1-1-2012

Variable Length K-SVD: A New Dictionary Learning Approach and Multi-Stage OMP Method for Sparse Representation

Mahdi Marsousi
Ryerson University

Recommended Citation

Marsousi, Mahdi, "Variable Length K-SVD: A New Dictionary Learning Approach and Multi-Stage OMP Method for Sparse Representation" (2012). *Theses and dissertations.* Paper 1284.

VARIABLE LENGTH K-SVD: A NEW DICTIONARY LEARNING APPROACH AND MULTI-STAGE OMP METHOD FOR SPARSE REPRESENTATION

by

Mahdi Marsousi
MASc., K. N. Toosi University of Technology, Tehran - Iran, 2008

A thesis
presented to Ryerson University
in partial fulfillment of the
requirements for the degree of

Master of Applied Science

in the program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis or dissertation to other institutions or individuals for the purpose of scholarly research.

Signature

I further authorize Ryerson University to reproduce this thesis or dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Ryerson University requires the signatures of all persons using or photocopying this thesis.

Please sign below, and give address and date.

VARIABLE LENGTH K-SVD: A NEW DICTIONARY LEARNING APPROACH AND MULTI-STAGE OMP METHOD FOR SPARSE REPRESENTATION

Master of Applied Science

2012

Mahdi Marsousi
Electrical and Computer Engineering
Ryerson University

Abstract

The Sparse representation research field and applications have been rapidly growing during the last 15 years. The use of overcomplete dictionaries in sparse representation has gathered extensive attraction. Sparse representation was followed by the concept of adapting dictionaries to the input data (dictionary learning). The K-SVD is a well-known dictionary learning approach and is widely used in different applications. In this thesis, a novel enhancement to the K-SVD algorithm is proposed which creates a learnt dictionary with a specific number of atoms adapted for the input data set. To increase the efficiency of the orthogonal matching pursuit (OMP) method, a new sparse representation method is proposed which applies a multi-stage strategy to reduce computational cost. A new phase included DCT (PI-DCT) dictionary is also proposed which significantly reduces the blocking artifact problem of using the conventional DCT. The accuracy and efficiency of the proposed methods are then compared with recent approaches that demonstrate the promising performance of the methods proposed in this thesis.

Acknowledgements

I would like to extend my sincere appreciation to Professor Javad Alirezaie, my MASc. supervisor in Department of Electrical and Computer Engineering, Ryerson University for helping me to research in my interested area, image and signal processing, and for his advices through my researches and studies.

I would like to express my appreciation to Dr. Paul Babyn for his support and comments on my papers and my research trend.

I am very grateful to my spouse, Ms. Hoda Mofidinasrabadi for accompanying me and for her great emotional support during my study at the Ryerson University.

Also, I would like to thank my parents for their encouragement and support and also, many thanks to my friends at the Ryerson University to help me better achieve my purpose.

Table of Contents

List of Tables

List of Figures

List of Acronyms

DCT: Discrete Cosine Transform

DFB: Directional Filter Bank

i.i.d.: Independent and Identically Distributed

KLT: Karhunen-Loeve Transform

K-SVD: Generalized K-Means, Sigular Value Decomposition

LASSO: Least Absolute Shrinkage and Selection Operator

MAP: Maximum a-Posteriori

MP: Matching Pursuit

MS-OMP: Multi-Stage Orthogonal Matching Pursuit

OMP: Orthogonal MAtching Pursuit

PCA: Principle Component Analysis

PI-DCT: Phase Included Discrete Cosine Transform

PSNR: Peak Signal to Noise Ratio

SAD: Sum of Abs Differences

SCoTLASS: Simplified Component Technique LASSO

STFT: Short Time Fourier Transform

St-OMP: Stagewise Orthogonal Matching Pursuit

SVD: Singular Value Decomposition

TOC: Time of Computation

USE: Uniform Spherical Ensemble

VLK-SVD: Variable Length K-SVD

Chapter 1. Introduction

A digitized signal or image consists of a finite number of samples taken with the Delta function. Although this kind of signal representation is useful to playback the signal or image, more efficient mathematical functions may be used to extract desired characteristics out of the signal. For example in order to have smarter signal representation it is possible to separate noise and signal from one another thereby representing a large signal with only a few coefficients.

The signal representation task requires a set of functions in order to linearly combine them so as to approximate the given signal or image. The set of functions is called as a Dictionary and each function is an Atom. The dictionary is orthogonal if all inner products of different atom pairs are zero. In this case, the inner product of the given signal or image with all dictionary atom provide coefficients of the signal representations. In the other case, the signal representation coefficient is obtained by the product of the signal and the dictionary inverse. The effectiveness of the signal representation with square size dictionaries is limited [1]. The use of over-complete dictionaries has resulted in a more effective signal representations. The number of atoms in an over-completed dictionary is much more than the size of its atoms. The use of over-completed dictionaries leads to have many zero coefficients in the signal representation. In this case, the signal representation is called the Sparse Representation because it has many zero coefficients. The sparse representation engages an optimization algorithm to efficiently reconstruct the input signal by reducing a cost function.

The over-completed dictionary can be created in either analytic-based way or learn-based way. In an analytic-based dictionary, the atoms are created using a stationary function such as cosine function, wavelet function and etc. In contrast, a learn-based dictionary is generated based on a training set. A learning algorithm is derived to take a small subset of the training set and modify it in an iterative process to find an optimal solution. A new method to create dictionaries has been recently emerging which makes a bridge between analytical-based and learn-based dictionaries.

An over-complete dictionary $D \in \mathbb{R}^{n \times K}$ of K atoms $\{d_K\}$. The sparse vector $x \in \mathbb{R}^K$ represents an input signal $y \in \mathbb{R}^n$ with a weighted summation of a few dictionary atoms. The aim is to approximate the input signal $y \approx Dx$ with only a few non-zero coefficients of vector x. This condition maintains the sparsity of the vector x. This idea can be formulated as a minimization problem (1.1) or (1.2), which is the basic concept of the sparse representation theory (Figure 1.1).

$$\arg\min_{x} \|x\|_0 \quad \text{subject to} \quad \|y - Dx\| < \varepsilon \tag{1.1}$$

and,

$$\arg\min_{x,D} \|y - Dx\|_2 \quad \text{subject to} \quad \|x\|_0 < \varepsilon \tag{1.2}$$

where the $\|.\|_0$ is the norm-zero. In the first equation, the subject is to optimize the problem to achieve a smaller error than ε, while having the minimum number of non-zero coefficients in the vector x. In the second equation, the subject is to minimize the error with a specific number of non-zero coefficients in the vector x. The implementation of the second equation is easier.

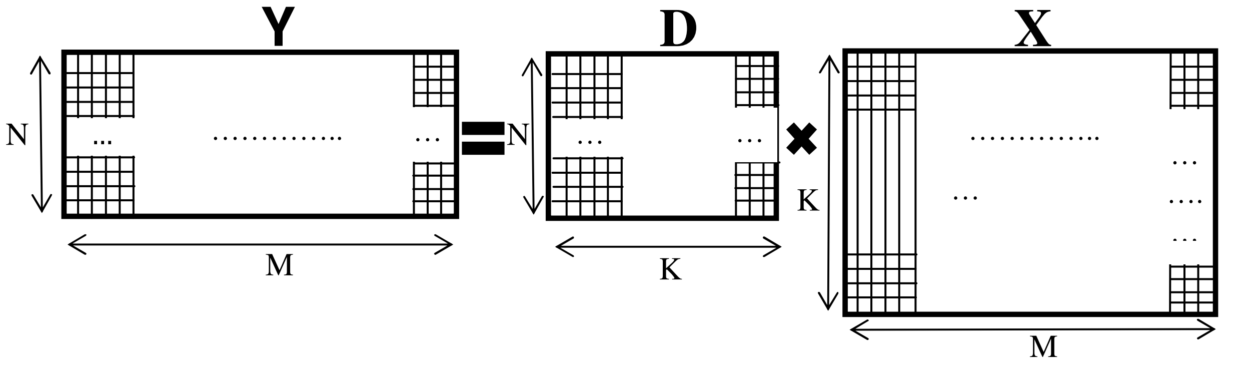


Figure 1.1. In this figure, the sparse representation concept and the size of each matrix are depicted. The Y is the input signal, D is the Dictionary and X is the Sparse Matrix.

The human visual system analyzes image information by taking localized, directional and band-pass features [2]. We are going to present a comprehensive study in order to show how accurately and effectively the three aforementioned visual characteristics are supported by different image representation methods. The important features in the image representation are listed as follows [3]:

1) Multiresolution Representation: It should cover all image information from coarse to fine resolutions.
2) Localized Information: Features should be extracted based on localized information both in spatial and frequency domain.
3) Sampling Window: It determines if the representation takes overlapping sampling windows.
4) Directionality: It specifies whether the representation is able to sense orientation of image data or not.

The first part of this thesis is dedicated to describe a set of existing analytical dictionaries, sparse representation methods and different dictionary learning approaches. In the second part, our contributions to the sparse representation and dictionary learning areas are presented along with comprehensive comparisons with existing methods.

In this thesis, a new analytical dictionary is presented based on the conventional DCT. A set of evenly selected phases between 0 to 2π is involved to create DCT dictionary atoms and is therefore called phase included DCT (PI-DCT) dictionary. It is important to note that the conventional DCT dictionary consists of cosine functions with only different frequencies in vertical and horizontal directions. The conventional way to create DCT atoms is not sufficiently representative for input signal components which have both frequency and phase information. This lack of representation leads to the problem of blocking artifacts. The proposed PI-DCT dictionary addresses this problem by mapping phase information to a specific atom with defined phase and frequency. In other words, for each vertical and horizontal frequency we have a set of atoms with different phases to cover input signal components with non-zero phase information.

This analytical dictionary is followed with a new sparse representation method which applies a multistage approach to the orthogonal matching pursuit (MS-OMP) method. The proposed MS-OMP method selects a set of atoms per each stage whereas the orthogonal matching pursuit (OMP) only adds 1 atom per each iteration. The OMP method calculates a pseudo-inverse for each iteration. Therefore, for T_0 selected atoms, T_0 pseudo-inverse computations are needed. In each stage, it selects M atoms, and thus it needs only T_0 / M stages to opt T_0 atoms. The MS-OMP method performs 1 pseudo-inverse transform for a set of M added atoms. Therefore, the number of pseudo-inverse transforms is reduced in our proposed approach. Similar to the matching pursuit based methods, the proposed approach tries to reduce the signal residual of the previous stage and send it to the next stage. In each stage, the MS-OMP selects a set of m atoms with higher correlations with the residual. Then the conventional matching pursuit (MP) method is applied on the set of selected atoms to opt M descriptive atoms where $M < m$. After selecting new atoms, the sparse vector is updated in the same way that the OMP method performs by computing the pseudo-inverse of the sub-dictionary consisting of all selected atoms in the s^{th} stage. The last step of each stage is formed by updating the signal residual.

The third proposed approach is a novel enhancement to the existing K-SVD dictionary learning method. The conventional K-SVD method iteratively updates a fixed number of atoms, subject to minimizing the reconstruction error. However, it is not clear how many atoms are needed for each input data set. The other problem of using the conventional K-SVD method is that it does not find the best solution to the optimization problem. Instead it minimizes the error until it converges to a local minimum point this is highly dependent on the initial selection of atoms. The proposed novel dictionary learning method starts with only 1 atom and spreads until a convergence is achieved. For example suppose, each atom represents a set of input patches, these represented input patches fall into a high-dimensional volume in space. The size of this volume determines the low-pass characteristic of the representation using this atom. Thus, if the volume is too large, it fails to properly represent details of all involved input patches. In this case, this atom is divided into more atoms in which each one has a smaller volume of involved patches. This operation iteratively repeats for all dictionary atoms until all atoms represent a limited volume in high-dimensional space. This procedure optimally creates atoms to cover all information in the space, and therefore it solves the problem associated with the conventional K-SVD method. Thus, this method is called variable length K-SVD (VLK-SVD) method. In each iteration of this approach, all dictionary atoms are updated one by one (In the same way that the K-SVD approach performs). The procedure of dividing each atom is performed right after the atom updating process. After all atoms are updated in each iteration, insignificant atoms with a small number of representing atoms are removed from the dictionary, this is to maintain the efficiency of the dictionary. After some iterations, the proposed approach converges to a number of atoms which sufficiently cover the input data set. If the input data set contains more details, the VLK-SVD method adds more atoms to cover all input data contents. A comprehensive study is presented in this thesis to draw a relation between the frequency domain information and the number of added atoms using the VLK-SVD approach.

This thesis is organized as follows. In Chapter 2, an introduction to featured analytical dictionaries is presented. In Chapter 3, the sparse representation problem is described and some of its existing solutions are introduced, in detail. Chapter 4, provides an introduction to the dictionary learning problem and two recently proposed approaches are introduced in this chapter. In Chapter 5, the PI-DCT and the MS-OMP method are introduced in detail. In Chapter 6, the VLK-SVD method is presented along with its evaluations and results. Chapter 7 provides conclusion and discussion of the new proposed methods and the possible future works.

Chapter 2. Analytical Dictionaries

2.1. Time-Frequency Dictionaries

Since early research in the area of signal processing, the Fourier transform has been widely focused to extract signal characteristics. The Fourier transform represents a signal as its frequency domain components. It is inferred from the concept that sinusoidal functions are pairwise orthogonal and all signals can be represented as a linear combination of these orthogonal basis. The coefficients of the signal representation are obtained using the inner product of the given signal with the Fourier basis (2.1).

$$x(t) = \int_{-\infty}^{+\infty} X(f) e^{i2\pi ft} df \tag{2.1}$$

The Fourier basis is used to create K atoms of a Time-Frequency dictionary related to low frequencies to approximate signals. Hence, the resultant representation is the smoothed signal and has a noise-reduction effect.

2.1.1. Discrete Cosine Transform

In order to produce non-complex coefficients, the signal is anti-symmetrically extended. This transform is called the Discrete Cosine Transform (DCT) and was introduced by [4], [5] and [6]. A digital signal is decomposed with the DCT as follows [7]:

$$X(m) = \frac{1}{\sqrt{2}} G_x(0) + \sum_{k=1}^{M-1} G_x(k) \cos\frac{(2m+1)k\pi}{2M}, \qquad m = 1, 2, \ldots, M-1 \tag{2.2}$$

where $G_x(k)$ is the k^{th} coefficient. For a specific value of $M = 8$, eigenvectors are calculated as follows:

$$\{\frac{1}{\sqrt{2}}, \cos\frac{(2m+1)k\pi}{16}\}, \quad k = 1, 2, \ldots, 7 \quad \text{and} \quad m = 0, 1, \ldots, 7 \tag{2.3}$$

By expanding the equation (2.2) in the 2D space, this formula is modified as follows [6]:

$$X(m,n) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} G_{xy}(k,l) \cos\frac{(2k+1)u\pi}{2n} \cos\frac{(2y+1)v\pi}{2m} \quad u = 1, \ldots, N-1 \quad v = 1, \ldots, M-1 \tag{2.4}$$

In Figure 2.1, a set of 64 atoms taken from DCT basis for $M = 8$ and $N = 8$ is displayed.

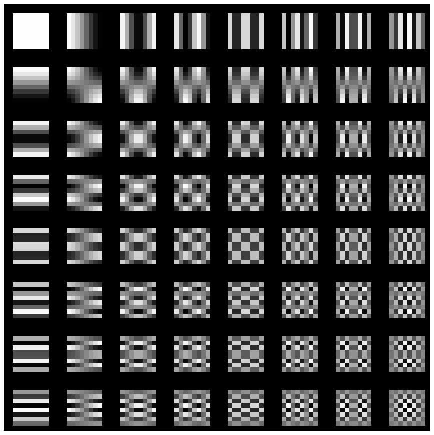


Figure 2.1. Displaying a dictionary of 64 atoms of size 8*8, based on DCT basis. [5]

In the Fourier basis, all of the atoms are created with a similar scale and represent a localized frequency response of a signal. This is referred to the Short Time Fourier Transform (STFT).

2.1.2. Gabor Transform

The other similar Time-Frequency signal decomposition is the Gabor Transform. Indeed, Gabor offers optimal localized windows using the Gaussian function. The Gabor transform is defined by

$$F_{GT}(\omega) = \int_{-\infty}^{\infty} [e^{-j\omega t} f(t)] g_a(t-b)dt \tag{2.5}$$

and,

$$g_a(t) = \frac{1}{2\sqrt{\pi a}} e^{-t^2/4a} \tag{2.6}$$

The Gabor Transform (Figure) localizes the Fourier Transform around $t=b$ where the parameter a determines the window width. Hence, it is inferable to say that the STFT is a generalized form of the Gabor Transform [8]. Due to the size limitation, these introduced basis functions are not adequately able to present the frequency response of the image. In the other words, the limited size of the sampling window contributes to fail these methods to describe the frequency characteristics of larger or smaller structures inside images corresponding to very high and very low frequency components, respectively [9].

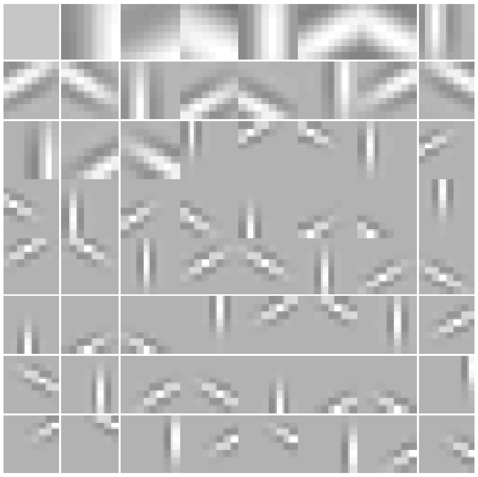


Figure 2.2. Displaying atoms (basis functions) of a typical Gabor Transform [10].

2.1.3. Wavelet Transform

To address the problem of different structure sizes, the Wavelet transform offers a resizable structure for atoms in which the frequency is related to each atom size. In the other words, the flexible time-frequency windows in the wavelet transform provides a non-uniform frequency bandwidth in which the frequency resolution is higher at lower frequencies and vice versa. The wavelet basis functions, called wavelets, are generated by dilation and translation of the basic wavelet $\psi(t)$, as [8],

$$\psi_{ab}(t) = |a|^{-\frac{1}{2}} \psi(\frac{t-b}{a}) \quad . \tag{2.7}$$

Based on the above definition, the wavelet transform is obtained by the product of basis functions using the following Integral form as,

$$F_{CWT}(a,b) = |a|^{-1/2} \int_{-\infty}^{\infty} f(t)\psi^*(\frac{t-b}{a})dt \quad . \tag{2.8}$$

It is shown that if the dilation parameter is $a = 2^{-m}$ and the translation parameter is $b = 2^{-m}k$, the signal $f(t)$ is recoverable using the wavelet series as follows,

$$f(t) = \sum_{m,k \in Z} < f, \psi_{mk}(t) > \psi_{mk}(t) \quad \cdot \tag{2.9}$$

where the $<.,.>$ is the inner product operator and wavelet basis functions, $\{\psi_{mk}\}$, are supposed to be orthonormal [8]. The wavelet transform of a discrete signal is calculated with a combination of multi-layer filter banks combined with the decimation blocks (Figure). This filter bank structure divides the frequency domain of the signal $x(n)$ into 4 analytical parts.

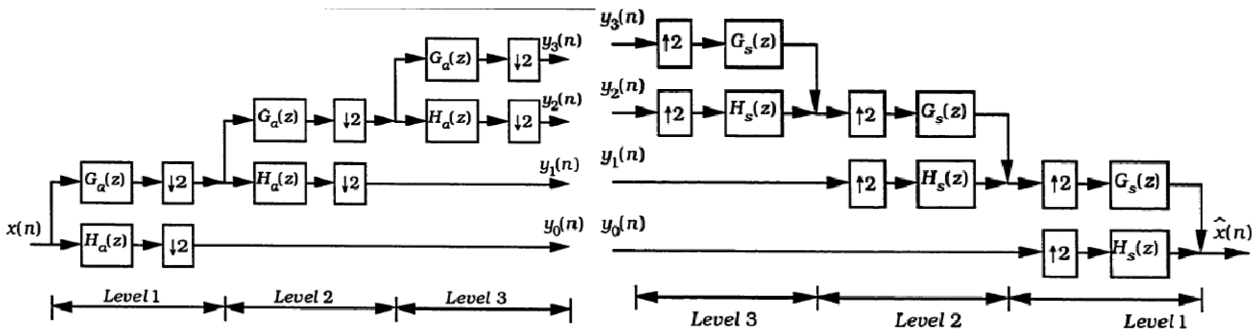


Figure 2.3. Displaying the three-level tree structure of forward (left) and reverse (right) Discrete Wavelet Transform [8]. The $H_a(z)$ block is a high-pass filter which selects the upper-half part of the frequency response of the signal whereas the $G_a(z)$ block selects the lower-half part [8].

In the synthetic phase, 4 parts are combined to constitute the frequency response of the given signal. In Figure , 2 dictionaries are displayed in which their atoms are basis functions of Daubechies and Har wavelets. The JPEG comparison uses the DCT transform whereas the JPEG 2000 is designed based on the wavelet transform.

The success of an image representation stands with its ability to capture visual information using a few descriptions. The wavelet transform perfectly represents 1-D signals. It is sensitive to high-frequency changes while it detects low-frequency terms in the signal. But when the wavelet transform comes to the higher dimensionalities, it fails to perfectly grab all directional information except vertical and horizontal directions. Moreover, the wavelet transform is sensitive to discontinuities in the edge points. But it fails to accurately represent smoothness along the contours in images.

