] for an elaborate review of sparse coding methods.

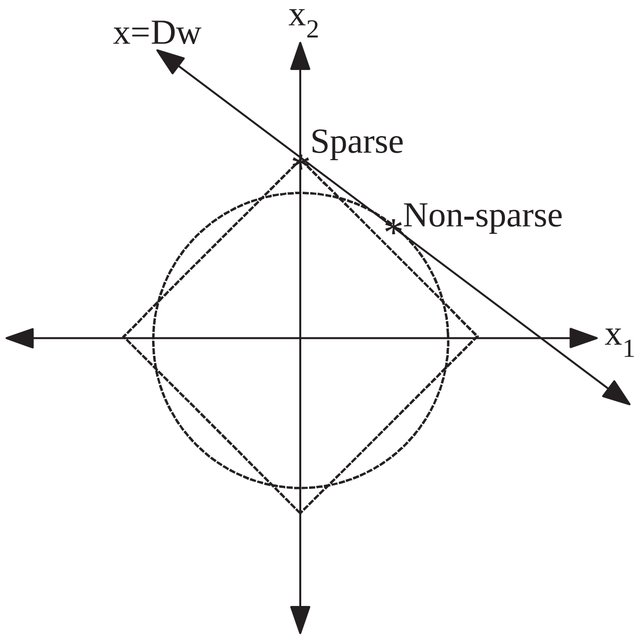


Figure 2.1. Coefficient space and solutions of an under-determined system of equations with K=2 and n=1.

2.3.2 Convex Relaxation Algorithms

The optimization in (2.2) is not convex in the l_0 norm. However, the l_0 norm minimization can be relaxed by replacing this norm with the l_1 norm. In fact, the l_1 norm is shown to be the closest convex surrogate function to the original objective of the l_0 minimization. This convex relaxation can be cast as follows.

$$\underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|_1 \; subject\ to \; \mathbf{x} = \mathbf{D}\mathbf{w}. \tag{2.6}$$

The principal idea of convex relaxation came from observing the geometry of the solution space to an under-determined system of equations which is composed of infinitely many vectors. A toy example is shown in Fig 2.1 with n=1 and K=2. Let us consider this example based on which the idea can be generalized. Linking this example to the optimizations in (2.2) and (2.6), a solution vector $\mathbf{w} \in \mathbf{R}^2$ is required to the problem of having a single equation with two unknowns such that only one variable is non-zero. In view of Fig. 2.1, the sparsest solution can be given as the one whose x_1 or x_2 component is zero. This solution has the minimal possible l_0 norm. Let us consider the cases of having solutions with minimized l_1 and l_2 norms, and examine whether such solutions are potentially consistent with the l_0 solution. In Fig. 2.1, the circle shape represents contour lines with a fixed l_2 norm, whereas the diamond shape corresponds to contour lines with a fixed l_1 norm. Besides, the diagonal line represents the l_0 solution to the problem $\mathbf{Dw} = \mathbf{x}$. Clearly, an l_0 minimized solution is the one that intersects with the axes x_1 or x_2. It is geometrically evident that an l_1 norm minimized solution can still be l_0 minimized, as the diamond shape intersects with the solution line and the axes x_1 and x_2. However, the l_2 solution can not intersect with the solution line and any of the axes at the same point. Therefore, the l_2 minimization violates sparsity in the l_0 sense. In other words, minimizing the l_1 norm is potentially consistent with minimizing the l_0 norm, which is not the case for l_2 minimization.

2.3.2.1 Basis Pursuit. In line with the above discussion, Chen *et al.* proposed the BP algorithm [21] as the convex relaxation formulated in (2.6). In this approach, the deployment of l_1 eases the problem of sparse coding in such a way that it can be solved using linear programming optimization. This is achieved by defining an augmented dictionary $\bar{\mathbf{D}} = [\mathbf{D}, -\mathbf{D}]$ which includes negative copies of the atoms in its columns.

Then, sparse coding boils down to the following optimization problem.

$$\bar{\mathbf{w}}* = \underset{\bar{\mathbf{w}} \in \mathbf{R}^K}{\operatorname{argmin}} 1^T \bar{\mathbf{w}} \ subject\ to\ \mathbf{x} = \bar{\mathbf{D}}\bar{\mathbf{w}},\ \ \mathbf{w} \succeq 0, \tag{2.7}$$

where $\bar{\mathbf{w}} \in \mathbf{R}^{2K}$ is an augmented coefficient vector whose elements are constrained to be greater than zero (here we used the notation $\succeq$ to indicate element-wise inequality) and 1 indicates a vector of ones and is introduced to express the l_1 norm as an inner product $< 1, \bar{\mathbf{x}} >= \|\bar{\mathbf{x}}\|_1$. This way, re-formulating the problem makes it possible to use a standard convex optimization method [22] resulting in an optimal solution to the new problem $\bar{\mathbf{x}}^*$. This solution can be directly translated to the solution of the original problem in (2.6) ($\mathbf{x}^*$). This is done easily by splitting $\bar{\mathbf{x}}^*$ in two consecutive vectors of length K as $\bar{\mathbf{x}}^* = [\mathbf{v}^*, \mathbf{u}^*]$ and subtracting the second vector from the first one $\mathbf{x}^* = \mathbf{v}^* - \mathbf{u}^*$.

The problem formulated in (2.6) can be extended to include the case where an exact solution does not exist, and an approximate solution is required. This is known as the basis pursuit denoising problem (BPD), i.e., there is no more any zero representation error constraint. The unconstrained problem can be posed as

$$\mathbf{w}^*_\lambda = \underset{\mathbf{w} \in \mathbf{R}^K}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\mathbf{w}\|_2 + \lambda\|\mathbf{x}\|_1,\ \ \mathbf{w} \succeq 0, \tag{2.8}$$

in this setting, the parameter λ is introduced to balance the trade-off between the sparsity of the solution in the l_1 sense, and the representation fidelity. It is worth mentioning that λ is set to be proportional to variance of the noise in case of additive Gaussian noise. This means that (2.8) is consistent with (2.6) for the noiseless case.

Similar to the case of (2.6), the optimization in (2.8) is a quadratic program that can be solved with any standard convex optimization algorithm [22] and has a convenient Bayesian interpretation as the maximum a posteriori (MAP) estimate of the signal under the assumptions that the noise follows a Gaussian distribution and that the coefficients follow a Laplacian distribution.

2.3.2.2 Least Absolute Shrinkage and Selection Operator. In [23] Tibshirani proposed the least absolute shrinkage and selection operator (LASSO) algorithm attempting at solving the l_1 relaxation of the sparse coding problem. This algorithm was extended by Osborne *et al.* in [24]. In the context of LASSO, the sparse coding problem formulation becomes as follows.

$$\mathbf{w}_p^* = \underset{\mathbf{w}\in\mathbf{R}^K}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{D}\mathbf{w}\|_2^2 \; \textit{subject to} \; \|\mathbf{w}\|_1 \leq p, \tag{2.9}$$

where the parameter p controls the sparsity of the solution.

The homotopy algorithm [24] introduced to solve (2.9) is an iterative method that starts form the solution $\mathbf{w}_0^* = 0$ and traces a solution path which follows increasing values of the parameter p until the desired constraint is reached. Enlarging the feasible set by increasing the value of p causes new atoms to enter the active set Λ and may result in other atoms to exit it. The least angle regression (LARS) algorithm proposed by Efron *et al.* [25] is a simplification of the homotopy idea where atoms are only allowed to enter the active set every time the solution gets updated.

2.4 Dictionary Learning

The sparse signal model is based on the availability of a dictionary $\mathbf{D}$ whose columns can give an approximation to a given signal $\mathbf{x}$. A dictionary can be obtained in two ways. The first way defines a dictionary as an analytic function. Examples of this sort include the standard algebraic basis functions such the Fourier basis functions, wavelets, contourlets, bandlets, etc. Sparse coding over such dictionaries is quite easy. This requires performing a simple inner product operation between $\mathbf{x}$ and $\mathbf{D}$. Then, sparsity can be imposed by thresholding the representation coefficients. However, such a sparse enforcement paradigm is shown not to fit a large set of signals [6]. In other words, pre-defined dictionaries are not adaptive to the signals to be sparsely represented. The second alternative is to learn dictionary over a set of training signals. The work by Mallat and Zhang [15] is the first to suggest performing a signal expansion based on a small subset of basis functions selected from a general dictionary of basis functions. Then, Chen *et al.* proposed the idea of basis pursuit (BP) [21] for successive sparse coding over a given dictionary. It is pointed out that these two pioneering works have together set the motivation for the recent shift of signal representation from transforms to dictionaries [26]. Learned dictionaries are shown to be more adaptive to local signal structures. Therefore, they have a better signal-fitting capability [7]. Nevertheless, sparse coding over a learned dictionary is no more carried out as an inner product process. Rather, a vector selection algorithm is necessary for this purpose, which is more computationally expensive than inner product and thresholding. It is worth mentioning that a dictionary and a frame are often regarded as the same thing, but the (tiny) difference is that a frame spans the signal space while a dictionary does not have to do so.

Dictionary learning (DL) is the process of learning or training a dictionary **D** over a certain training set **X**. This set is often composed of example signals. The common practice in the context of image processing is to obtain a training set as a column-stacking of patches extracted from natural images and reshaped into the vector form. **D** is first initialized with randomly selected training vectors. Then, it undergoes a DL process that aims at two purposes; first giving a loyal representation to the vectors in **X**, and second is keeping the representation sparse. Given $\mathbf{X} \in \mathbf{R}^{n\times m}$ as a training set composed of m training vectors $\mathbf{x} \in \mathbf{R}^{n}$, the DL process can be formally posed as finding a solution pair $\mathbf{D} \in \mathbf{R}^{n\times K}$, $\mathbf{W} \in \mathbf{R}^{K\times m}$ for the following inverse problem.

$$\mathbf{X} \approx \mathbf{DW}, \tag{2.10}$$

where the matrix **W** has as its columns the sparse coding coefficient vectors of the training data, thus, it is sparse column-wise. In other words, each column in **W** is the sparse coding of the corresponding column in **X**, which needs to be sparse.

There is an ambiguity concerning the above DL model in the sense that if $(\mathbf{D}, \mathbf{W})$ is a solution pair, there exists another equivalent solution $(\mathbf{D}' = \mathbf{DA}, \mathbf{W}' = \mathbf{BW})$ where the matrices $\mathbf{A}$ and $\mathbf{B}$ are related as $\mathbf{AB} = \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix. It has been shown that imposing a unit-$l_2$norm constraint on the columns of $\mathbf{D}$ overcomes this ambiguity, as $\mathbf{D} = \mathbf{D} : \|\mathbf{d}_k\|_2 = 1, 1 \leq k \leq K$. Therefore, it is customary to learn dictionaries whose atoms have a unit 2-norm.

In view of the fact that the DL process is a two-fold optimization problem, it can have two possible formulations in analogy with the case of sparse coding. The first formulation considers the DL problem as an error-constrained sparsity minimization problem as follows.

$$\underset{\mathbf{D},\mathbf{W}}{\operatorname{argmin}} \|\mathbf{w}_i\|_0 \; subject \; to \; \|\mathbf{X}_i - \mathbf{D}\mathbf{W}_i\|_F^2 < \epsilon \; \forall \; 1 \leq i \leq m, \tag{2.11}$$

where ϵ again denotes the representation error tolerance, $\| \;\; \|_F$ is the Frobenius matrix norm and $\mathbf{W} = [\mathbf{w}_1\mathbf{w}_2 ... \mathbf{w}_m]$ is the matrix of sparse coding vectors such that $\mathbf{X} \approx \mathbf{DW}$. The other DL problem formulation views it as a sparsity-constrained representation error minimization problem, as follows.

$$\underset{\mathbf{D},\mathbf{W}}{\operatorname{argmin}} \|\mathbf{X}_i - \mathbf{D}\mathbf{W}_i\|_F^2 \; subject \; to \; \|\mathbf{W}_i\|_0 < S \; \forall \; 1 \leq i \leq m. \tag{2.12}$$

Analogous to the sparse coding problem, the DL process can be formulated as an unconstrained error minimization problem, as follows.

$$\underset{\mathbf{D},\mathbf{W}}{\operatorname{argmin}} \|\mathbf{X}_i - \mathbf{D}\mathbf{W}_i\|_F^2 + \lambda\|\mathbf{W}_i\|_0 \;\forall\; 1 \leq i \leq m, \tag{2.13}$$

where the parameter λ balances the trade-off between sparsity and representation fidelity.

In view of the NP-hard nature of sparse coding, the formulations in (2.11), (2.12) and (2.13) are all non-convex. This is because optimizing in $\mathbf{D}$ and $\mathbf{W}$ can not be convex at the same time, even if the l_0 norm minimization is relaxed to l_1 minimization as done with sparse coding. A common remedy to handle this obstacle is to tackle the optimizations in a block-coordinate descent fashion. This means performing the DL process as a successive alternation between a sparse coding stage and a dictionary update stage. This is advantageous in the sense that the DL optimization can be made convex in one of the two variables $\mathbf{D}$ and $\mathbf{W}$ while keeping the other one fixed. In summary, the DL process starts with an initial dictionary $\mathbf{D}^0$, and performs the following two steps successively at each iteration t, as follows:

I. **Sparse coding**: given a fixed dictionary $\mathbf{D}^t$, the matrix of spare representation coefficients $\mathbf{W}^t$ can be computed as a standard sparse approximation problem using any solver that is suitable to the particular formulation. For example, if dictionary learning is defined as a sparsity-constrained optimization, then any method that seeks a best S-term approximant to the training signals can be employed, such as OMP or LARS.

II. **Dictionary update**: given a fixed matrix of sparse approximation coefficients $\mathbf{W}^t$, the dictionary $\mathbf{D}^t$ is updated to $\mathbf{D}^{t+1}$ in order to improve the objective of the dictionary learning optimization, subject to optional constraints.

It is noted that the solution space to (2.11) does not necessarily contain dictionaries with unit-l_2 norm. Therefore, a normalization step is often carried out at the end of each DL iteration, as follows.

$$\mathbf{D}^{t+1} \leftarrow \mathbf{D}^{t+1}\mathbf{E}^{-1}, \tag{2.14a}$$

$$\mathbf{W}^{t} \leftarrow \mathbf{E}\mathbf{W}^{t}, \tag{2.14b}$$

Where $\mathbf{E}$ is a diagonal matrix whose elements $e_{k,k} = \|\mathbf{d}_k^t\|_2$ contain the norm of the dictionary. This way, every atom in the updated dictionary is normalized and the coefficients in the matrix $\mathbf{W}^t$ are updated such that the product $\mathbf{D}^{t+1}\mathbf{E}^{-1}\mathbf{E}\mathbf{W}^t = \mathbf{D}^{t+1}\mathbf{W}^t$ remains unchanged.

Noting that the DL process is a succession of a sparse coding stage and a dictionary update stage, many DL algorithms exist in the literature. It can be noted that the core difference between them lies in the way a dictionary is updated, while sparse coding is not much influential in the DL process. The reader is referred to the paper by Rubinstein *et al.* [26] for more details about other DL algorithms. In the next subsections, some well-known DL algorithms are briefly reviewed.

2.4.1 The MOD Algorithm

One of the well-known DL algorithms is the method of optimal directions (MOD), proposed by Engan *et al.* [27]. This algorithm is based on the DL formulation specified in (2.11). In this algorithm, the authors proposed using any sparse coding method (e.g. OMP) for the first stage. Then, they performed the dictionary update by calculating the pseudo-inverse of the DL equation using the calculated sparse representation coefficient matrix. This gives a locally optimal solution to the problem $\underset{\mathbf{D},\mathbf{W}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{DW}\|_F^2$. This algorithm is summarized in Algorithm 3.

Algorithm 3 MOD Dictionary Learning

```
INPUT: a training set $\mathbf{X}$, and initial dictionary $\mathbf{D}^0$, S, number of iterations $Num$.
OUTPUT: $\mathbf{D}$, $\mathbf{W}$.
Initialization: $\mathbf{D} \leftarrow \mathbf{D}^0$ and $i \leftarrow 1$
while $i \leq Num$ do
    for $j = 1$ to $m$ do
        set $\mathbf{W}_j \leftarrow \underset{\mathbf{W}_j}{\operatorname{argmin}} \|\mathbf{W}_j\|_0 \; subject\ to\ \mathbf{DW}_j = \mathbf{X}_j$ } Sparse Coding
    end for
    $\mathbf{D} \leftarrow \mathbf{XW}^\dagger$ } Dictionary Update
    $\mathbf{D} \leftarrow \mathbf{DE}$ } Dictionary Normalization
    $i \leftarrow i + 1$
end while
```

2.4.2 The K-SVD Algorithm

Aharon *et al.* proposed the K-SVD algorithm in [28]. It adopts the same DL formulation the MOD uses. However, K-SVD differs in the way the dictionary update stage is performed. The objective function of (2.11) is $C(\mathbf{D}, \mathbf{W}) = \|\mathbf{X} - \mathbf{DW}\|_F^2$. The approximant term can be re-written as a sum of rank-1 matrices, as $C(\mathbf{D}, \mathbf{W}) = \|\mathbf{X} - \sum_{k=1}^{K} \mathbf{D}_k \mathbf{W}_k\|_F^2 = \|\mathbf{X} - \sum_{k' \neq k} \mathbf{D}'_k \mathbf{W}'_k - \mathbf{D}_k \mathbf{W}_k\|_F^2$.

Let us define a partial residual matrix, $\mathbf{E}_k$, as $\mathbf{E}_k = \mathbf{X} - \sum_{k' \neq k} \mathbf{D}'_k \mathbf{W}'_k$, then the atom $\mathbf{D}_k$ and the corresponding row of sparse approximation coefficients $\mathbf{W}_k$ can be jointly optimized to locally minimize the cost function C by calculating the best rank-1 approximation to $\mathbf{E}_k$. Therefore, K-SVD updates $\mathbf{D}$ one atom at a time. This can be summarized as follows.

I. For each dictionary atom $\mathbf{D}_k$, determine the set Λ_k of nonzero elements of the k-th row of $\mathbf{W}$. (that is, the set of training data which use the k-th atom in their approximation).

II. A partial residual matrix is calculated and its columns are restricted to the active set of signals that use the k-th atom for their sparse approximation.

III. The atom $\mathbf{D}_k$ and the coefficients $\mathbf{W}^k_{\Lambda_k}$ are updated using the solution of the best rank-1 approximation of the matrix $\mathbf{E}_k$, which can be calculated using its SVD.

Since the support of the sparse approximation coefficients should not be modified during the dictionary update step, $\mathbf{E}_k$ and its rank-1 approximation are restricted to the columns corresponding to the signals that use the k-th atom in their sparse approximation, that is, the indices corresponding to non-zero elements of the vector $\mathbf{W}_k$. A summary of the K-SVD algorithm is presented in Algorithm 4.

2.4.3 Online Dictionary Learning

Instead of learning a dictionary over a set of training vectors, it is possible to continuously tune it with every upcoming signal in an online manner. An approach to online dictionary learning is proposed by Mairal *et al.* [5, 6] in their online dictionary learn-

Algorithm 4 K-SVD Dictionary Learning

```
INPUT: $\mathbf{X}, \mathbf{D}^0, S, Num.$
OUTPUT: $\mathbf{D}, \mathbf{W}$.
Initialization: $\mathbf{D} \leftarrow \mathbf{D}^0$ and $i \leftarrow 1$.
while $i \leq Num$ do
    for $m = 1$ to $m$ do                                                    } Sparse
        set $\mathbf{W}_m \leftarrow \underset{\mathbf{W}_m}{\operatorname{argmin}} \|\mathbf{W}_m\|_0 \ subject\ to\ \mathbf{DW}_m = \mathbf{X}_m$   } Coding
    end for
    for $k = 1$ to $K$ do
        set $\Lambda_k \leftarrow i \subseteq 1, 2, m \ subject\ to\ \mathbf{W}_{k,i} \neq 0$        }
        set $\mathbf{E}_k \leftarrow [\mathbf{X} - \sum_{k' \neq k} \mathbf{D}'_k \mathbf{W}'_k]_{\Lambda_k}$   }
        $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] \leftarrow SVD(\mathbf{E}_k)$                    } Dictionary Update
        $\mathbf{D}_k \leftarrow \mathbf{u}_1$                                           }
        $\mathbf{W}_{\Lambda_k} \leftarrow \sigma_{1,1} \mathbf{v}_1^T$                                }
    end for
    $\mathbf{D} \leftarrow \mathbf{DE}$ } Dictionary Normalization
    $i \leftarrow i + 1$
end while
```

ing algorithm (ODL). In their approach, Mairal *et al.* use the DL formulation in (2.11) while relaxing the l_0 norm with the l_1 norm. However, they attempt to solve the optimization for each incoming training signal in an online manner. This can be viewed as solving the following optimization problem.

$$(\mathbf{D}^*, \mathbf{W}^*) = \underset{\mathbf{D}, \mathbf{W} \in \mathbf{R}^{n \times K}}{\operatorname{argmin}} \|\mathbf{X}^{T_0} - \mathbf{DW}^{T_0}\|_F + \lambda \|\mathbf{W}^{T_0}\|_1 \ forall\ 1 \leq i \leq m, \tag{2.15}$$

where the super-script T_0 indicates that $\mathbf{X}$ and $\mathbf{W}$ contain online data acquired at discrete times $t = 1, ..., T_0$. Sparse coding is performed using any vector selection technique that uses l_1 minimization such as the LARS algorithm, and the dictionary update stage is carried out atom-wise. It has been shown that handling the training atoms in batches of certain training vectors contributes to a better representation quality of the learned dictionary.

In [29], Skretting and Engan proposed an online extension to the MOD algorithm. In this setting, all the dictionary atoms are updated with each training signal. Their approach uses recursive least-squares to solve for the dictionary update equation. This is done by using a forgetting factor allowing for increasing the convergence speed without sacrificing the optimization optimality.

2.5 Sparse Representation over Multiple Learned Dictionaries

It is well-known that the success of the sparse models depends on how closely the columns (atoms) of **D** can approximate a given signal [6]. A major challenge to the DL process is the need for an extensive training signal set. This is because signals have high dimensionality. They can thus possess many structural features such as directional edges, textures, etc. Naturally, some image features are more common, while others are less. DL algorithms such as the K-SVD [28] and the ODL [5, 6] favor the more common features to be fit by the learned dictionary. Traditional DL and sparse coding algorithms do not adequately address the problem of learning dictionaries which possess a certain geometric structure [30].

In view of the above observations, researchers tended to consider splitting the signal space into several classes, and learning a compact dictionary for each class. Since the variability of class signals is less than the general signal variability, a class-dependent dictionary requires a lower degree of redundancy to keep the representation quality in acceptable levels. Several works exist in the literature aiming at this purpose. These differ in the criteria applied to define the signal classes. This definition is important for separating the training data into different classes or clusters. Besides, it determines on which dictionary to select for the sparse coding of a given signal. This process is referred to as model selection. The idea behind this approach is that a class dictionary needs not to be highly redundant. This allows for designing compact class dictionaries. This means a good representation quality at minimized redundancy and computational complexity levels.

Examples of the above research trend include clustering the training and testing data as proposed by Dong *et al.* [31], where they applied K-means clustering for separating signals into several clusters, and learned cluster sub-dictionaries. The same clustering criterion is used to classify a signal into a cluster and use its dictionary for the purpose of sparse coding. Along this line, researchers aimed at designing directionally-structured dictionaries that correspond to directionally-selective signal classes. In [30], Yang *et al.* employed multiple geometric cluster dictionaries. Each cluster is concerned with a certain directional structure. Sparse coding of a signal is carried out over the cluster dictionary that best fits this signal based on its structure. Another approach by Feng *et al.* [32] employs subspace segmentation-based DL for embedding structures in the learned dictionary. In [33], Yu *et al.* designed a structural dictionary composed of several orthogonal bases that correspond to different structures. Again, sparse coding of a signal is done by first selecting the best fitting basis (model) according to the signal's structure, and then calculating the sparse coding coefficient with respect to this basis.

2.6 Classical Applications of Sparse Representation in Image Processing

A quick revision to the problems of single image super-resolution and denoising via sparse representation over learned dictionaries is presented in the following two subsections. These are benchmark image processing applications, and will be used as the basic applications to test the ideas proposed on this thesis.