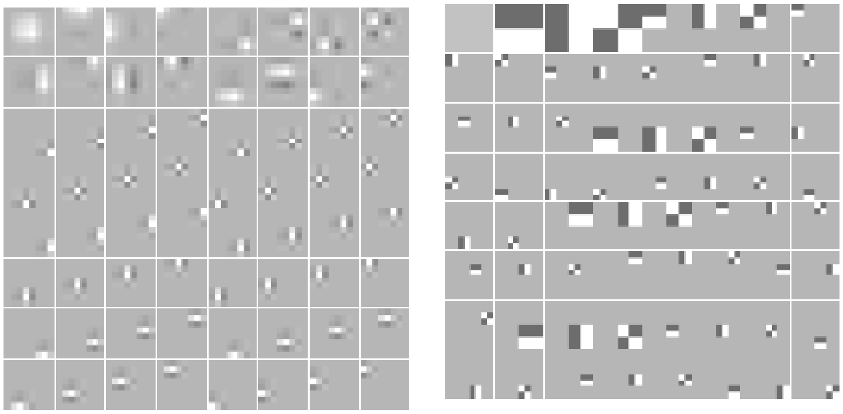


Figure 2.4. Demonstration of atoms (wavelets) of the Wavelet Transform. The left side and right side figures show basis functions of the Daubechies and Har, respectively [10].

2.1.4. Contourlet Transform

To address the abovementioned problem, the Contourlet transform is used to sparsely represent visual directional information in images [3]. If the smooth contour, shown in Figure 2.5, is represented with the Wavelet transform, a large number of square blocks are needed to represent its shape whereas the Contourlet transform easily describes it using 6 directional blocks. Apparently, the efficiency and performance of the Contourlet representation is higher.

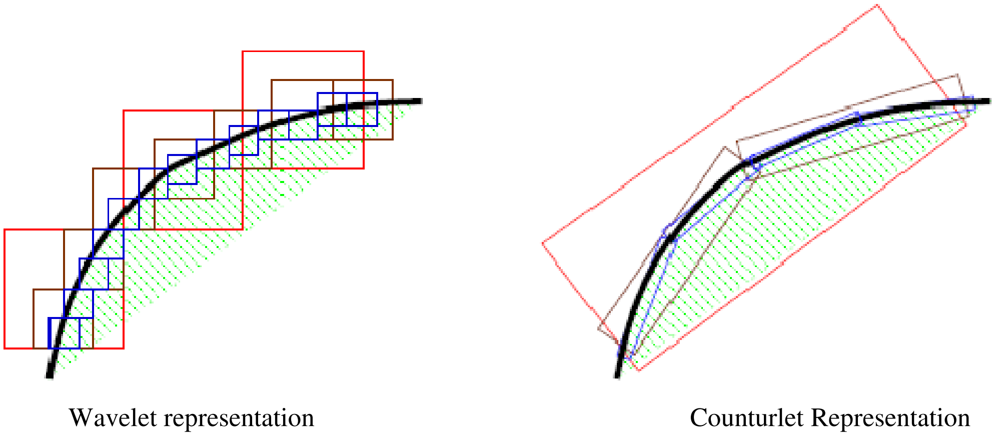
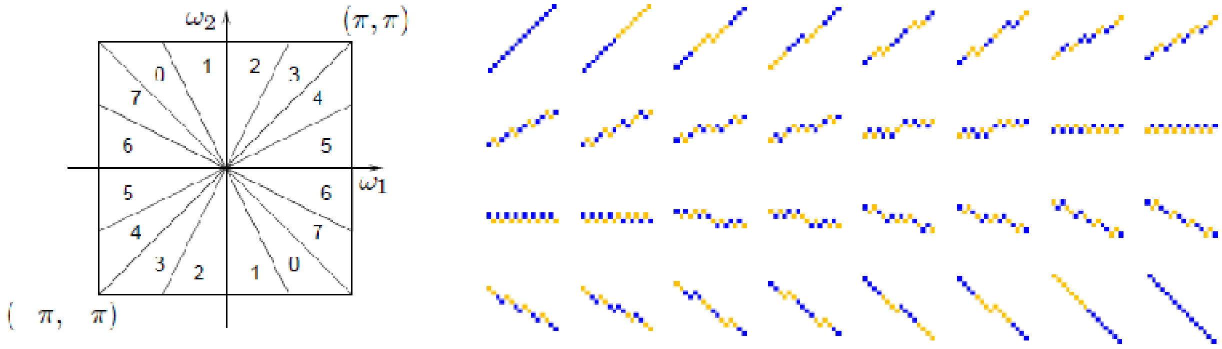


Figure 2.5. This figure illustrates how the Contourlet transform has a dominant performance comparing with the Wavelet transform [3].

As mentioned above, the Contourlet approach defines a filter bank, inspired from the Multiresolution property of the Wavelet Transform, combined with the directional image segmentation using contour model. Hence, this approach is called the directional Multiresolution analysis framework.

In the Contourlet sparse expansion, the first step is to use a multi-scale transform (wavelet-like) for edge detection. In the second step, a local directional transform is applied for contour segment detection. Since the multi-scale and directional transform are decoupled, a variety of combinations can be defined providing flexibility for the Contourlet transform. The directional filter bank (DFB), introduced by [14], is used to derive basis functions of the directional transform by taking the impulse responses of the filter bank. The directional divisions and basis functions are displayed in the Figure .



(a) (b)

Figure 2.6. Displaying the directional filtering concept of the Contourlet method. (a) It shows the directions related to $l=3$ which makes $2^3=8$ directional divisions. (b) It displays 32 basis functions which are generated using the Haar filter [3].

According to the down-sampling of a 1-D signal, $x_d[n]=x[a\cdot n]$ provides samples and the frequency response is $X_d(\omega)=\frac{1}{|a|}X(\frac{\omega}{a})$. For down-sampling of a 2-D signal, a sampling matrix is needed [11] and the resampled signal is obtained by

$$x_d\begin{bmatrix} n'_x \\ n'_y \end{bmatrix}=x[M\cdot\begin{bmatrix} n_x \\ n_y \end{bmatrix}],\ M=\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \tag{2.10}$$

$$X(\begin{bmatrix} \omega'_1 \\ \omega'_2 \end{bmatrix})=\frac{1}{|\det(M)|}X((M^T)^{-1}\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix})$$

where M is the sampling matrix. n_x and n_y refer to samples in the spatial domain while ω_1 and ω_2 represent the 2-D frequency domain. This sampling matrix is a transform which maps the frequency domain area specified by $\{-\pi\le m_{11}\omega_1+m_{21}\omega_2\le\pi\}\cap\{-\pi\le m_{12}\omega_1+m_{22}\omega_2\le\pi\}$ into $\{-\pi\le\omega_1\le\pi\}\cap\{-\pi\le\omega_2\le\pi\}$ [12]. According to this definition, directional frequency domain signal decomposition is implemented which divides the frequency domain into two hour-glass-shaped spectral regions (2.7). The resampling matrix is a combination of the scaling and rotation matrix (-45 degree) as,

$$M = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad . \tag{2.11}$$

Using this transform the input image is divided into two images in which the first one and second one are formed with the spectral regions shown in (Figure 2.7.a) and (Figure 2.72.7.b), respectively. In order to attain wedged-shaped frequency domain decomposition, each split is divided into finer oriental decompositions, refer to Figure .a.

The proposed FDB structure by [12] has a great problem in which its oriental decompositions are distorted and repositioned in sub-bands images (Frequency Scrambling). Therefore, this interesting method has not been widely used until an improved method was proposed by [13].

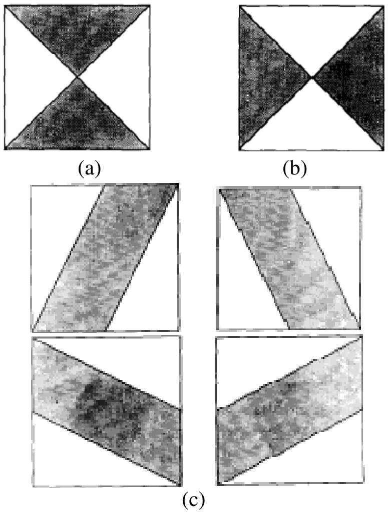


Figure 2.7. (a) and (b) display the selected spectral regions of two directional Hour-Glass-Shaped filters. (c) shows the frequency response of the parallelogram filters [12] for finer oriental decompositions.

Before introducing the improved DFB method, we need to introduce different sampling matrixes as follows [13]:

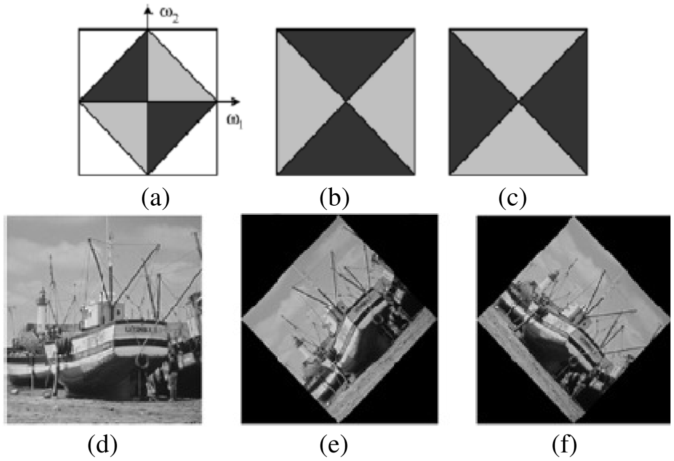
1) Resampling Matrix: 2×2 integer matrix with non-zero determinant.
2) Diagonal Sampling Matrix: a resampling matrix in which its principal diagonal values are in the order of 2^n and 2 other elements are zero.

3) Generalized Quincunx Sampling Matrix: a sampling matrix with ± 1 elements with determinant 2.

$$q_1 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \text{ and } q_2 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \tag{2.12}$$

The q_1 and q_2 are the most common quincunx matrixes and they will be used to implement directional filters (Figure 2.8).

4) Unimodular Matrix: a resampling matrix in which its determinant is ± 1 with a Unimodular inverse.



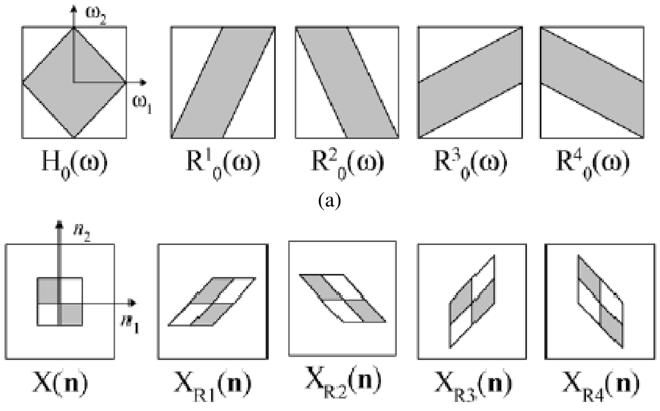
(a) (b) (c)

(d) (e) (f)

Figure 2.8. (a) shows the Fourier transform of the input, (b) shows the down-sampled input with q_1, (c) shows the down-sampled input with q_2. (d), (e) and (f) show a typical image rotated using q_1 and q_2. [13]

5) Diamond-Conversion Matrix: it is a Unimodular matrix, $\{R_i, i = 1,2,3,4\}$, which can be used with the $H_0(\omega)$ (band-pass filters described in the Wavelet transform) to create four parallelogram band-pass filters, $R_0^i(\omega)$ (Figure 2.9).

$$\begin{matrix} R_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} & R_2 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \\ R_3 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} & R_4 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \end{matrix} \tag{2.13}$$



(a)

(b)

Figure 2.9. (a) Diamond shaped pass-band and four parallelogram pass-bands by R_i. (b) An input image and its resampled output with R_i matrixes [13].

A simple two-band DFB is designed using the quincunx sampling matrixes and the diamond filters $H_0(\omega)$ and $H_1(\omega)$ (Figure).

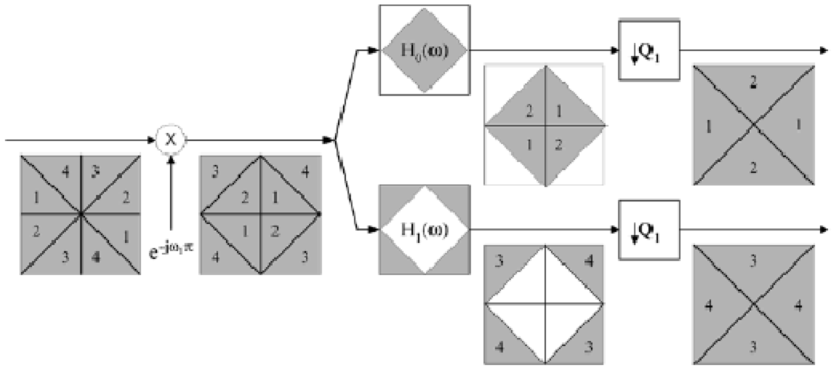


Figure 2.10. Displaying the two-band DFB image decomposition. Oriental frequency components are divided into two down-sampled images [13].

It is easy to create a four-band directional decomposition by cascading the same module on the two outputs obtained with two-band directional decomposition. In order to implement an eight band DFB, the four diamond-conversion matrixes (2.13) are applied to outputs of four-band directional decomposed images (Figure 3).

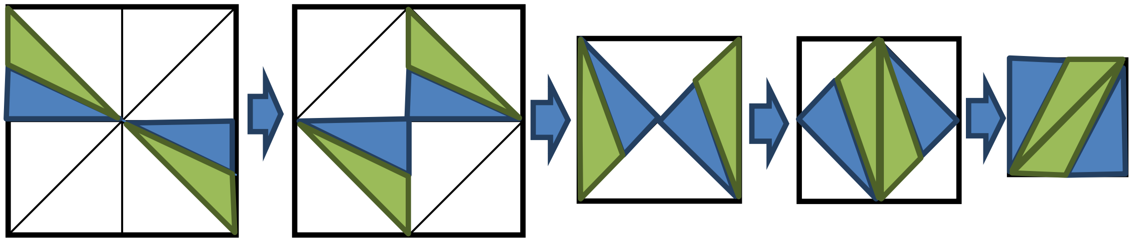


Figure 3. Showing the sequence of the 2 layers of a two-band directional decomposition. This image shows why a diamond-conversion matrix is needed to decompose green (lighter) and blue (darker) regions in the right side resultant image for further decomposition into eight-bands.

The tree-structure used for eight band decomposition can be replaced with a simpler form which consists of a converted filter, $H_i^3(\omega)$, and a single down-sampler matrix, D_i^3,

$$D_i^3 = q_1 \cdot q_1 \cdot R_1 \cdot q_1 = \begin{bmatrix} -2 & 2 \\ 0 & -4 \end{bmatrix} \quad . \tag{2.14}$$

This down-sampler matrix is a non-diagonal matrix, therefore it results in unwanted geometrical transformations in the frequency domain, and this is not desired. To get rid of this problem, a Unimodular matrix is used to convert the overall matrix to a diagonal matrix. The post-sampling matrix, B_i^3, is applied which the $S_i^3 = D_i^3 \cdot B_i^3$ provides a proper decimation.

$$S_i^l = \begin{cases} diag(2^{l-1}, 2) & 0 \le i \le 2^{l-1} \\ diag(2, 2^{l-1}) & 2^{l-1} \le i \le 2^l \end{cases} \tag{2.15}$$

The impulse response of the equivalent synthesis filter is resampled and the result provides basis functions which span all directions.

$$\left\{ d_i^l \left[n - S_i^l m \right] \right\}_{0 \le i \le 2^l, m \in \mathbb{Z}^2} \tag{2.16}$$

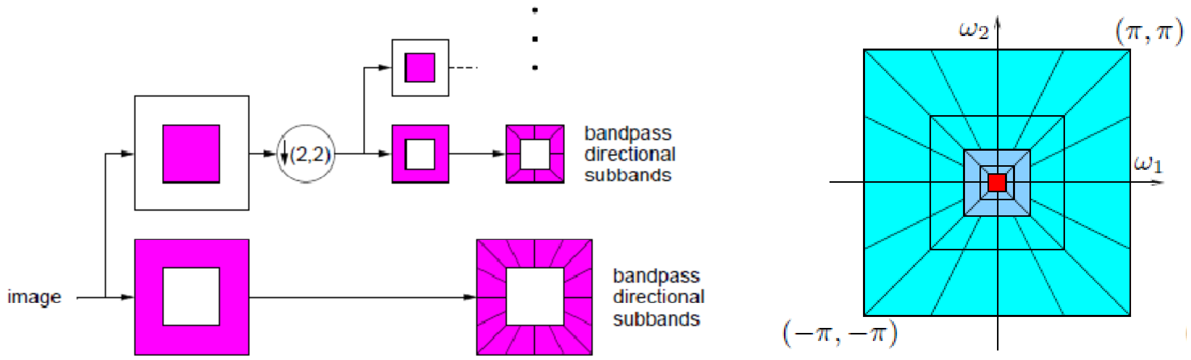
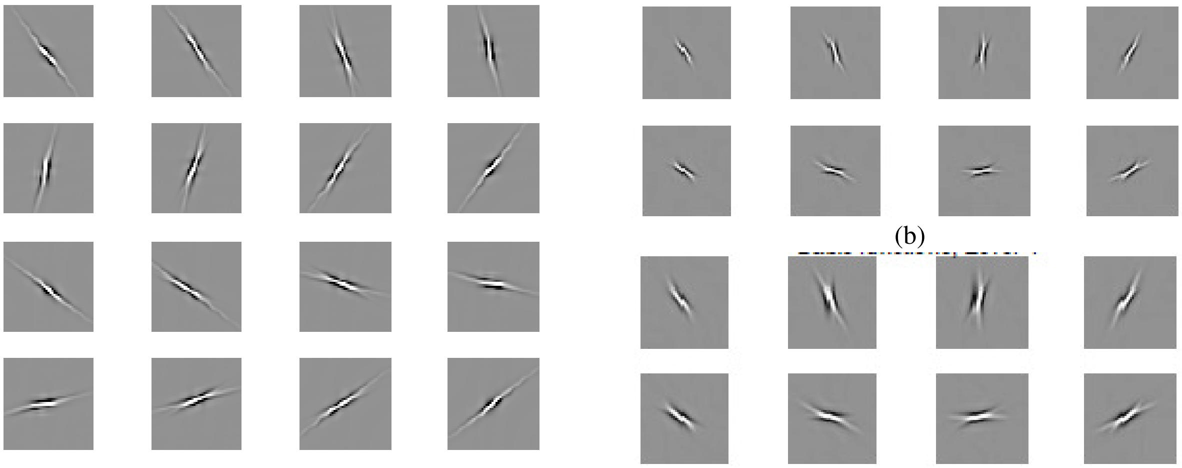


Figure 2.124. Displaying the multi-scare directional decomposition. In figure (a) the concept of multi-scale directional filtering is displayed as a two steps process. In figure (b) the Contour packets are specified for a particular selection of layers and directional divisions [3].



(b)

(a) (c)

Figure 2.13. Displaying Contourlet atoms [14]. (a) basis functions of level 2, (b) level 3 and (c) level 4.

Now, the multi-scale decomposition is combined with the directional filtering. The block diagram of the described approach is displayed in Figure 2.124 and the result is shown in Figure. Band-passed images are fed into the FDB and repeated to the coarse resolution by iteratively decimating the filtered image.

2.2. Karhunen-Loeve Transform

In 1977, a new statistical approach for restoring images degraded by Gaussian noise was proposed by [15], based on the Karhunen-Loeve Transform (KLT). The KLT is a linear Transform which provides a statistical tool to adapt the signal representation based on the signal data. Its basis functions are eigenvectors of the covariance matrix which are uncorrelated and maintains the maximum compression. This was the first step to use the signal's data to create basis functions. The KLT dictionary atoms are the first K eigenvectors of the data covariance matrix. In Figure 2.14, a close relation between the DCT atoms and KLT atoms is displayed. The advantage of the KLT transform is to provide an adapted representation with the given signal whereas its complexity is higher than the DCT transform.

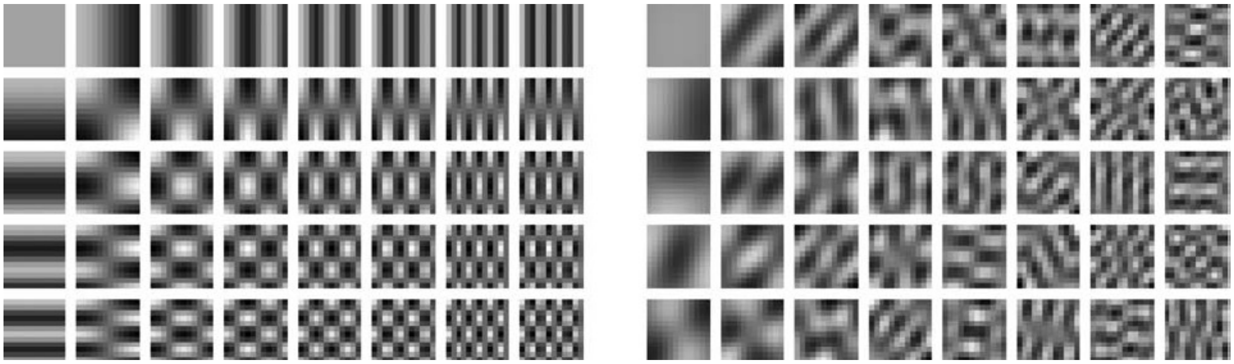


Figure 2.14. Displaying the under-completed DCT dictionary atoms (left) versus the KLT atoms.

2.3. Principal Component Analysis

The Principle Component Analysis (PCA) is an unsupervised method to reduce the dimensionality of a data set with a large number of interrelated variables. In essence, the idea behind the PCA is similar to the KLT transform in which they both are built based on the signal data. The PCA method transforms the space represented by the current variables into a new set of variables which are uncorrelated. Only the most uncorrelated variables are kept to reduce the dimensionality while maintaining the data set variation [16]. The PCA can be obtained using the Singular Value Decomposition (SVD) of the Data Matrix, or using eigenvalue decomposition of the data covariance matrix [17].

Suppose X^T is data with a zero empirical mean value (mean of columns are zero). The singular value decomposition of X is $X = UDV^T$ in which U is a $m \times m$ matrix of eigenvectors of XX^T, D is a $m \times n$ rectangular diagonal matrix and V is a $n \times n$ matrix of eigenvectors of $X^T X$. The principal components are columns Z_i of $Z = UD$ with the variance of D_{ii}^2 / n. V is the corresponding loading vector of PCs. According to the concept of the PCA, the data can be approximated having only $q\{q \ll \min(n, p)\}$ principal components and it is the clue to reduce the data dimensionality. In addition, principal components are uncorrelated which provides a better analysis on the data with converting the data into separable modes.

According to the advantages of using the PCA, a large number of applications have been derived based on this method (Figure) such as handwritten character recognition [18], human face recognition [19], gene expression data analysis [20] and etc.

The basic PCA defines the principal components as a linear combination of all the original variables which have non-zero values. This problem is known as the loading problem and results in a difficult interpretation of the results [21]. A very simple way to address this problem is to define a threshold to set loadings below the threshold equal to zero, and is called *Simple Thresholding*.

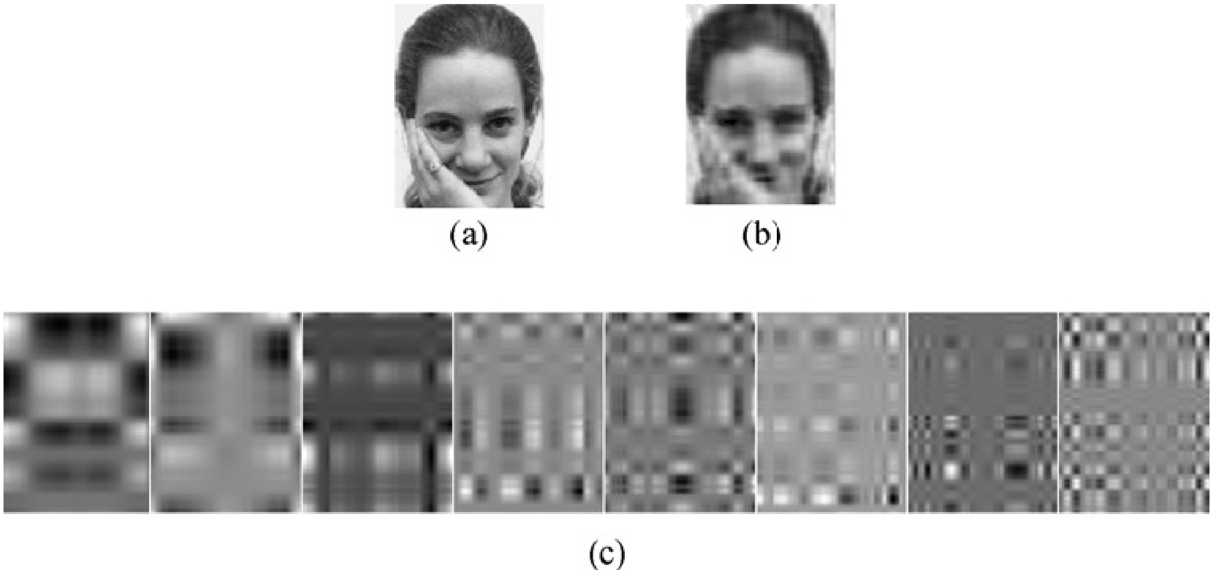


Figure 2.15. Displaying the face representation using PCA method. (a) is the original image, (b) is the result of summing first 8 principal components and (c) shows 8 principal components [22].

Sparse PCA is proposed by [21] to address this loading problem which is based on a linear regression method, Lasso, proposed by [23]. We first introduce the Lasso method and then continue to describe the sparse PCA approach.

2.3.1. Multiple Linear Regression, Lasso and Elastic Net

Multiple linear regression is a method which models the relationship between an measured variable (response vector), $Y = (y_1, ..., y_n)^T$, and a set of explanatory variables (predictors), $X_j = (x_{1j}, ..., x_{nj})^T, j = 1, ..., p$. In the multiple linear regression method the data is modeled using linear functions which its parameters are estimated from data. Linear regression methods are implemented using the least squares approach or minimizing the penalized least squares loss function. The linear regression can be applied to predict the new value of Y having an additional value of X, or to find the relative strength between Y and X_j.

$$y = X\beta + \varepsilon \tag{2.17}$$

where β is a p-dimensional parameter vector (regression coefficients) while ε models an additive noise (disturbance term). The Lasso method combines the regression model with an additional constraint on the regression coefficients to maintain the sparseness of the result.

$$\beta_{lasso} = \arg\min_{\beta} \left| Y - \sum_{j=1}^{p} X_j \beta_j \right|^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.18}$$

where the λ is the Lagrange coefficient which mixes the linear regression model with an l_1 optimization problem on the regression coefficients.

The number of variables selected by the lasso method is limited by the number of observations. Later on, a generalized form of the lasso was proposed by [24], Elastic Net, which adds the l_2 norm of the loading coefficient to the lasso problem definition.

$$\beta_{en} = \arg\min_{\beta} \left| Y - \sum_{j=1}^{p} X_j \beta_j \right|^2 + \lambda_2 \sum_{j=1}^{p} \left| \beta_j \right|^2 + \lambda_1 \sum_{j=1}^{p} \left| \beta_j \right| \qquad (2.19)$$

The defined penalty in the elastic net is a convex combination of the lasso penalty and a ridge penalty. It is shown that the elastic net solves the limitation of the lasso method in terms of the number of selected variables [24].

2.3.2. Sparse Principal Component Analysis

An interesting approach, SCoTLASS, has been proposed by [25] which directly uses the l_1 norm on the PCA to achieve a sparse loading coefficients. The SCoTLASS method doesn't provide a convex optimization problem, and therefore its high computational cost makes it an impractical method. The SCoTLASS optimization problem is,

$$a_k = \arg\min_{a_k} \left\{ a_k (X^T X) a_k \right\} \quad \text{subject to} \quad a_k^T a_k = 1 \quad \text{and} \quad \sum_{j=1}^{p} | a_{k,j} | \leq t \quad . \qquad (2.20)$$

To realize the SCoTLASS as a feasible approach, the Sparse PCA method was first introduced by [21]and was later used by other researchers. The Sparse PCA tries to engage the multiple linear regression method, Lasso, and combine it with the concept of SCoTLASS to sparsify the loading coefficients while holding the maximum data variance.

The sparse PCA is performed in two steps. In the first step the PCA is performed using the SVD method. In the second step, a suitable sparse approximation based on the Lasso method is fulfilled using the following equation:

$$\hat{\beta} = \arg\min_{\beta} \left\| Z_i - X\beta \right\|^2 + \lambda \left\| \beta \right\|^2 + \lambda_1 \left\| \beta \right\|_1 \tag{2.21}$$

where $Z_i = U_i D_{ii}$ is the i^{th} principal component and $\left\| \beta \right\|_1 = \sum_{j=1}^{p} \left| \beta_j \right|$ are the 1-norm. The i^{th} approximated loading $\hat{V}_i = \frac{\beta}{\left\| \beta \right\|}$ which is a sparse approximated model of V_i. In fact, the term $X\hat{V}_i$ approximates Z_i. The larger value of λ leads to the production of more zero coefficients in β .In this approach the principle components should be determined individually and then sparsify the loadings in the second step. The reader is referred to [21] in which a numerical solution is provided for this problem.

Chapter 3. Sparse Coding Methods

3.1. Projection Pursuit Regression

In 1981, a new method called Projection Pursuit Regression was proposed by [26] based on the nonparametric multiple regression in which an iterative procedure (Successive Refinement) finds a smooth representation of an input data. Although, other techniques existed to address the nonparametric regression (kernel, nearest-neighbor, spline smoothing), they all fail to model a high-dimensional sparse data. On the other hand, polynomial regression methods, which provide surface regression for high-dimensional data, need high order of polynomials resulting in a high complexity. This problem is addressed in the Projection Pursuit Regression method by implementing a flexible surface regression without using polynomial functions.

Suppose X is a p-dimensional predictor variables and a random variable Y (response) consists of n measurements. The surface regression method tries to approximate the response by a linear combination of univariate functions S_{α_m} of predictors.

$$\varphi(X) = \sum_{m=1}^{M} S_{\alpha_m}(\alpha_m \cdot X) \tag{3.1}$$

where $\alpha_m \cdot X$ is an inner product. α_m is a coefficient vector. S_{α_m} is a smoothing function and is generally described with

$$S(z_i) = \underset{i-k \le j \le i+k}{AVE}(z_j)\,,\ z_i = \alpha \cdot x_i \tag{3.2}$$

where AVE is an averaging function (mean, median, or other ways). The approximation problem is solved in an iterative manner.

1) Initialize the residual $r_i^1 = y_i$, $i = 1, 2, \ldots, n$
2) Find coefficient vector of iteration M, α_M by maximizing

$$\alpha_M = \arg\max_{\alpha} \left\{ 1 - \sum_{i=1}^{n} \left(r_i^M - S_\alpha(\alpha \cdot x_i) \right)^2 \right\} \tag{3.3}$$

3) Calculate the residual as $r_i^{M+1} = r_i^M - S_{\alpha_M}(\alpha_M \cdot x_i)$, $i = 1, 2, \ldots, n$
4) Terminate if the residual is acceptable, unless repeat steps 2 & 3.

3.2. Matching Pursuit

In 1993, a method proposed by [27] to decompose any signal into a linear expansion of waveforms which describe time-frequency properties of the signal. Time-frequency atoms are selected to best match the signal structure and are created with dilations, translations and modulations of a single window function. In essence, the matching pursuit is closely related to the projection pursuit regression. The matching pursuit is a greedy method which iteratively decomposes the signal into its representing waveforms (atoms). Basically, its definition is inspired with the Hilbert transform.

$$< f, g >= \int_{-\infty}^{+\infty} f(t)\overline{g}(t)dt \tag{3.4}$$

where it describes the inner product of functions f and g. The atoms are obtained by scaling, translating and modulating of a real function, $g(t)$,

$$g_{\gamma}(t) = \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right)e^{i\xi t} \tag{3.5}$$

where the coefficient $1/\sqrt{s}$ normalizes the norm of $g_{\gamma}(t)$ to 1. In order to have an acceptable representation, a comprehensive countable set of atoms should be derived using (3.5) to create a dictionary. Then , the signal is represented as a linear combination of created atoms as,

$$f(t) = \sum_{n=1}^{N} a_n g_{\gamma_n}(t) \quad . \tag{3.6}$$

The matching pursuit seeks for a linear expansion of f which best match its inner structure. This is performed by successive refinements of f under the Hilbert transform as,

$$f = \left\langle f, g_{\gamma_0} \right\rangle g_{\gamma_0} + Rf \quad \cdot \tag{3.7}$$

where Rf is a residual vector. It is conspicuous Rf is perpendicular to g_{γ_0}. Therefore, the energy preservation of f is satisfied by,

$$\|f\|^2 = \left|\left\langle f, g_{\gamma_0} \right\rangle\right|^2 + \|Rf\|^2 \quad . \tag{3.8}$$

To minimize the error (norm of $\|Rf\|$), γ_0 should be selected to maximize $\left|\left\langle f, g_{\gamma_0} \right\rangle\right|$. In the next step, the above procedure is repeated for the residual as,

$$R^n f = \left\langle R^n f, g_{\gamma_n} \right\rangle g_{\gamma_n} + R^{n+1} f \quad \cdot \tag{3.9}$$

This procedure should be repeated until the residual of the iteration m falls into an acceptable error. Now, the function f is approximately decomposed by the matching pursuit as,

$$f = \sum_{n=0}^{m-1} \left\langle R^n f, g_{\gamma_n} \right\rangle g_{\gamma_n} + R^m f \quad . \tag{3.10}$$

3.3. Orthogonal Matching Pursuit

The matching pursuit is able to optimally decompose f in the case that atoms are pairwise orthogonal. However, there is no guarantee that it converges into an efficient response if the basis waveforms are not orthogonal. The Orthogonal Matching Pursuit (OMP), proposed by [28], is an alternative solution based on the matching pursuit which provides a fast convergence with non-orthogonal dictionaries. The key to its enhancement is to update all obtained coefficients a_k (3.6) to be used in the next iteration. Having this concept, suppose the decomposing of f is formulated for the iteration k as,

$$f = \sum_{n=1}^{k} a_n^k x_n + R_k f \quad . \tag{3.11}$$

The objective is to update the coefficient a_n^k into a_n^{k+1},

$$f = \sum_{n=1}^{k+1} a_n^{k+1} x_n + R_{k+1} f \quad . \tag{3.12}$$

As the dictionary atoms are not necessarily orthogonal, an auxiliary model of the new x_{k+1} based on previously selected atoms are needed,

$$x_{k+1} = \sum_{n=1}^{k} b_n^k x_n + \gamma_k \quad . \tag{3.13}$$

The new created term using a weighted summation of previously selected atoms, $\sum_{n=1}^{k} b_n^k x_n$, is a new projection of x_{k+1} which is unexplainable using $\{x_0, \ldots, x_k\}$. Having this modification, the non-orthogonal dictionary problem of the matching pursuit method is omitted. In the next iteration, these updates should be applied to maintain independency of the next adding term to existing descriptors as,

$$a_n^{k+1} = a_n^k - \alpha_k b_n^k , \quad n = 1,\ldots,k \tag{3.14}$$

and $a_{k+1}^{k+1} = \alpha_k$

where $\alpha_k = \frac{\langle R_k f , x_{k+1} \rangle}{\langle \gamma_k , x_{k+1} \rangle} = \frac{\langle R_k f , x_{k+1} \rangle}{\|\gamma_k\|^2} = \frac{\langle R_k f , x_{k+1} \rangle}{\|x_{k+1}\|^2 - \sum_{n=1}^{k} b_n^k \langle x_n , x_{k+1} \rangle}$.

The OMP method resolved the convergence problem of the matching pursuit method with the cost of adding complex computations which proportionally arises by increasing the number of iterations. In general, the OMP algorithm can be explained as (Table 3.1. OMP algorithm. A lot of techniques have been developed to address the computational complexity related to OMP iterative process specifically due to the calculation of $\{b_n^k\}$. To illustrate, lets rewrite (3.13) in a matrix form as,

$$v_k = A_k b_k \tag{3.15}$$

$$v_k = \left[\langle x_{k+1}, x_1 \rangle, \langle x_{k+1}, x_2 \rangle, \ldots, \langle x_{k+1}, x_k \rangle \right]^T$$
$$b_k = \left[b_1^k, b_2^k, \ldots, b_k^k \right]^T$$
$$A_k = \begin{bmatrix} \langle x_1, x_1 \rangle & \langle x_2, x_1 \rangle & \cdots & \langle x_k, x_1 \rangle \\ \langle x_1, x_2 \rangle & \langle x_2, x_2 \rangle & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \langle x_1, x_k \rangle & \langle x_2, x_k \rangle & \cdots & \langle x_k, x_k \rangle \end{bmatrix} .$$

Having the third step of the OMP algorithm, the matrix A_k is nonsingular. Therefore, b_k is calculated using this equation,

$$b_k = A_k^{-1} v_k \tag{3.16}$$

And A_{k+1} is obtained as follows,

$$A_{k+1} = \begin{bmatrix} A_k & v_k \\ v_k^* & 1 \end{bmatrix} \tag{3.17}$$

And the inverse matrix of A_k can be recursively calculated,

$$A_{k+1}^{-1} = \begin{bmatrix} A_k^{-1} + \beta b_k b_k^* & -\beta b_k \\ -\beta b_k^* & \beta \end{bmatrix}, \beta = \frac{1}{(1 - v_k^* b_k)} \quad . \tag{3.18}$$

This formula shows an iterative calculation method instead of finding A_k in each iteration.

Table 3.1. OMP algorithm

```
1) Initialize $R_0 f = f$ , $D_0 = \{\}$ , $a_0^0 = 0$ , $k = 0$ , $x_0 = 0$
2) Find $\arg\max_j \left|\langle R_k f, x_j \rangle\right|; x_j \in D$
3) If $\left|\langle R_k f, x_j \rangle\right| < threshold$ then stop
4) Compute $\{b_n^k\}$
5) Update $\{a_n^{k+1}\}, n = 1, \ldots, k+1$
6) Update the model $f_{k+1} = \sum_{n=1}^{k+1} a_n^{k+1} x_n$ , $R_{k+1} f = f - f_{k+1}$ , $D_{k+1} = D_k \cup \{x_{k+1}\}$
7) Go back to step 2
```

3.3.1. Cholesky Orthogonal Matching Pursuit

The orthogonal update step of the OMP method computes the pseudo-inverse of the local dictionary D_k. The complexity of pseudo-inverse transform arises by increasing the number of iterations. An intelligent substitution for the pseudo-inverse calculation is proposed by [29] which engages Cholesky factorization to update the sparse vector coefficients, $\overline{\gamma}$, for the following sparse representation problem,

$$\overline{\gamma} := \arg\min_{\overline{a}} \|\overline{x} - D\overline{a}\| \text{ subject to } \|\overline{a}\|_0 < \Gamma \tag{3.19}$$

where Γ is the limit of non-zero coefficients. Now, consider the iteration $k+1$, we have $D_{k+1} = [D_k \quad d_n]$, where d_n is the selected atom which maintains the maximum correlation with the residual, $\overline{r}_k$. One straightforward formulation to obtain the sparse vector, $\overline{\gamma}$ is,

$$\left[D_{k+1}^T D_{k+1}\right] \overline{\gamma} = D_{k+1}^T \overline{x} \tag{3.20}$$

$$A_{k+1}\overline{\gamma} = \left[\begin{bmatrix} D_k^T \\ d_n^T \end{bmatrix} \begin{bmatrix} D_k & d_n \end{bmatrix}\right]\overline{\gamma} = \begin{bmatrix} D_k^T D_k & D_k^T d_n \\ d_n^T D_k & d_n^T d_n \end{bmatrix}\overline{\gamma} = \begin{bmatrix} A_k & v \\ v^T & c \end{bmatrix}\overline{\gamma} = \overline{\alpha}_{k+1}$$

$$A_{k+1} = \begin{bmatrix} A_k & v \\ v^T & c \end{bmatrix}$$

and,

$$A_{k+1}\overline{\gamma} = \overline{\alpha}_{k+1} \tag{3.21}$$

where $\overline{\alpha}_{k+1} = D_{k+1}^T \overline{x}$ is a column vector of the iteration $k+1$, consisting of inner products of local dictionary atoms and the input signal, $\overline{x}$. In each iteration, a column and a row is added to the A_k to create the A_{k+1}. Using the Cholesky decomposition, the $A_{k+1} = LL^T$ is derived assuming that L is a lower triangular matrix which helps to reduce computation costs of solving (3.21),

$$\begin{aligned} &L_{k+1}L_{K+1}^T\overline{\gamma} = \overline{\alpha} \\ &\overline{y} := solve \quad L\overline{y} = \overline{\alpha} \\ &\overline{\gamma} := solve \quad L^T\overline{\gamma} = \overline{y} \end{aligned} \tag{3.22}$$

The point to use the aforementioned calculations instead of the pseudo-inverse is the simple relation between the L_{k+1} and L_k,

$$L_{k+1} = \begin{bmatrix} L_k & 0 \\ \overline{w}^T & \sqrt{c - \overline{w}^T\overline{w}} \end{bmatrix}, \quad \overline{w} = L_k^{-1}\overline{v} \tag{3.23}$$

and,

$$\overline{w} := solve \quad L_k\overline{w} = \overline{v} \tag{3.24}$$

Supposing that the L_k is a lower triangular matrix, the L_{k+1} will remain a lower triangular matrix. The complete algorithm of the Cholesky-OMP method is presented in Table 3.1.