2.6.1 Single Image Super-Resolution via Sparse Representation

Super-resolution is the problem of reconstructing a high resolution (HR) image from given low resolution (LR) image(s) of the same scene. The worst-case scenario is the case of having one LR image, and this is called as single-image super-resolution (SISR). Various approaches to the SISR problem have been proposed. The sparse representation-based approach is one of the most outstanding approaches. An algorithm for SISR using sparse representation is proposed by Yang *et al.* [34]. This algorithm is based on using patches of a LR image to reconstruct the corresponding patches of the unknown HR image. Sparsity of the representation is applied as a generic natural image property that is invariant with respect to scale. In this setting, they apply two constraints on the reconstructed HR image. First is a reconstruction constraint enforcing the blurred and downsampled version of the HR image estimate to be consistent with the given LR image. Second, is a sparsity constraint which assumes that the sparse coding coefficients of a LR patch with respect to a LR dictionary are similar to those of the underlying HR one with respect to a HR dictionary. More precisely, this means that the patch sparse representation coefficients are invariant to scale (resolution level). In [34], the authors used a set of sampled patch pairs as a dictionary pair for sparse coding. However, this paradigm is very slow in practice. Therefore, in [35], Yang *et al.* employed a pair of coupled dictionaries learned from such patch pairs in a coupled manner, i.e., in such a way that the invariance of the sparse approximation coefficients of LR and HR patches are kept very close to each other. They first calculated the sparse coding coefficients of a LR patch with respect to the LR dictionary. Then, they imposed these coefficients on the HR dictionary to find a HR patch estimate. For the sake of local consistency of reconstructed HR patches, it is advantageous to divide the LR image into overlapping patches [34, 35]. Correspondingly, the reconstructed HR patches are also overlapped. They are then reshaped and merged at the overlap locations to generate a HR image estimate.

The LR and HR dictionaries are simultaneously learned in a coupled manner. Patches of HR training images are extracted and column-stacked to form an array of HR training patches. The mean value of each patch is subtracted to allow for a better training. At the same time, LR images are obtained by applying a blurring and downsampling operation of the HR ones. Then, LR images are upsampled to the so called middle resolution (MR) level to allow for better feature extraction. Afterwards, first and second order gradient filters are operated over the MR images to extract the features. The next step is to combine the extracted features of each MR patch into a single column, and then to combine feature vectors column-wise to obtain the corresponding LR training patch array. Eventually, LR and HR training patches are used to simultaneously train for a pair of coupled LR and HR dictionaries, respectively. This coupling is vital for the validity of the sparse coding coefficient invariance assumption.

Given a HR vector patch $\mathbf{x}_H$, one may find the sparse coding coefficient vector $\mathbf{w}_H$ of this patch over a dictionary in the same resolution level $\mathbf{D}_H$. Vector selection techniques such the LASSO [23] can be applied for this purpose. A sparse approximation of $\mathbf{x}_H$ can be written as

$$\mathbf{x}_H \approx \mathbf{D}_H \mathbf{w}_H. \tag{2.16}$$

Analogously, one may obtain a sparse approximation for the corresponding LR patch $\mathbf{x}_L$. This requires a sparse coding coefficient vector $\mathbf{w}_L$ over a LR dictionary $\mathbf{D}_L$. This approximation can be written as

$$\mathbf{x}_L \approx \mathbf{D}_L \mathbf{w}_L. \tag{2.17}$$

A blurring and downsampling operator $\boldsymbol{\Psi}$ can be used to relate $\mathbf{x}_L$ and $\mathbf{x}_H$. With the assumption that $\boldsymbol{\Psi}$ also relates the atoms of $\mathbf{D}_L$ and $\mathbf{D}_H$, one may write

$$\mathbf{x}_L \approx \boldsymbol{\Psi}\mathbf{x}_H \approx \boldsymbol{\Psi}\mathbf{D}_H \mathbf{w}_H \approx \mathbf{D}_L \mathbf{w}_H. \tag{2.18}$$

Based on the above analysis, the following result in concluded $\mathbf{w}_H \approx \mathbf{w}_L$. Consequently, a reconstruction for $\mathbf{x}_H$ can be obtained using $\mathbf{D}_H$ and $\mathbf{w}_L$, as follows

$$\mathbf{x}_H \approx \mathbf{D}_H \mathbf{w}_L. \tag{2.19}$$

2.6.2 Image Denoising with the K-SVD Algorithm

An example denoising technique via sparse representation is proposed in [36] by Aharon *et al.* based on the K-SVD DL algorithm. In this approach, the noisy image is first divided into small overlapping patches. These are then reshaped into the vector form. Let us denote by $\mathbf{Y}$ a column stack of patches extracted from the noisy image. Let us further denote by $\mathbf{X}$ the corresponding patches of the noise-free image. It is customary to model the relationship between $\mathbf{X}$ and $\mathbf{Y}$ as follows.

$$\mathbf{Y} = \mathbf{X} + \eta, \tag{2.20}$$

where η is the added noise. Reconstructing an approximation to $\mathbf{X}$ from $\mathbf{Y}$ can be formulated as follows.

$$(\hat{\mathbf{X}}, \hat{\mathbf{Y}}) = \underset{\mathbf{X},\mathbf{Y}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{Y}\|_F^2 + \lambda \|\mathbf{D}\mathbf{W} - \mathbf{Y}\|_F^2 + \sum_i \|\mathbf{W}_i\|_0, \tag{2.21}$$

where λ balances the trade-off between the representation fidelity of sparse representation and the distance between $\mathbf{X}$ and $\mathbf{Y}$. This can be viewed as solving a set of smaller optimization problems. Each problem can be viewed as

$$(\hat{\mathbf{X}}_i, \hat{\mathbf{Y}}_i) = \underset{\mathbf{X}_i, \mathbf{Y}_i}{\operatorname{argmin}} \|\mathbf{X}_i - \mathbf{Y}_i\|_F^2 + \lambda \|\mathbf{D}\mathbf{W}_i - \mathbf{R}\mathbf{X}\|_F^2 + \sum_i \|\mathbf{R}_i\|_0, \tag{2.22}$$

where $\mathbf{R}_i$ is the matrix which selects the i-th patch ($\mathbf{X}_i$) from $\mathbf{X}$ i.e. $\mathbf{X}_i = \mathbf{R}_i\mathbf{X}$. Minimizing the above cost function minimizes the error between each sub-image in the true image $\mathbf{X}_i$ and the corresponding one $\mathbf{Y}_i$ in the noisy one. This is based on the assumption that each patch in the input image can be represented sparsely as a linear combination of a few atoms in a dictionary $\mathbf{D}$. Ideally, for denoising the first term should be rewritten as $\|\mathbf{X} - \mathbf{Y}\|_F^2 < C\sigma^2$ where C is a constant and σ^2 is the variance of the noise. However, this term is implicitly incorporated into the cost function in the selection of the parameter which will depend on the noise variance. The closed-form solution to this cost function is given by

$$\hat{\mathbf{X}} = \frac{\lambda \mathbf{Y} + \sum_i \mathbf{R}_i^T \mathbf{D}\mathbf{W}_i}{\lambda \mathbf{I} + \sum_i \mathbf{R}_i^T \mathbf{R}_i}. \tag{2.23}$$

The solution to this problem thus involves averaging of overlapping patches after each patch has been sparsely reconstructed along with a weighted sum of the original noisy patch. Each pixel in a patch is hence a weighted linear combination of different pixels, the weights being derived from the sparse coding. Since the patches are overlapping, the final value of each pixel is thus an average of all representations obtained from the sparse coding stage.

Chapter 3

WAVELET-DOMAIN DICTIONARY LEARNING AND SPARSE REPRESENTATION

3.1 Introduction

Wavelets have been widely used as orthogonal basis functions. They possess several desirable features such compactness, directionality and analysis in many scales. Sparsity is another desirable character of wavelets. One can impose sparsity on a wavelet representation by hard or soft thresholding the representation coefficients. However, such enforcement is shown not fit a wide class of signals. This means degrading the representation quality. This chapter presents an attempt to combine the aforementioned desirable wavelet features with the representative power of learned dictionaries. Particularly, the appeal of a learned dictionary in being locally adaptive to the signals of the class it is trained on. The proposed paradigm is tested over the problem of single-image super-resolution, as proposed in [37]. Experiments conducted on benchmark images show an outstanding super-resolution performance. This is because the designed subband dictionaries inherit the directional nature of their respective wavelet subbands.

In the upcoming sections, the proposed wavelet-domain dictionary learning and sparse coding super-resolution algorithm will be presented with experimental results investigating its performance. The peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) are used as quantitative quality metrics along with visual comparison as a qualitative measure.

3.2 The Proposed Wavelet-Domain Super-Resolution Approach

Recalling the review in Chapter 2, sparse coding over multiple compact dictionaries has been shown as a better alternative to designing a single highly redundant dictionary, in terms of both representation quality and computational complexity [38]. Motivated by these observations, it seems advantageous to perform sparse coding in the wavelet domain, over wavelet-domain learned dictionaries. This can be viewed in the following points.

I. Wavelet decomposition filters are in charge of performing the signal classification process. A signal is separated into wavelet subbands concerning the directional nature.

II. A dictionary learned in a certain wavelet subband is expected to inherit the directional nature of this underlying subband.

III. With wavelet analysis filters, there is no need to apply feature extraction filters.

IV. The variability of subbands within a certain wavelet subband is less than the general signal variability. This makes it possible to learn more compact dictionaries in the wavelet domain.

V. Wavelet synthesis filters are used to build up a signal reconstruction from its subbands. This means that that there will be no need for applying a sparse model selection criterion.

3.2.1 Coupled Dictionary Learning in the Wavelet Domain

Sparse representation-based super-resolution requires the availability of dictionaries in the two resolution levels. The proposed algorithm thus requires the availability of wavelet domain training subband signals in both resolution levels. Given a training image set, one can perform a two-level wavelet decomposition as depicted in Fig. 3.1. In this work, we assume that level-1 detail subbands are the training signals of the HR wavelet dictionaries, whereas the level-2 details are the training signals of the LR dictionaries of the same subbands. There is a variety of possible analysis filters to be used for this purpose. In this work, we employ the nearly symmetric symlet wavelet [39] of order 29. Borders are treated with periodic extension.

A common practice in wavelet domain-based super-resolution approaches such as the works in [40, 41, 42] is to assume that the wavelet filters can model the blurring and downsampling operator. This means that the given LR image is assumed to be the approximation subband of the target HR image. It is noted that more investigation can be directed towards designing filterbanks that can more accurately model the blurring and downsampling operator. Therefore, the super-resolution problem is to estimate the detail subbands of the unknown HR image. The corresponding wavelet subbands of the given LR image are thus used to reconstruct their counterparts of the HR one. In this work, we adopt this assumption. This means that there will be no reconstruction and thus no DL in the approximation wavelet subband. Accordingly, the proposed algorithm requires three pairs of coupled dictionaries for the three detail wavelet subbands; the horizontal, vertical and diagonal detail subbands.

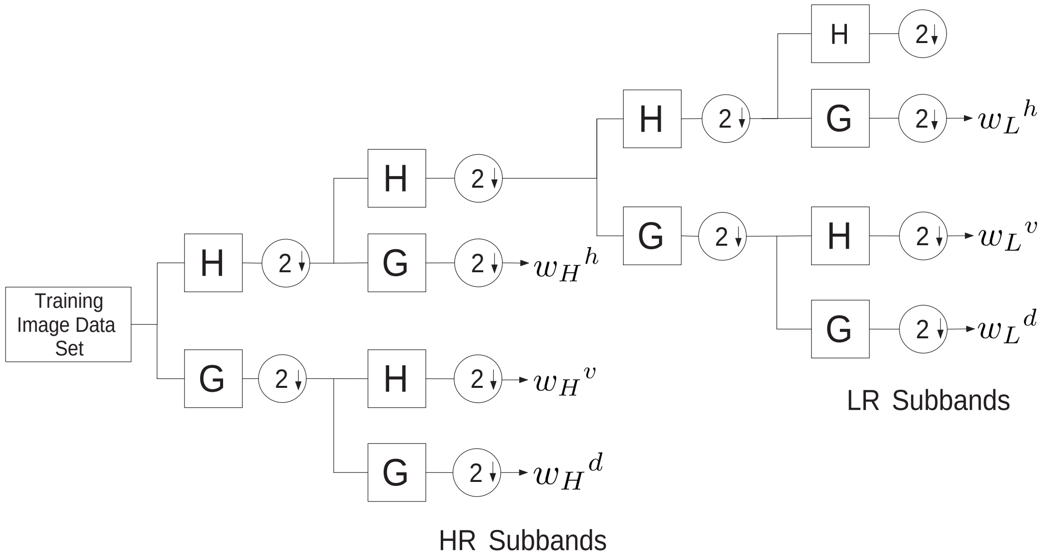


Figure 3.1. Two-level wavelet decomposition of the training image data set. Filters H and G are the scaling and wavelet filters, respectively.

An easy way to impose the coupling between a LR and HR dictionary is to first learn the LR one, and then calculate the HR one such that they are related with the blurring and downsampling operator, as proposed by Zeyde *et al.* in [43]. In this setting, LR training images are interpolated to the MR level. Then, features are extracted from these MR images and used as the training data for the LR dictionary. The next step is to use the corresponding patches in the HR training images, along with the sparse coding coefficients of the LR features over the LR dictionary to calculate a corresponding HR dictionary.

In this work, we adopt the above approach for learning the dictionary pairs in a coupled manner. Training data for the LR wavelet subbands are obtained by feeding the training images to a 2-level wavelet decomposition, as explained earlier. The details $\mathbf{w}_L{}^h, \mathbf{w}_L{}^v$ and $\mathbf{w}_L{}^d$ are interpolated to the MR level. Each of $\mathbf{w}_L{}^h, \mathbf{w}_L{}^v$ and $\mathbf{w}_L{}^d$ is interpolated separately by feeding it to a one-level inverse wavelet transform, while setting the other three subbands to zero. The wavelet-interpolated subbands at the MR level are labeled as $\mathbf{w}_M{}^h, \mathbf{w}_M{}^v$, and $\mathbf{w}_M{}^d$. This interpolation helps in maintaining the same directional nature of the respective wavelet subband. Also, it increases the size of the LR patches, allowing for sparse representation vectors that are dimensionally compatible with the HR dictionaries.

Since one deals with wavelet details, there is no need to apply feature extraction filters. This means that the wavelet subbands of each training LR image at the MR level ($\mathbf{w}_M{}^h, \mathbf{w}_M{}^v$, and $\mathbf{w}_M{}^d$) are divided into overlapping patches. In this work, we use a 2-pixel overlap. Extracted patches are then stacked column-wise to form a LR training array. Let $\mathbf{W}_M{}^h, \mathbf{W}_M{}^v$, and $\mathbf{W}_M{}^d$ denote the training array of all the LR training images in the horizontal, vertical and diagonal detail subbands, respectively. Then, one can view the learning process of the three LR subband dictionaries over these training data as follows.

$$\underset{\mathbf{D}_L{}^y, \boldsymbol{\alpha}_M{}^y}{\operatorname{argmin}} \|\mathbf{W}_M{}^y - \mathbf{D}_L{}^y \boldsymbol{\alpha}_M{}^y\|_2{}^2 \, subject\, to \, \|\boldsymbol{\alpha}_M{}^y\|_0 < S, \tag{3.1}$$

where the subscripts L and M denote the LR and MR levels, respectively. The superscript $y = \{h, v, d\}$ stands respectively for the horizontal, vertical and diagonal detail wavelet subbands. $\mathbf{D}_L{}^y$ denotes the three LR subband dictionaries learned with $\mathbf{W}_M^y$, respectively. $\boldsymbol{\alpha}_M{}^y$ denotes the sparse representation coefficients of $\mathbf{W}_M{}^y$ as coded over $\mathbf{D}_L^y$, respectively. A DL algorithm such as the K-SVD algorithm [28] can be used to solve for the above optimization problem.

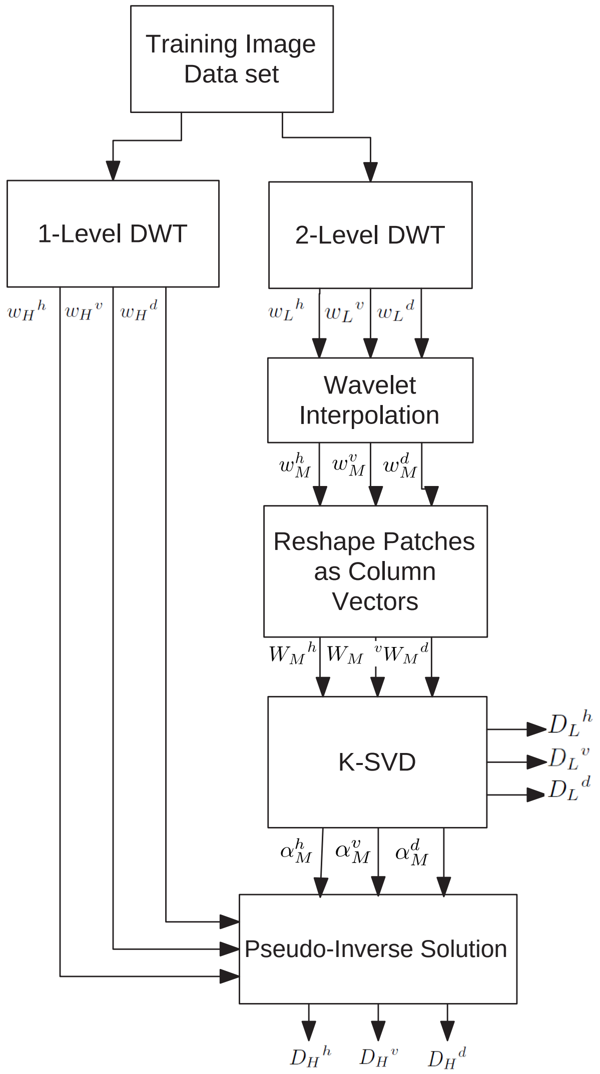

Training Image Data set
1-Level DWT
2-Level DWT
$w_H{}^h$ $w_H{}^v$ $w_H{}^d$
$w_L{}^h$ $w_L{}^v$ $w_L{}^d$
Wavelet Interpolation
w_M^h w_M^v w_M^d
Reshape Patches as Column Vectors
$W_M{}^h$ $W_M{}^v$ $W_M{}^d$
K-SVD
$D_L{}^h$
$D_L{}^v$
$D_L{}^d$
α_M^h α_M^v α_M^d
Pseudo-Inverse Solution
$D_H{}^h$ $D_H{}^v$ $D_H{}^d$

Figure 3.2. The proposed wavelet-domain DL Algorithm.

As mentioned earlier, HR wavelet training subbands ($\mathbf{w}_H^y$) are obtained as the 1-level wavelet subbands of the training images. The next step is to divide each subband into patches, reshape them into the vector form and stack them column-wise to form a training array of HR patches. Similar to the DL algorithm of Zeyde *et al.*, HR training subbands $\mathbf{W}_H^y$ and the sparse coding coefficients $\boldsymbol{\alpha}_M^y$ on the respective LR subbands are together used to calculate a coupled HR dictionary. This calculation can be viewed as:

$$
{\mathbf{D}_H}^y = {\mathbf{W}_H}^y {\boldsymbol{\alpha}_H}^{y\dagger} \approx {\mathbf{W}_H}^y {\boldsymbol{\alpha}_M}^{yT} \left({\boldsymbol{\alpha}_M}^y {\boldsymbol{\alpha}_M}^{yT}\right)^{-1}, \tag{3.2}
$$

where the superscripts †, T and -1 denote the Moore-Penrose pseudo-inverse, algebraic transpose and inverse operators, respectively. The proposed wavelet-domain DL algorithm is illustrated in Fig. 3.2. In this work, we employed a training set composed of natural and computer-generated images. 6×6 patches are extracted from the training subbands. Three couples of LR and HR wavelet subband dictionaries are learned with the proposed DL algorithm. The K-SVD algorithm with twenty iterations and sparsity S=3 is used to carry out the LR dictionary learning process. Figure 3.3 shows example reshaped atoms of the horizontal, vertical and diagonal detail HR dictionaries $\mathbf{D}_H^h$, $\mathbf{D}_H^v$ and $\mathbf{D}_H^d$, in subfigures (a), (b) and (c), respectively. It is notable that the designed dictionaries inherit the directional nature of their respective subbands.

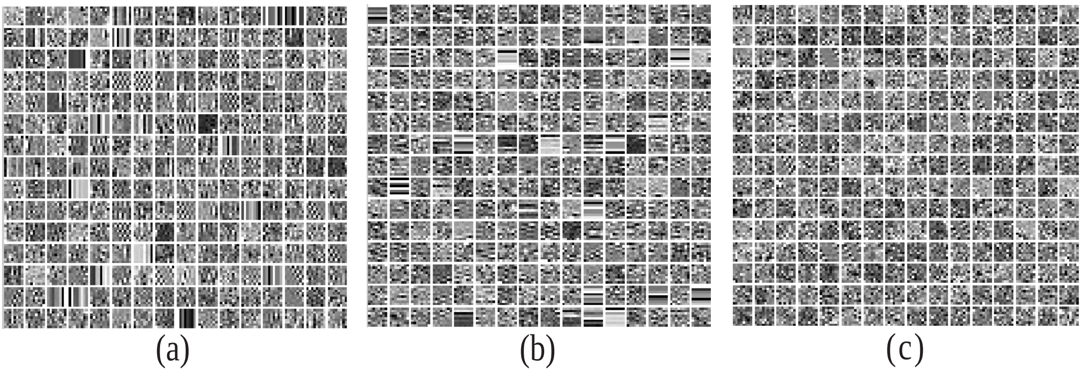


Figure 3.3. Example HR subband dictionary portions. (a): horizontal detail subband dictionary, (b): vertical detail subband dictionary and (c): diagonal detail subband dictionary.

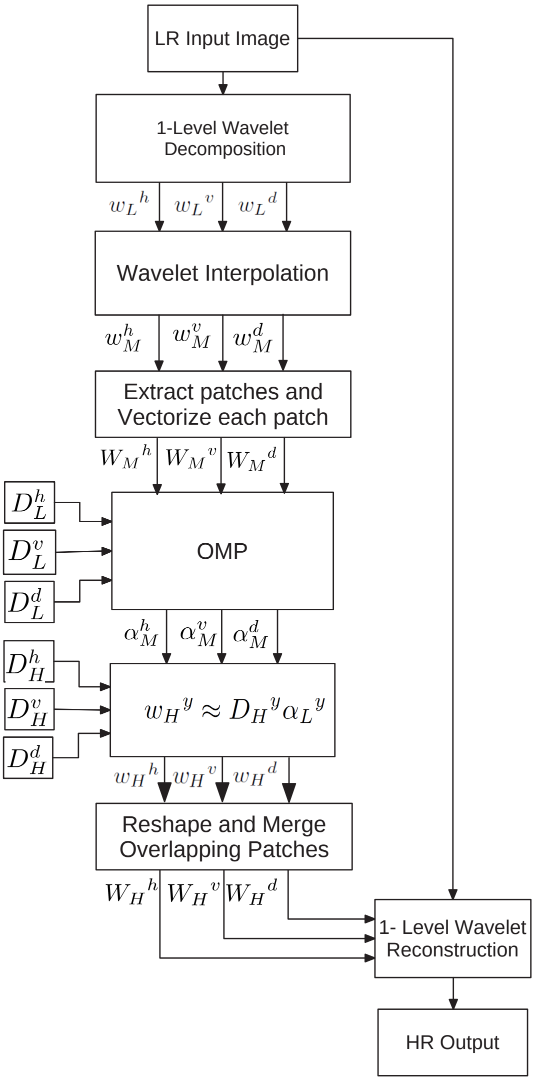


Figure 3.4. The proposed wavelet subband-based image reconstruction algorithm

3.2.2 Reconstructing the HR Wavelet Subbands

The proposed super-resolution algorithm aims at estimating the detail subbands of the unknown HR image from their counterparts in the LR image. Once these are estimated, a wavelet synthesis stage uses them along with the input LR image as the approximation subband and reconstructs a HR image estimate. This means two usages of the given LR image. The given LR image is first decomposed to give its wavelet subband coefficients. The same wavelet filters used in the training process are to be used in the reconstruction stage, as well. This decomposition gives the wavelet subbands $\mathbf{w}_L{}^y$ of the LR image. The next step is to upsample each subband to the MR level as $\mathbf{w}_M{}^y$. Then, it is divided into overlapping patches. These are then reshaped into the vector form and stacked column-wise to form a LR feature array $\mathbf{W}_M^y$.

The next step is to find the sparse coding coefficients of $\mathbf{W}_M^y$ over the corresponding LR subband dictionary $\mathbf{D}_L{}^y$. This can be formulated as the following sparse coding problem which can be solved using an algorithm such as OMP.

$$\underset{\boldsymbol{\alpha}_M{}^y}{\operatorname{argmin}} \|\mathbf{W}_M{}^y - \mathbf{D}_L{}^y \boldsymbol{\alpha}_M{}^y\|_2 \;\; s.t. \;\; \|\boldsymbol{\alpha}_M{}^y\|_0 < S. \tag{3.3}$$

Applying the basic assumption of sparse representation-based super-resolution, the sparse coding of a LR subband over a LR subband dictionary can be used along with a HR subband dictionary to reconstruct the corresponding HR wavelet subband. Therefore, an array of HR subband patches can be reconstructed as follows.

$$\mathbf{W}_H{}^y \approx \mathbf{D}_H{}^y \boldsymbol{\alpha}_H{}^y \approx \mathbf{D}_H{}^y \boldsymbol{\alpha}_M{}^y. \tag{3.4}$$

Each column of $\mathbf{W}_H{}^y$ is then reshaped as a two-dimensional (2-D) patch. Next, overlapping patches are merged to form a 2-D wavelet subband $\mathbf{w}_H{}^y$. Since patches of $\mathbf{W}_H{}^y$ are overlapping, each pixel value in $\mathbf{w}_H{}^y$ is taken as the average of its values belonging to the overlapping patches that include this pixel. To this end, the algorithm reconstructs ${\bf w}_H{}^h$, ${\bf w}_H{}^v$ and ${\bf w}_H{}^d$ as 2-D wavelet subbands. Given the LR image and the reconstructed HR wavelet subbands, a HR image estimate can be obtained by performing a one-level inverse wavelet transform. Figure 3.4 illustrates the reconstruction process.

3.2.3 Sparse Coding and Dictionary Learning Computational Complexity Reduction

As presented in Chapter 2, the DL process is based on alternating between a sparse coding stage and a dictionary update stage. It is well-known that the sparse coding stage requires a vector selection algorithm, and is thus more computationally demanding than the dictionary update [44]. On the other hand, the computational complexity of sparse coding depends on the dictionary dimensions. In this regard, designing multiple compact dictionaries means reducing the computational complexity, as compared to the case of using a single highly redundant dictionary. In return, this means significantly reducing the DL computational complexity, since sparse coding causes the biggest computational complexity overhead in this process.

Let us analyze the computational complexity of OMP as an example vector selection technique, and compare it for the cases of using a single highly redundant dictionary and using multiple more compact dictionaries. It is shown in [45] that the computation complexity of OMP to find the sparse coding of an n-dimensional signal over an K-atom dictionary with a sparsity S is $\mathcal{O}(KSn)$. Therefore, it can be shown that if one uses three subband dictionaries smaller than a single dictionary with a factor γ, the computational complexity of sparse coding will be $\mathcal{O}(\frac{3KSn}{\gamma})$.

It is thus evident that using multiple compact wavelet subband dictionaries instead of a single highly redundant one reduces the sparse coding and DL computational complexities. Basically, it is intuitively expected that a subband dictionary can be compact, i.e., is not needed to be highly redundant. This is because a subband dictionary is, in essence, a class dictionary, as it is responsible only for representing the signals of its class which have a common similarity. Comparing, for example, the computational complexity of learning a single large dictionary in the spatial domain of size 1000, to learning three smaller dictionaries in the wavelet domain of size 216, one approximately reduces the computational complexity of the OMP vector selection stage by a factor of 1.54.

3.3 Simulations and Results

In this section, we present a study of the impact of the patch size and dictionary redundancy on the proposed algorithm's performance. It is shown that a relatively large patch size is can be employed while using a small training set. Next, the performance of the proposed algorithm is compared to that of several other surer-resolution techniques. This is done in terms of peak signal-to-noise ratio (PSNR) [46] and structural similarity index (SSIM) [47] as quality metrics, along with visual comparisons.

Given the true image $\mathbf{y}$ and its estimate $\widehat{\mathbf{y}}$, both being 8-bit (gray level) with $N_1 \times N_2$ pixels, the PSNR is defined as

$$PSNR(\mathbf{y}, \widehat{\mathbf{y}}) = 10 \log_{10} \frac{255^2}{MSE(\mathbf{y}, \widehat{\mathbf{y}})}, \tag{3.5}$$

Table 3.1. Kodak set PSNR (dB) and SSIM comparisons of bicubic interpolation, the baseline algorithm of Zeyde *et al.*, wavelet interpolation and the proposed algorithm.

Image	Bic.	Zeyde *et al.*	Wav. Int.	P_{6x6}	Image	Bic.	Zeyde *et al.*	Wav. Int.	P_{6x6}
1	26.69 0.9117	27.78 0.9673	27.30 0.9345	30.10 0.9726	13	24.71 0.9265	25.50 0.9735	25.08 0.9495	27.85 0.9726
2	34.03 0.8939	34.98 0.957	34.42 0.9503	36.59 0.9834	14	29.89 0.9487	31.19 0.9826	30.21 0.9653	33.26 0.9879
3	35.03 0.9103	36.54 0.9579	35.58 0.9681	36.92 0.9818	15	32.88 0.9073	34.39 0.955	34.43 0.9674	36.78 0.9902
4	34.57 0.9434	35.94 0.9787	35.09 0.9726	39.59 0.9929	16	32.05 0.9263	32.82 0.9692	32.31 0.9564	35.40 0.9843
5	27.13 0.9538	28.77 0.9864	27.62 0.9657	30.31 0.9869	17	32.86 0.9536	34.24 0.9813	33.26 0.9785	37.19 0.9928
6	28.27 0.904	29.21 0.9548	28.70 0.9493	30.16 0.9729	18	28.78 0.9358	29.80 0.9761	29.40 0.9623	31.23 0.9816
7	34.27 0.9425	36.25 0.9825	34.47 0.9782	39.98 0.9969	19	28.79 0.9322	30.02 0.9718	28.95 0.9569	30.35 0.9764
8	24.31 0.92	25.36 0.9675	24.46 0.9312	27.18 0.9706	20	32.36 0.8217	33.90 0.8843	32.63 0.9672	34.50 0.9869
9	33.13 0.8922	34.89 0.9505	33.17 0.9606	35.07 0.9836	21	29.26 0.8961	30.29 0.9545	29.59 0.9639	31.43 0.98
10	32.94 0.9173	34.47 0.9649	33.27 0.9618	36.11 0.9901	22	31.36 0.9295	32.45 0.9733	31.60 0.9615	33.68 0.9849
11	29.93 0.8925	31.05 0.9518	30.13 0.9472	32.26 0.9804	23	35.92 0.945	37.78 0.9738	36.00 0.9818	39.04 0.9857
12	33.56 0.908	35.13 0.9605	34.40 0.9581	37.09 0.9835	24	27.62 0.9365	28.57 0.9744	28.20 0.9598	30.59 0.9855
					Average.	30.85 0.9187	32.14 0.9646	31.26 0.9603	33.86 0.9835

where $MSE(\mathbf{y}, \widehat{\mathbf{y}})$ denotes the mean-squared error between **y** and $\widehat{\mathbf{y}}$, which is defined as

$$MSE(\mathbf{y}, \widehat{\mathbf{y}}) = \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (\mathbf{y}_{ij} - \widehat{\mathbf{y}}_{ij})^2. \tag{3.6}$$

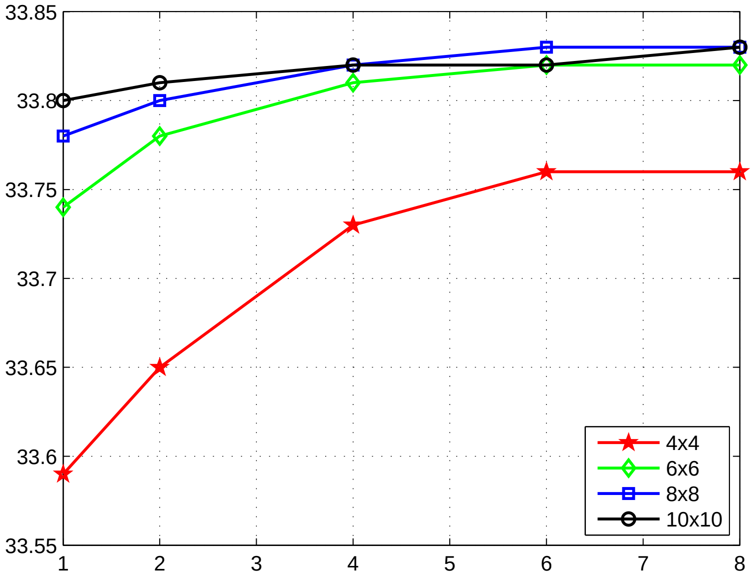


Figure 3.5. Average Kodak set PSNR vs. dictionary length-to-width ratio for different patch sizes.

3.3.1 The Effect of Patch Size and Dictionary Redundancy on the Representation Quality

In view of the existing trade-off between the dictionary redundancy and the representation quality [48]. It is interesting to investigate the performance of the proposed algorithm with different levels of redundancy. In this context, one may loosely define redundancy as the ratio between the number of dictionary atoms to the dimension of each atom. In case of sparse coding of image patches, the atom size is equal to the patch size. This means that one can define the redundancy as the ratio between the number of dictionary atoms to the patch size employed. In this regard, the patch dimension is also an influential factor; a large patch size is generally desirable as it is better able to describe image features and structures more distinctly. On the other hand, this means the need for a larger training set to account for these possible structures. To sum up, there is an upper limit on effective patch sizes, as a large patch size means requiring an extensive training set. It also means a larger atom dimension. For the

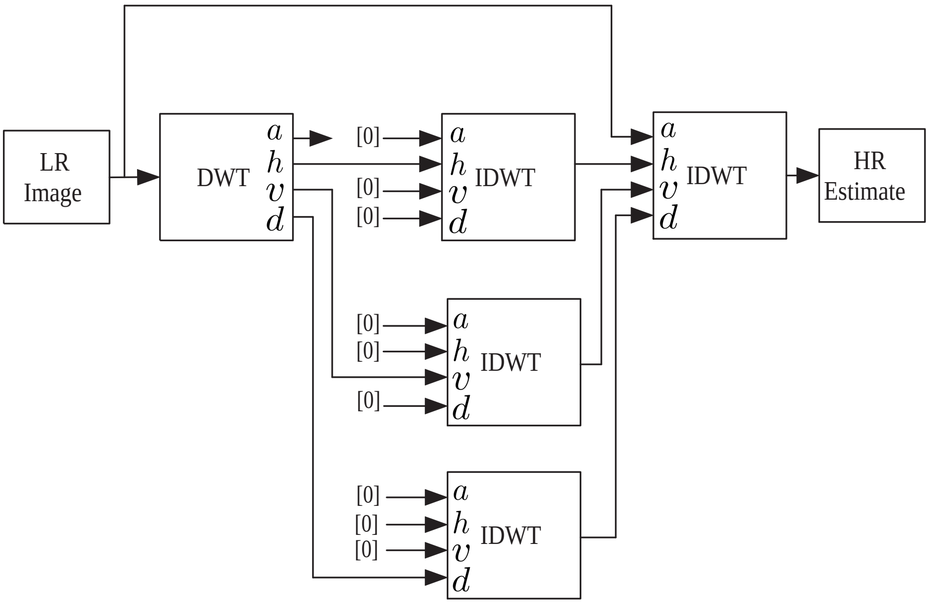


Figure 3.6. Wavelet Interpolation. Symbols a, h, v, d and [0] denote the approximation, horizontal detail, vertical detail and diagonal detail wavelet subbands, and zero matrix respectively.

purpose of investigating the proposed algorithm's performance with different redundancies, the following experiment is conducted. The proposed algorithm is run with different patch sizes and different redundancy levels over the images of the Kodak set [49] as test images. DL is done with the K-SVD algorithm with 20 iterations and S=3. Besides, it is made sure that the training set does not include any of the test images. The average PSNR value for the whole set is plotted in Fig. 3.5 for each case. It can be seen in view of Fig. 3.5 that a redundant dictionary performs better than a complete one, for all patch sizes. Besides, it can be seen that a redundancy of 6 gives a good performance for all patch sizes, beyond this value, there is no significant performance improvement. Furthermore, a patch size of 6×6 forms a good compromise between computational complexity and representation quality inferred in terms super-resolution

PSNR performance.

3.3.2 Wavelet-Domain Dictionary Learning and Sparse Coding for Single-Image Super-resolution

In this section we investigate the performance of the proposed algorithm as compared to the baseline algorithm of Zeyde *et al.* [43] and bicubic interpolation. It is also interesting to compare the performance of the proposed algorithm with that of wavelet interpolation. In this context, wavelet interpolation interpolates the wavelet subbands of the given LR image while conserving their directional nature. This is achieved by first decomposing the image into wavelet subbands. Then the three detail wavelet subbands of this image are individually interpolated. Each detail subband in interpolated by feeding it to a discrete wavelet transform (DWT) synthesis stage while setting the other three subbands to zero. This means that the reconstruction wavelet filters will be used to interpolate this subband while preserving its subband (directional) nature. After interpolating the wavelet subbands individually, they are fed along with the LR image assumed as the approximation subband, to an inverse wavelet transform (IDWT) stage using the same reconstruction filters. The output of this stage is the wavelet interpolation of the LR image. This interpolation scheme is depicted in Fig. 3.6.

In view of the conclusion made about Fig. 3.5, the proposed algorithm is set to use a patch size of 6×6 with a redundancy level of 6. This means that the designed subband dictionaries are of the dimension 36×216. A sparsity S=3 is used throughout the simulations. The baseline algorithm uses a patch size of 4×4, with 1000-atom dictionaries. It is noted that the baseline algorithm uses 986,981 patches, while the proposed algo-

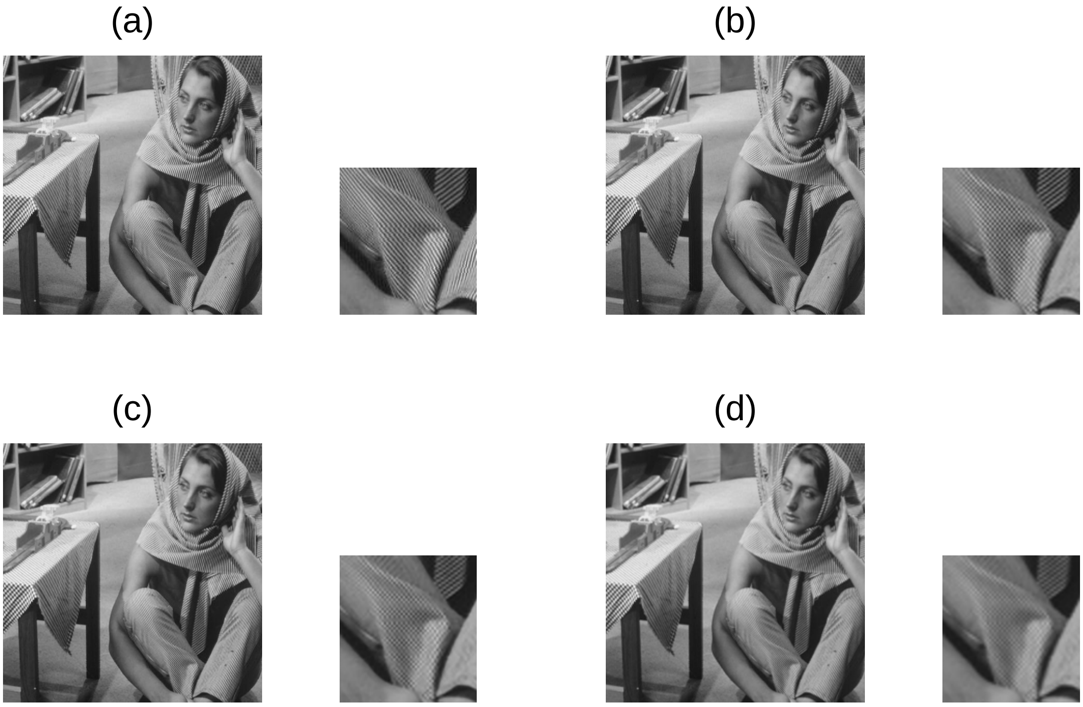


Figure 3.7. Visual comparison of the Barbara image. (a) original image, (b) the proposed algorithm's result, (c) the baseline algorithm's result, (d) bicubic interpolation's result.

rithm uses only 98,538 patches to give satisfactory results. This comes in accordance with proposition that DWT sparsifies a given training set allowing for using smaller data sets. Besides, a wavelet subband is a signal class of less variability as compared to the general signal case. Therefore, fewer training vectors are required to train for a wavelet subband. The 24 images of the Kodak set are used as test images. PSNR and SSIM measures as used in this test. Table 3.1 shows the PSNR and SSIM values of the aforementioned approaches. These are denoted by "Bic.", "Zeyde *et al.*", "Wav. Int." and "$P_{6\times6}$", respectively.

In view of Table 3.1, the proposed algorithm has an average PSNR improvement of 1.71 dB over Zeyde *et al.*'s algorithm. Besides, the average PSNR improvement over wavelet interpolation is 2.6 dB, and there is a significant improvement over bicubic

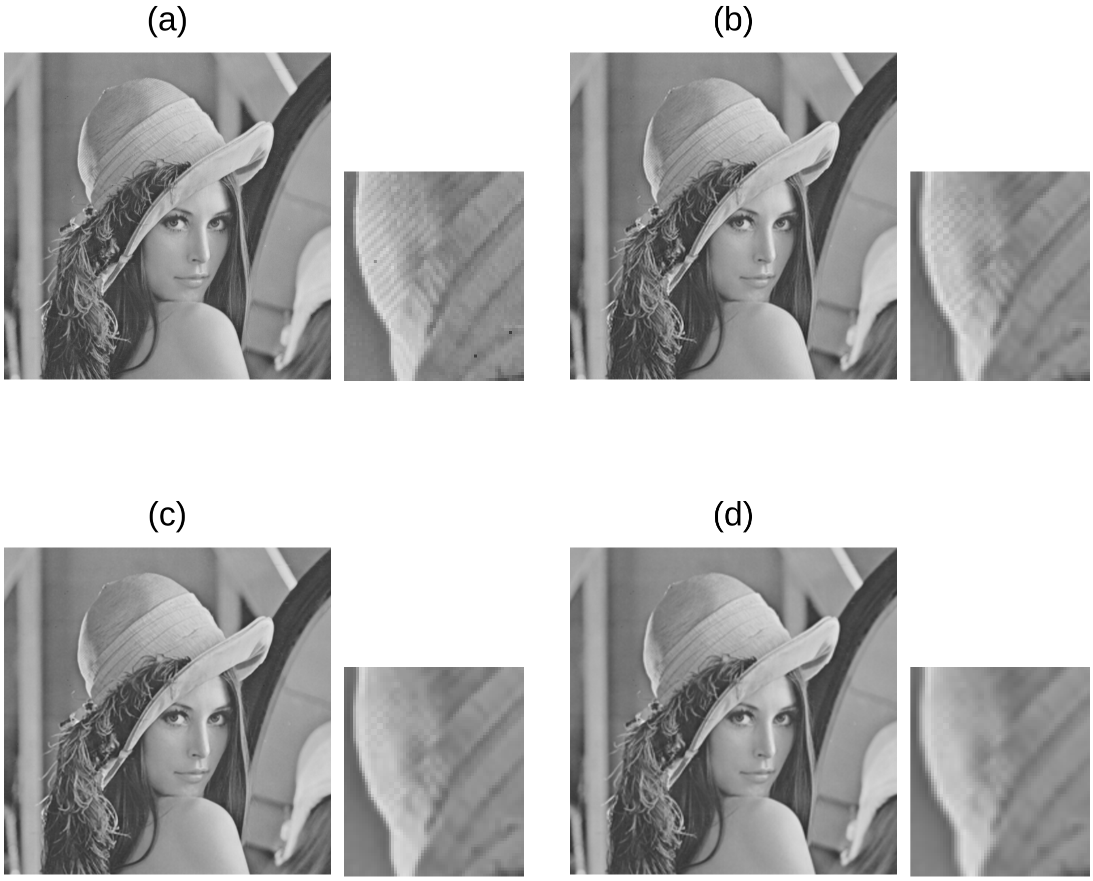


Figure 3.8. Visual comparison of the Lena image. (a) original image, (b) the proposed algorithm's result, (c) the baseline algorithm's result, (d) bicubic interpolation's result.

interpolation. These results point out that the proposed algorithm is better able in reconstructing the high frequency (HF) image contents. The same observations can be made in terms of the SSIM measure.

As another quantitative assessment, the performance of the proposed algorithm is compared to those of bicubic interpolation and the algorithms of Zeyde *et al.*, Timezel [42], and the nonlocal autoregressive model (NARM) algorithm of Dong *et al.* [50] which are more recent super-resolution approaches. This test is conducted over a set of well-

Table 3.2. PSNR (dB) and SSIM comparisons of bicubic interpolation, Temizel's algorithm, NARM algorithm, the baseline algorithm and the proposed algorithm with benchmark images.

Image	Bic	Temizel	NARM	Zeyde *et al.*	P_{6x6}
Barbara	25.27	-	23.86	**25.73**	**25.73**
	0.9117	-	0.8242	0.9622	**0.9680**
Elaine	31.06	**33.4**	30.38	31.32	31.45
	0.9088	-	0.6733	0.9462	**0.9687**
Baboon	22.98	24.24	23.74	24.50	**24.61**
	0.9330	-	0.7004	0.9653	**0.9796**
Peppers	30.28	34.18	**35.40**	35.20	34.52
	0.9505	-	0.9060	0.9753	**0.9831**
Fingerprint	31.95	-	33.13	33.97	**34.98**
	0.9911	-	0.9613	**0.9979**	0.9974
Lena	34.70	34.68	35.01	36.18	**36.80**
	0.9566	-	0.9238	0.9807	**0.9902**
Zone-plate	11.40	-	10.87	11.98	**12.72**
	0.6923	-	0.5875	**0.8678**	0.7978
Boat	29.93	-	32.61	31.25	**33.76**
	0.9276	-	0.9189	0.9626	**0.9741**
Avg.	27.20	-	28.13	28.77	**29.32**
	0.9090	-	0.8119	0.9573	**0.9574**

known benchmark images. Again, a scale-factor of 2 is considered. The algorithm of Temizel is included in this comparison because it is wavelet-based and does assume that the LR image is the approximation subband of the HR image. The NARM algorithm is chosen since it has a state-of-the-art performance and is compared with several outstanding super-resolution techniques, as reported in [50]. PSNR and SSIM Results of this test are reported in Table 3.2.