Table 3.1. The Algorithm of Cholesky-OMP

```
1. Initialize: $D$, input signal $\bar{x}$, number of non-zero coefficients $\Gamma$, $I = ()$, $L = [1]$,
   $\bar{r} = \bar{x}$, $\bar{\gamma} = \bar{0}$, $\bar{\alpha} = \bar{0}$, $n = 1$
2. $j := \arg\max \left\langle d_j^T, \bar{r} \right\rangle$
3. If $n > 1$ then
      $\bar{w} := solve \quad L\bar{w} = D_I^T \bar{d}_j$
      $L = \begin{bmatrix} L & 0 \\ \bar{w}^T & \sqrt{1 - \bar{w}^T \bar{w}} \end{bmatrix}$
4. Update $I = I \cup j$
5. Calculate $\bar{\alpha}_n = D_I^T \bar{x}$
6. $\bar{y} := solve \quad L\bar{y} = \bar{\alpha}_n$
7. $\bar{\gamma} := solve \quad L^T \bar{\gamma}_I = \bar{y}$
8. $r = x - D_I \bar{\gamma}_I$
9. $n = n + 1$
10. If $n < \Gamma$ then go to step 2
```

3.3.2. Batch Orthogonal Matching Pursuit

As discussed previously, Cholesky method reduces the computations needed for the orthogonal update step of each iteration. The second approach proposed by [29], applies another technique to reduce the computational cost of calculating the correlation (inner product) of the residual with dictionary atoms. In this situation they are aiming to find the best matching atom with the residual. A pre-calculated matrix, $G = D^T D$, is stored in the memory to eliminate redundant computations.

$$\bar{\alpha}' = D^T x \tag{3.25}$$

$$\bar{\alpha} = D^T r = D^T (x - D_I (D_I^T D_I)^{-1} D_I{}^T x) = \bar{\alpha}' - G_I (G_{I,I})^{-1} \bar{\alpha}'_I$$

It shows that the correlation between the residual and dictionary atoms is calculated based on the matrix G, without knowing the residual. The stopping factor can be applied based on the number of non-zero coefficients or the norm of the residual (error). For the second case, the error in the k^{th} iteration is calculated based on the error of the $k-1^{th}$ iteration,

$$r_k = \overline{x} - D\overline{\gamma}_k = x - D\overline{\gamma}_k + D\overline{\gamma}_{k-1} - D\overline{\gamma}_{k-1} = r_{k-1} + D\left(\overline{\gamma}_{k-1} - \overline{\gamma}_k\right) \tag{3.26}$$

The orthogonality property of the OMP method maintains that the residual is perpendicular to the current signal approximation [29], $(\overline{r}_k)^T D\gamma_k = 0$. Now, the norm of the residual is obtained as,

$$\begin{aligned}
\|\overline{r}_k\|_2^2 &= (\overline{r}_k)^T \overline{r}_k = (\overline{r}_k)^T (\overline{r}_{k-1} + D\left(\overline{\gamma}_{k-1} - \overline{\gamma}_k\right)) = \overline{r}_k^T r_{k-1} + \overline{r}_k^T D\overline{\gamma}_{k-1} \\
&= (\overline{r}_{k-1} + D\left(\overline{\gamma}_{k-1} - \overline{\gamma}_k\right))^T \overline{r}_{k-1} + (\overline{r}_k)^T D\gamma_{k-1} \\
&= \|\overline{r}_{k-1}\|_2^2 - (\overline{r}_{k-1})^T D\overline{\gamma}_k + (\overline{r}_k)^T D\overline{\gamma}_{k-1} \\
&= \|\overline{r}_{k-1}\|_2^2 - (\overline{x} - D\overline{\gamma}_{k-1})^T D\overline{\gamma}_k + (\overline{x} - D\overline{\gamma}_k)^T D\overline{\gamma}_{k-1} \\
&= \|\overline{r}_{k-1}\|_2^2 - \overline{x}^T D\overline{\gamma}_k + \overline{x}^T D\overline{\gamma}_k \\
&= \|\overline{r}_{k-1}\|_2^2 - (\overline{r}_k + D\overline{\gamma}_k)^T D\overline{\gamma}_k + (\overline{r}_{k-1} + D\overline{\gamma}_{k-1})^T D\overline{\gamma}_{k-1} \\
&= \|\overline{r}_{k-1}\|_2^2 - (\overline{\gamma}_k)^T D^T D\overline{\gamma}_k + (\overline{\gamma}_{k-1})^T D^T D\overline{\gamma}_{k-1} \\
&= \|\overline{r}_{k-1}\|_2^2 - (\overline{\gamma}_k)^T G\overline{\gamma}_k + (\overline{\gamma}_{k-1})^T G\overline{\gamma}_{k-1}
\end{aligned} \tag{3.27}$$

By defining $\delta_k = (\overline{\gamma}_k)^T G\overline{\gamma}_k$, the error of each iteration is updated as,

$$\varepsilon_k = \varepsilon_{k-1} - \delta_k + \delta_{k-1} \tag{3.28}$$

According to the fact that $G\overline{\gamma} = G_I\overline{\gamma}_I = G_I\left(G_{I,I}\right)^{-1}\overline{\alpha}_I^0$ which is calculated in each iteration, and therefore the error update step takes a low computational time. The algorithm of this method is presented in Table 2.3.

Table 2.3. The Algorithm of Batch-OMP

```
1. Initialize: $D, G = D^T D$, input signal $\overline{x}$, $\overline{\alpha}' = D^T\overline{x}$, number of non-zero coefficients $\Gamma$,
   $I = (), L = [1], \varepsilon^0 = \overline{x}^T\overline{x}$, $\overline{\gamma} = \overline{0}$, $\overline{\alpha} = \overline{\alpha}'$, $\delta_0 = 0$, $n = 1$
2. $j := \max\left(\overline{\alpha}\right)$
3. If $n > 1$
   $\overline{w} := solve \quad L\overline{w} = G_{I,j}$
   $L = \begin{bmatrix} L & 0 \\ \overline{w}^T & \sqrt{1 - \overline{w}^T\overline{w}} \end{bmatrix}$
4. Update $I = I \cup j$
5. $\overline{y} := solve \quad L\overline{y} = \overline{\alpha}_I$
```

```
6. $\overline{\gamma} := solve \quad L^T \overline{\gamma}_I = \overline{y}$
7. $\overline{\beta} = G_I \overline{\gamma}_I$
8. $\overline{\alpha} = \overline{\alpha}' - \overline{\beta}$
9. $\delta_n = \gamma_I^T \overline{\beta}_I$
10. $\varepsilon_n = \varepsilon_{n-1} - \delta_n + \delta_{n-1}$
11. $n = n + 1$
12. If $\varepsilon^{n-1} > \varepsilon^n$ then go to step 2
```

3.3.3. Stagewise Orthogonal Matching Pursuit

Stagewise Orthogonal Matching Pursuit (St-OMP), is a fast technique proposed by [30] to provide a sparse solution for extremely underdetermined sparse representation problems. The purpose is to reduce the representation error in a Stagewise fashion by approximating the solution of $y = \Phi \overline{x}_0$, while $\overline{x}_0$ is the best sparse solution. This proposed method is intended to reduce the tedious computations needed for solving large size sparse problems. The basic idea is to extract multiple atoms in each stage, called Stagewise OMP, whereas the traditional OMP method finds exactly one atom per each iteration. Therefore, the number of iterations needed to extract a certain number of non-zero coefficients is significantly reduced. In this method, the process of selecting atoms uses the matching filter, $\tilde{x} = \Phi^T y$, similar to the OMP method, and $z = \tilde{x} - x_0$ measures the reconstruction accuracy. This method assumes that the dictionary, $\Phi \in \mathbb{R}^{n \times N}$, is randomly taken from the uniform spherical ensemble (USE) and its columns are independent and identically distributed (i.i.d.) points on the unit sphere. Using this definition, if both n and N are adequately large, then the entries of vector z approximately have a Gaussian distribution with the following standard deviation [30],

$$\sigma \approx \frac{\|x_0\|_2}{\sqrt{n}} \tag{3.29}$$

In the same behavior to the OMP method, the St-OMP initiates the residual with the input signal. In each stage, the correlations between the residual, $\overline{r}_s$, and dictionary atoms are calculated, $\overline{c}_s = \Phi^T \overline{r}_s$

(Matching Filter). Then, a hard thresholding is applied to the output of the matching filter, $J_s = \{ j : |c_s(j)| > t_s\sigma \}$ where t_s is the threshold, aiming to select a subset of atoms with higher correlation. The index of selected atoms are gathered, $I_s = I_{s-1} \cup J_s$, and then, the input signal is projected to all atoms of the selected sub-set,

$$(x_s)_{I_s} = \left(\Phi_{I_s}^T \Phi_{I_s}\right)^{-1} \Phi_{I_s}^T y \tag{3.30}$$

The result is the sparse vector of the s^{th} stage. Then, the residual is updated by $\overline{r}_s = y - \Phi x_s$. This process is repeated for a pre-defined number of stages or stopped earlier if a desired reconstruction error is achieved. The threshold in each iteration is calculated for the residual based on considering noise to be as a Gaussian distribution with standard deviation, σ. The offering value for threshold is $2 < t_s < 3$. The St-OMP block diagram is depicted in Figure. In order to have more detail to select a good threshold value, readers are referred to [30].

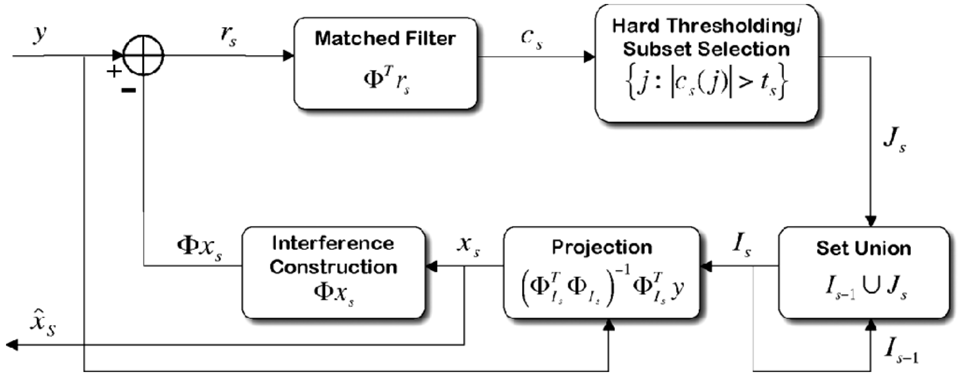


Figure 3.1. Schematic Representation of the St-OMP method [30].

3.4. Basis Pursuit

The basis pursuit [31] is a method to decompose a signal specifically using over-completed dictionaries. An over-complete dictionary can be created by concatenating multiple dictionaries (DCT, wavelet, Gabor and etc.) in order to combine their characteristics. This fact contributes to an extreme desire of a lot of works, specifically the basis pursuit technique, on the over-completed dictionaries. For an input signal s and an over-completed dictionary D which consists of waveforms ϕ_γ (γ is a parameter), the representation is defined as,

$$s=\sum_{i=1}^{m}\alpha_{\gamma_i}\phi_{\gamma_i}+R^{(m)}, \tag{3.31}$$

where $R^{(m)}$ is the residual. The basis pursuit method uses a convex optimization problem which minimizes the l_1 norm of the representation coefficients [32]. The l_1 norm creates a nonlinear optimization problem which leads to provide a higher sparsity. On the other hand, this method is based on the global optimization and therefore it stably finds the global optimum representation whereas the MP method cannot.

$$\min\|\alpha\|_1 \text{ subject to } \Phi\alpha=s. \tag{3.32}$$

To introduce the solution to this problem, we should first describe the primal-dual interior point algorithm which is a popular linear programming method.

3.4.1. Primal Dual Interior Point Algorithm for linear programming

The primal dual Interior point described by [33] is a linear programming method which provides a solution for the standard form primal problem as,

$$x=\arg\min_{x} \quad c^T x \text{ subject to } Ax=b\ x\geq 0, \tag{3.33}$$

where $c,x\in\mathbb{R}^n$, $b\in\mathbb{R}^m$ and A is a matrix with the size of $m\times n$. There is a dual problem associated with the primal problem,

$$\arg\max_{(y,s)} \quad b^T y \qquad \text{subject to} \quad A^T y+s=c, \quad s\geq 0, \tag{3.34}$$

where $y\in\mathbb{R}^m$ and $s\in\mathbb{R}^n$ is called the dual slack. The quantity $c^T x-b^T y$ is called the duality gap which is the termination factor in the linear programming. The problem is rewritten as,

$$\begin{aligned}&A^T y + s = c \\ &Ax = b \\ &XSe = 0 \\ &(x,s) \geq 0\end{aligned} \tag{3.35}$$

where $e \in \mathbb{R}^n, [e_i] = 1, i = 1, \ldots., n$, X and S are diagonal matrixes defined as,

$$X = \begin{pmatrix} [x]_1 & 0 & \cdots & 0 \\ 0 & [x]_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & [x]_n \end{pmatrix}, \ S = \begin{pmatrix} [s]_1 & 0 & \cdots & 0 \\ 0 & [s]_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & [s]_n \end{pmatrix} . \tag{3.36}$$

To solve the problem (3.35), the variant of Newton's method is used. Considering the result (x^*, y^*, s^*), the problem is reformulated as follows,

$$F(x,y,s) = \begin{bmatrix} A^T + s - c \\ Ax - b \\ XSe \end{bmatrix} = 0, \ (x,s) \geq 0 \quad . \tag{3.37}$$

According to the Newton's method, the difference of the result between the current iteration and the next iteration $(\delta x, \delta y, \delta s)$ is calculated as the following linear model,

$$F'(x,y,s) \begin{bmatrix} \delta x \\ \delta y \\ \delta s \end{bmatrix} = -F(x,y,s) \tag{3.38}$$

where F' is the Jacobean of F . The final formula is obtained according to the equation (3.38),

$$\begin{bmatrix} 0_{n \times m} & A^T_{n \times m} & 0_{n \times n} \\ A_{m \times n} & 0_{m \times m} & 0_{m \times n} \\ S_{n \times n} & 0_{n \times m} & X_{n \times n} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta s \end{bmatrix} = \begin{bmatrix} 0_{n \times 1} \\ 0_{m \times 1} \\ -XSe \end{bmatrix} \quad . \tag{3.39}$$

The value of $(x,y,s)^{k+1} = (x,y,s)^k + (\delta x, \delta y, \delta s)$ should be iteratively modified until the value of the duality gap gets smaller than a desired error. The introduced method is the simplest form of the interior point method and a lot of enhancements have been applied to accelerate the convergence and to increase the accuracy of the result.

3.4.2. Solving the Basis Pursuit method using the linear programming

After introducing the prime dual method, it is now time to utilize it to solve (3.19). The solution is supposed to be sparse with n non-zero coefficients corresponding to n columns of Φ . These columns constitute the basis of a space $\mathbb{R}^n$. After finding the optimal basis, the solution is uniquely identified [32]. To use the primal dual method, the signal patch, s, is split into $s=u-v$, while $u,v \geq 0$. Now, define $x=\begin{bmatrix}u & v\end{bmatrix}^T$ and assume $A=[\Phi \quad -\Phi]$. we have,

$$\|s\|_1 = \|u\|_1 + \|v\|_1 = \sum u_i + \sum v_i = c^T x \quad . \tag{3.40}$$

Using this transformation, it is possible to use the dual prime method to solve the BP (Table).

Table 3.3. The pseudo-algorithm of solving the BP using the dual primal interior point

```
1. Initialize $x, y$ and $s$
2. Solve equation (3.39) to find $[\Delta x \quad \Delta y \quad \Delta s]$
3. Update $x, y$ and $s$
4. Check convergence criteria, if not converged go to step 2
5. End
```

Chapter 4. Dictionary Learning

The selection of a dictionary influences the quality of the sparse representation in terms of the level of sparsity, the error of representation, the desired characteristic extractions and etc. As discussed previously, different analytical dictionaries have emerged to address the different applications in which each dictionary highlights a special property of the input signal. For example the DCT dictionary extracts localized frequency domain information, whereas the wavelet conducts a multi-resolution decomposition. All these dictionaries share the same fact that they are independent of input signal contents. However, it is of great interest to create a dictionary well adapted to the input signal, aiming to increase the sparsity of the representation. To address this matter, the dictionary learning has emerged and developed since the last decade. Dictionary learning intends to create a dictionary, D, which best represents the input set, $Y = \{ y_i \in \mathbb{R}^n \mid i = 1, \dots, N \}$.

The first approach to solve the dictionary learning problem is based on the maximum likelihood method. [34] proposed a probabilistic model based on the input set for learning an overcomplete dictionary. This approach considers the error of representation as an additive Gaussian noise,

$$y = Dx + \varepsilon \tag{4.1}$$

where $D \in \mathbb{R}^{n \times L}$ is an overcomplete basis (dictionary), $L > n$. x and ε are sparse vector and reconstruction error, respectively. The data likelihood is derived as,

$$l(Y) = \log\left(\prod_{i=1}^{N} P(y_i \mid D, x) \right) \propto \sum_{i=1}^{N} -\frac{1}{2\sigma^2}(y_i - Dx)^2 \tag{4.2}$$

where σ^2 is the reconstruction error variance. The problem of the above definition is that the overcomplete representation has many solutions. A remedy to this problem is to define prior probability for the sparse coefficients, $P(x)$, and the optimization problem is derived as,

$$x = \arg\max_{x} \quad \prod_{i=1}^{N} P(x_i \mid y_i, D) = \arg\max_{x} \quad \prod_{i=1}^{N} P(y_i \mid D, x_i) P(x_i) \tag{4.3}$$

One possible selection for $P(x)$ is the Laplace distribution, $P(x_i) \propto \exp\left(-\theta \left|x_i\right|\right)$, which puts a great emphasis on zero values and forces the problem to maintain the sparsity condition. The total formulation turns into the following minimization problem [35],

$$D = \arg\min_{D} \quad \sum_{i=1}^{N} \min_{x_i} \left\{ \left\| Dx_i - y_i \right\|_2^2 + \lambda \left\| x_i \right\|_1 \right\} \tag{4.4}$$

A solution to the above optimization is obtained by the use of the gradient descend procedure,

$$D_{n+1} = D_n - \eta \sum_{i=1}^{N} \left(D_n x_i - y_i \right) x_i^T \tag{4.5}$$

where η adjusts the update step. The other method to learn a set of atoms is the method of optimal directions (MOD) which is proposed by [36] and works in the similar way of the K-means method.

In this paper, it is stated that for an overcomplete set of vectors, namely frames, because $L > n$, frames are dependent and therefore can't be taken as the basis for the space. In this method, the author avoids to use the term, dictionary, because it implies the vector quantization or classification whereas the term frame covers both basis and an overcomplete set. Each signal, $y \in \mathbb{R}^n$, can be represented as a linear combination of frames. The training process is iteratively performed in which each iteration is combined with a sparse coding update and dictionary atoms update procedures. The sparse coding update step uses OMP or Basis Pursuit methods. For the dictionary update method, the error of reconstruction, r_i, for all input sets are calculated based on the Frobenius norm, $R = [r_1, r_2, \ldots, r_N]$, where R is the residual of representing all input signals [35] and is calculated as,

$$\|R\|_F^2 = \left\| \sum_{i=1}^{N} r_i \right\|_F^2 = \|Y - DX\|_F^2 \quad . \tag{4.6}$$

Taking the derivative of (4.6) with respect to D, we reach to the point $\left(Y - DX\right)X^T = 0$. Suppose $\tilde{R}_{r,x} = RX^T$ is the estimation of the cross correlation between the residual and vector of sparse coefficients. $\tilde{R}_{x,x} = XX^T$ is the auto correlation of the sparse vector. Then the set of frames is iteratively updated as follows [37],

$$D_i = D_{i-1} + \tilde{R}_{r,x} \cdot \tilde{R}_{x,x}^{-1} \tag{4.7}$$

The next category of dictionary learning approaches applies the maximum a-posterior probability, $P(D \mid Y) \propto P(Y \mid D)P(D)$. This category is similar to the maximum likelihood method, but the prior probability, $P(D)$, brings more flexibility to derive a lot of formulations. For a prior probability that forces the dictionary atoms to have unit Frobenius norm, the dictionary update is derived as follows,

$$D_{n+1} = D_n + \eta EX^T + \eta \cdot tr(XE^T D_n) D_n \tag{4.8}$$

In the following, two recent dictionary learning methods, l_1-regularized and K-SVD, are introduced in detail.

4.1. Sparse coding using l_1-regularized least squares and learning using Lagrange dual

A new learning scheme based on the maximum a-posteriori (MAP) estimation is proposed by [38]. In this model, it is assumed that the reconstruction error has a zero-mean Gaussian distribution with covariance $\sigma^2 I$. The prior probability distribution for each coefficient in the sparse vector is selected to be Laplacian, $P(x_j) \propto \exp(-\beta\phi(x_j))$, where $\phi(.)$ is the sparsity function. $\phi(x_j) = \|x_j\|_1$ is the L1-Penalty which maintains sparsity while still being robust to irrelevant features. Considering the uniform distribution for the dictionary atoms, the solution to the maximum posterior estimation is obtained as,

$$\min_{\{d_j\},\{x_i\}} \sum_{i=1}^{N} \frac{1}{2} \left\| y_i - \sum_{j=1}^{L} d_j x_{i,j} \right\|^2 + \beta \sum_{i=1}^{N} \sum_{j=1}^{L} \phi(x_{i,j}) \text{ subject to } \|d_j\|^2 \leq c, \forall j = 1,\ldots,L \tag{4.9}$$

The above optimization problem can be more easily written in matrix form as follows,

$$\min_{D,X} \|Y - DX\|_F^2 + \beta \sum_{i,j} \|x_{i,j}\|_1 \text{ subject to } \sum_i D_{i,j}^2 \leq c, \forall j = 1,\ldots,L \tag{4.10}$$

The above optimization problem is convex if the problem is optimized in terms of D while X is fixed or visa versa. The optimization problem to simultaneously update both of them is not a convex problem. For updating D (fixed X), the optimization problem is a least squares problem with quadratic constraints. The solution proposed by [38] is to use the Lagrange dual which is claimed to be much more efficient than other gradient descend approaches. For learning X (fixed D), the optimization problem is a regularized least squares problem. The proposed method by [38] to address this updating step is the L1-regularized least squares problem.

For solving the problem to find the optimum value of atoms, $\{x_j\}$, for fixed D, the following minimization problem should be solved,

$$\min_x \left\| y_i - \sum_j d_j x_{i,j} \right\|^2 + \left(2\sigma^2\beta\right)\sum_j \left|x_{i,j}\right| \tag{4.11}$$

Considering only non-zero coefficients, $x_{i,j}$, the problem is simplified to an unconstrained quadratic optimization problem, by replacing $-x_{i,j}$ for $x_{i,j} < 0$. Thus, the proposed method tries to make a guess on the sign of coefficients, $x_{i,j}$, and then change the sign if it is not correct. X presents the solution to the following simplified notation,

$$\min_x \|y - Dx\|_2^2 + \gamma\|x\|_1 \quad . \tag{4.12}$$

During the iterative optimization solution, the set of non-zero coefficients are kept and updated in the active set.

The dual Lagrange method is applied to derive the dictionary update step. In this case, the sparse matrix, X, is obtained and fixed and the minimization is over D,

$$\min \|Y - DX\|_F^2 \text{ subject to } \sum_{i=1}^{k} B_{i,j}^2 \le c, \forall j = 1,\ldots,n \tag{4.13}$$

This problem is a least squares with quadratic constraints. The Lagrange definition to (4.13) is,

$$l(D,\lambda) = trace\left(\left(Y - DX\right)^T \left(Y - DX\right)\right) + \sum_{j=1}^{L} \lambda_j \left(\sum_{i=1}^{n} D_{i,j}^2 - c\right) \tag{4.14}$$

$\lambda_j \ge 0$ is a dual variable. The minimization problem over the Lagrange dual is,

$$D(\lambda) = \min_B l(D,\lambda) = trace(Y^T Y - YX^T (XX^T + \Lambda)^{-1} (YX^T)^T - c\Lambda),\ \Lambda = diag(\lambda) \tag{4.15}$$

The Newton's method solution to (4.15) is,

$$D^T = (XX^T + \Lambda)^{-1}(YX^T)^T \tag{4.16}$$

Solving the dual problem tackles much less number of variables in contrast with the primal problem. The pseudo-algorithm of the L$_1$-regularized least squares is displayed in Table .

Table 4.1. The pseudo-algorithm of the L$_1$-regularized least squares

```
1. Initialize $x = \overline{0}$, $\theta = \overline{0}$, *active set={}*, where $\theta_i \in \{-1, 0, 1\}$
2. From zero coefficients of $x$, select $i = \arg\max_i \left| \frac{\partial \|y - Dx\|_2^2}{\partial x_i} \right|$
   Add $x_i$ to the *active set* if:
   a. If $\frac{\partial \|y - Dx\|_2^2}{\partial x_i} > \gamma$, then set $\theta_i = -1$ and *active set = active set* $\cup i$,
   b. If $\frac{\partial \|y - Dx\|_2^2}{\partial x_i} < -\gamma$, then set $\theta_i = 1$ and *active set = active set* $\cup i$,
3. Feature-Sign Step:
   Select $\hat{D}$ (sub matrix of $D$), $\hat{x}$ (sub-vector of $x$) and $\hat{\theta}$ (sub-vector of $\theta$) corresponding to the *active set.* Compute $\hat{x}_{new} = (\hat{D}^T \hat{D})^{-1}(\hat{D}^T y - \gamma \frac{\hat{\theta}}{2})$
   Perform a discrete line search on the close line from $\hat{x}$ to $\hat{x}_{new}$:
   a. Check the objective value at $\hat{x}_{new}$ and all points where any coefficient changes sign
   b. Update $\hat{x}$ and the corresponding value in $x$ to point with the lowest objective value.
   Remove zero coefficients from the active set and update $\theta = sign(x)$
4. Check the optimality conditions:
   a. Optimality condition for nonzero coefficients: $\frac{\partial \|y - Dx\|^2}{\partial x_j} + \gamma sign(x_j) = 0, \forall x_j \neq 0$
      If condition (a) is not satisfied, go to step (3) without new activation, unless check (b)
   b. Optimality condition for zero coefficients: $\left| \frac{\partial \|y - Dx\|^2}{\partial x_j} \right| \leq \gamma, \forall x_j \neq 0$
      If condition (b) is not satisfied, go to step (2), unless return $x$ as the solution.
```

4.2. K-SVD method

The K-SVD method is proposed by [35], generalizes the K-means method to address the dictionary learning problem. In the K-means method a dictionary of codewords,

$C = [c_1, \ldots, c_L], \{c_i\} \in \mathbb{R}^n$, is calculated using a training algorithm and each input signal is represented to its closest code word such that $\|y - c_i\|_2^2 \geq \|y - c_k\|_2^2, \forall i \neq k$ which is the extreme case of the sparse representation problem with only one nonzero coefficient and the coefficient is 1. The K-SVD method proposes a generalized K-means method in which each input signal can be represented by more nonzero coefficients with arbitrary values between 0 to 1. For this case, the minimization problem is,

$$\min_{X,D} \|Y - DX\|_F^2 \text{ subject to } \|x_i\|_0 \leq T_0, \forall i \tag{4.17}$$

At each iteration, two steps are performed. The first step considers that the dictionary, D, is fixed and determines the sparse vector, X, using any possible method such as OMP. In the second step, the obtained sparse vector is fixed and the dictionary is updated to minimize the optimization problem (4.17). To perform this, dictionary columns are updated, $\tilde{d}_i$, individually in which each column is calculated to minimize the mean square error of the input signal reconstruction using all other dictionary columns (atoms). The problem of updating only one column can be addressed using the singular value decomposition (SVD). For updating only the column, d_k, the following optimization problem should be solved,

$$\|Y - DX\|_F^2 = \left\| Y - \sum_{j=1}^{L} d_j X_j^T \right\|_F^2 = \left\| \left(Y - \sum_{j \neq k} d_j X_j^T \right) - d_k X_k^T \right\| \tag{4.18}$$
$$= \left\| E_k - d_k X_k^T \right\|_F^2$$

The column, d_k, should be updated to reduce the reconstruction error, E_k, obtained by all other atoms. Using the SVD, new values for both d_k and X_k^T are obtained. However, X_k^T may have a lot of nonzero coefficients which is against the sparsity constraint. The key to solve this problem is to consider only nonzero coefficients and corresponding input signals. Then, the obtained updated result corresponds to nonzero coefficients of existing sparse vector and it maintains the sparsity condition. For each atom update, a set of indexes of atoms which are involved to represent the updating atom is defined as,

$$\omega_k = \left\{ i \mid 1 \le i \le L, X_k^T(i) \ne 0 \right\} \tag{4.19}$$

Now, a matrix $\Omega_k \in \mathbb{R}^{N \times length(\omega_k)}$ is formed with ones on $\left[col : \omega_k(i) \quad row : i \right]$ positions and zeros elsewhere. Now, the transform $x_k^R = x_k^T \Omega_k$ only keeps nonzero coefficients. The same thing is applied on the input signals, $Y_k^R = Y\Omega_k$, which shrinks the input signals to relative input signals which are represented with the k^{th} atom. Similarly, it is applied on the error, $E_k^R = E_k \Omega_k$, and the minimization problem is modified,

$$\tilde{d}_k = \arg\min_{d_k} \left\| E_k^R - d_k x_k^R \right\| \tag{4.20}$$

Now, we can decompose E_k^R using the SVD into $E_k^R = U \Delta V^T$. $\tilde{d}_k$ is the first column of U and x_k^R is the first column of V multiplied by $\Delta(1,1)$. In Table 4.2, the pseudo-algorithm of the K-SVD method is presented which shows the simplicity of this method.

Table 4.2. The pseudo-algorithm of the K-SVD method.

```
1. Initialize dictionary with $D \in R^{n \times L}$ which all columns are normalized
2. Use any pursuit algorithm to solve $\arg\min_X \| Y - DX \|$ subject to $\| X \|_0 \le T_0$
3. For each column in $D$, $d_k$,
    a. Find $\omega_k$, according to (4.19).
    b. Compute the representation error $E_k^R$
    c. Apply SVD decomposition and find $\tilde{d}_k$ and $x_k^R$
4. Check the convergence criteria and if it is not achieved go to step 2.
```

Chapter 5. Phase Included DCT and Multi-Stage OMP

5.1. New Overcomplete DCT dictionary: Phase Included-DCT

To our best knowledge, all previously used DCT dictionaries were only defined based on the changes of the frequency with no consideration on the phase of the signal. Therefore, for a signal whose frequency components have a nonzero phase, causes to a high reconstruction error. To illustrate, consider a simple example that the input signal is $f(x,y)\in \mathbb{R}^{8\times 8} = \cos(2\pi f_x n_x + \pi/2)$. The representation of this signal using the OMP with 5 nonzero coefficients based on the conventional completed DCT dictionary is depicted in Figure. The calculated accuracy based on the sum of absolute differences for reconstructing this patch using the conventional DCT dictionary is 43.37%. It illustrates that the conventional DCT dictionary is not representative for this patch, however it is created using a simple cosine function.

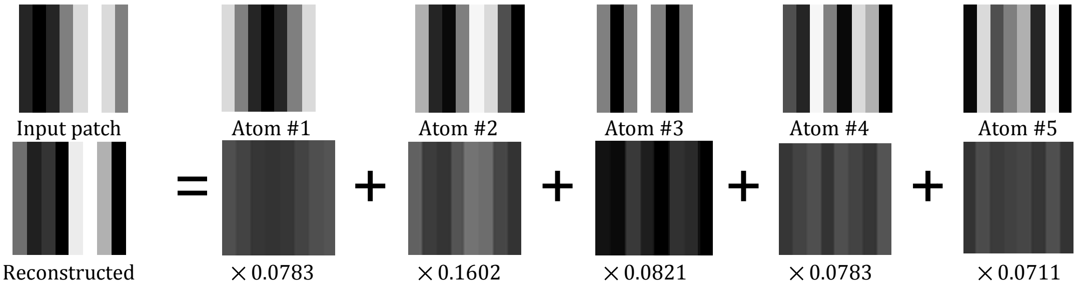
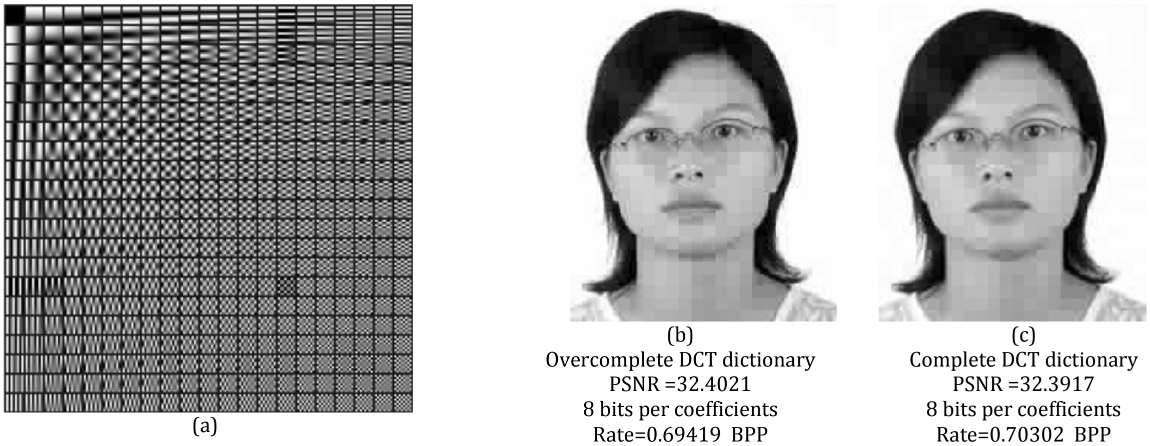


Figure 5.1. Reconstruction of a cosine function (input patch) with a non-zero phase. It is sparsely represented using the OMP method with 5 nonzero coefficients. Selected atoms are displayed in the first row and the result of multiplying them with their corresponding coefficients are shown in bottom row. The reconstructed atom error obtained by the sum of abs differences (SAD) is 43.37%.

One possible way to decrease the representation error is to apply overlapping when selecting input signal patches. This, results in tremendously increasing the computational cost. [35] applies an overcomplete DCT dictionary by defining more frequency divisions and comparing the result with the complete DCT dictionary. Figure shows that the representation error using the overcomplete DCT with more frequency divisions is not reduced [35]. The reason is that the phase information of input patches are lost using the overcomplete DCT and complete DCT dictionaries used by [35].


(a)

(b)
Overcomplete DCT dictionary
PSNR =32.4021
8 bits per coefficients
Rate=0.69419 BPP

(c)
Complete DCT dictionary
PSNR =32.3917
8 bits per coefficients
Rate=0.70302 BPP

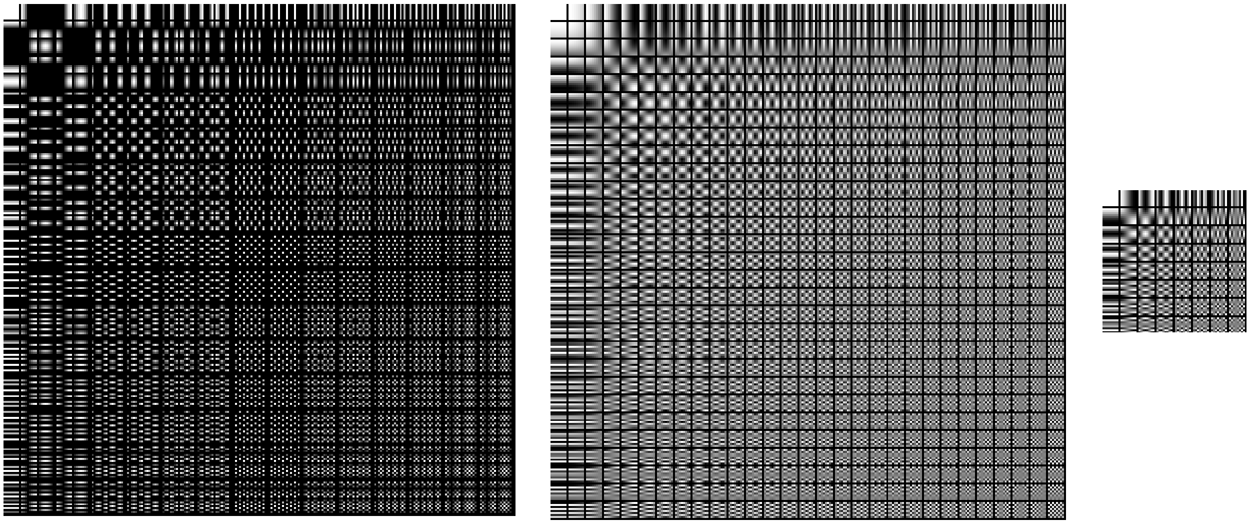
Figure 5.2. Demonstration of the insignificance of using the overcomplete DCT with more frequency divisions [35]. (a) shows the overcomplete dictionary, (b) and (c) show the sparse representation results obtained using overcomplete DCT and complete DCT dictionaries, respectively. The PSNR obtained using the overcomplete DCT is not better than the representation using complete DCT dictionary.

We propose the phase included DCT (PI-DCT) dictionary which is the extension of the complete DCT dictionary by including the phase information. For each frequency in vertical direction, f_x, or horizontal direction, f_y, N_φ phase divisions, $\varphi \in \{0, 2\pi / N_\varphi, 4\pi / N_\varphi, ..., 2(N_\varphi - 1)\pi / N_\varphi\}$, are added to the dictionary. The size of the dictionary is calculated by $L = ((n_x - 1) \times N_\varphi) \times ((n_y - 1) \times N_\varphi) + ((n_x - 1) \times N_\varphi) + ((n_y - 1) \times N_\varphi) + 1$ where n_x and n_y are number of frequency divisions (equal to the patch size $n_x \times n_y$) in vertical and horizontal directions, respectively. For example, consider that the size of each patch is 8×8 and therefore,

$n_x = 8$ and $n_y = 8$. Then for $N_\varphi = 4$ ($\overline{\varphi} = \left[0, \pi/2, \pi, 3\pi/2\right]$), the number of dictionary atoms is 841 (Figure 5). Atoms are created using the following equation,

$$\overline{d}(f_x, f_y, \varphi_x, \varphi_y) = \left[f_{\overline{\theta}}(1,1) \quad f_{\overline{\theta}}(1,2) \quad \cdots \quad f_{\overline{\theta}}(1,n_x) \quad f_{\overline{\theta}}(2,1) \quad f_{\overline{\theta}}(2,2) \quad \cdots \quad f_{\overline{\theta}}(n_y,n_x) \right]^T \tag{5.1}$$
$$f_{\overline{\theta}}(i_x, i_y) = f_{f_x, f_y, \varphi_x, \varphi_y}(i_x, i_y) = \left(\cos(2\pi f_x i_x + \varphi_x)\right) \times \left(\cos(2\pi f_y i_y + \varphi_y)\right)$$

where $\overline{\theta} = [f_x, f_y, \varphi_x, \varphi_y]$ is the set of parameters. This can be interpreted that nonzero coefficients obtained of the sparse representation using this overcomplete DCT dictionary reflect phase and magnitude of dominant components of the patch frequency transform. The coefficient values show the magnitude of the frequency components while their phases are defined based on $\overline{\theta}$ used to create corresponding atoms.



PI-DCT
(841 atoms), $N_\varphi = 4$, $n_y = 8$, $n_x = 8$

Overcomplete DCT
(841 atoms)

DCT
(64 Atoms)

Figure 5. Displaying PI-DCT, Overcomplete DCT and conventional DCT dictionaries.

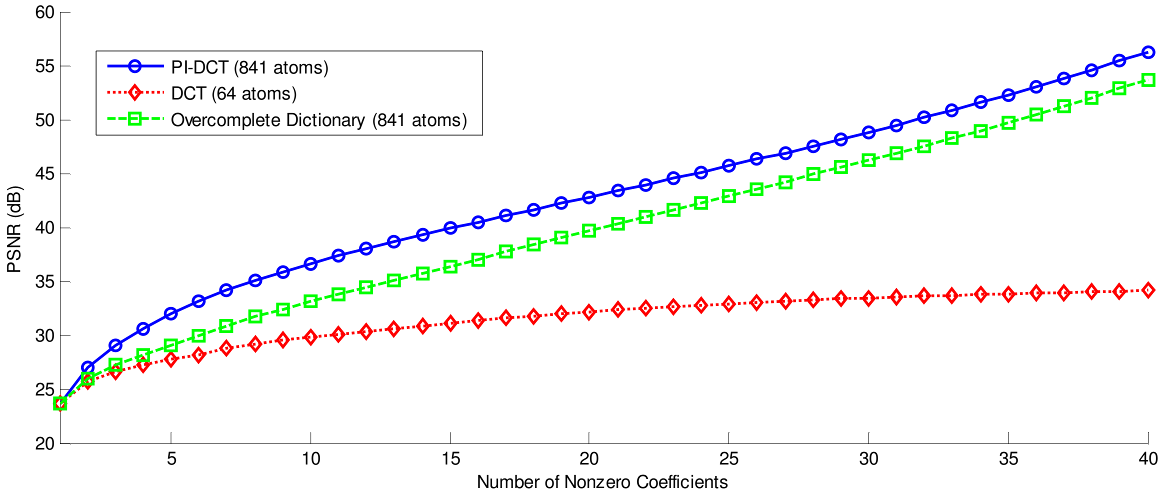


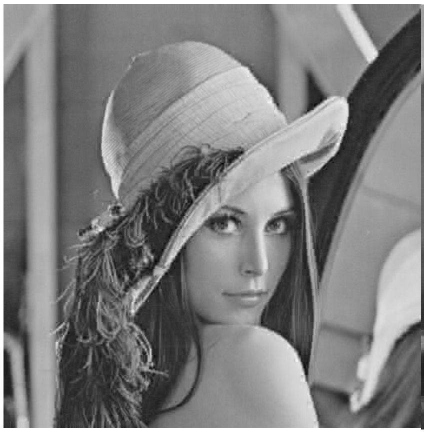
Figure 5.4. Demonstration of the reconstruction quality using the OMP method for PI-DCT, DCT and overcomplete DCT dictionaries. The patches of size 8*8 are extracted from the Lena photo (512*512). This figure shows that the PI-DCT offers a higher PSNR for the same number of atoms.

To evaluate the PI-DCT, 4096 patches are extracted from the Lena photo in which each patch is 8×8 pixels. The overcomplete DCT dictionary (Figure 5) is created assuming 29 frequency divisions from 0 to 2π without phase equal to 0. The PI-DCT is created with $N_\varphi = 4$, $n_y = 8$, $n_x = 8$. The OMP sparse coding method is applied because of its simplicity and fast performance with different number of nonzero coefficients. Figure shows a comparison between PI-DCT, DCT and overcomplete DCT dictionaries based on the number of nonzero-coefficients versus the PSNR calculated as shown below [35],

$$PSNR = 10\log_{10}(\frac{1}{MSE}) \tag{5.2}$$

where MSE is the mean square error between the original image and the reconstructed image. According to Figure5.4, the PI-DCT provides at least 2.5 dB more value of PSNR than the overcomplete DCT. The DCT dictionary fails to compete with PI-DCT and overcomplete DCT dictionaries. To have more illustration, Figure shows the Lena image reconstructed using the PI-DCT, DCT and overcomplete DCT dictionaries. Reconstructed images using the DCT and overcomplete DCT dictionaries suffers from high blocking artifacts, because they don't support phase information, whereas the reconstructed image of Lena using the PI-DCT eliminates this problem and provides a higher PSNR.

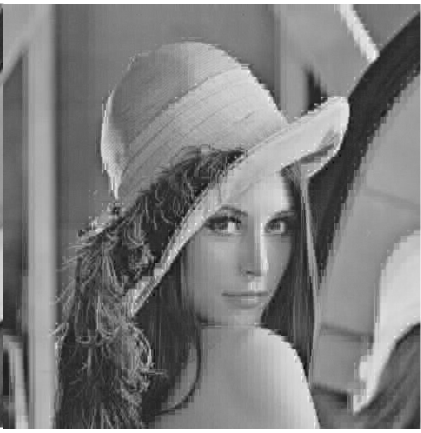
In Figure5.6, an investigation over the influence of $N_\varphi$on the reconstruction quality is shown with a different number of nonzero-coefficients. A significant improvement is attained by increasing N_φ from 1 (conventional DCT) to 2 which implies the importance of using at least two phase divisions to create a DCT dictionary. For $N_\varphi = 3$, the results are averagely improved by 1 dB in contrast with $N_\varphi = 2$. However, the number of atoms for $N_\varphi = 2$ and $N_\varphi = 3$ are 225 and 484, respectively. For $N_\varphi = 4$, the PSNR is less than 0.5 dB improved while the dictionary size is 841 atoms. For $N_\varphi = 5$, the dictionary size is increased to 1296 atoms and it improves the PSNR value to 1.3 dB more than the PSNR obtained with $N_\varphi = 3$. However, this is achieved by sacrificing the efficiency of representation. After $N_\varphi = 5$, the size of the dictionary increases while the reconstruction quality is not significantly improved. Therefore, the offered selections of N_φ are 2, 3 and 5. Figure shows the reconstruction results of 2 other standard images using OMP, St-OMP and MS-OMP methods.