As noted in Table 3.2, the proposed algorithm is clearly superior to bicubic interpolation and generally performs better than the baseline algorithm, with an average PSNR improvement of 0.83 dB. Also, the proposed algorithm's performance is comparable to that of Temizel's algorithm, and is generally better than that of the NARM algorithm. In accordance with the PSNR performance, SSIM values of the proposed algorithm are comparable to those of the baseline algorithm. Besides there is an average SSIM im-

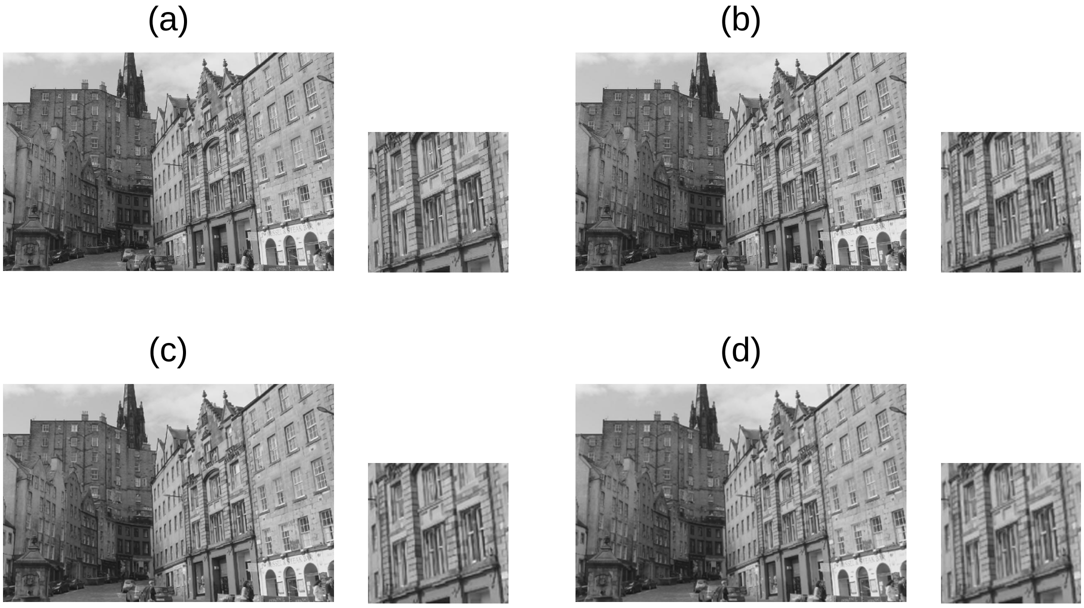


Figure 3.9. Visual comparison of perspective image. (a) original image, (b) the proposed algorithm's result, (c) the baseline algorithm's result, (d) bicubic interpolation's result.

provement of 0.1455 and 0.0484 over the NARM algorithm and bicubic interpolation, respectively.

As an empirical indication to the computational complexity of the proposed algorithm as compared to that of the baseline algorithm, we measured the execution times of both algorithms. As run on an Intel Core i7 2.00 GHz laptop PC under Matlab R2009a environment, the proposed algorithm requires 1.01 S to train for a 36×216 dictionary, with the aforementioned settings. However, training for a 36×1000 dictionary with the same settings is 150.68 S. This result comes in accordance with the expected reduction of computational complexity as a result of designing more compact dictionaries, as explained in subsection 3.2.3.

For a qualitative assessment, Figures 3.7, 3.8, 3.9 and 3.10 compare the ground-truth

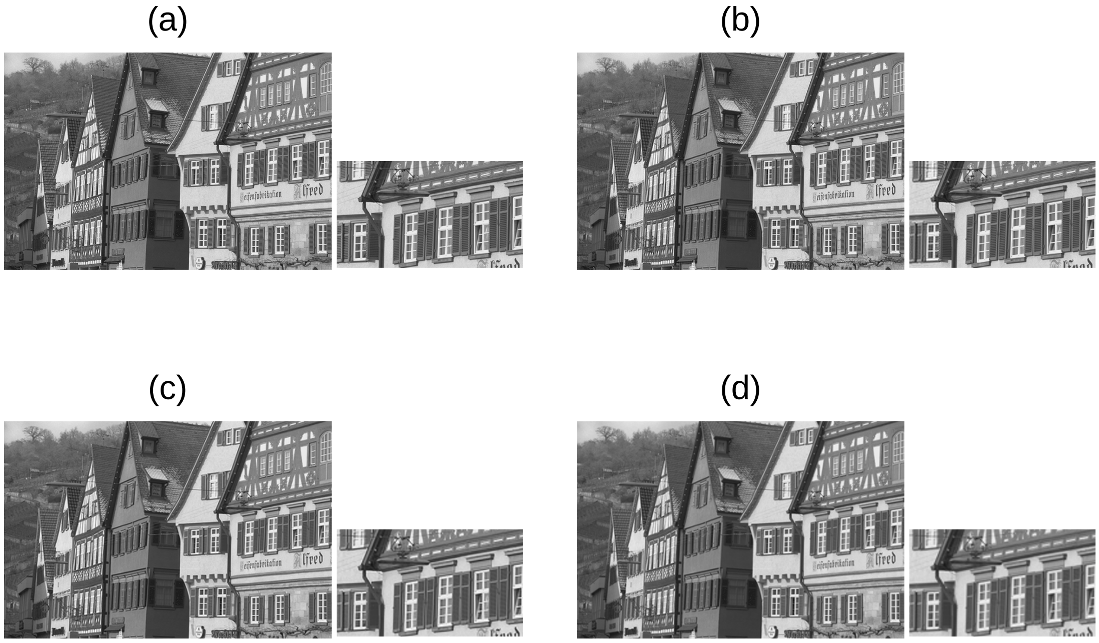


Figure 3.10. Visual comparison of image number 8 in the Kodak set. (a) original image, (b) the proposed algorithm's result, (c) the baseline algorithm's result, (d) bicubic interpolation's result.

image to its reconstructions obtained with the proposed algorithm, the baseline algorithm of Zeyde *et al.* and bicubic interpolation on the images Barbara, Lena, a perspective image and Kodak set image number 8, respectively. It can be seen in Fig. 3.7 the proposed algorithm's reconstruction is the best to approximate the true image. It is slightly better than that of Zeyde *et al.*'s and bicubic interpolation's reconstructions. Particularly, the insets provided show that Zeyde *et al.* and bicubic interpolation's reconstructions exhibit aliasing. However, the proposed algorithm's reconstruction exhibits less aliasing. Similar conclusions can be made about Fig. 3.8. It is again observed that the proposed algorithm's reconstruction has the smallest amount of artifacts. The same observations are made with the case of Fig. 3.9. It is seen in view of Fig. 3.10 that the proposed algorithm is better able to reconstruct the edges.

This can be particularly seen in observing the details on the window object. Overall, the aforementioned visual comparisons point out that the proposed algorithm is better in terms of reconstructing the image HF details such as edges and textures and thus exhibits less artifacts.

For the purpose of examining the ability of the proposed algorithm in reconstructing the detail wavelet subbands, Figures 3.11, 3.12 and 3.13 show the detail subbands of the original image, along with the corresponding ones in the reconstructions obtained with the proposed algorithm, the baseline algorithm and bicubic interpolation. This test is conducted on the inset shown in Fig. 3.10 as it is rich of texture and other HF contents. In view of these figures, it is noted that the proposed algorithm superior in reconstructing the detail wavelet subbands on an image.

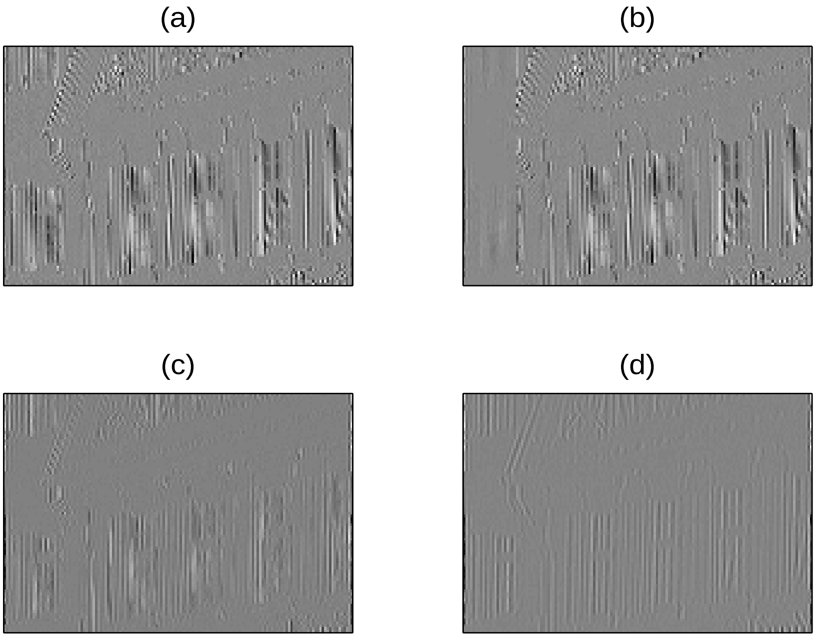


Figure 3.11. Horizontal detail subband reconstruction. (a) original and reconstructions with (b) the proposed algorithm, (c) the baseline algorithm and(d) bicubic interpolation.

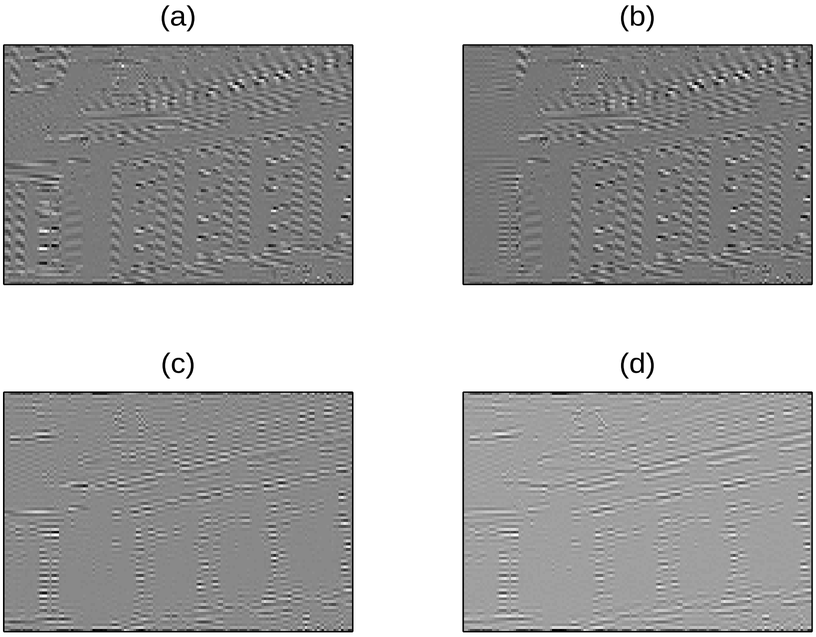


Figure 3.12. Vertical detail subband reconstruction. (a) original and reconstructions with (b) the proposed algorithm, (c) the baseline algorithm and(d) bicubic interpolation.

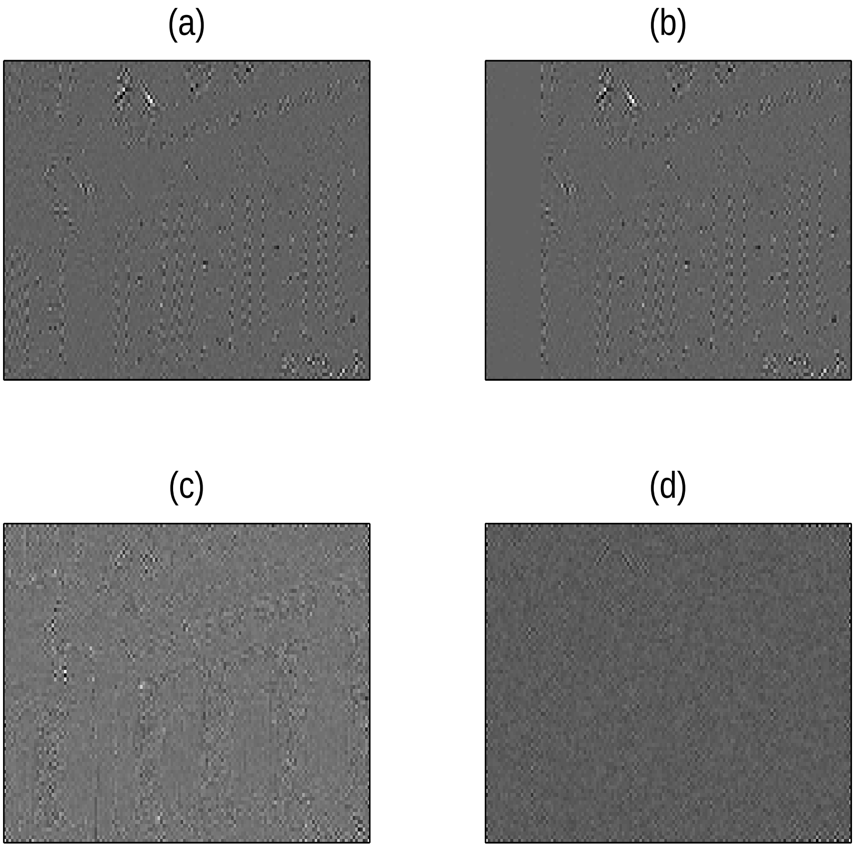


Figure 3.13. Diagonal detail subband reconstruction. (a) original and reconstructions with (b) the proposed algorithm, (c) the baseline algorithm and(d) bicubic interpolation.

Chapter 4

DIRECTIONALLY-STRUCTURED DICTIONARY LEARNING AND SPARSE CODING BASED ON SUBSPACE PROJECTIONS

4.1 Introduction

Recalling the discussion made in Chapter 2, it is noted that sparse coding over multiple dictionaries came as an attempt towards improving the representation quality while reducing the computational cost of sparse representation. In this approach, each dictionary is concerned with a certain signal class. DL is done over the class signals, and sparse coding of a signal in a specific class is then carried out over the class dictionary. However, there are still open ended questions concerning the definition of a class, and the accuracy of the model selection process in adopting a certain class to a given test signal. Accordingly, the work presented in this chapter forms an attempt to address these questions. In this chapter, we propose a strategy for designing multiple structured dictionaries in such a way that the signal as a whole is represented as a summation of structural components. This work is published in [51].

In our approach, we divide the signal space into directionally-selective structured (directional) subspaces. This signal division process is carried out using the projection operation. Projection operators are specially designed to fit for this task. In this setting, a signal class is in fact a signal subsapce obtained as the projection of the signal space using a certain projection operator that is concerned with a certain directionality.

For each signal subsapce, a compact dictionary is learned over the subsapce training set which is obtained by projecting the training set onto that subsapce. On the sparse coding side, a signal is decomposed into its structural components using the suitable projection operators. Then, the signal's sparse approximation is obtained as the direct sum of the sparse codings of each of these components, each coded over its repressive subsapce dictionary. Since projection operators are designed to exactly decompose and reconstruct a signal, the signal as a whole is represented. This is essentially the advantage of this proposed strategy over standard sparse coding over multiple dictionaries with a single model selection, e.g., [31, 32, 33]. In these approaches, a model (dictionary) is selected once for each signal based on its major information content. However, this approach over-looks the possibility that this signal may possess structural components that are not suitably fit with the selected model.

The proposed strategy is shown to improve the sparse representation quality as compared to standard sparse coding over multiple dictionaries. This result is validated in terms of the PSNR measure over the tests of image representation. Moreover, the designed subspace dictionaries are shown to inherit the intended directional nature of their respective directional subspaces.

4.2 The Proposed Dictionary Learning and Sparse Approximation Strategy

Signal and Image features are inherently directional. Redundant dictionary-based sparse representation approaches most often fail to take the advantage of the directional image contents. In the standard DL approach, the training data is treated as a whole without emphasizing the structural image content. Better representation of this directional content promises to improve the overall image representation quality. The scope of this work is to generate a multiplicity of directional structured subspace dictionaries. Such dictionaries are expected to better represent various image patch structures, especially the least common ones. The same idea can be extended towards designing dictionaries based on other image features, such textures.

4.2.1 Subspace dictionary learning

Let us denote by $\mathbf{U}$ the signal space in $\mathbf{R}^n$. Let us further denote by $\mathbf{u}_1$ and $\mathbf{u}_2$ the subspace of horizontally and vertical aligned signals in $\mathbf{R}^n$. It is noted that these two subspaces are selected to have orthogonal directionalities.This orthogonality comes in harmony with the nature of OMP as a vector selection algorithm. In fact, OMP initializes a so-called residual vector with the signal, and successively approximates this vector by selecting more dictionary atoms. This residual is updated after each atom selection. In this update process, the residue is orthogonal to the atoms selected, as detailed in Chapter 2. If $\mathbf{u}_3$ denotes the remainder of signal space of signals being neither horizontal nor vertical, then $\mathbf{U}$ can be written as the direct summation of the three subspaces as $\mathbf{U} = \mathbf{u}_1 \oplus \mathbf{u}_2 \oplus \mathbf{u}_3$. Intuitively, the projection of a signal onto a certain subspace is the signal's component that belongs to this subspace. A projection operator can thus be designed to characterize the directional nature of its corresponding subspace. The three subspace can be written as projections of $\mathbf{U}$ onto their respective column spaces, as $\mathbf{u}_1 = \phi_1 \mathbf{U}$, $\mathbf{u}_2 = \phi_2 \mathbf{U}$ and $\mathbf{u}_3 = \phi_3 \mathbf{U}$, where ϕ_i is a projection operator defining the i-th subspace. Complete division of $\mathbf{U}$ into these three subspaces requires that $\phi_1 + \phi_2 + \phi_3 = \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix.

To this end, one can think about designing such projection operators to serve for the division purpose. If $\mathbf{s}_1$ denotes a set of horizontally-aligned signals (vectorzied image patches), one can obtain a projection operator that corresponds to the directionality of these patches as $\mathbf{p}_1 = \mathbf{s}_1\mathbf{s}_1{}^{\dagger}$ where the superscript † denotes the Moore-Penrose pseudo-inverse. In this setting, pre-multiplying $\mathbf{U}$ with $\mathbf{p}_1$ yields the projection of $\mathbf{U}$ onto the column space of the vectors in $\mathbf{s}_1$. In other words, if $\mathbf{s}_1$ contains enough vectors, one may assume that this set represents almost all horizontally-aligned patches. Correspondingly, if another set $\mathbf{s}_2$ contains enough vertically-aligned patches, then one may similarly obtain the vertical projection operator as $\mathbf{p}_2 = \mathbf{s}_2\mathbf{s}_2{}^{\dagger}$. To this end, the remainder subspace $\mathbf{u}_3$ can be defined as the projection of $\mathbf{U}$ onto the complements of the column spaces of the vectors in $\mathbf{s}_1$ and $\mathbf{s}_2$. This can be written as $\mathbf{u}_3 = [\mathbf{I}-\mathbf{p}_2][\mathbf{I}-\mathbf{p}_1]\mathbf{U}$. One can readily assume that this projection contains signal portions that have neither horizontal nor vertical components. Besides, it is clear that the tree subspaces add up to $\mathbf{U}$.

According to the above discussion, a signal $\mathbf{x} \in \mathbf{R}^n$ can be decomposed into three subspace structural components $\mathbf{x}_1{\in}\mathbf{u}_1$, $\mathbf{x}_2{\in}\mathbf{u}_2$ and $\mathbf{x}_3{\in}\mathbf{u}_3$. The same projection operators can be used for this decompositions as $\mathbf{x}_1{\approx}\phi_1\mathbf{x}$, $\mathbf{x}_2{\approx}\phi_2\mathbf{x}$ and $\mathbf{x}_3 = \phi_3\mathbf{x}$. If $\mathbf{x}$ is known to be more strongly horizontal, then $\phi_1 = \mathbf{p}_1$, $\phi_2 = \mathbf{p}_2[\mathbf{I}-\mathbf{p}_1]$ and $\phi_3 = [\mathbf{I}-\mathbf{p}_2][\mathbf{I}-\mathbf{p}_1]$. On the other hand, if $\mathbf{x}$ has more vertical structures than horizontal, then one may write $\phi_2 = \mathbf{p}_2$, $\phi_1 = \mathbf{p}_1[\mathbf{I} - \mathbf{p}_2]$ and $\phi_3 = [\mathbf{I} - \mathbf{p}_1][\mathbf{I} - \mathbf{p}_2]$. This means that a model selection is required to identify the major directionality a signal has, in order to accordingly arrange the succession of the projections.

To establish the sets of directional patches $\mathbf{s}_1$ and $\mathbf{s}_2$, one may use a directional classifier such as the dominant angle in the gradient operator's phase as proposed in [30] to obtain patches of the desired directionality. Then, a suitable dimensionality reduction operation can be carried out on such patches to extract the sets. For example, K-means clustering can be applied for this purpose. Moreover, directional patches can be extracted from the image of interest and used accordingly to design the projection operators. In this setting, projection operators are specially designed for the image in hand.

To this end, if $\mathbf{X}$ is a set of training signals, a training set for the subspace $\mathbf{u}_1$ can be obtained by projecting $\mathbf{X}$ onto that subsapce. This can be easily achieved by left multiplying $\mathbf{X}$ with the relevant projection operator $\mathbf{p}_1$. Similarly, the training sets of the other two subspaces can be obtained using the corresponding projection operators. This can be written as $\mathbf{X}_1 \approx \mathbf{p}_1 \mathbf{X}$, $\mathbf{X}_2 \approx \mathbf{p}_2 \mathbf{X}$ and $\mathbf{X}_3 = [\mathbf{I} - \mathbf{p}_2][\mathbf{I} - \mathbf{p}_1]\mathbf{X}$. Any standard DL method such as K-SVD or ODL can be used for the training process.

The above mentioned setting defines three directional subspaces $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$. This forms a triplet of subspaces well-suited for the representation of patches with horizontal or vertical directional nature. The remainder subspace $\mathbf{u}_3$ is dedicated for the representation of image components that are neither horizontally nor vertically oriented. In the same manner, the setting can be extended to be well-suited for patches of other orientations. Using similar steps, a diagonally-oriented subspace $\mathbf{u}_4$ can be defined, and along with an anti-diagonal subspace denoted by $\mathbf{u}_5$. Considering these two subspace, a remainder subspace $\mathbf{u}_6$ can be defined to correspond to patched structures that are neither diagonal, nor anti-diagonal structures. It can thus be seen that the second subspace triplet $\mathbf{u}_4$, $\mathbf{u}_5$ and $\mathbf{u}_6$ can be effectively used to sparsely represent patches of diagonal or anti-diagonal dominant directional nature. Subspace training sets for this triplet can be obtained after designing the underlying projection operators $\mathbf{p}_4$ and $\mathbf{p}_5$. This can be written as $\mathbf{X}_4 \approx \mathbf{p}_4\mathbf{X}$, $\mathbf{X}_5 \approx \mathbf{p}_5\mathbf{X}$ and $\mathbf{X}_6 = [\mathbf{I} - \mathbf{p}_5][\mathbf{I} - \mathbf{p}_4]\mathbf{X}$. Obtaining the training data for the aforementioned subspaces is illustrated in Fig. 4.1. The proposed DL strategy is illustrated in Algorithm 5.

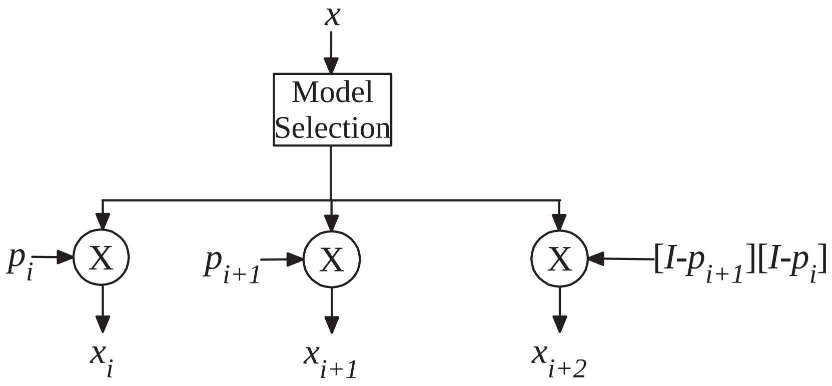


Figure 4.1. Subspace projection of a training signal, i=1 for the first triplet and 4 for the second one.

4.2.2 Component-wise sparse representation

By means of the above mentioned projection operators, a signal $\mathbf{x}$ can be exactly decomposed as the summation of three subspace components. If the dominant directional structure of $\mathbf{x}$ is horizontal or vertical, then it will be decomposed in the subspaces of the first triplet. If it is diagonally or anti-diagonally structured, it will be decomposed using the subspaces of the second triplet. This decomposition can be written as follows.

$$\mathbf{x} = \oplus_{i=j}^{i=j+3} \mathbf{x}_j, \tag{4.1}$$

where $\oplus$ denotes direct sum, and j=1 for the case of the first triplet, and j=2 for the case of the second one. Given that $\mathbf{x}_i \approx \mathbf{D}_i \boldsymbol{\alpha}_i$ gives the sparse approximation of the component $\mathbf{x}_i$ over its subspace dictionary $\mathbf{D}_i$ with the sparse approximation vector $\boldsymbol{\alpha}$, one can write the sparse approximation of $\mathbf{x}$ as follows.

$$\hat{\mathbf{x}} = \sum_{i=j}^{i=j+3} \hat{\mathbf{x}}_j, \tag{4.2}$$

where j=1 for the case of the first triplet, and j=2 for the case of the second one.

Algorithm 5 The Proposed Subsapce DL Strategy

```
**INPUT:** Training signal set $\mathbf{X}$.
**OUTPUT:** Subspace dictionaries ($\mathbf{D}_1$ through $\mathbf{D}_6$).
Classify patches in $\mathbf{X}$ (or the test image) into directional clusters.
Apply K-means on the directional clusters and put the centroids as columns of $\mathbf{s}_1$ through $\mathbf{s}_4$.
Calculate projection operators $\mathbf{p}_1$ through $\mathbf{p}_4$.
Project $\mathbf{X}$ onto the six subspaces to find the corresponding training data sets $\mathbf{X}_1$ though $\mathbf{X}_6$.
Learn a compact dictionary for each subspace with its training data set.
```

4.2.3 Representation quality of subspace component-wise sparse approximation

Recalling the signal decomposition in (4.2), it is interesting to trace the successive approximation of a signal $\mathbf{x}$ during sparse approximation. In (4.2), $\mathbf{x}_1 = \mathbf{p}_1\mathbf{x}$, $\mathbf{x}_2 = \mathbf{p}_2\mathbf{x}$ and $\mathbf{x}_3 = [\mathbf{I} - \mathbf{p}_2][\mathbf{I} - \mathbf{p}_1]\mathbf{x}$. Let us use the following notation $\phi_1 = \mathbf{p}_1$, $\phi_2 = \mathbf{p}_2$ and $\phi_3 = [\mathbf{I} - \mathbf{p}_2][\mathbf{I} - \mathbf{p}_1]$. Considering OMP as an example vector selection technique, it successively selects that atom in a dictionary which has the maximum projection onto a so-called residual vector. This residual vector is initialized with $\mathbf{x}$, and is updated with each iteration as the difference between $\mathbf{x}$ and its resulting sparse approximation. OMP first computes a one-atom approximation of $\mathbf{x}$ as $\mathbf{x} = \mathbf{Dw} \oplus \mathbf{r}$, where α is the coefficient corresponding to the first selected atom (denoted by $\mathbf{d}^1$), and $\mathbf{r}$ is the residue which characterizes the error of this one-atom representation. OMP calculates $\mathbf{r}$ as $\mathbf{r} = [\mathbf{I} - \mathbf{d}^1\mathbf{d}^{1^\dagger}]\mathbf{x}$. Intuitively, effective sparse approximation requires minimizing the resulting residue after selecting each atom. Now, effective selection of the atom $\mathbf{d}^1$ should result in minimizing $\mathbf{r}$. To this end, it seems interesting to analyze the residue that remains after one-atom sparse approximation in the standard sparse representation case, and how does it compare with the case of the proposed strategy. Similar to the decomposition of $\mathbf{x}$, one may decompose $\mathbf{r}$ as $\mathbf{r} = \mathbf{r}_1 \oplus \mathbf{r}_2 \oplus \mathbf{r}_3$, where $\mathbf{r}_1 = \phi_1\mathbf{r}$, $\mathbf{r}_2 = \phi_2\mathbf{r}$ and $\mathbf{r}_3 = \phi_3\mathbf{r}$. Following a few algebraic steps, one can write $\mathbf{r}_1 = [\phi_1 - \phi_1\mathbf{d}^1\mathbf{d}^{1^\dagger}]\mathbf{r}$, $\mathbf{r}_2 = [\phi_2 - \phi_2\mathbf{d}^1\mathbf{d}^{1^\dagger}]\mathbf{r}$ and $\mathbf{r}_3 = [\phi_3 - \phi_3\mathbf{d}^1\mathbf{d}^{1^\dagger}]\mathbf{r}$. For the standard sparse coding scenario, it is clear that minimizing $\mathbf{r}_1$ requires the quantity $\mathbf{d}^1\mathbf{d}^{1^\dagger}$ to approximate ϕ_1. On the other hand, minimizing $\mathbf{r}_2$ and $\mathbf{r}_3$ requires $\mathbf{d}^1\mathbf{d}^{1^\dagger} \approx \phi_2$ and $\mathbf{d}^1\mathbf{d}^{1^\dagger} \approx \phi_3$, respectively. It is thus clear that the selected atom $\mathbf{d}^1$ can not minimize the three components of the residual at the same time. Only one component can potentially be minimized. The same argument holds for the other atoms to be selected.

When the same scenario is re-made with the proposed strategy, the difference is that one now selects three atoms at the same time from three different subspace dictionaries. Let us denote the first selected atom in $\mathbf{D}_1$ through $\mathbf{D}_3$ by $\mathbf{d}_1^1$ through $\mathbf{d}_3^1$, respec-

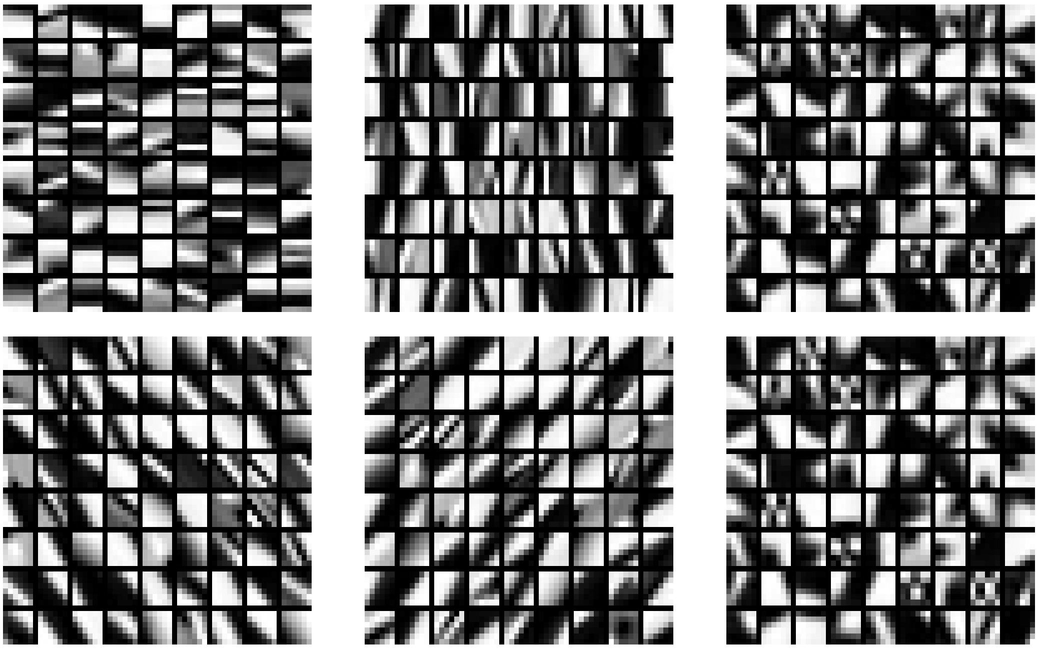


Figure 4.2. Reshaped atoms of triplet dictionaries designed with the proposed strategy. The first/second triplet in the first/second row.

tively. A similar analysis reveals that the requirements for minimizing $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ are $\mathbf{d}_1^1\mathbf{d}_1^{1^\dagger} \approx \phi_1$, $\mathbf{d}_2^1\mathbf{d}_2^{1^\dagger} \approx \phi_2$ and $\mathbf{d}_3^1\mathbf{d}_3^{1^\dagger} \approx \phi_3$, respectively. It is thus possible to minimize these three components of the residue at the same time. In conclusion, the proposed strategy is potentially able to loyally represent different signal structures at the same time, whereas standard DL and sparse approximation is unable to loyally represent them at the same time. This is because loyalty in representing a certain structure causes a certain disability in representing the other one.

4.2.4 Computational complexity reduction

As discussed in Chapter 3, designing multiple compact dictionaries instead of a single highly redundant one is advantageous for reducing the computational complexity of sparse coding, and that of the DL process, as well. In this work, using 3 triplet dictionaries where each has sixth the number of atoms in a single dictionary (K) reduces the computational complexity of OMP to be $\mathcal{O}(\frac{KSN}{2})$. Therefore, the proposed strategy serves for the purpose of reducing the computational complexity of sparse coding.

4.3 Experimental Validation

This experiment aims at examining the representation quality of the proposed strategy. Patches extracted from natural images are reshaped into the vector form and used as training and testing signals. Testing images are divided into fully-overlapping 6×6 patches. These are then sparsely represented. The sparse approximations of image patches are then reshaped to the two-dimensional case and merged to form an image estimate. This estimate is compared to the ground-truth image in terms of PSNR.

Three scenarios of dictionary learning and sparse approximation are considered in this experiment. 40,000 patches are used as the training set for all the cases. In the first scenario S_1, a 36×360 dictionary is learned over the whole training set with ODL. Sparse approximation of a patch is done via OMP with S=4. In the second scenario (S_2), the directional clustering method of Yang *et al.* [30] is used to cluster the training set into five clusters. Five 36×72 cluster dictionaries are learned with ODL. These include directional dictionaries $\mathbf{D}_1$ through $\mathbf{D}_4$ corresponding to orientations of 0°, 45°, 90° and 135° and $\mathbf{D}_5$ as a non-directional dictionary. Sparse approximation is done by first selecting a dictionary for each patch using the same clustering criterion, and then the patch is coded over the selected dictionary using OMP with $S = 4$. In the third scenario (S_3), patches of each test image are used to calculate the corresponding projection operators. Then, the aforementioned directional dictionaries are used as initial dictionaries and updated with the proposed DL strategy, as specified in the previous section. Two dictionary triplets are established, where the remainder dictionary in each tripled is the non-directional dictionary $\mathbf{D}_5$ as it is. Figure 4.2 shows two-dimensional reshaped atoms of the five dictionaries trained with the proposed strategy. It is clearly noted that the dictionaries are directional as their atoms inherit the directional natures of their respective subspaces. Then, sparse approximation of a patch is done according to the proposed strategy, using OMP with $S = 4$. Representation PSNR values of the three scenarios are reported in Table 4.1. In view of Table 4.1, it is clear that standard sparse coding over multiple dictionaries (S_2) has better representation compared to sparse representation over a highly redundant dictionary (S_1) with an average PSNR improvement of 1.41 dB. Besides, the representation quality of the proposed strategy (S_3) is superior to the case of (S_2). The average PSNR improvements of the proposed strategy over standard sparse representation over single and multiple dictionaries are 1.96 dB and 0.55 dB, respectively.

Table 4.1. Image Representation PSNR (dB) with S_1, S_2 and S_3.

Image	S_1	S_2	S_3
Animalroi	36.66	38.06	**38.31**
Barbara	31.45	30.98	**31.81**
Butterfly	31.89	33.84	**34.51**
Fingerprint	33.73	34.45	**35.70**
Leaves	32.42	35.67	**36.11**
Peppers256	34.01	35.02	**35.46**
ppt3	32.91	34.93	**35.28**
Starfish	34.34	35.77	**35.98**
Average	33.43	34.84	**35.39**

Chapter 5

VARIABLE PATCH SIZE SPARSE REPRESENTATION

5.1 Introduction

In the context of sparse representation of images, two-dimensional information are converted into the form of a vector $\mathbf{x}$. Due to computational complexity concerns, it is impractical to reshape the whole image as a single vector. Rather, an image is divided into a set of patches of a small size [34, 52, 53]. Each patch is then reshaped into the vector form, for the purpose of sparse coding. It is quite intuitive to observe that, as compared to an entire image, a small image patch is a simpler, more local entity, and hence can be more accurately represented by means of a smaller number of bases [54]. Therefore, image partitioning into patches allows for feasible dictionary learning and sparse coding with an acceptable level of computational complexity. In the sparse representation literature, a trade-off exists between the patch size and the computational complexity. It is well known that a large patch size is desirable in the sense that it allows access to more of the intrinsic image features. However, because a patch size governs the dictionary atom size, a large size increases the computational complexity of sparse coding and dictionary learning.

It is customary to set the patch size to a specific dimension depending on the performance in the problem considered. Recently, researchers tended to select the patch size to be inversely proportional to the image's spatial frequency. Other attempts aimed at setting a patch size more effectively, e.g., Zhou *et al.* [55] formulate the patch size determination as an optimization problem that minimizes an objective function involving image gradients and features. In a recent work in the sparse representation context, Levin *et al.* [8] address the patch size issue, and indicate the importance of employing a variable patch size for sparse representation, in terms of representation quality and computational complexity.

From other contexts, researchers point out the advantage of employing a variable patch size. As an example, De Smet *et al.* [56] proposed using non-uniformly resized image patch exemplars to find the best patch size and aspect ratio. Additionally, Li at al. [57] report that it is difficult to determine the appropriate patch-size for patch-based abnormality detection in colonoscopic images, and use multi-size patches to simultaneously represent image regions.

In this chapter, we preset a variable patch size sparse representation strategy that is proposed in [58]. The proposed strategy chooses the best representation of each small image region from a set of possible representations of different sizes. This set includes direct representation of this region over a corresponding dictionary and other approximations obtained by extracting them from the representations of larger patches that contain this region. Linear extraction operators are used for this purpose. The most appropriate representation is the one being closest to the true image region. This way, sparse representation error is the patch size selection criterion.

Simulations verify the ability of the proposed strategy to improve the representation quality as compared to the standard case of employing a fixed patch size. For the image representation problem, the proposed strategy is shown to have an average PSNR improvement of 0.99 dB over the standard case for a set of benchmark images. Besides, it is presented as a promising image denoising framework. Simulations show that the proposed strategy, with correct sparse model selection, is competitive to the state-of-the-art denoising algorithms.

5.2 The Proposed Variable Patch Size Sparse Representation Strategy

This work attempts to devise an answer to the fundamental question of: which is better, to sparsely code a small patch over a corresponding dictionary, or to think of it as a portion of a larger patch which is coded over a dictionary of a larger atom size? Figure 5.1 can suggest a justification as to why this question arises. A sample natural image is divided into patches of sizes 4×4 and 8×8. Two cases are then considered. In the first case, each 4×4 image patch is coded over a 16×256 dictionary. In the second case, the representation of each 4×4 patch is extracted as a part of the sparse approximation of the 8×8 patch containing it, as coded over a 64×256 dictionary. In both cases, the true patch is compared to its approximation in terms of the MSE measure. Then, the difference in MSE of the first and the second cases is calculated.

A histogram of the MSE difference is plotted in Fig. 5.1. The histogram points out that this difference is positive for some patches, meaning that the second case is better in representation. However, it is negative for some other patches meaning the contrary.

For a limited number of patches, there is no difference. This result suggests the need for a means of a patch size selection criterion to optimize sparse representation.

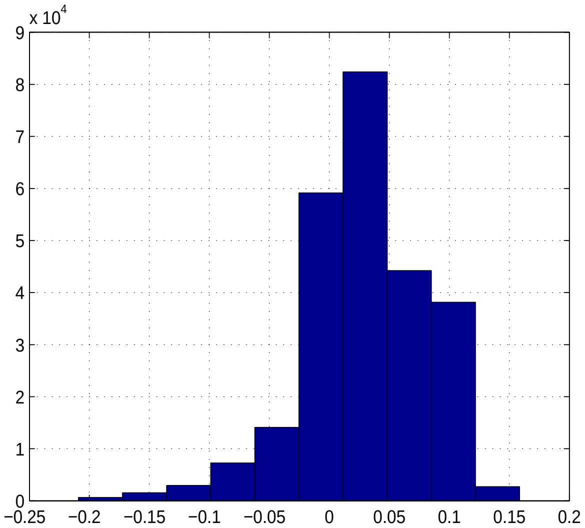


Figure 5.1. Histogram of the MSE difference with the cases of 4×4 and 8×8 patch sizes.

To achieve variable patch size sparse representation, the proposed strategy chooses the most appropriate patch size for each small image region. Figure 5.2 demonstrates the basic idea of this strategy. For illustrative purposes, only three patch sizes are considered. From left to right, large, medium and small-sized patches are shown. The shaded area represents the image region to be sparsely coded. In this setting, three dictionaries are to be used: large, medium and small-sized, corresponding to the three patch sizes. The small shaded square box represents the smallest-size patch of interest. To this end, the sparse coding of this smallest-size patch can be obtained in three ways: directly coding it over the small dictionary, extracting its representation out of the sparse approximation of the medium patch, or extracting it out of the sparse approximation of the large patch. The representation MSE can be used as the patch size selection criterion.

The proposed strategy requires dividing an image into patches with different patch sizes. Each patch size requires a corresponding dictionary. Then, each patch at each size is sparsely coded over its corresponding dictionary. The sparse representation of each of the smallest-size patches can be obtained by directly coding it over the corresponding smallest-size dictionary, or, extracting it from the sparse approximation of each of the larger patches that contain it. Since patches are one-dimensional, linear extraction operators can be conveniently used for this purpose. Patch extraction is carried out by pre-multiplying a larger patch with the suitable extraction operator. At this stage, the MSE between the true patch and its several representations is calculated. In this work, the MSE measure is used to decide on the most appropriate representation of a patch.

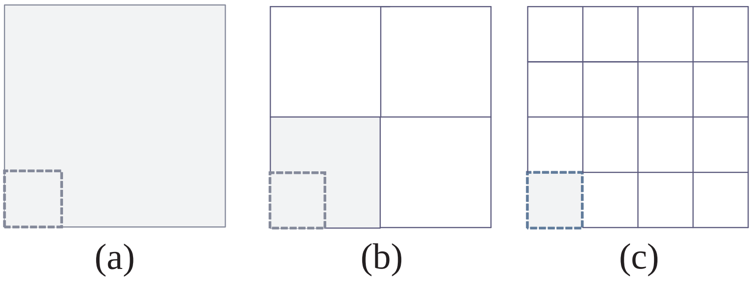


Figure 5.2. Different patch representation possibilities.

Having a set of z possible patch sizes put in an ascending order, the j-th patch size can be labeled with j ranging from 1 to z. The smallest patch size thus corresponds to $j = 1$. In this context, the smallest-size image region is denoted by $\mathbf{x}_1$, and the j-th size patch that contains it is denoted by $\mathbf{x}_j$. One can directly obtain a sparse representation of $\mathbf{x}_1$ over a dictionary $\mathbf{D}_1$, as $\hat{\mathbf{x}_1} \approx \mathbf{D}_1\mathbf{w}_1$. The patch $\mathbf{x}_1$ can be extracted from the j-th size patch using an extraction operator $\mathbf{R}_j$, as $\mathbf{x}_1 = \mathbf{R}_j\mathbf{x}_j$. If $\mathbf{x}_j$ is sparsely represented over a corresponding dictionary $\mathbf{D}_j$ as $\hat{\mathbf{x}_j} \approx \mathbf{D}_j\mathbf{w}_j$, then the sparse approximation of $\mathbf{x}_1$ can be extracted from that of $\mathbf{x}_j$, as $\hat{\mathbf{x}_1} \approx \mathbf{R}_j\hat{\mathbf{x}}_j \approx \mathbf{R}_j\mathbf{D}_j\mathbf{w}_j$. From these representations, the next step is to determine the optimal patch size j_o that minimizes the MSE, between $\mathbf{x}_1$ and its several approximations, as shown in (5.1).

$$j_o = \underset{j}{\operatorname{argmin}}\, MSE(\mathbf{x}_1, \mathbf{R}_j\mathbf{D}_j\mathbf{w}_j),\ j = 1, 2, 3....z, \tag{5.1}$$

where the linear extraction operator at j=1 is the identity matrix. The optimally-sized sparse approximation of the patch $\mathbf{x}_1$, denoted by $\hat{\mathbf{x}}_{1_o}$, can be found as in (5.2).

$$\hat{\mathbf{x}}_{1_o} = \mathbf{R}_{j_o}\mathbf{D}_{j_o}\mathbf{w}_{j_o}, \tag{5.2}$$

where $\mathbf{R}_{j_o}$, $\mathbf{D}_{j_o}$ and $\mathbf{w}_{j_o}$ denote the extraction operator, the dictionary and the sparse coding vector corresponding to j_o, respectively. After finding the most appropriate representation of each smallest-size patch as $\hat{\mathbf{x}}_{1_o}$, these representations are to be spatially concatenated to form an image estimate at the patch size $j = z$. This concatenation is the opposite of the function of the linear extraction operators corresponding to that patch size.

A plurality of sparse representations has recently been shown to be more advantageous as compared to the sparsest one alone [38]. The proposed setting can be thought of as a plurality of representations of each of the smallest-size patches, where the weights of the representations are all zeros expect for the most appropriate patch size which is given a unity weight. This is in fact a binary weighting which can be extended toward adopting more suitable weights assuring better participation of several representations belonging to different patch sizes. This more detailed weighting needs further investigation and can be the scope of another work.

The proposed sparse representation strategy is outlined in Algorithm 6.

Algorithm 6 The Proposed Variable Patch Size Sparse Representation Strategy

```
INPUT: The test image, the set of possible patch sizes and their corresponding dictionaries.
OUTPUT: An array of sparse approximations with variable patch sizes.
Divide the image with several patch sizes.
for each of the smallest-size patches do
    Calculate the MSE of representing the patch over its dictionary.
    Calculate the MSE of representing the patch as a part of each of the larger patches over its dictionary.
    Determine the best patch size $j_o$ as in (5.1).
    Set the representation of this patch as in (5.2).
end for
Concatenate the approximations $\hat{\mathbf{x}}_{1_o}$ to form a patch array of the size $j = z$.
```

5.3 Experimental Validation

In this section, numerical and visual experimentations are presented to investigate the performance of the proposed strategy with the problems of image representation and denoising.

5.3.1 Image Representation

In this experiment, patch sizes of 4×4 and 8×8 and two dictionaries of 16×256 and 64×256 sizes are employed. Dictionaries are trained over a set of natural images using the K-SVD algorithm [28] with 20 iterations and sparsity S=3. These parameter values are typically used in several works.

Table 5.1. Image representation PSNR (dB) Results.

Image	Standard	Proposed
Baboon	25.58	**29.38**
Barbara	30.26	**33.49**
Bridge	25.02	**26.56**
Coastguard	25.76	**26.29**
Comic	22.76	**23.55**
Face	26.34	**26.62**
Flowers	29.20	**30.28**
Foreman	23.94	**24.05**
Lena	27.05	**27.29**
Man	26.94	**27.99**
Monarch	28.93	**29.32**
Pepper	28.07	**28.33**
Ppt3	22.98	**23.31**
Zebra	24.57	**24.82**
Average	26.24	**27.23**

For a set of benchmark images, the problem is to divide an image into fully overlapping patches, sparsely code the patch array, reconstruct it, reshape the reconstructed patches into the two-dimensional form and eventually average overlapping patches to form an image estimate. Then, each estimate is compared to the ground-truth image. Sparse coding herein is done with the OMP algorithm [16] with $S = 3$. For the standard case, an 8×8 patch size and a 64×256 dictionary are used. The proposed strategy divides an image into overlapping patches of 4×4 and 8×8 sizes, and selects the best patch size to code and reconstruct each patch. Size-optimized patches are concatenated to form patches of an 8×8 eventual size. Comparisons are done in terms of the PSNR measure [46].

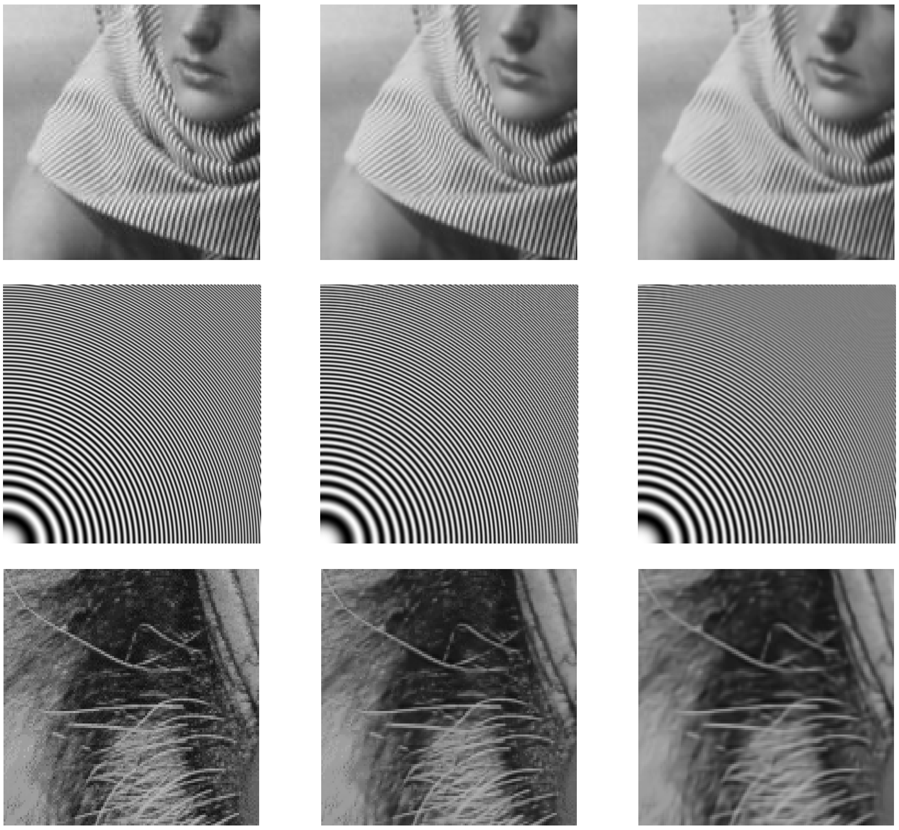


Figure 5.3. From left to right: the original scene, reconstruction with the proposed strategy and reconstruction with standard sparse coding.