PI-DCT

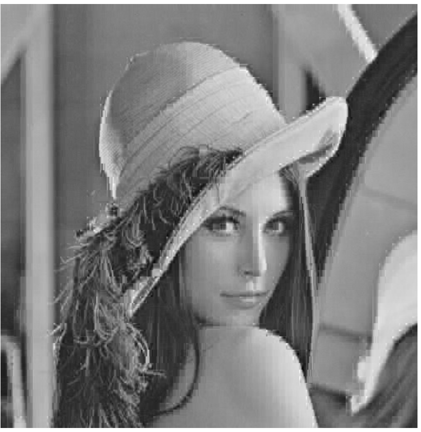
$N_\varphi = 4$, $n_y = 8$, $n_x = 8$, (841 atoms)

5 nonzero coefficients
PSNR = 31.98 dB



DCT

5 nonzero coefficients
PSNR =27.69 dB



Overcomplete DCT

5 nonzero coefficients
PSNR = 29.06 dB

Figure 5.5. Displaying reconstruction results using PI-DCT, DCT and overcomplete DCT dictionaries with 5 nonzero coefficients. Blocking artifacts can be seen for results obtained by the DCT and overcomplete DCT dictionaries whereas the PI-DCT perfectly reconstructed the Lena image without the blocking artifact.

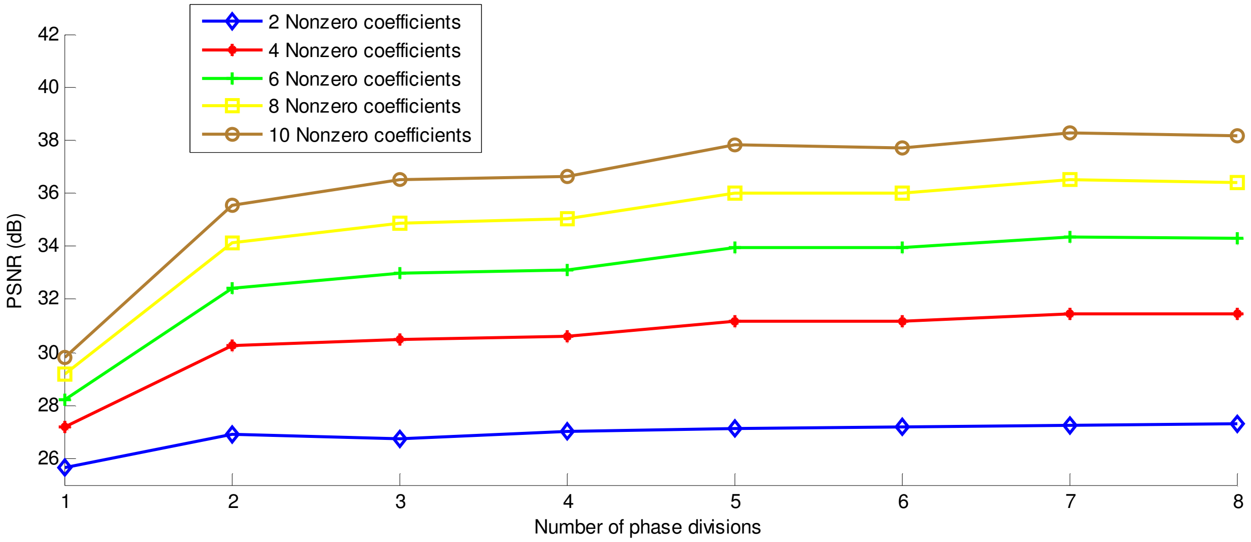
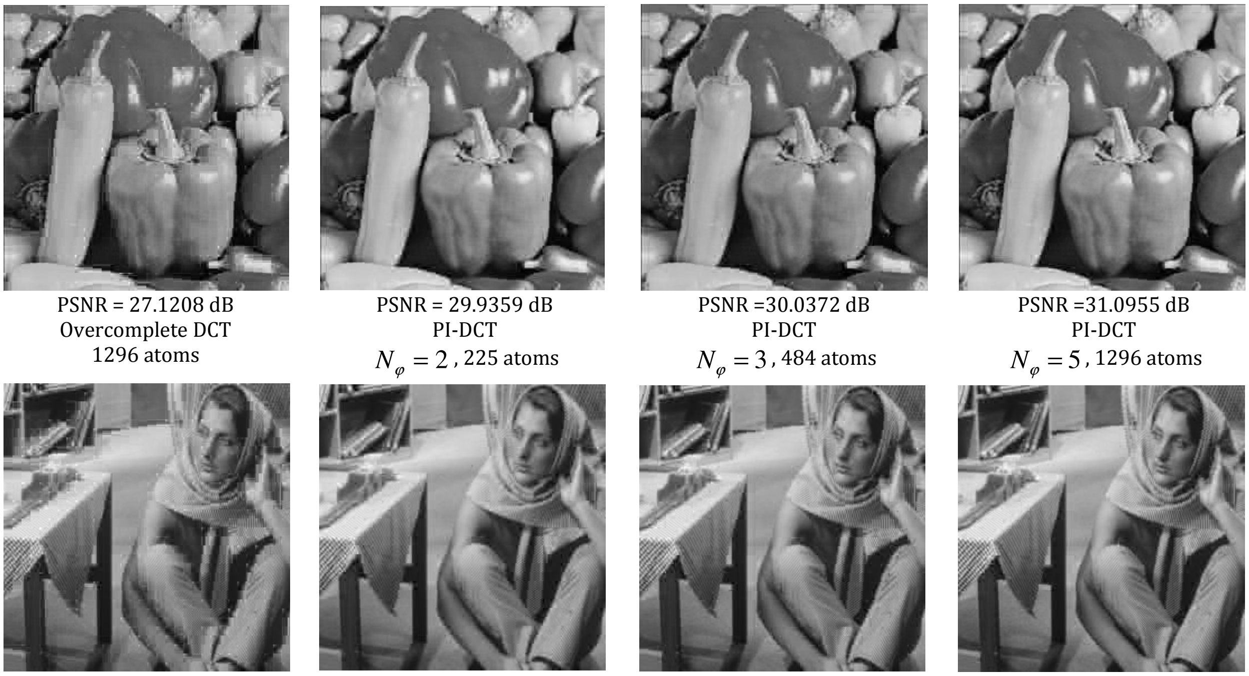


Figure 5.6. Demonstration of the reconstruction quality versus the number of phase divisions, N_φ. The sparse coding method is OMP and 4096 patches (8×8 pixels) are extracted from the Lena image to obtain these results. The PSNR increases by more than 5 dB when N_φ changes from 1 (conventional DCT) to 2.



PSNR = 27.1208 dB
Overcomplete DCT
1296 atoms

PSNR = 29.9359 dB
PI-DCT
$N_\varphi = 2$, 225 atoms

PSNR =30.0372 dB
PI-DCT
$N_\varphi = 3$, 484 atoms

PSNR =31.0955 dB
PI-DCT
$N_\varphi = 5$, 1296 atoms

PSNR = 27.1614 dB
Overcomplete DCT
1296 atoms

PSNR = 29.1307 dB
PI-DCT
$N_\varphi = 2$, 225 atoms

PSNR =29.834 dB
PI-DCT
$N_\varphi = 3$, 484 atoms

PSNR =30.8909 dB
PI-DCT
$N_\varphi = 5$, 1296 atoms

Figure 5.7. Demonstration of the quality of reconstruction using the PI-DCT and overcomplete DCT dictionaries. Both images are reconstructed using the OMP method with 6 nonzero-coefficients.

5.2. New Sparse Representation Method: MS-OMP

In section 3.3. Orthogonal Matching Pursuit, a brief introduction is presented on the orthogonal matching pursuit (OMP) and its derivatives which all try to increase the speed of the OMP method. We are proposing a new scheme to reduce the computational cost of the OMP method by selecting more than 1 atom per each stage. This implies a close relation with the St-OMP method [30]. However, an essential difference exists for selecting atoms in each stage which leads our proposed method to behave more accurate than the St-OMP. Indeed, the St-OMP uses a hard-thresholding approach to opt better matching atoms from the dictionary. This results in a high dependency with the threshold value which highly affects the quality and speed of this algorithm. To illustrate, the higher threshold selection causes less entering atoms while it increases the number of stages needed to maintain a certain sparsity condition, T_0. It improves the quality of representation by reducing the representation error, $\overline{r}$, in more stages. On the other hand, each stage needs pseudo-inverse computation these are computationally demanding. Therefore, selecting higher threshold values improves the quality by sacrificing the computational cost. By selecting a lower threshold value, the number of entering atoms increases leading to an uncontrollable increment in nonzero coefficients. Thus, the strong dependency of the threshold selection complicates solving the sparse representation problem conditioned with a defined sparsity level. In contrast, the proposed MS-OMP approach eliminates such a hard-threshold step and replaces it with a more robust group of atom selections. This gated entrance of atoms decreases the solution speed, but it provides more robustness and controllability while maintains a higher efficiency than the OMP method.

The proposed multi-stage approach finds a sorted set, of m atoms with the highest correlations to the reconstruction residual at each stage. Then, the MP method is employed to efficiently extract M atoms from the sorted set of m atoms, $m > M$ .The coefficients updating step computes the pseudo-inverse of the corresponding atoms multiplied with the input patch which is inspired from the OMP method. As in each stage M nonzero coefficients are added to the sparse vector, the number of stages is determined based on the sparsity condition, T_0. We first present a theoretical justification to show the mathematical background of the MS-OMP method. Then, more technical details are presented followed by comprehensive experimental evaluations.

5.2.1. Component-Based Signal Extraction

Similar to the St-OMP method, the input of each stage is the residual of the previous stage. The basic idea behind it is to minimize the stage error before passing it to the next stage. Each stage error is assumed to be a function which can be represented with a linear combination of atoms and the input patch is the summation of these functions (Signal Components). Suppose for an input patch, $\overline{P} \in \mathbb{R}^N$, there exists a linear combination of signal components, $\{\overline{f}_k \in \mathbb{R}^N\}$,

$$\overline{P} = \sum_{k=1}^{K} \overline{f}_k \tag{5.3}$$

In the matching step, the inner product of the residual and dictionary atoms feature those atoms which better project the residual. Therefore, for each stage, a set of atoms can be selected which maintains the following condition,

$$\left| \left\langle \overline{f}_k \cdot d_{ia} \right\rangle - \left\langle \overline{f}_k \cdot d_{ib} \right\rangle \right| < t_1^k,\ ia, ib \in U_k \tag{5.4}$$

The above condition only considers atoms of each stage. Another condition applies between atoms of k^{th} and $(k+1)^{th}$ stages as,

$$\left\langle \overline{f}_k \cdot d_i \right\rangle - \left\langle \overline{f}_{k+1} \cdot d_j \right\rangle > t_2^k,\ \ j \in U_{k+1} \text{ and } i \in U_k \tag{5.5}$$

where t_1^k and t_2^k are two values, $t_2^k \gg t_1^k$. $\langle . \rangle$ stands for the inner product of two vectors and U_k is a set of dictionary atom indexes which satisfies the above condition for the k^{th} component, $\overline{f}_k$. Having this definition, the inner products of the input patch and the dictionary atoms result in maximum values corresponding to $\bar{f}_1$ which are members of U_1. Maximum values related to the second component, $\bar{f}_2$, which are members of U_2 are achieved using the inner product of $res_1 = \bar{P} - \bar{f}_1$ and dictionary atoms. Other $\{U_k, k > 2\}$ are obtained using the inner product of res_{k-1} and atoms,

$$res_{k-1} = \bar{P} - \sum_{n=1}^{k-1} \bar{f}_n \tag{5.6}$$

Having $U = \{U_k, k = 1, \ldots, K'\}$ where $K' < K$, an approximation of $\bar{P}$, $\tilde{P} = \sum_{k=1}^{K'} \bar{f}_n$, is attained using the pseudo-inverse equation as,

$$\tilde{\alpha} = \left(D_U^T D_U\right)^{-1} D_U^T P \text{ and } \tilde{P} = D_U \tilde{\alpha} \tag{5.7}$$

5.2.2. Sparse coding solution using the Multi-Stage OMP

The proposed multi-stage OMP method, selects a set of atoms in each stage to minimize the previous stage residual. At the stage S, the inner product of dictionary atoms and res_{S-1}, assuming $res_0 = \bar{P}$, is computed to find candidate atoms entering the U_S to update $U = \{U_n, n = 1, \ldots, S\}$. The m highest values of the inner product corresponds to the candidate atoms. The candidate atoms are obtained and sorted using a fast sorting method. These candidate atoms have higher matching with the k^{th} component and they form a local dictionary, D_L. MP method [27] seeks among the D_L to find a combination of fix number, M, of atoms ($M \le m$) to efficiently represent res_{S-1} in each stage to approximate the corresponding component. After selecting M representing atoms of S^{th} stage, $U = \{U_n, n = 1, \ldots, S\}$ is updated and $\tilde{\alpha}$ is calculated using (5.7). This step resembles the update step of OMP while finding the representing atoms are handled through MP method atom finding strategy. Please note that for $M = 1$, the result of this method is exactly equal to the OMP method. After a specific limit, increasing M is not effective to reduce the component representing error and hence, a good selection of M is a tradeoff between accuracy and computational cost. Figure 5.8 shows the block diagram of the proposed sparse representation method. The pseudo-algorithm of the proposed method is presented in Table .

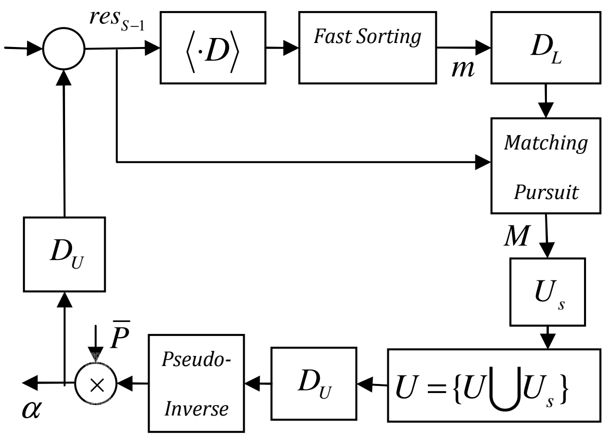


Figure 5.8. Demonstration of the block diagram of the proposed method (MS-OMP).

Table 5.1. Pseudo-algorithm of the MS-OMP method.

```
Init: $\overline{P}$, $res_0 = \overline{P}$, $\overline{\alpha} \to \overline{0}$, $D_U$
    for S=1:N
        $proj = \langle res_{S-1}, D \rangle$
        $ind = fastSorting(proj)$
        $D_L = \{d_{ind(1)}, ...., d_{ind(L)}\}$
        $res2 = res_{S-1}$
        $U_s \to empty$
        for n=1: $M$
            $m2 = \langle res, D_L \rangle$
            $ind2 = \arg\max(m2)$
            $U_s = \{U_s \bigcup ind(ind2)\}$
            $d' \to D_L(ind2)$
            $res2 = res2 - \frac{\langle res2, d' \rangle}{\langle d', d' \rangle} d'$
        end
        $U = U \bigcup U_s$
        $\overline{\alpha} = (D_U^T D_U)^{-1} D_U \overline{P}$
        $res_{S-1} = \overline{P} - D_U \overline{\alpha}$
    End
```

5.2.3. Fast Sorting Method

This sorting method is developed to efficiently find L highest values among all inner products of atoms and res_{S-1} at the S^{th} stage. First, a set of indexes and values $T \in \mathbb{R}^{2\times L}$ is initialized with all values equal to zero where the first row refers to maximum values and the second row consists of corresponding atom indexes. The inner product values are checked one by one and if the inner product value corresponding to the atom number n satisfies $v^n > T(1,m)$, this atom enters into T and its order within the set is determined using the bubble sort method. The column m containing the lowest value is eliminated at the end of sorting procedure.

5.2.4. Evaluations and Results

As described previously, our method is inspired by the OMP method intended to increase the efficiency of the sparse representation for large dictionaries. The St-OMP method is basically a close definition to our proposed method. However, there exist some essential differences between them. In order to make a realistic computational time comparison between our proposed method, OMP and St-OMP, we have developed these algorithms using the Microsoft C#. The result of implementation in MATLAB shows a non-reliable inconsistent behavior because of several issues regarding memory allocation, interpretational-based execution and etc. which may vary in speed in different computers. In other words, the way that the code is written highly affects the performance of the code execution in MATLAB. Instead, C# is a general purpose programming language which compiles the code before execution and provides a realistic framework to compare different methods in terms of computational time. The PI-DCT dictionary (Figure 5), consisting of 841 atoms is used to sparsely represent images using the conventional OMP method, St-OMP approach and the proposed MS-OMP method. Patches are taken without overlapping and each patch is 8×8 pixels. The comparison is based on the time of computations (TOC) and the Peak Signal to noise ratio (PSNR) computed as follows,

$$PSNR = 10\log_{10}\frac{1}{\sqrt{MSE^2}} \qquad (5.8)$$

$$MSE^2 = \frac{1}{N}\sum_i \left\| y_i - Dx_i \right\|_2^2, \quad i = 1,\dots,N$$

The proposed method is controlled using three parameters including: number of entering atoms to the MP block (m), number of selecting atoms per each stage (M) and number of non-zero coefficients T_0. The number of iterations is determined $iter = \left[T_0/M\right]$ where [.] takes the floor of T_0/M. The complexity order of the sorting module is specified by m. The effect of increasing m is displayed in Figure 5.9 which shows the representation quality (PSNR) in terms of changing m. Accordingly, its quality is increased until $m = 40$ and then no improvement is achieved after that. However, increasing m leads to proportional increment of the sorting computational time. The offered selected value is $m = 10$.

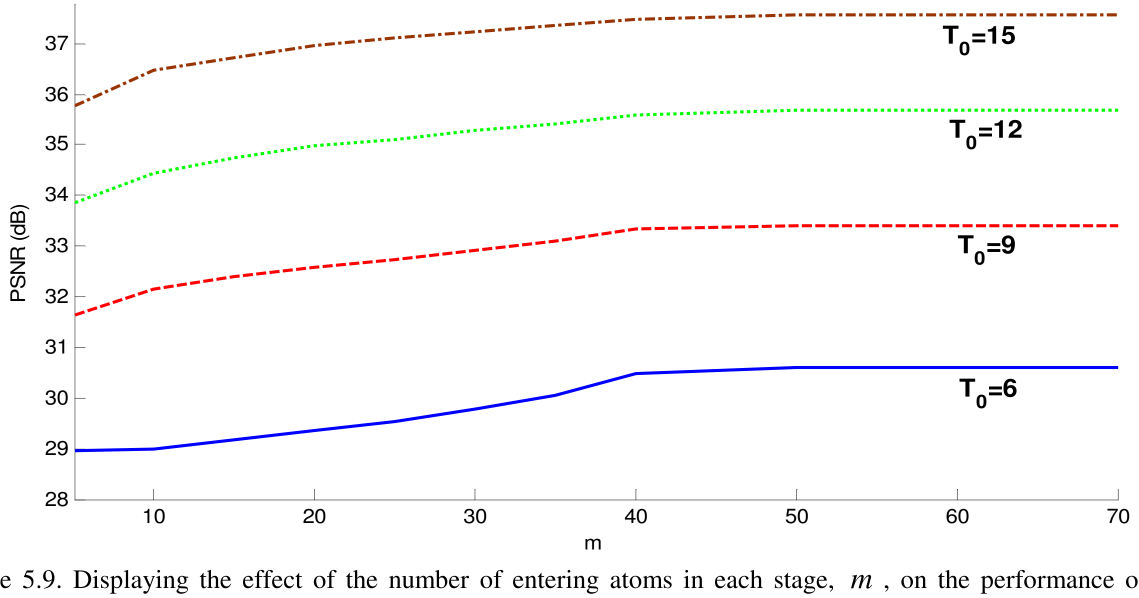


Figure 5.9. Displaying the effect of the number of entering atoms in each stage, m, on the performance of the proposed method. The number of selecting atoms in each stage, M, is set to 3. This is repeated for 4 different numbers of non-zero coefficients, T_0.

The second parameter of the proposed method which plays an important role in the performance of the proposed method is the number of selecting atoms in each stage, M. Increasing

M results in fewer iterations and reduces the computational time, because it performs fewer stages including MP block and pseudo-inverse calculation. In Figure 5.10, the PSNR changes of the MS-OMP method versus M is displayed for 4 different values of T_0. According to Figure 5.10, increasing M results in reducing the representation quality. Good selections for M are 3 and 4 which keeps the PSNR still high while these values suggest a good computational time reduction.