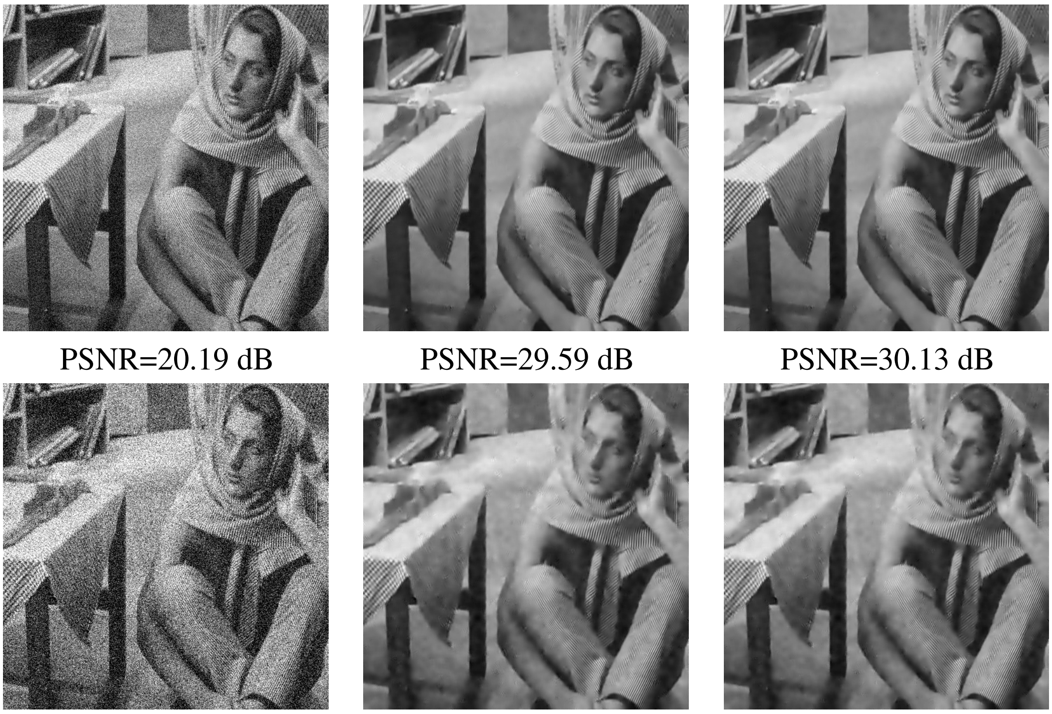
Table 5.1 lists the PSNR values for reconstructions obtained with the standard and the proposed cases, respectively. It shows that the proposed representation strategy is better able to represent images with an average PSNR improvement of 0.99 dB. Besides, it can be concluded that the PSNR improvement is better for images of rich directional and structural contents, e.g., the Baboon and Barbara images. As a visual comparison, Fig. 5.3 shows particular scenes belonging to the Barbara, Zone-Plate and Baboon benchmark images. Each true scene is compared to its reconstructions obtained with the proposed strategy and the standard sparse representation case. It is clearly seen in Fig. 5.3 that the proposed representation strategy is better able to reconstruct image regions with high frequency contents, e.g., the stripes and the curvy lines.

Table 5.2. Denoising PSNR (dB) Results.

Image	Noise Level	Stand.	Prop.	BM3D	Ram *et al.*
Lena	10	35.57	**35.95**	35.93	35.39
	25	31.30	31.70	**32.05**	31.80
	50	27.88	28.07	**28.96**	**28.96**
Barbara	10	34.51	**35.02**	34.93	34.39
	25	29.57	30.13	**30.61**	30.47
	50	25.42	25.66	27.16	**27.35**
Boats	10	33.65	**34.14**	33.94	33.70
	25	29.36	29.82	**29.89**	29.70
	50	25.90	26.20	**26.71**	26.69
House	10	36.07	36.37	**36.63**	35.80
	25	32.08	32.43	**32.79**	32.54
	50	28.05	28.32	29.54	**29.64**
Peppers	10	34.35	**34.77**	34.70	34.26
	25	29.80	**30.37**	30.26	30.01
	50	26.19	26.59	26.69	**26.75**

5.3.2 Image Denoising

The performance of the proposed strategy as a framework for K-SVD image denoising is investigated, and compared to that of K-SVD denoising with a fixed patch size [36]. The proposed strategy serves the same denoising methodology of [36], but with two patch sizes of 4×4 and 8×8. Herein, patch size selection is carried out in terms of the representation error of sparse approximations as compared to the true image patches. This scenario improves the performance compared to the case of using a fixed patch sizes of 8×8 as done in [36]. Adopting the proposed strategy as a means of denoising thus requires developing an effective patch size selection mechanism that uses only the noisy image or any of its features.



PSNR=20.19 dB PSNR=29.59 dB PSNR=30.13 dB

PSNR=14.14 dB PSNR=25.41 dB PSNR=25.65 dB

Figure 5.4. From left to right: the noisy image, K-SVD reconstruction with standard sparse representation and the proposed K-SVD reconstruction.

Figure 5.4 shows the noisy Barbara image and its reconstructions obtained with K-SVD denoising used with standard sparse coding and with the proposed strategy, respectively. Noise variances are 25 and 50 for the images shown the upper and lower rows of Fig. 5.4, respectively. Results show a slight visual improvement obtained when the proposed strategy is applied.

Results reported herein are preliminary, and the development of such a mechanism is left as a future work. The comparison also includes the Block-matching and 3-D filtering (BM3D) algorithm of Dabov *et al.* [59] and the work of Ram *et al.* [60], as they have the state-of-the-art performances. For some benchmark images, PSNR values of the aforementioned algorithms are reported in Table 5.2. Zero-mean additive white Gaussian noise levels of 10, 25 and 50 dB variances are considered.

It can be clearly seen in view of Table 5.2 that the proposed strategy has a good PSNR improvement over the case of employing a fixed patch size. It can also be seen that the results obtained with the proposed strategy are comparable with those obtained by the algorithms of [59] and [60], particularly for the cases of relatively small noise levels.

Chapter 6

A STRATEGY FOR RESIDUAL COMPONENT-BASED MULTIPLE STRUCTURED DICTIONARY LEARNING

6.1 Introduction

In almost all DL algorithms, vector selection plays a crucial role for the sparse approximation stage. Greedy sparse approximation algorithms such as the orthogonal matching pursuit (OMP) [16] are initialized with the original (training or test) signal and each residual component is successively approximated by selecting one atom at a time from a given dictionary. When multiple dictionaries are to be learned, most approaches first cluster data into several clusters and then learn a dictionary for each cluster, using only the cluster data. It is noted that using a signal in a specific cluster to train for the cluster dictionary does not take into account the structure of its residual components. In other words, the residual after each single atom approximation may not necessarily belong to the selected model. Therefore, a dictionary can possibly be updated with a residual component of an irrelevant structure.

In this work, we show that a signal and its residual after one-atom approximation do not necessarily possess the same geometric structure. In the multiple DL setting, this observation calls for a model selection for each residual component. Therefore, a strategy for selecting a model for each residual during the multiple structured DL process is proposed [61]. This allows the residual components of a training signal to update only relevant dictionaries. The advantage of the proposed strategy over the conventional multiple DL algorithms ([31, 33, 30]) is that a given training signal can update atoms from different dictionaries. Compared to the conventional sparse approximation methods based on single or multiple dictionaries, the proposed strategy is shown to significantly improve the sparse representation quality. Simulations over natural images indicate that the proposed strategy using the online dictionary learning (ODL) algorithm [5] based dictionary update improves the standard ODL results by averages of 5.04 dB and 4.71 dB (in terms of PSNR) for the cases of sparsity 2 and 3, respectively. The PSNR improvement over conventional multiple DL based representation is also significant with an average of 0.85 dB and 1.24 dB for sparsity levels of 2 and 3, respectively.

6.2 The proposed multiple structured dictionary learning strategy

Most DL algorithms such as ODL [5] and K-SVD [28] require the use of a vector selection algorithm. Such an algorithm selects a few atoms from a given dictionary **D** to approximate a given vector signal **x**. Considering OMP with sparsity S as an example, it initializes a so-called residual vector $\mathbf{r}_i$ ($i = 0, 1, .., S-1$) as $\mathbf{r}_0 = \mathbf{x}$ and iterates S times to approximate it. In each iteration, one atom is selected for the current residual. The residual after each atom selection is updated. This process is recreated S times.

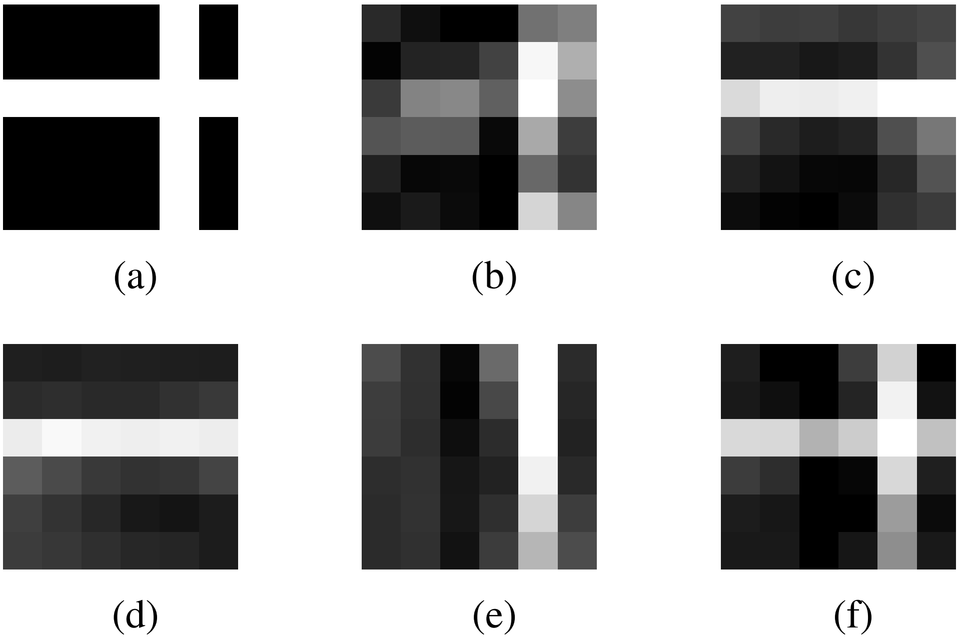
Compared to **x**, the residual may possess a different structure. This fact can not be effectively exploited when one uses a single dictionary. Let us consider the case of multiple structured dictionaries. Such dictionaries can be trained over data which is clustered based on the desired structure. In such a setting, training data in one cluster regardless of their residuals can only update the dictionary of that specific cluster. This practice in the multiple dictionary setting ([31, 33, 30]) ignores the possibility that **x** and each of its residual components may possess different directional structures.

Table 6.1. Orientation of the signal $x = r_0$ and its first residual component r_1.

			$\mathbf{r}_1$				
			C^0	C^{45}	C^{90}	C^{135}	C^{nd}
$\mathbf{r}_0 = \mathbf{x}$	C^0	1667	30.0 %	16.3 %	18.2 %	16.6 %	19.0 %
	C^{45}	2194	16.8 %	20.4 %	22.7 %	17.0 %	23.2 %
	C^{90}	2405	16.7 %	23.4 %	19.8 %	18.4 %	21.8 %
	C^{135}	2134	17.9 %	17.9 %	19.4 %	21.2 %	23.6 %
	C^{nd}	1600	17.8 %	19.6 %	19.4 %	20.0 %	23.3 %

The following experiment empirically validates the above observation. A training set is obtained by randomly sampling 10^5 patches of size 6×6 from the image set [62]. Using the clustering approach in [30], this set is clustered into five clusters. The first four C^0, C^{45}, C^{90} and C^{135} correspond to orientations of 0°, 45°, 90° and 135°, and the fifth cluster C^{nd} is non-directional. Using ODL with S=2, a 36×72 dictionary is learned for each cluster. Cluster dictionaries are denoted by $\mathbf{D}^1$ through $\mathbf{D}^5$, respectively. We then used 10^4 randomly selected test patches to investigate the validity of the above observation.

First, based on the dominant orientation of a test signal $\mathbf{x} = \mathbf{r}_0$, it is sparsely approximated over the corresponding cluster dictionary with sparsity S=1. Then, the first residual component $\mathbf{r}_1$ is calculated and its dominant orientation is determined. The selected cluster for each residual component is identified and the results are presented



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

Figure 6.1. Graphical illustration of reconstructing a patch. (a) Original, (b) Case 1 reconstruction, (c) Case 2 reconstruction , (d) reconstruction of r_0 (S=1), (e) reconstruction of r_1 (S=1), and (f) Case 3 reconstruction (f=d+e).

in Table 6.1. The third column of Table 6.1 shows the number of patches ($\mathbf{x} = \mathbf{r}_0$) in each cluster. The next columns show the distribution of $\mathbf{r}_1$. For example, 1,667 patches of $\mathbf{r}_0$ belong to cluster C^0. Only 30.0 % of these patches have a residual $\mathbf{r}_1$ which also belongs to cluster C^0. The remaining 70.0 % belong to other clusters. A similar conclusion is valid for all clusters.

Let us further motivate this observation with a toy example. Consider the 6×6 patch in Fig. 6.1 (a). It is composed of a horizontal stripe and a vertical stripe. We study three cases. Case 1: sparse approximation with S=2 over a single 36×360 dictionary trained using the dataset of the previous experiment. Case2: sparse approximation using multiple dictionaries ($\mathbf{D}^1$ through $\mathbf{D}^5$ of the previous experiment) with S=2 over the

Algorithm 7 The Proposed Multiple DL Strategy.

```
INPUT: Training set $\mathbf{X} \in \mathbf{R}^{n \times m}$, structured random initializations of dictionaries $\mathbf{D}^1$ to $\mathbf{D}^M \in \mathbf{R}^{n \times K}$, sparsity $S$, number of iterations $Num$.
OUTPUT: A set of structured dictionaries $\mathbf{D}^1$ through $\mathbf{D}^M$.
Initialize $\mathbf{A}^j \in \mathbf{R}^{K \times K} \leftarrow I, \mathbf{B}^j \in \mathbf{R}^{n \times K} \leftarrow \mathbf{D}^j, \forall j \in \{1, 2, ..., M\}$.
for $j$=1 to $Num$ do
  for Each vector $\mathbf{x}$ in $\mathbf{X}$, do
    Initialize $\mathbf{r}_0 \leftarrow \mathbf{x}$.
    for $i = 0, 1, ..., S-1$, do
      Find the projection of $\mathbf{r}_i$ onto every atom in each dictionary.
      Select the dictionary $\mathbf{D}_i^b$ that maximizes this projection. Also note the single-atom sparse representation vector $\mathbf{w}_i$.
      Update $\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{D}_i^b \mathbf{w}_i$.
      Update $\mathbf{A}^b \leftarrow \mathbf{A}^b + \mathbf{w}_i {\mathbf{w}_i}^T$.
      Update $\mathbf{B}^b \leftarrow \mathbf{B}^b + \mathbf{r}_i {\mathbf{w}_i}^T$.
      Using $\mathbf{A}^b$ and $\mathbf{B}^b$, update $\mathbf{D}_i^b$ using Algorithm 2 of [5].
    end for
  end for
end for
```

selected dictionary. This case assumes that the directional structure of the first residual component is the same as that of the original signal. Thus two atoms are selected from the same dictionary. Case 3: sparse approximation using multiple dictionaries (the dictionaries of Case 2) where $\mathbf{r}_0$ and $\mathbf{r}_1$ are allowed to be represented with the most fitting dictionary. The sparse representation of the patch is thus the sum of the sparse (one-atom) representations of $\mathbf{r}_0$ and $\mathbf{r}_1$. This residual component-based representation can be achieved in two ways. A simple and naive way is to join the directional dictionaries into a single dictionary and rely on OMP to select the two most appropriate atoms. The second way is to apply a model selection for each residual component. If model selection is based on maximizing the residual projection, then the two approaches tend to be the same. In this experiment, Case 3 joins the learned dictionaries of Case 2 into a single dictionary, over which sparse representation is done.

Figure 6.1 (b) shows the reconstruction for Case 1, which is a moderate approximation with a mean-squared error (MSE) of 0.1167. The reconstruction with Case 2 is shown in Fig. 6.1 (c). The horizontal stripe is better reconstructed. However, the vertical stripe is almost totally lost. The MSE is 0.1137. This is because $\mathbf{D}^1$ is selected based on the patch directionality, which is well-suited to represent horizontal structures, whereas it severely fails in representing vertical ones. Fig. 6.1 (d) and (e) respectively show the one-atom approximations of $\mathbf{r}_0$ and $\mathbf{r}_1$. Herein, OMP picks one atom from dictionary $\mathbf{D}^1$ and another from $\mathbf{D}^3$ to represent $\mathbf{r}_0$ and $\mathbf{r}_1$, respectively. Fig. 6.1 (f) shows the reconstruction of Case 3 where both signal portions are effectively reconstructed. The MSE of this reconstruction is 0.0396 which is significantly lower.

6.2.1 Residual component-based dictionary learning and sparse representation

The above observations suggest a strategy for learning multiple structured dictionaries based on the structure of not only the original training signal, but the residual components, as well. Given an initial set of M dictionaries, $\mathbf{D}^1$ through $\mathbf{D}^M$, each initialized with randomly selected data of a specific structure, one can use the residual components of a signal in a training set $\mathbf{X}\in\mathbf{R}^{n\times m}$ to train such dictionaries. It is noted that no clustering of the original training set $\mathbf{X}$ is required. One only needs a model selection to determine which structure a given residual component possesses.

Given a training signal $\mathbf{x}$, the residual is initialized as $\mathbf{r}_0 = \mathbf{x}$. The objective is to identify the best fitting dictionary for each residual $\mathbf{r}_i$ and then to use it to update the dictionary. In this work, the model selection for the residual component signals is based on maximum projection. Having M dictionaries, this requires calculating the projection of the residual component signal $\mathbf{r}_i$ onto each atom from all the dictionaries.

The maximum projection determines the atom and thus the dictionary to be updated.

Once the best dictionary $\mathbf{D}_i^b$ corresponding to $\mathbf{r}_i$ is determined, the next residual can be obtained by projecting $\mathbf{r}_i$ onto the orthogonal complement of the selected atom in $\mathbf{D}_i^b$. This can be equvalently written as $\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{D}_i^b\mathbf{w}_i$, where $\mathbf{w}_i$ is a one-atom sparse representation coefficient vector. Then, $\mathbf{r}_i$ and $\mathbf{w}_i$ can be used to update $\mathbf{D}_i^b$. In this work, we chose the update stage of the ODL algorithm [5]. First, the matrices $\mathbf{A}^j$ and $\mathbf{B}^j$ [5] for each dictionary $\mathbf{D}^j$ $(j{\in}\{1, 2, ..., M\})$ are initialized. They are then updated with $\mathbf{r}_i$ and $\mathbf{w}_i$ using $\mathbf{A}^b \leftarrow \mathbf{A}^b + \mathbf{w}_i\mathbf{w}_i^T$ and $\mathbf{B}^b \leftarrow \mathbf{B}^b + \mathbf{r}_i\mathbf{w}_i^T$, where b denotes the selected model and T is the transpose operator. The matrices $\mathbf{A}^j$ and $\mathbf{B}^j$ are then used to update $\mathbf{D}_i^b$ [5].

At this stage, the next residual $\mathbf{r}_{i+1}$ is considered as a new signal. This process is repeated S times. In contrast to the standard multiple DL [31, 33, 30], the proposed DL strategy allows each training signal to potentially contribute to the update of several dictionaries. The proposed DL strategy is outlined in Algorithm 7.

The M dictionaries obtained with the proposed DL strategy can be combined to constitute a single dictionary. A signal $\mathbf{x}$ can be sparsely represented using OMP over the combined dictionary. Another alternative is to perform a model selection process for each residual component. In this setting, a model $\mathbf{D}_i^b$, $(b{\in}\{1, 2, ..., M\})$, is selected for each residual $\mathbf{r}_i$ based on its structure. Then, $\mathbf{r}_i$ is sparsely represented with a single atom picked from the selected model. This gives an approximation to $\mathbf{r}_i$ as $\hat{\mathbf{r}}_i = \mathbf{D}_i^b\mathbf{w}_i$.

Then, the signal **x** can be approximated by summing the sparse approximations of its residuals as $\hat{\mathbf{x}} = \sum_{i=0}^{S-1} \hat{\mathbf{r}}_i$.

6.2.2 Computational complexity of the proposed strategy

For each training signal, standard DL employs one sparse approximation stage with sparsity S and one dictionary update, whereas the proposed strategy employs S one-atom sparse approximations and S dictionary updates. Considering the fact that the dictionary update stage is not computationally demanding [29], we compare the DL computational complexity in terms of the computational complexity of the sparse approximation stage.

The approximate computational cost of the OMP algorithm is $\mathcal{O}(kSn)$ [45]. For the proposed strategy, since dictionaries are of the size K/M, the computational complexity is $\mathcal{O}(\frac{K}{M}Sn)$. However, a model selection is required for each residual. With a simple model selection criterion, the proposed strategy reduces the computational cost. With projection-based model selection, the computational cost of the proposed strategy is $\mathcal{O}(KSn + \frac{KSn}{M})$.

6.2.3 Convergence of the proposed strategy

The convergence of the ODL algorithm is well-established [5]. The proposed residual component-based DL strategy employs single-atom representation for each residual component and ODL-type dictionary update. Thus, the convergence of the proposed strategy is certain. Figure 6.2 shows the signal-to-noise ratio (SNR) convergence behavior of the proposed strategy in comparison to standard ODL. For this experiment, 10^5 patches of the size 6×6 randomly selected from the image database used in [43]

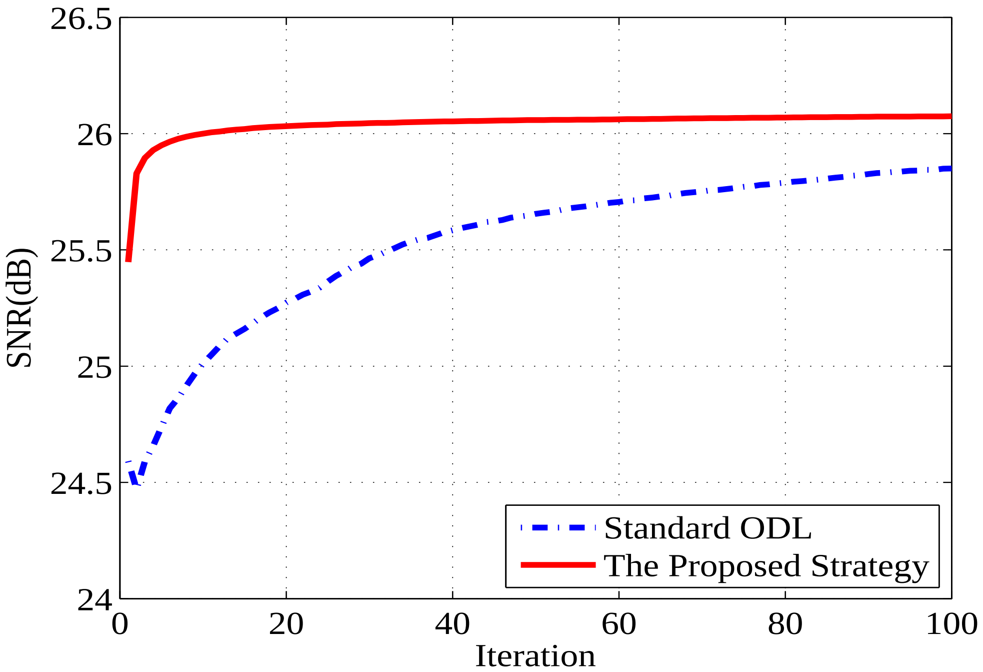


Figure 6.2. SNR convergence of the proposed strategy and standard ODL.

and used as a data set. Sparsity is set to S=3. The preliminary results in Fig. 6.2 show that the proposed strategy converges faster and achieves a higher SNR value in comparison to standard ODL.