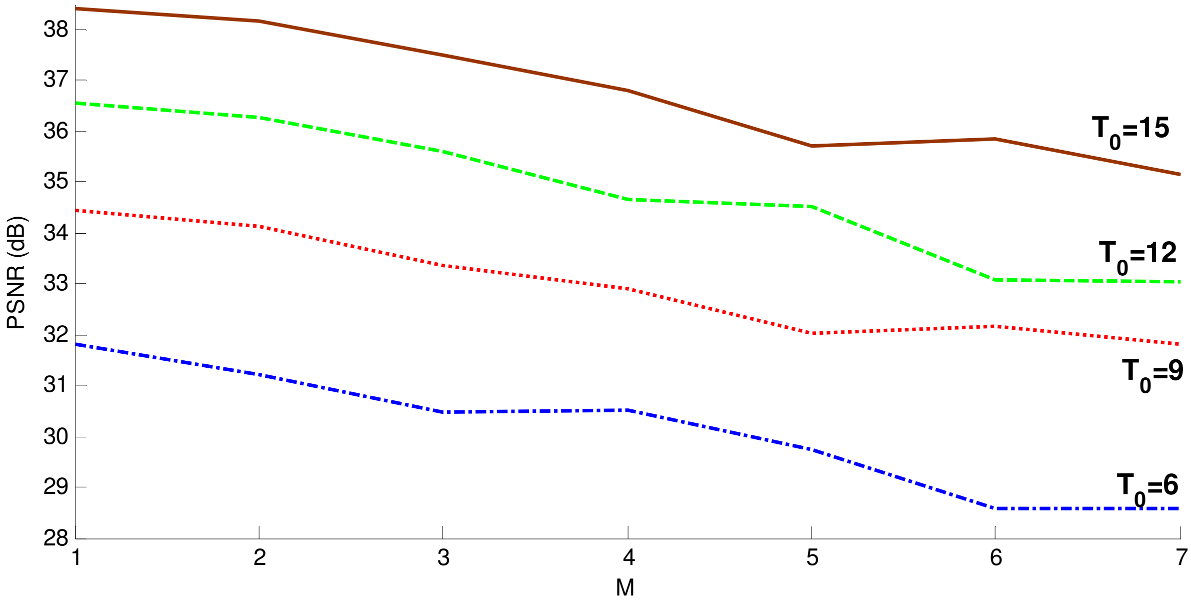


Figure 5.10. Demonstration of the influence of the number of non-zero coefficients, M, on the PSNR of the MS-OMP method. This graph is presented for 4 different number of non-zero coefficients, T_0.

The proposed method is faster than the OMP approach, because it performs fewer pseudo-inverse calculations to obtain the same number of nonzero coefficients, T_0. The St-OMP method applies hard-thresholding which is extremely dependent to the selection of the threshold; however it is faster than the MS-OMP, because it does not need any sorting block to select a set of m atoms with the highest correlations. This comes with an uncontrollable number of entering atoms to the sparse vector per each iteration and ruins the sparsity of the result. In Figure 5.11, the efficiency and accuracy of the proposed method is compared with conventional OMP and St-OMP methods for different T_0 (Other parameters are $m = 10$ and $M = 3$). It shows that the St-OMP method is much faster than our proposed method whereas the quality of MS-OMP stands much higher than the St-OMP, having the same sparsity level.

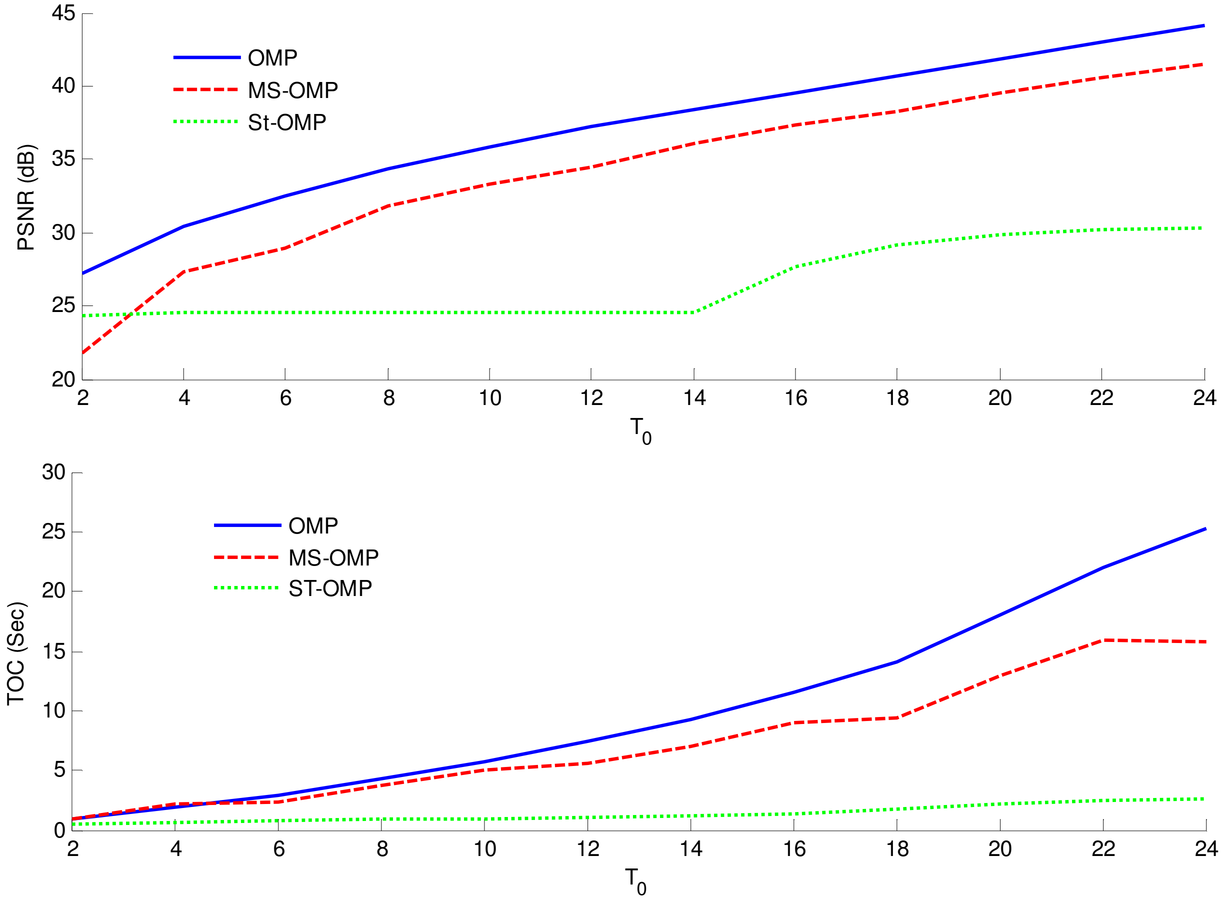
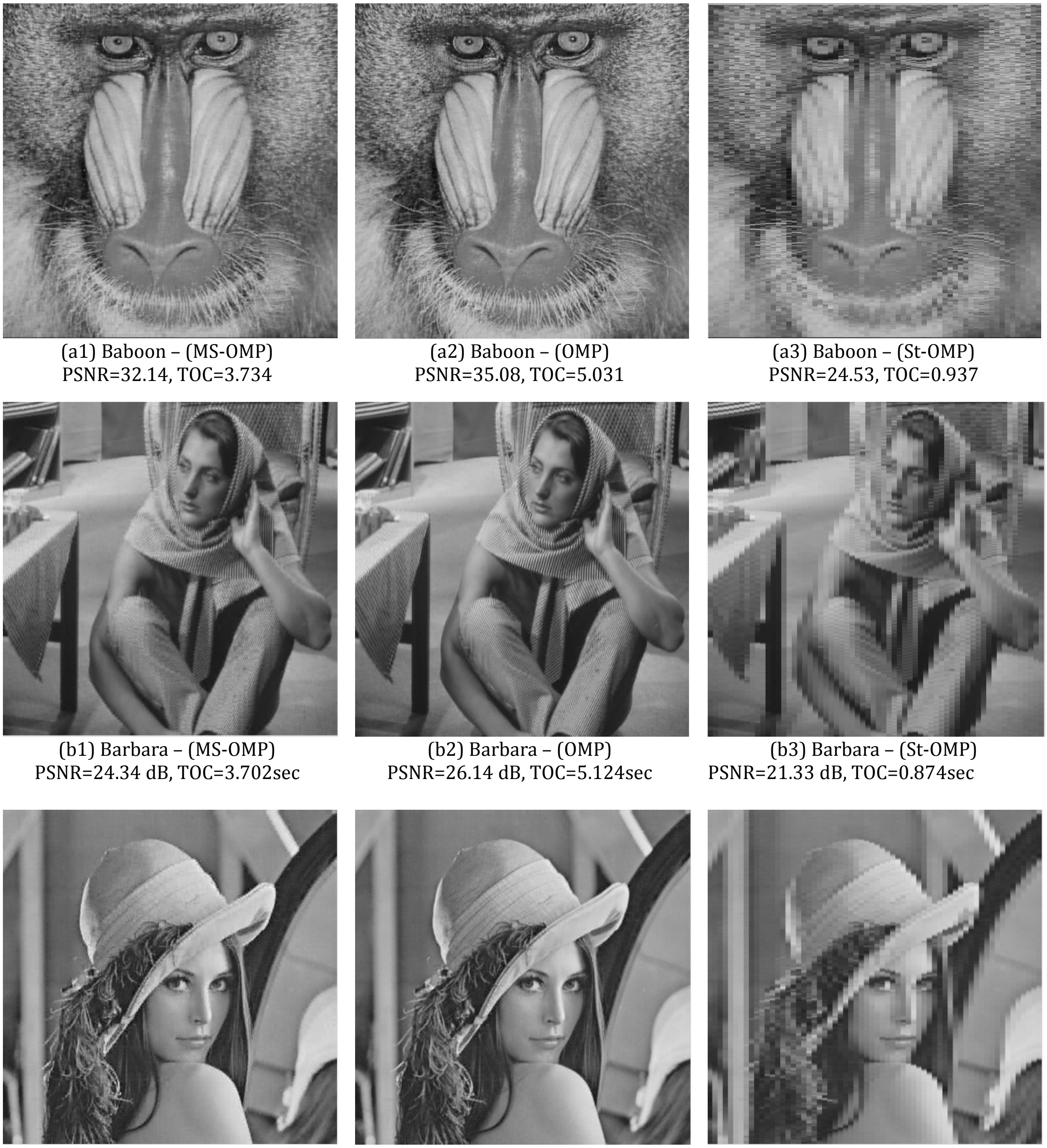


Figure 5.11. Demonstrating the result of comparison of time of computation and PSNR between OMP, MS-OMP, St-OMP versus the number of non-zero coefficients.

The reconstruction result of 3 standard images obtained using MS-OMP, OMP and St-OMP is shown in Figure 6 for $T_0 = 18$, $m = 10$ and $M = 3$. The importance of the proposed method is highlighted for large size dictionaries which the conventional OMP takes a long time to calculate the sparse vector. Instead, the designed method offers a better alternative with higher speed while maintaining a satisfactory sparse level. Figure 7 illustrates how the proposed method provides an advantage over the conventional OMP approach for larger dictionaries in terms of the computational time (TOC).



(a1) Baboon – (MS-OMP)
PSNR=32.14, TOC=3.734

(a2) Baboon – (OMP)
PSNR=35.08, TOC=5.031

(a3) Baboon – (St-OMP)
PSNR=24.53, TOC=0.937

(b1) Barbara – (MS-OMP)
PSNR=24.34 dB, TOC=3.702sec

(b2) Barbara – (OMP)
PSNR=26.14 dB, TOC=5.124sec

(b3) Barbara – (St-OMP)
PSNR=21.33 dB, TOC=0.874sec

(c1) Lena – (MS-OMP)
PSNR=28.52 dB, TOC=3.812sec

(c2) Lena – (OMP)
PSNR=31.64 dB, TOC=4.937sec

(c3) Lena – (St-OMP)
PSNR=22.47 dB, TOC=0.953sec

Figure 6. Displaying the result of reconstruction three standard images using MS-OMP, OMP and St-OMP.

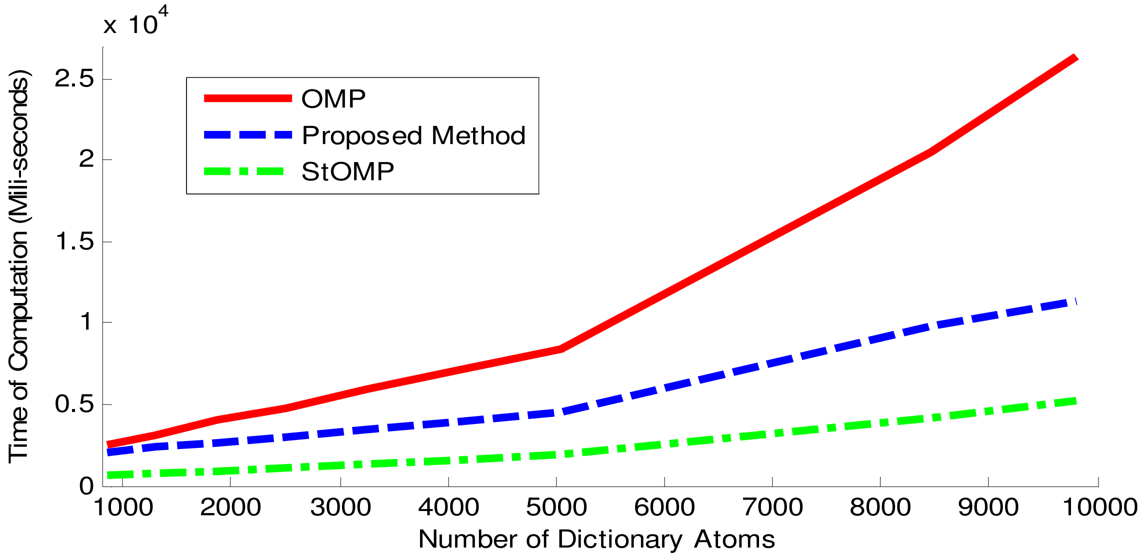


Figure 7. Comparing Time of computations of OMP, MS-OMP and St-OMP methods versus the number of atoms.

Chapter 6. New Dictionary Learning Approach: Variable Length K-SVD

Analytical dictionaries extract some characteristics of the input signal. They provide generic basis functions for all signals without any preference for some specific images. Learnt-based dictionaries evolved to provide more adaptation to the signal content leading to achieve better representation quality and higher sparsity level. The dictionary learning problem is formulated as,

$$\arg\min_{D,X}\left\{\left\|Y-DX\right\|_F^2\right\}, \text{ subject to } \left\|x_i\right\|_0 < T_0 \text{ and } X=\{x_i \mid i=1,\ldots,N\} \tag{6.1}$$

where $D \in \mathbb{R}^{n \times L}$ is the dictionary, $Y \in \mathbb{R}^{n \times N}$ is a set of input patches and X includes all sparse vectors representing input patches. Among all proposed algorithms for learning dictionaries, the K-SVD method has attracted a great deal of attention during last years, because of its simplicity and flexibility in working with different sparse coding methods. The optimization problem starts with an initiated dictionary, $D \in \mathbb{R}^{n \times L}$, and separately updates each dictionary atom, subject to minimize the reconstruction error calculated using the rest of the dictionary. The size of the predefined dictionary affects the time of computation and representation error. In essence, the number of atoms needed to perfectly represent all input patches relates to the content detail level of input patches. In other words, more patterns existing in input patches demands larger dictionary to be covered. Now, the question arises how we can determine the number of dictionary atoms for different applications.

6.1. Methodology

We are proposing new dictionary learning based on the K-SVD method which starts with only 1 atom and iteratively spreads until it converges to an efficient number of atoms adapted for each specific application. After each dictionary atom update, a weighted variance vector is calculated as,

$$v_j(l) = \frac{1}{N_L}\sum_{i=1}^{N_L} x_i(j)^2(\hat{y}_i(l) - d_j(l))^2 \ , \ \ \overline{v}_j = [v_j(1),\ldots,v_j(l),\ldots.v_j(n)] \tag{6.2}$$

where $\hat{y}_i(l)$ denotes the mean removed i^{th} input patch, $\overline{v}_j$ is the weighted variance vector and j refers to the updating atom index. The weighted variance considers the calculated sparse coefficients to determine how much a distance between $\hat{y}_i$ and d_j affects the variance calculated for d_j. In other words, patches which have greater coefficients corresponding to d_j, have more influence in calculating the weighted variance vector of d_j, and vise versa. The maximum value of the weighted variance vector is searched, $l_{\max} = \max\{v\}$, and if $v_j(l_{\max}) > th_2$ then the j^{th} atom is split into 2 atoms. In other words, the j^{th} atom is covering a wide volume in the $\mathbb{R}^n$ which works like a low-pass filter (averaging window) and it causes to loss input patch details. To solve this problem, this atom is divided into 2 atoms in the direction of maximum weighted variance and the dictionary size is increased by one atom as,

$$\begin{aligned} d^{(1)} &= d_j - \gamma v_j(l_{\max}) \\ d^{(2)} &= d_j + \gamma v_j(l_{\max}) \end{aligned} \tag{6.3}$$

$D = [d_1, ..., d_{j-1}, d^{(1)}, d^{(2)}, d_{j+1}, ..., d_L]$

where γ is a constant value. The dictionary is updated and its size is increased and thus, X should be updated as,

$$\bar{g}^T = X_j^T \qquad (6.4)$$
$$X = [X_1^T, X_2^T, ..., X_{j-1}^T, 0.5 \times \bar{g}^T, 0.5 \times \bar{g}^T, X_{j+1}^T, ..., X_L^T]$$

At each iteration the size of the dictionary changes and therefore this method is named, Variable Length k-SVD method (VLK-SVD). The presented atom insertion procedure reduces the sparse representation error. This is because the maximum weighted variance of the distance of patches with $\bar{d}^{(1)}$ and $\bar{d}^{(2)}$ are less than the maximum weighted variance of distances of patches with $\bar{d}_j$. Therefore, patches are more likely to be better represented after the atom insertion step. It implies that the new K-SVD method moves toward a better representation. The pseudo-algorithm of the proposed method is proposed in Table 6.1.

Table 6.1. The pseudo-Algorithm of the proposed method

```
1. Initialize $D = [ones(n,1), mean(Y)] \in \mathbb{R}^{n\times 2}$, $Y \in \mathbb{R}^{n,N}$ and $X = zeros(2,N)$, $L = 2$
2. $X = OMP(D,Y)$
3. For k=2 to $L$
       $D' = \{d_1, ..., d_{k-1}, d_{k+1}, ..., d_L\}$, $X' = [X_1^T, ..., X_{k-1}^T, X_{k+1}^T, ..., X_L^T]$, $g = X_k^T$
       $Ind$ := Find index of nonzero coefficients for $k^{th}$ atom
       $Y_R = \{y_j \mid j \in Ind\}$, $X_R = \{x_j \mid j \in Ind\}$, $E_R = Y_R - D'X_R$
       $[U, \Delta, V]$ := SVD decomposition
       $d_k = U_1$ and $g(\{ind\}) = V_1^T$
       Calculate $\bar{v}_j$ and find $l_{\max} = \max\{v\}$
       If $v_j(l_{\max}) > th_2$
           $d^{(1)} = d_j - \gamma v_j(l_{\max})$
           $d^{(2)} = d_j + \gamma v_j(l_{\max})$
           $D = [d_1, ..., d_{j-1}, d^{(1)}, d^{(2)}, d_{j+1}, ..., d_L]$
           $X = [X_1^T, X_2^T, ..., X_{j-1}^T, 0.5 \times \bar{g}^T, 0.5 \times \bar{g}^T, X_{j+1}^T, ..., X_L^T]$
           $k = k + 1$
           $L = L + 1$
       end
   end
```

```
4. $X = OMP(D,Y)$
5. For k=2 to $L$
       If $\sum_{j=1}^{N} X_{k,j}^{T} < th_1$ then remove $k^{th}$ atom from the dictionary
   end
6. If convergence is not achieved then go to step 2
7. End
```

6.2. Evaluations and Results

The proposed method is implemented in MATLAB (version 7.10) to evaluate its performance. Standard images including Lena, Barbara, Peppers, Goldhill and Baboon are used at the size of 512×512 pixels to show efficiency and accuracy of the proposed algorithm. As discussed in 6.1. Methodology, the proposed model is sensitive to three parameters including T_0, th_1 and th_2. T_0 is the number of nonzero coefficients of the sparse vector obtained in the sparse coding step. th_2 has an influence on the new atom creating process whereas th_1 controls the non-important atom removing procedure. We are targeting two purposes for preparing results in this section. The other existing parameter is γ, which is not deterministic to the result as much as the other three parameters. We arbitrarily selected γ to equal 1. Firstly, we want to provide a simple reasoning to show why we selected specific values for the parameters mentioned above. The second purpose which we target to address in this section is to illustrate the relationship between the input data complexity and the converging number of atoms in the proposed dictionary learning method. The accuracy of sparse representation is presented with the PSNR (5.8).

Some results are provided based on the Lena image because of its general usage with other works. In addition, other standard images are involved to demonstrate how the proposed method works with different images. At the end, a frequency domain analysis is performed to show how it affects the size of the learnt-based dictionary obtained by the proposed algorithm.

Figure6.1 shows how th_2 affect the performance of the VLK-SVD method for $th_1 = 15$ and $T_0 = 18$. This is obtained by changing th_2 from 0.8 to 2. Increasing th_2 decreases the number of dividing atoms which satisfy $v_j(l_{\max}) > th_2$. This fact is demonstrated in Figure where the less number of atoms is attained by selecting $th_2 = 2$ or $th_2 = 1.8$. According to Figure, there is a good trade of between the number of atoms and PSNR is obtained by selecting $th_2 = 1.6$.