6.3 Experimental Validation

The proposed DL strategy is compared to standard DL of single and multiple dictionaries, in experiments of image representation and reconstruction of a known dictionary.

6.3.1 Image representation test

In this test, a test image is divided into overlapping 6×6 patches. Then, the sparse approximations of the patches are obtained and merged using the overlap-add method [3] to form an image estimate. This estimate is then compared to the ground-truth image in terms of the PSNR measure.

We study and compare three scenarios. Each scenario is tested with sparsity S=2 and S=3. In the first Scenario (S_1), a single 36×360 dictionary is initialized with randomly selected data vectors and trained over the training set of the previous experiment using ODL and with 100 iterations. In the representation stage, each patch is sparsely approximated using OMP. In the second scenario (S_2), the same training set is clustered into the five aforementioned clusters using the dominant phase angle measure [30]. For each cluster, a 36×72 dictionary is initialized by randomly selected data vectors and is trained over the cluster data with 100 ODL iterations. It is noted that the number of atoms in each dictionary in S_2 is one fifth of the number of atoms in the dictionary of S_1. In the reconstruction stage, we use maximum projection to determine the best dictionary for each patch. The patch is then sparsely approximated by picking all of the S atoms from the selected dictionary. In the third scenario (S_3), five dictionaries are initialized by randomly selected data from the corresponding clusters in S_2. Then, they are trained over the training set of S_1, with 100 iterations according to Algorithm 1. Then, we combine the five dictionaries into a single dictionary, and perform sparse approximation on it using OMP.

Table 6.2 shows the PSNR values of the three aforementioned scenarios for two-atom (S=2) and three-atom ($S = 3$) approximations. It is clearly seen that the proposed strategy (S_3) is better able to represent image patches as compared to the multiple dictionary-based approximation (S_2) and sparse approximation over a single dictionary (S_1). The average PSNR improvement S_3 has over S_1 and S_2 are, respectively, 4.61 dB and 0.85 dB for the case of $S = 2$, and are 5.02 dB and 1.58 dB for the case of

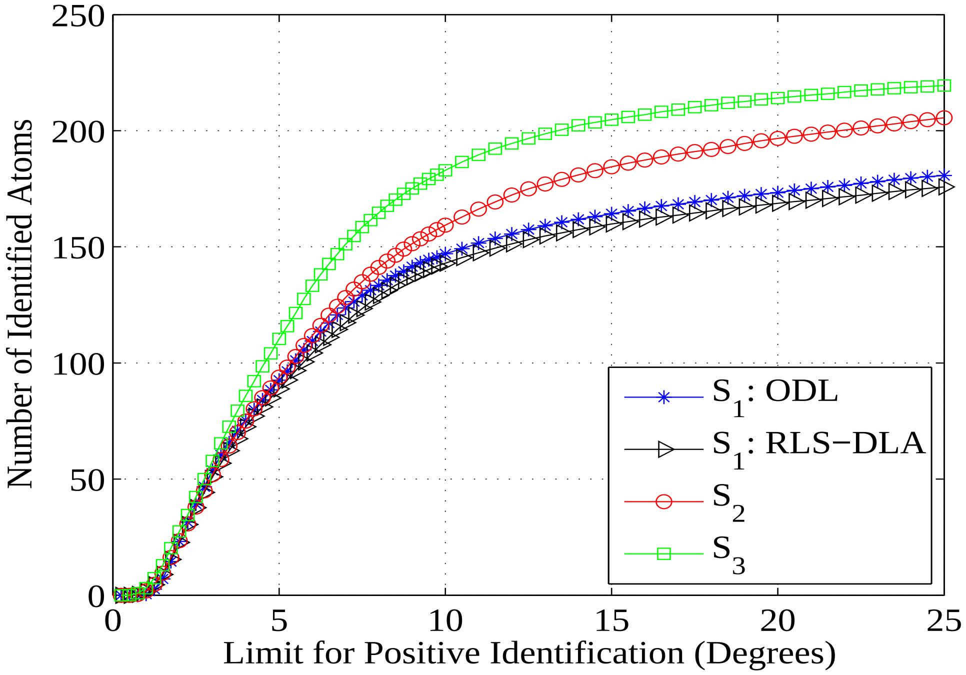


Figure 6.3. Number of identified dictionary atoms versus inner angle tolerance values (in degrees) averaged over 50 trials. SNR= 20 dB.

$S = 3$, respectively. It is noted that the improvement for images which are structurally rich (such as Fingerprint and Raccoon) is more significant.

6.3.2 Reconstruction of a known dictionary

To evaluate the ability of the proposed strategy in reconstructing a known dictionary, the following experiment is conducted. A data set of 10^5 patches is clustered into the aforementioned clusters C^1 through C^{nd}. Then, 50 vectors are randomly selected from each cluster and normalized to unit column norm, to form the structured dictionaries. These dictionaries are combined to form a dictionary $\mathbf{D}_c$ with 250 atoms. A synthetic dataset $\mathbf{X}_s$ of 2000 vectors is generated by superposing 3 atoms randomly selected from $\mathbf{D}_c$ with random weights. Then, additive white Gaussian noise is added such that the SNR is 20 dB. We consider the three DL Scenarios S_1 through S_3. All scenarios use 100 DL iterations with S=3.

For S_1, the dictionary is initialized by randomly selecting 250 vectors from $\mathbf{X}_s$. Then, it is trained over $\mathbf{X}_s$ using ODL. For reference, we also include Scenario S_1 using the recursive least squares dictionary learning algorithm (RLS-DLA) of [29]. For S_2, $\mathbf{X}_s$ is clustered into the aforementioned five clusters [30]. Then, 50 data vectors are randomly selected from each cluster to serve as an initialization for the corresponding dictionary. In S_2, each dictionary is trained over the corresponding cluster data via ODL. S_3 uses $\mathbf{X}_s$ to train for the five dictionaries according to Algorithm 1. In S_2 and S_3, the five trained dictionaries are combined to form a dictionary estimate. For each scenario, $\mathbf{D}_c$ is compared to its estimate in terms of the similarity measure used in [29], which is based on the inner angle between matched atoms. In this setting, an atom in the dictionary estimate is identified if the inner angle between this atom and its match in $\mathbf{D}_c$ is less than a prescribed inner angle tolerance. For a set of inner angle tolerances, Fig. 6.3 shows the number of identified atoms, averaged over 50 trials. When the inner angle between the matched atoms is 10°, the proposed strategy identifies 36 and 40 more atoms compared to standard ODL and RLS-DLA, respectively.

Table 6.2. Image representation PSNR (dB) comparison for Scenarios S_1, S_2 and S_3, with 2 and 3-atom representations.

Sparsity	2-Atom Representation			3-Atom Representation		
Image	S_1	S_2	S_3	S_1	S_2	S_3
Barbara	27.25	30.50	**31.25**	28.69	31.90	**34.06**
Boat	31.12	34.32	**35.24**	32.45	35.54	**37.13**
Butterfly	28.19	32.99	**34.02**	30.04	34.33	**36.15**
Cameraman256	27.62	30.90	**31.63**	28.98	32.05	**33.51**
Comic	26.86	30.10	**30.91**	28.40	31.48	**33.00**
Face	35.25	36.88	**37.41**	36.25	37.66	**38.69**
Fence	28.00	32.87	**33.59**	29.75	34.62	**36.09**
Fingerprint	28.48	32.61	**34.14**	31.09	34.42	**36.58**
Foreman	35.60	40.46	**40.95**	37.93	41.82	**42.80**
Leaves	27.74	33.81	**35.08**	30.15	35.59	**37.81**
Lena	34.44	37.89	**38.66**	36.04	39.15	**40.54**
Man	30.13	32.65	**33.28**	31.35	33.68	**34.89**
Parrot	27.52	31.09	**31.85**	29.09	32.26	**33.63**
Peppers	29.93	34.72	**35.41**	31.54	35.91	**37.12**
Raccoon	31.60	34.33	**35.24**	33.08	35.78	**37.52**
Starfish	30.71	34.47	**35.55**	32.36	36.04	**37.96**
Average	30.03	33.79	**34.64**	31.70	35.14	**36.72**

Chapter 7

IMPROVED DICTIONARY LEARNING BY CONSTRAINED RE-TRAINING OVER RESIDUAL COMPONENTS

7.1 Introduction

Signals contain inherently repetitive structures. It has been shown that the DL process favors repetitive patterns and structures in the learned dictionary. There are several attempts at improving the representation power of a learned dictionary by making a better use of the training set $\mathbf{X}$. These include the work of Mairal *et al.* [5] which replaces each dictionary atom with a rank-1 approximation of the vectors in $\mathbf{X}$ that use it. Another attempt is the stage-wise K-SVD approach of Russo and Dumitrescu [63] where a dictionary is first learned over $\mathbf{X}$ and then updated with vectors in $\mathbf{X}$ that are worst represented by this dictionary. Besides, Zepeda *et al.* [64] addressed the residual spaces of $\mathbf{X}$ subject to greedy pursuit algorithms for sparse representation. They learned dictionaries in residual domains, where each dictionary is adapted to the characteristics of its residual domain. Each dictionary is obtained by applying K-means clustering on the i-th residual of the training set.

In this chapter, we propose a strategy for performing a second pass of dictionary learning. This pass updates a dictionary obtained in a first DL pass using the residuals of the training set subject to sparse representation over this dictionary. However, the representation fidelity of the original training set is imposed as a constraint. The overall update process is formulated as a constrained error minimization problem. This problem is solved using the Lagrange multiplier method. A line-search step is added to optimize the multiplier. Experimental results conducted over natural images validate that dictionaries learned with the proposed strategy have better representation capabilities as compared to dictionaries trained with the standard DL approach. This result is validated in terms of the PSNR measure.

7.2 The proposed dictionary learning strategy

It is well-known that the sparse representation stage is almost common to all DL algorithms [65]. An algorithm such as the orthogonal matching pursuit (OMP) [16] can be applied for this purpose. Therefore, such algorithms differ in the way they update **D**. More precisely, in the way the representation fidelity is imposed. After calculating **W**, the DL process solves for a dictionary **D** that suffices the representation fidelity constraint. This fidelity is addressed by solving for **D** that minimizes the following objective function.

$$\Phi(\mathbf{D}, \mathbf{W}) = ||\mathbf{X} - \mathbf{DW}||_F^2. \tag{7.1}$$

This gives an update equation for **D** after each DL iteration.

Using the same training set, the DL process alternates between the sparse representation and the dictionary update stages for a certain number of iterations, or until a specific stopping criterion is met. Standard DL algorithms use the same training set in the whole DL loop. However, some recent works came to question the effectiveness of using the same training set in the DL process. Along this line, Russo and Dumitrescu [63] proposed updating the dictionary with the data that is worst represented by this dictionary, instead of the whole training set. At this point, it is interesting to notice that given a certain $\mathbf{D}$, a signal $\mathbf{x}$ can be decomposed into two components in terms of its sparse representation over this dictionary, as $\mathbf{x} = \mathbf{x}_z + \mathbf{x}_r$. In this notation $\mathbf{x}_z$ represents the zero-error part of $\mathbf{x}$ that is exactly represented as a sparse approximation over $\mathbf{D}$. This can be written as $\mathbf{x}_z = \mathbf{D}\mathbf{w}$. On the other hand, $\mathbf{x}_r$ represents the residual component of $\mathbf{x}$ which can be obtained as $\mathbf{x}_r = \mathbf{x} - \mathbf{D}\mathbf{w}$. Let us consider the case of performing the DL process on a data set $\mathbf{X}^0$ to obtain a dictionary $\mathbf{D}^0$. Let us further denote by $\mathbf{X}^1$ the matrix of residuals components of the columns in $\mathbf{X}$ as sparsely represented over $\mathbf{D}^0$. Intuitively, the vectors in $\mathbf{X}^1$ characterize the signals that $\mathbf{D}^0$ is unable to represent. Following the same logic, the training set $\mathbf{X}^0$ can be written as $\mathbf{X}^0 = \mathbf{X}_z + \mathbf{X}^1$. A dictionary should be able to represent both components of $\mathbf{X}^0$. It is guaranteed that $\mathbf{D}^0$ is loyal to representing $\mathbf{X}_z$ and disloyal to that of $\mathbf{X}^1$. Therefore, performing a second DL pass on $\mathbf{D}$ over $\mathbf{X}^1$ is expected to improve its representation capability given that it is constrained not to degrade the representation fidelity of $\mathbf{X}^0$. The above-mentioned DL pass can be stated as follows.

$$\begin{aligned} \underset{\mathbf{D},\mathbf{W}^1}{\arg\min} \quad & \|\mathbf{X}^1 - \mathbf{D}\mathbf{W}^1\|_F^2 \\ s.t. \quad & \|\mathbf{w}_i^1\|_0 \leq S \; \forall \; 1 \leq i \leq m \\ & \|\mathbf{X}^0 - \mathbf{D}\mathbf{W}_0^1\|_F^2 \leq \|\mathbf{X}^0 - \mathbf{D}^0\mathbf{W}^0\|_F^2, \end{aligned} \tag{7.2}$$

where $\mathbf{W}^1$ and $\mathbf{W}_0^1$ denote the sparse coding coefficients of $\mathbf{X}^1$ and $\mathbf{X}^0$ over $\mathbf{D}$, respectively and $\mathbf{w}_i^1$ is the i-th column in $\mathbf{W}^1$. Also, $\mathbf{D}^0$ represents a dictionary obtained with a first pass DL process over $\mathbf{X}^0$, and $\mathbf{W}^0$ is the sparse representation of $\mathbf{X}^0$ over this dictionary.

The above optimization is similar to the standard DL problem stated in (1), with imposing that the dictionary update stage will not degrade the representation of the original training set $\mathbf{X}^0$. This means that the problem is still convex in $\mathbf{D}$ and $\mathbf{W}^1$ individually. Therefore, it can be solved by alternating between calculating the coefficients $\mathbf{W}^1$ and updating $\mathbf{D}$. Any sparse representation technique, such as the OMP, can be used for calculating the coefficients $\mathbf{W}^1$. However, the fidelity constraint of $\mathbf{X}^0$ is only imposed during the dictionary update stage. While keeping the coefficients $\mathbf{W}^1$ fixed, the dictionary update stage can thus be viewed as the following inequality-constrained error minimization problem.

$$\begin{array}{ll} \underset{\mathbf{D}}{\arg\min} & \|\mathbf{X}^1 - \mathbf{D}\mathbf{W}^1\|_F^2 \\ s.t. & \|\mathbf{X}^0 - \mathbf{D}\mathbf{W}_0^1\|_F^2 \leq \|\mathbf{X}^0 - D^0\mathbf{W}^0\|_F^2. \end{array} . \tag{7.3}$$

This problem can be efficiently solved using Lagrange multipliers. If the scalar ϵ denotes the term $\|\mathbf{X}^0 - \mathbf{D}^0\mathbf{W}^0\|_F^2$, the Lagrangian of this problem can be expressed as

$$L(\mathbf{D}) = \|\mathbf{X}^1 - \mathbf{D}\mathbf{W}^1\|_F^2 + \gamma[\|\mathbf{X}^0 - \mathbf{D}\mathbf{W}_0^1\|_F^2 - \epsilon], \tag{7.4}$$

where γ denotes the Lagrange multiplier. Following the assumption that the error minimization is convex in $\mathbf{D}$, and noting that $L(\mathbf{D})$ is differentiable with respect to $\mathbf{D}$, its gradient with respect to $\mathbf{D}$ can give its critical points. A minimal point is obtained by equating the gradient of the Lagrangian to zero, as shown in (6).

$$\begin{aligned}\frac{d}{d\mathbf{D}}L(\mathbf{D}) &= \frac{d}{d\mathbf{D}}tr[(\mathbf{X}^1 - \mathbf{D}\mathbf{W}^1)^T(\mathbf{X}^1 - \mathbf{D}\mathbf{W}^1)] \\ &+\gamma\frac{d}{d\mathbf{D}}tr[(\mathbf{X}^0 - \mathbf{D}\mathbf{W}_0^1)^T(\mathbf{X}^0 - \mathbf{D}\mathbf{W}_0^1)] = 0, \qquad (7.5)\end{aligned}$$

where, T denotes the transpose operation. The solution to (6) gives the critical points to the problem in (4). This gives dictionary solution $\mathbf{D}^*$ as

$$\mathbf{D}^* = [\mathbf{X}^1\mathbf{W}^{1^T} + \gamma\mathbf{X}^0\mathbf{W}_0^{1^T}][\mathbf{W}^1\mathbf{W}^{1^T} + \gamma\mathbf{W}_0^1\mathbf{W}_0^{1^T}]^{-1}. \qquad (7.6)$$

In the above formulation, γ is a scalar and therefore its determination is easy. A convenient line-search procedure can be applied. For this purpose, a set of γ values can be used, such that $\mathbf{D}^*$ is calculated for every γ value. Then, $\mathbf{D}^*$ is said to be feasible if it meets the fidelity constraint of $\mathbf{X}^0$, i.e., if $||\mathbf{X}^0 - \mathbf{D}^*\mathbf{W}_0^1||_F^2 \leq \epsilon$. Then, the optimal solution of $\mathbf{D}^*$ amongst the feasible solutions is chosen as the one that best minimizes the objective function $||\mathbf{X}^1 - \mathbf{D}\mathbf{W}^1||_F^2$. The proposed DL strategy is outlined in Algorithm 8.

To analyze the impact of the parameter γ on the DL process, let us compare the dictionary update equation of (7) with that of the standard DL problem. Let us consider

Algorithm 8 The Proposed Residual Retraining DL Strategy

```
INPUT: A training set $\mathbf{X}^0 \in R^{n \times m}$, an initial dictionary $\mathbf{D}^0 \in R^{n \times k}$ trained over $\mathbf{X}^0$,
sparsity $S$, the number of iterations $Num$ and a set of $\gamma$ values.
OUTPUT: $\mathbf{D}$.
Find the sparse representation coefficients of $\mathbf{X}^0$ over $\mathbf{D}^0$ as $\mathbf{W}^0$.
Calculate the residual set as $\mathbf{X}^1 = \mathbf{X}^0 - \mathbf{D}^0\mathbf{W}^0$.          } Initialization
Initialize $\mathbf{D}_0 \leftarrow \mathbf{D}^0$.
while $0 \leq j \leq Num - 1$ do
    Find $\mathbf{W}^1$ as the sparse representation of $\mathbf{X}^1$ over $\mathbf{D}_j$.   } Sparse Approximation
    Find $\mathbf{W}_0^1$ as the sparse representation of $\mathbf{X}^0$ over $\mathbf{D}_j$.
    For each $\gamma$ value calculate $\mathbf{D}$ according to (7.6).
    From the set of feasible $\mathbf{D}$ solutions, select $\mathbf{D}^*$ that best
    minimizes the objective function in (7.3)                                   } Dictionary Update
    Find $\mathbf{W}_0^1$ as the sparse representation of $\mathbf{X}^0$ over $\mathbf{D}_j$.
    Set $\mathbf{D}_{j+1} \leftarrow \mathbf{D}^*$
    Normalize $\mathbf{D}_{j+1}$
    $j \leftarrow j + 1$
end while
```

the method of optimal directions (MOD) [27] as a standard DL algorithm. MOD minimize the objective function in (2) using a pseudo-inverse solution as $\mathbf{D}^* = \mathbf{X}\mathbf{W}^\dagger = \mathbf{X}\mathbf{W}^T[\mathbf{W}\mathbf{W}^T]^{-1}$, where † denotes the Moore-Penrose pseudo inverse. It is clear that setting γ to 0 transforms the update equation of (7) into $\mathbf{D}^* = \mathbf{X}^1\mathbf{W}^{1^T}[\mathbf{W}^1\mathbf{W}^{1^T}]^{-1} = \mathbf{X}^1\mathbf{W}^{1^\dagger}$. This is in fact the standard MOD update equation with the residue as the training set. That is, the proposed dictionary update stage becomes the MOD update stage where the training set is the residual $\mathbf{X}^1$ and no constraint is applied on this update process. However, setting γ to a large value means that $\mathbf{D}^* \approx \mathbf{X}_0^1\mathbf{W}_0^{1^T}[\mathbf{W}_0^1\mathbf{W}_0^{1^T}]^{-1} \approx \mathbf{X}_0^1\mathbf{W}_0^{1^\dagger}$ .This gives more importance to continuing the learning process over the original training set $\mathbf{X}^0$. More precisely, this means disregarding the residue in the dictionary update. In view of this, γ balances the impact of the representation fidelities of $\mathbf{X}^1$ and $\mathbf{X}^0$ on the dictionary update. Besides, the proposed DL stage gives a regularized version of the MOD solution. This regularization comes in order to preserve the representation fidelity of $\mathbf{X}^0$, while optimizing the representation of $\mathbf{X}^1$.

7.3 Experimental Validation

This section presents experiments investigating the performance of dictionaries learned with the proposed strategy, compared to the standard DL approach with the MOD and K-SVD [28] DL algorithms.

7.3.1 MSE and SNR convergence

To compare the convergence of the proposed strategy as compared to standard MOD and K-SVD DL, the following experiment is conducted. A training set $\mathbf{X}^0$ is obtained by randomly sampling 10^3 patches of the size 6×6 from the image set used in [43]. Then, the MOD algorithm is used to train for a 36×72 dictionary over this set, with sparsity S=3 and 100 iterations. This is called a first-pass dictionary $\mathbf{D}_{fp}$. To this end, a residual training set is obtained as $\mathbf{X}^1 = \mathbf{X}^0 - \mathbf{D}_{fp}\mathbf{W}^0$, where $\mathbf{W}^0$ denotes the sparse representation coefficients of $\mathbf{X}^0$ with respect to $\mathbf{D}_{fp}$. Then, two scenarios are considered. In the first scenario, the DL process is continued with the same training set for another 100 MOD iterations. In the second scenario, the proposed strategy is used to update the dictionary $\mathbf{D}_{fp}$ using the residual $\mathbf{X}^1$, as explained in Algorithm 1. Besides, a third scenario is considered where the same training data set is used to train for a 36×72 with K-SVD and the same parameters for 200 iterations.

The proposed strategy determines γ using a line-search procedure. In this search, γ is set to values ranging between 0 and 99. For each γ value, $\mathbf{D}$ is calculated, and the set of feasible $\mathbf{D}$ solutions is established. From this set, the dictionary that best minimizes the objective function in (4) is chosen as the optimal solution. For each DL iteration, the signal-to-noise ratio (SNR) and mean-squared error (MSE) measures are calculated

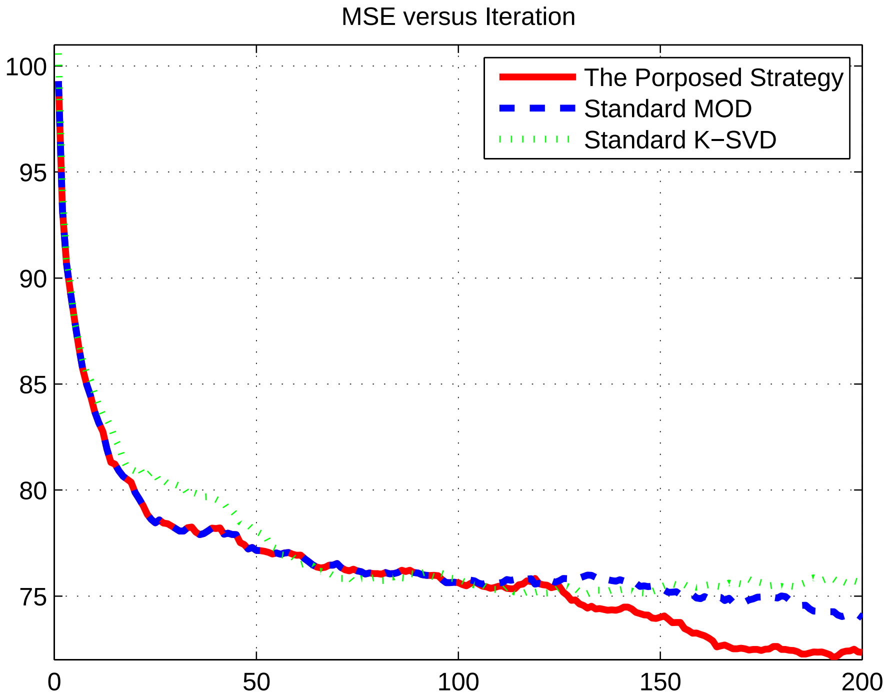


Figure 7.1. Convergence of the MSE between X_0 and its sparse approximation for standard MOD, standard K-SVD and the proposed DL strategy.

between the original training set $\mathbf{X}^0$ and its sparse approximation. SNR is defined as $SNR(\mathbf{X}^0, \hat{\mathbf{X}^0}) = 10\log_{10}\frac{\sum_{i=1}^{M}\|\mathbf{x}_i^0\|_2^2}{\sum_{i=1}^{M}\|\mathbf{x}_i^0 - \hat{\mathbf{x}_i^0}\|_2^2}$. In this notation, $\mathbf{x}_i^0$ denotes the i-th vector in $\mathbf{X}^0$, and $\hat{\mathbf{x}_i^0}$ denotes its sparse approximation. This approximation is calculated as $\hat{\mathbf{x}_i^0} = \mathbf{D}\mathbf{w}_i$. Also, MSE is calculated between $\mathbf{X}^0$ and its sparse approximation $\hat{\mathbf{X}^0}$ over the current dictionary.

Figure 7.1 shows the MSE of representing the training set $\mathbf{X}^0$ using the dictionary in each scenario after each iteration. It is noted that the MSE of the proposed strategy converges to a value lower than that of standard MOD and K-SVD. The SNR performance of the two scenarios is plotted in Fig. 7.2. It is noted that the proposed strategy converges to a higher SNR value as compared to standard DL with MOD and K-SVD.

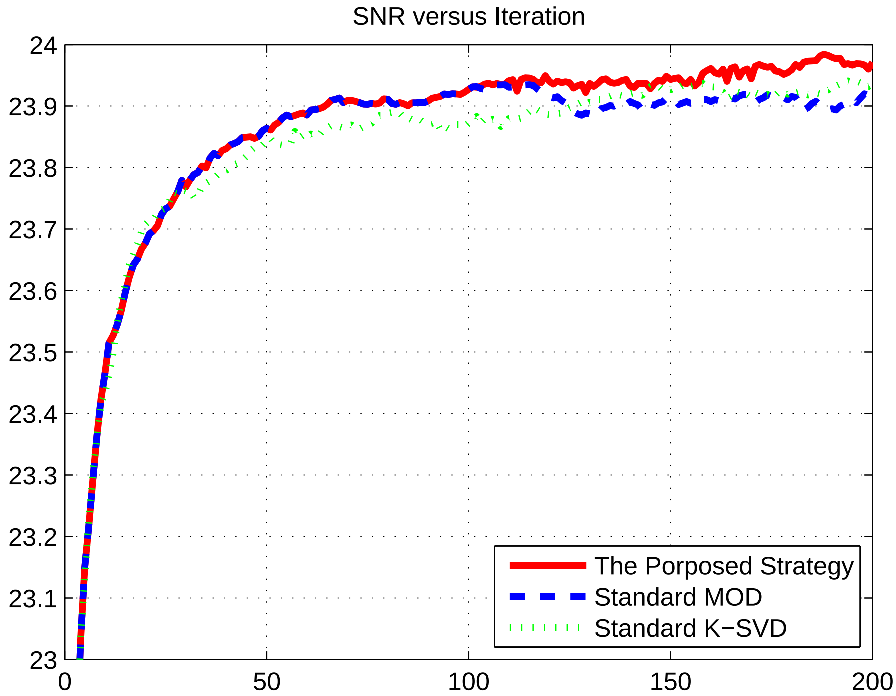


Figure 7.2. SNR Convergence for the cases of standard MOD, standard K-SVD and the proposed DL strategy.

7.3.2 Image Representation

In this experiment, we compare the quality of sparse approximations of image patches over the three dictionaries designed in the first experiment. The objective is to compare each test image to its estimate obtained by merging sparse approximations of its overlapping patches using each dictionary. Each test image is divided into fully-overlapping 6×6 patches. Each patch is then sparsely represented over each dictionary with S=3 to give a patch sparse approximation. Sparse approximations of all patches of an image are reshaped and merged to obtain an image estimate that is compared to the ground-truth one in terms of the peak signal-to-noise ratio (PSNR). PSNR values for some benchmark images are listed in Table 7.1 for the cases of using standard MOD, the proposed strategy and standard K-SVD, repressively. It is noted that sparse approximation over the dictionary designed with the proposed strategy is better than the cases of using standard MOD and K-SVD dictionaries. On average, sparse approximation over the dictionary designed with the proposed strategy has PSNR improvements of 0.46 dB and 0.16 dB over the cases of using standard MOD and K-SVD dictionaries. It is noted that this improvement is stronger for images rich of texture and high frequency components such as the Barbara and Kodak images.

Table 7.1. Representation PSNR (dB) with a dictionary learned with standard MOD, the proposed strategy and standard K-SVD, respectively.

Image	MOD	Proposed DL	K-SVD
Baboon	26.41	26.63	26.42
Barbara	30.29	30.96	30.59
Butterfly	28.25	28.57	28.63
Fence	29.41	30.10	30.03
kodim01	29.34	29.70	29.47
kodim08	27.18	27.73	27.52
Parthenon	30.35	30.75	30.68
TextImage4	19.35	19.83	19.66
Average	27.57	28.03	27.87

These preliminary experiments indicate that the proposed constrained update with the residual signals improves the representation power of learned dictionaries. In this work, the proposed strategy comes as an extension to the MOD algorithm. Further work can be done on extending this idea to be applied to other standard DL algorithms such as K-SVD and other online DL methods such the online dictionary learning algorithm (ODL) [5].

Chapter 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

In this thesis, strategies for sparse representation over multiple structural dictionaries are proposed and investigated. The broad line of this work lies in aiming at improving the representation quality and reducing the computational complexity. Meeting these objectives requires first defining signal classes in order to design compact class-dependent dictionaries. Secondly, it is required to devise suitable sparse representation paradigms to make the best use of the defined class-dependent dictionaries. In this context, a class dictionary is better able to represent signals in its class as compared to a general one. Beside, this allows for designing class dictionaries of compact sizes whereby reducing the computational costs of dictionary learning and sparse representation. More specifically, the conclusions made through this study can be summarized as follows.

- Performing sparse coding on the wavelet domain over wavelet subband dictionaries is shown to serve for the above mentioned purposes. It has been shown that this sparse coding paradigm enhances the representation accuracy. At the same time, the computational complexity of sparse coding and dictionary learning is reduced. This is due to the fact that wavelet subbands contain directionally-structured signals. This allows for designing compact dictionaries, while using a relatively large patch size. Besides, small size training sets are required.

This sparse coding paradigm utilized as the framework for a super-resolution algorithm. Experiments conducted on several natural images point out that this algorithm is competitive to the state-of-the-art super-resolution algorithms and its performance is superior to the case of employing a single highly-redundant dictionary. Moreover, the designed wavelet subband dictionaries are shown to inherit the directional nature of their respective wavelet subbands. Another desirable characteristic of this paradigm is that it does not need a classification or a sparse model selection, as these tasks are done by the wavelet analysis and synthesis filterbanks, respectively.

- The idea of designing directionally-structured dictionaries via subspace projections is also shown to improve the representation quality at reduced computational cost. Projection operators are designed in such a way that they uniformly partition the signal space in a directional sense. Moreover, the projection process naturally separates signal features. This means that there is no need to perform classification to define the signal classes. Besides, the signal as a whole is represented as a composition of the sparse representation of its projections. This means that one needs not to employ a model selection process. In this work, the designed subspace dictionaries are shown to inherit the intended directional nature of their respective subspaces.

- A strategy for determining the optimal patch size is proposed. This is achieved by dividing an image into very small regions. The sparse coding of each region is extracted out of the sparse coding of the path containing this region such that the representation error is minimized. The proposed strategy is shown to improve the representation quality of sparse coding. This is validated with experiments conducted over the problems of image representation and denoising.

- A strategy for residual component-based sparse coding is shown to improve the quality of the representation while using compact dictionaries. This is motivated by the observation that a signal and its residual components may not necessarily be best represented with a certain dictionary. This proposition is empirically investigated. Splitting a signal into remedial component corresponding to a certain vector selection method is shown to allow for a better representation. This observation calls for a dictionary update process that exploits this usage. Experiments conducted over the representation of natural images validate the improvement in representation quality. Besides, the proposed dictionary update stage is shown to enhance the intended structure of the designed dictionaries.
- A strategy for improving standard dictionary learning by performing a second DL pass over the residual components of the training set. The second DL pass is constrained to preserve the representation loyalty for the original training set. A line-search algorithm is used to regularize the two objectives of this DL paradigm. This proposed strategy is shown to improve the representation power of learned dictionaries compared to the standard DL approach.

8.2 Future Work

Here is an account of possible extensions to the findings of this thesis.

- The super-resolution algorithm proposed in Chapter 3 can be further extended along the following directions.

  - The super-resolution algorithm in Chapter 3 assumes that a given low resolution image is the approximation subband of the unknown HR one. However, the validity of such an assumption relies on how closely the wavelet analysis filters can resemble the blurring and downsampling operator. Further investigation can be paid towards designing wavelet filterbanks that can have a better resemblance of that operator. The result of this resemblance is that the proposed super-resolution algorithm will be better applicable to real-life super-resolution problems.
  - one can also extended the work conducted in Chapter 3 by performing dictionary learning in the subband domain of the dual tree complex wavelet transform (DTCWT). DTCWT has been shown as a directionally selective transform, and can therefore serve for the purpose of separating image features based on their directional content.

- The proposed residual component-based DL strategy is shown to be superior to standard DL of a single divisionary and multiple dictionaries. However, it can still be further improved along the following directions.

  - Addressing the DL problem with a fixed representation error, rather than fixed sparsity. This allows for the applicability of the prosopic DL strategy as a framework for image denoising. This is because image denoising via sparse representation is, in essence, based on minimizing sparsity with a fixed sparse representation error.
  - Extending the proposed DL strategy for the case of coupled feature spaces.

With this extension, the prossed DL strategy can be used for the super-resolution application. Besides, the proposed DL strategy in coupled feature spaces can be used as the DL technique for designing wavelet subband dictionaries.

  - The proposed DL strategy starts with a set of initial structured dictionaries. It then updates these dictionaries using residual components of the relevant structure. Dictionaries have been successfully used for the purposes of classification and recognition [66]. In recent literature, the added benefit of employing structured dictionaries for the classification task in terms of label consistency and robustness has been well established [67]. In view of the ability of the prossed strategy to enhance the structure of a set of dictionaries, it seems promising to pursue work towards applying it for the classification task.

- The way projection operators can be investigated with more effort. This includes, for example, designing custom-made projection operators that are specially designed for a given image.

- Applicability of the proposed subspace sparse coding with subspace projections requires investigating the invariance of these operators with respect to a given degradation operator such as scale, noise and other factors. Having projection operators with invariance to the degradation operators means possibility application of the idea in real-life applications.

- Developing a better patch size selection criterion for the variable patch size representation strategy presented in Chapter 5. This means better applicability of this strategy in real-life applications.

- Extending the representation strategies proposed in Chapters 4 and 6 to be applied to coupled feature spaces. This allows for making use of these strategies in the super-resolution application.

REFERENCES

[1] M. Elad, M. Figueiredo, and Yi Ma. “On the Role of Sparse and Redundant Representations in Image Processing”, *Proc. IEEE*, vol. 98, no. 6, pp. 972-982, (2010).

[2] R. Baraniuk, E. Candes, M. Elad, and Yi Ma.“Applications of Sparse Representation and Compressive Sensing”, Proc. IEEE, vol. 98, no. 6, pp. 906-909, (2010).

[3] M. Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer. (2010).

[4] S. Mallat, and Guoshen Yu. “Super-Resolution With Sparse Mixing Estimators”,*IEEE Trans. on Image Process.*, vol. 19, no. 11, pp. 2889-2900, (2010).

[5] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. “Online dictionary learning for sparse coding”. *In Proceedings of the international conference on machine learning*, (2009).

[6] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. “Online learning for matrix factorization and sparse coding”, *Journal of Machine Learning Research*, vol. 11, pp.

19-60, (2010).

[7] G. Yu, G. Sapiro, and S. Mallat. “Solving Inverse Problems With Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity”, *IEEE Trans. on Image Process.*, vol. 21, no. 5, pp. 2481-2499, (2012).

[8] L. Anat, B. Nadler, F. Durand, and W.T. Freeamn. “Patch complexity, finite pixel correlations and optimal denoising”. Computer VisionECCV. Springer Berlin Heidelberg. PP.73-86, (2012).

[9] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde. “Model-Based Compressive Sensing”, *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982-2001, (2010).

[10] Y.C. Eldar, P. Kuppinger, and H. Blcskei. “Compressed sensing of block-sparse signals: Uncertainty relations and efficient recovery”. *arXiv preprint arXiv*, pp. 0906.3173, (2009).

[11] Y.C. Eldar, and M. Mishali. “Robust recovery of signals from a structured union of subspaces”. *Information Theory, IEEE Transactions on*, vol. 55, no. 11, pp. 5302-5316, (2009).

[12] J. Huang, T. Zhang, and D. Metaxas. “Learning with structured sparsity”. *The Journal of Machine Learning Research*, vol. 12, pp. 3371-3412, (2009).

[13] J. Rodolphe, J.Y. Audibert, and F. Bach. “Structured variable selection with sparsity-inducing norms”. *The Journal of Machine Learning Research*, vol.12, pp. 2777-2824, (2011).

[14] S. Mihailo, F. Parvaresh, and B. Hassibi. “On the reconstruction of block-sparse signals with an optimal number of measurements”. *Signal Processing, IEEE Transactions on*, vol. 57, no. 8, pp. 3075-3085, (2009).

[15] S. Mallat, and Z. Zhang. “Matching pursuits with time-frequency dictionaries”, IEEE Trans. Signal Process., vol. 41, no. 12, pp. 3397-3415, (1993).

[16] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition”. In: *Proceedings of the 27th Annu. Asilomar Conf. Signals, Systems, and Computers, Pacific Grove, CA*, Nov 1-3, Vol. vol. 43, no. 1, pp. 40-44, (1993).

[17] D. Needell, and R. Vershynin. “Signal recovery from inaccurate and incomplete measurements via regularized orthogonal matching pursuit”, *Preprint Dec*, (2007).

[18] D. Needell, and J. A. Tropp. “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples”. *Applied and Computational Harmonic Analysis*, vol. 26, pp. 301321, (2008).

[19] W. Dai, and O. Milenkovic. “Subspace pursuit for compressive sensing signal reconstruction”, *Preprint*, (2009).

[20] R. Gagan, and A. Sahoo. “A comparative study of some greedy pursuit algorithms for sparse approximation”.*Third Annual Symposium on Combinatorial Search*. Vol. 19, (2009).

[21] S.S. Chen, D.L. Donoho, and M.A. Saunders. “Atomic decomposition by basis pursuit”. *SIAM J. Sci. Comput*, vol. 20, no. 1, pp. 33-61, (1998).

[22] S. Boyd, and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, (2004).

[23] R. Tibshirani. “Regression shrinkage and selection via the LASSO”. *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 58, no. 1, pp. 267288, (1996).

[24] M.R. Osborne, B. Presnell and B.A. Turlach. “A new approach to variable selection in least squares problems”. IMA Journal of Numerical Analysis, vol. 20, no. 3, pp. 389404, (2000).

[25] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. “Least angle regression”. *The Annals of Statistics*, vol.32, no. 2, pp. 407451, (2004).

[26] R. Rubinstein, A. Bruckstein, and M. Elad. “Dictionaries for sparse representation modeling”. *Proceedings of the IEEE*, vol. 98, no. 6, pp. 10451057, (2010).

[27] K. Engan, S.O. Aase, and J.H. Husoy. “Method of optimal directions for frame design”. *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5, pp. 24432446, (1999).

[28] M. Aharon, M. Elad, and A.M. Bruckstein. “The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representations”. *IEEE Trans. Signal Processing*. vol. 54, no. 11, pp. 4311-4322, (2006).

[29] K. Skretting, and K. Engan. “Recursive least squares dictionary learning algorithm”. *IEEE Trans. on Signal Processing*, vol. 58, no. 4, pp. 21212130, (2010).

[30] S. Yang, M. Wang., Y. Chen, and Y. Sun. “Single-image super-resolution reconstruction via learned geometric dictionaries and clustered sparse coding”. *Image Processing, IEEE Transactions on*, vol. 21, no. 9, pp. 4016-4028, (2012).

[31] W. Dong, L. Zhand, G. Shi, and X. Wu. “Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization”. *IEEE Trans. Image Processing*, vol. 20, no. 7, pp. 1838-1857, (2011).

[32] J. Feng, L. Song, X. Yang, and W. Zhang. “Learning dictionary via subspace segmentation for sparse representation”, *In Proceedings of the ICIP*, pp. 1245-1248, (2011).

[33] G. Yu, G. Sapiro, and S. Mallat. “Image modeling and enhancement via structured sparse model selection”, *In Proceedings of the ICIP*, pp. 1641-1644, (2010).

[34] J. Yang, J. Wright, T. Huang, and Y. Ma. “Image super-resolution as sparse representation of raw image patches”. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), Anchorage, AK,* Jun. 23-28, pp. 1-8, (2008).

[35] J. Yang, J. Wright, T. Huang, and Y. Ma. “Image super-Resolution via sparse representation”. *IEEE Trans. Image Processing*. vol. 19, no. 11, pp. 2861-2873, (2012).

[36] M. Elad, and M. Aharon. “Image denoising via sparse and redundant representations over learned dictionaries”, *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736-3745, (2006).

[37] M. Nazzal, and H. Ozkaramanli. “Wavelet domain dictionary learning-based single image superresolution”. Signal, Image and Video Processing, vol. 9, no. 7, pp. 1491-1501, (2014).

[38] M. Elad, and I. Yavneh. “A Plurality of Sparse Representations Is Better Than the Sparsest One Alone”, IEEE Transactions on Information Theory, vol. 55, no. 10, pp. 4701-4714, (2009).

[39] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, PA, vol. 194, pp. 254-257, (1992).

[40] D.D. Muresan, and T.W. Parks. “Prediction of image detail”. In: *Proceedings of IEEE International Conference on Image Processing ICIP. Vancouver, BC, Canada*, Sept. 10-13, pp. 323-326, (2000).

[41] N. Mueller, L. Yue, and N.D. Minh. “Image interpolation using multiscale geometric representations”. *Electronic Imaging. International Society for Optics and Photonics*, (2007).

[42] A. Temizel. “Image resolution enhancement using wavelet domain hidden Markov tree and coefficient sign estimation”. In: *Proceedings of IEEE International Conference on Image Processing ICIP, San Antonio, TX*, Vol. 5, pp. 381-384, (2007).

[43] R. Zeyde, M. Elad, and M. Protter. “On single image scale-up using sparse representations”. Curves and Surfaces, *Avignon-France*, vol. 6920, pp. 711-730, (2010).

[44] B. Mailh, and M.D Plumbley. “Fixed points of dictionary learning algorithms for sparse representations”. Submitted to IEEE Transactions on Information Theory, (2013).

[45] B. Mailh, R. Gribonval, P. Vandergheynst, and F. Bimbot. “Fast orthogonal sparse approximation algorithms over local dictionaries”. In: *Processdings of Signal Processing (SIGPRO)*. Vol. 91, no. 12, pp. 2822-2835, (2011).

[46] Q. Huynh-Thu, and M. Ghanbari. “Scope of validity of PSNR in image/video quality assessment”, *Electronics Letters*, vol. 44, no. 13, pp. 800-801, (2008).

[47] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P Simoncelli. “Image quality assessment: from error visibility to structural similarity”. *IEEE Trans. Image Process*. vol. 13, no. 4, pp. 600-612, (2004).

[48] A.M. Bruckstein, D.L. Donoho, and M.Elad. “From sparse solutions of systems of equations to sparse modeling of signals and images”. *SIAM Rev*. vol. 51, no. 1, pp. 34-81, (2009).

[49] Kodak Lossless True Color Image Suite:
http://r0k.us/graphics/kodak/
date visited: 20-Jan-2014

[50] W. Dong, L. Zhang, R. Lukac, and G. Shi. “Sparse representation based image interpolation with nonlocal autoregressive modeling”. *IEEE Trans. Image Process*. vol. 22, no. 4, pp. 1382-1394, (2013).

[51] M. Nazzal, and H. Ozkaramanli. “Directionally-Structured Dictionary Learning and Sparse Representation Based on Subspace Projections”, IEEE Signal Processing and Communications Applications Conference (SIU 2015), Malatya, Turkey, (2015).

[52] A. Rangarajan. “Learning matrix space image representations”, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 153-168, (2001).

[53] Y. Jieping. “Generalized low rank approximations of matrices”, *Machine Learning*, vol. 61, no. 1-3, pp. 167-191, (2005).

[54] K. Gurumoorthy, A. Rajwade, A. Banerjee, and A. Rangarajan. “A Method for Compact Image Representation Using Sparse Matrix and Tensor Projections Onto Exemplar Orthonormal Bases”, IEEE Trans. on Image Process., vol. 19, no. 2, pp. 322-334, (2010).

[55] H. Zhou, and J. Zheng. “Adaptive patch size determination for patch-based image completion”, *In Proceedings of the IEEE International Conference on Image Processing*, pp. 421-424, (2010).

[56] v. De Smet, V.P. Namboodiri, and L. Van Gool. “Nonuniform image patch exemplars for low level vision”, *In Proceedings of the IEEE Workshop on Applications of Computer Vision*, pp. 23-30, (2013).

[57] P. Li, K. Luck Chan, and S.M. Krishnan. “Learning a multi-size patch-based hybrid kernel machine ensemble for abnormal region detection in colonoscopic images”, *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 670-675, (2005).

[58] M. Nazzal, and H. Ozkaramanli. “Variable patch size sparse representation over learned dictionaries”, Proceedings of 22nd IEEE Signal Processing and Communications Applications Conference (SIU 2014), Trabzon, Turkey, April (2014)

[59] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. “Image denoising by sparse 3-D transform-domain collaborative filtering”, *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080-2095, (2007).

[60] I. Ram, M. Elad, and I. Cohen. “Image processing using smooth ordering of its patches”, *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2764-2774, (2013).

[61] M. Nazzal, F. Yeganli, and H. Ozkaramanli. “A strategy for residual component-based multiple structured dictionary learning”, *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 2059-2063, (2015).

[62] Flickr Image Data Set
http://see.xidian.edu.cn/faculty/wsdong/wsdong_downloads.htm
Date visited: 1-March-2015

[63] C. Rusu, and B. Dumitrescu. “Stagewise K-SVD to Design Efficient Dictionaries for Sparse Representations”, *IEEE Signal Process. Lett.*, vol. 19, no. 10, pp. 631-634, (2012).

[64] J. Zepeda, C. Guillemot, and E. Kijak. “The iteration-tuned dictionary for sparse representations”. *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pp. 93-98, (2010).

[65] M. Sadeghi, M. Babaie-Zadeh M., and C. Jutten. “A new algorithm for learning overcomplete dictionaries”, in *Proc. of the 21st European Signal Processing Conference (EUSIPCO),* pp. 1-4, (2013).

[66] Z. Jiang, Z. Lin, and L.S. Davis. “Label Consistent K-SVD: Learning a Discriminative Dictionary for Recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651-2664, (2013).

[67] Y. Suo, M. Dao, T. Tran, H. Mousavi, U. Srinivas, and V. Monga. “Group structured dirty dictionary learning for classification”,*IEEE International Conference on (ICIP)*, pp. 150-154. (2014)