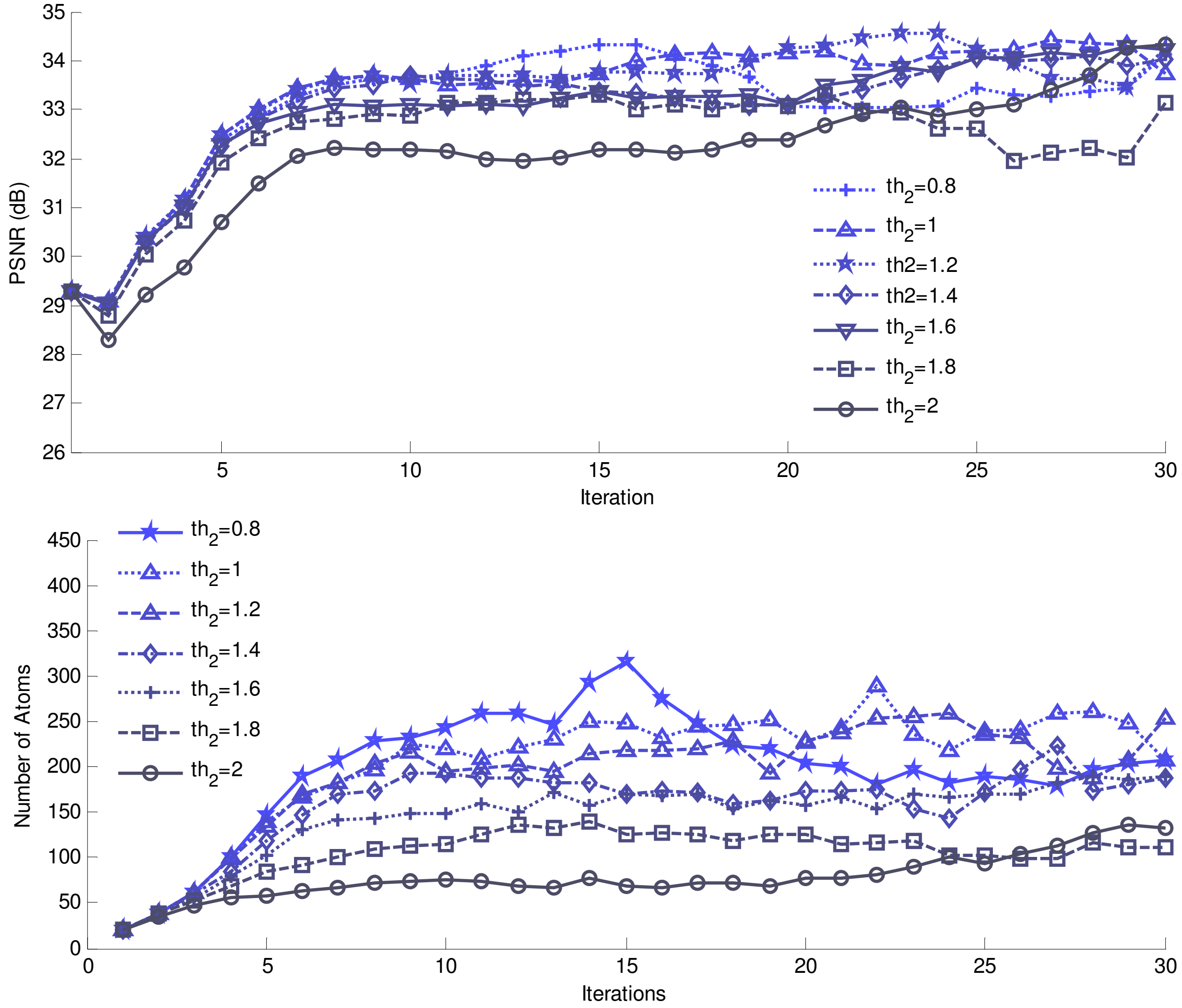


Figure 6.1. Demonstration of the influence of the threshold th_2 on the proposed method. The top figure shows the PSNR versus iterations while the bottom figure displays the number of atoms versus iterations. (Results are obtained based on the Lena Image)

In Figure, the influence of th_1 on the PSNR and number of atoms is displayed while $th_2 = 1.6$ and $T_0 = 18$. th_1 works as a threshold for determining insignificant atoms which should be removed from the dictionary. For a dictionary with a small number of atoms, each atom has a greater chance to be involved in representing input patches. As the size of the dictionary increases during iterations, the number of represented patches using some atoms decreases. This is the way that some atoms become insignificant after some iteration. th_1 is the threshold to select insignificant atoms and their removal increases the efficiency of the dictionary. According to the Figure, selecting $th_1 = 15$ provides more stability.

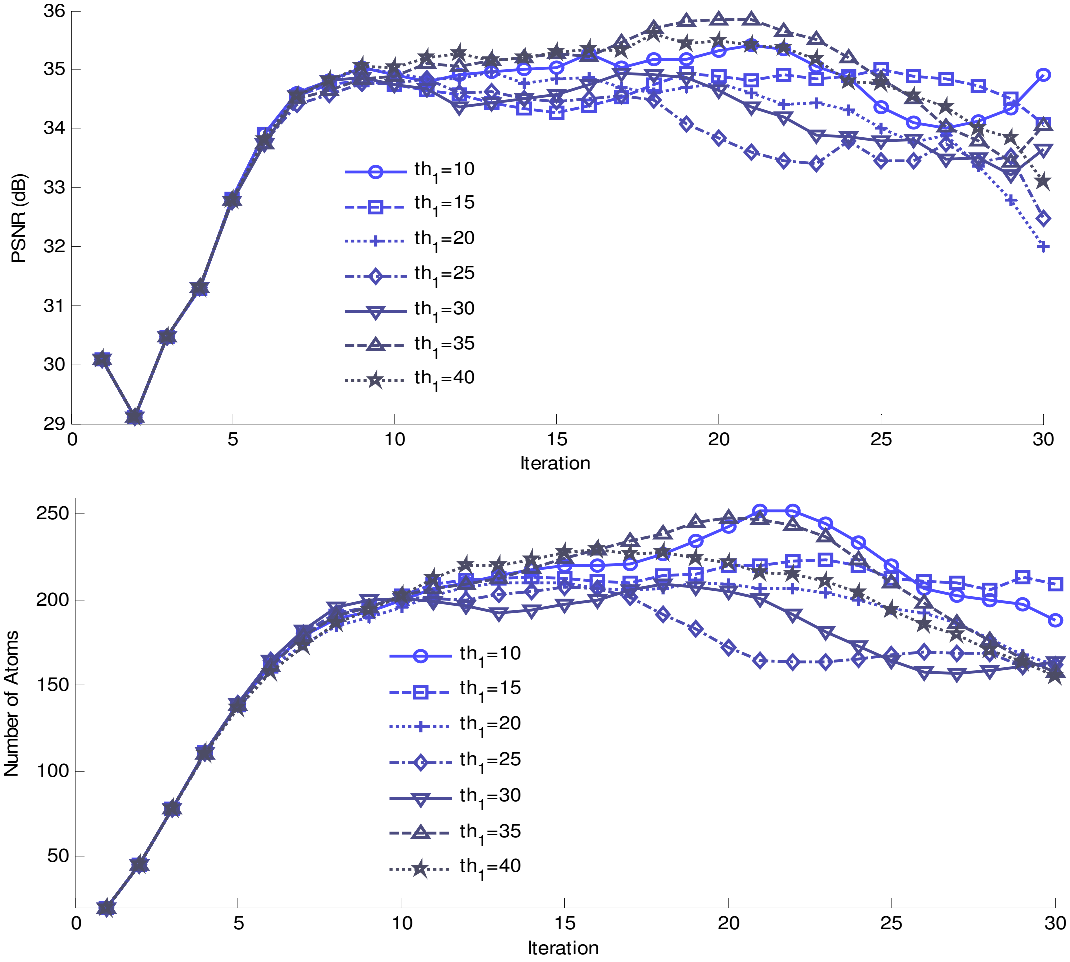


Figure 6.2. Displaying the result of changing th_1 and its effect on the PSNR and Number of atoms (Results are obtained based on the Lena Image)

The number of non-zero coefficients, T_0, has a direct control on the quality of sparse coding. It means that more non-zero coefficients imply more involving atoms from the dictionary to represent an input patch. Figure shows the result of iterative process of dictionary learning with different T_0 values, selecting $th_1 = 15$ and $th_2 = 1.6$. According to Figure, selecting $T_0 = 6$ results in a large number of inserted atoms to sufficiently cover all patches. Selecting $T_0 = 18$ provides a good SNR value while the number of non-zero coefficients stays similar to other T_0 values.

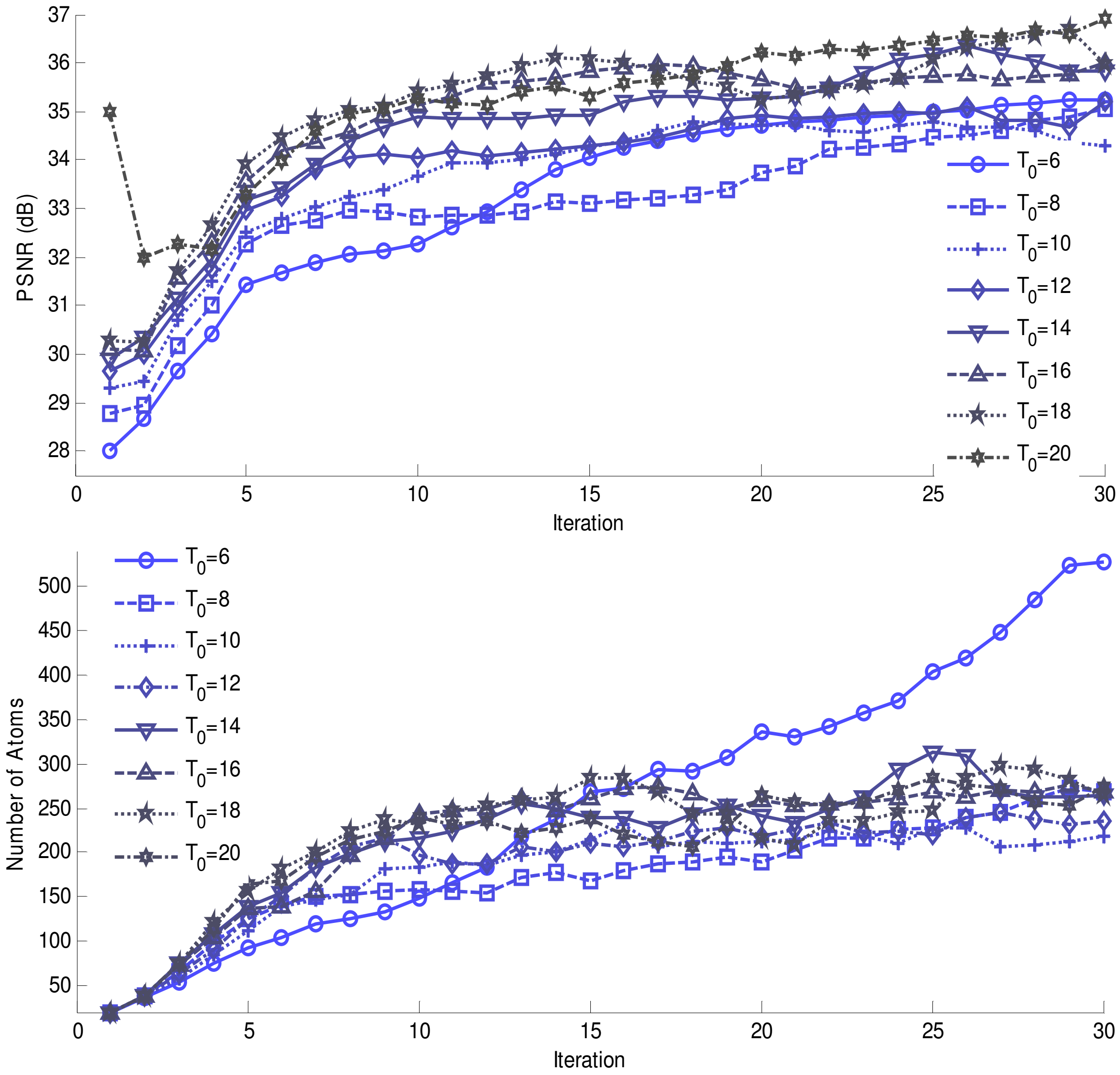
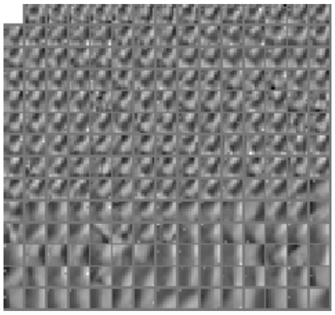


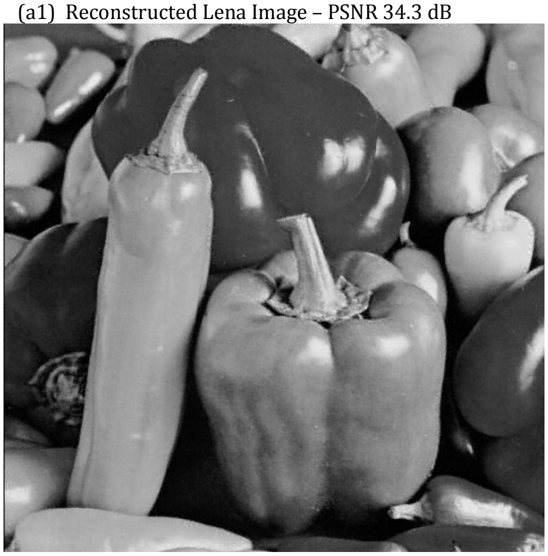
Figure 6.3. Displaying the effect of selecting different values for non-zero coefficients of the sparse coding step on the performance of the proposed method. (Results are obtained based on the Lena Image)

The selected values for parameters $th_1 = 15$, $th_2 = 2$ and $T_0 = 18$ are used to learn dictionaries for different images in 50 iterations, specified in Figure. The Baboon image has more details therefore demanding more dictionary atoms (624 atoms) to accurately cover all patches whereas the Goldhill photo is sufficiently represented using only 100 atoms. The other interesting point which proves the importance of our proposed method is the appearance of dictionaries in Figure for different images. In fact, the complexity of input patches determines how many atoms are needed to provide a representative dictionary for input patches. The conventional K-SVD method takes a fixed-size dictionary and there is no reasoning behind the size of the dictionary.

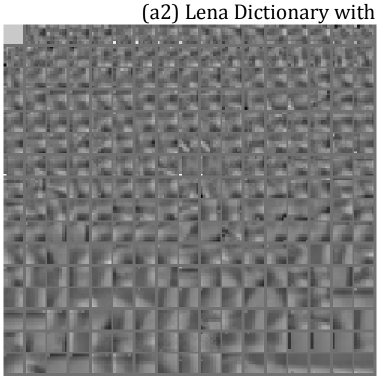
(a1) Reconstructed Lena Image – PSNR 34.3 dB



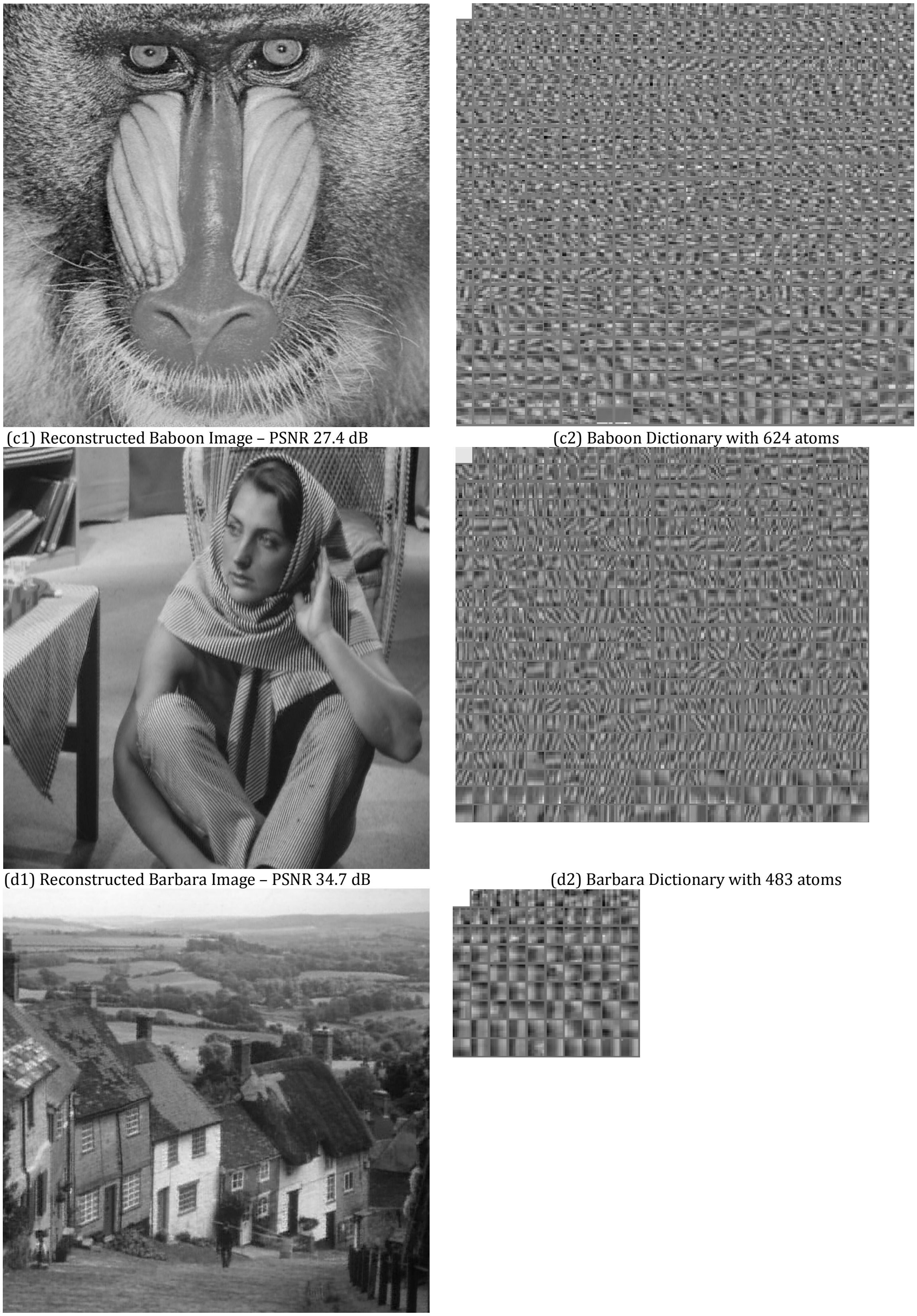
(a2) Lena Dictionary with 225 atoms



(b1) Reconstructed Peppers Image – PSNR 34.2 dB



(b2) Peppers Dictionary with 272 atoms



(c1) Reconstructed Baboon Image – PSNR 27.4 dB

(c2) Baboon Dictionary with 624 atoms

(d1) Reconstructed Barbara Image – PSNR 34.7 dB

(d2) Barbara Dictionary with 483 atoms

(e1) Reconstructed Goldhill Image – PSNR 31.6 dB

(e2) GoldhillDictionary with 90 atoms

Figure 6.4. Displaying results of 5 standard images using the proposed method.

To compare the proposed method with the conventional K-SVD method, the K-SVD method is applied to learn dictionaries for the 5 abovementioned standard images with 484 atoms in the dictionary created using the overcomplete DCT dictionary. For the K-SVD method, the number of non-zero coefficients is selected to be 18.

In Table 6.2, a comparison between the proposed VLK-SVD method and the K-SVD method is presented. The K-SVD method is highly dependent to the initial dictionary selection and it affects the final converged result. The K-SVD improved the reconstruction results obtained by the overcomplete DCT. In contrast, the VLK-SVD starts from only 20 atoms (It can start with only 1 atom) and converges to an optimum number of atoms needed to reconstruct the image. In the Goldhill image reconstruction, the result of VLK-SVD is astonishingly better than the K-SVD approach while the VL-KSVD creates a dictionary with only 90 atoms.

Table 6.2. Comparison of the proposed method (VLK-SVD) and the K-SVD dictionary learning method.

Image Name	VLK-SVD		K-SVD		Overcomplete DCT	
	PSNR (dB)	Number of Atoms	PSNR (dB)	Number of Atoms	PSNR (dB)	Number of Atoms
Lena	34.32	225	29.95	484	23.66	484
Peppers	34.24	272	26.91	484	26.59	484
Baboon	27.43	624	24.54	484	20.38	484
Barbara	34.71	483	24.62	484	23.66	484
Goldhill	31.62	90	28.89	484	25.51	484

6.3.1. Frequency Domain Analysis

It is to our interest to show the exciting relationship between the convergence of the proposed method and the frequency domain information of input patches. Higher frequency components correspond to edges which needs more atoms to be finely represented, whereas lower frequency components refer to homogeneous regions in the image which can be represented with a few atoms including the DC atom (which consists of all 1 elements).

The efficiency of the proposed method comes with the fact that the frequency domain information affects the number of atoms in the converged result. In other words, the dictionary size is automatically determined based on the detail level which is related to the frequency domain contents of input patches. In order to prove this relation, a Gaussian low-pass filter with a window size 15×15 and different sigma values, $\sigma^2 \in \{1,2,3,4,5\}$ from 1 to 5 is applied to the Lena image where higher values of σ^2 corresponds to lower passing frequency band. Then the dictionary learning procedure is performed to find the number of atoms for each low-passed image with different sigma values. The selected values for parameters $th_1 = 15$, $th_2 = 2$ and $T_0 = 18$ are used to learn dictionaries in 25 iterations. The initial dictionary contains 20 atoms of a completed size DCT dictionary.

The result of this evaluation is presented in Figure 6.5. Accordingly, the number of atoms of learnt-dictionaries decreases when σ^2 of the Gaussian low-pass filter increases. For the original image where no low-pass filtering is applied, the dictionary is converged to 150 atoms whereas for $\sigma^2 = 5$, the learnt dictionary contains only 22 atoms. It means that an image with lower frequency contents need fewer atoms to be sparsely presented and the proposed method is highly sensitive to this.

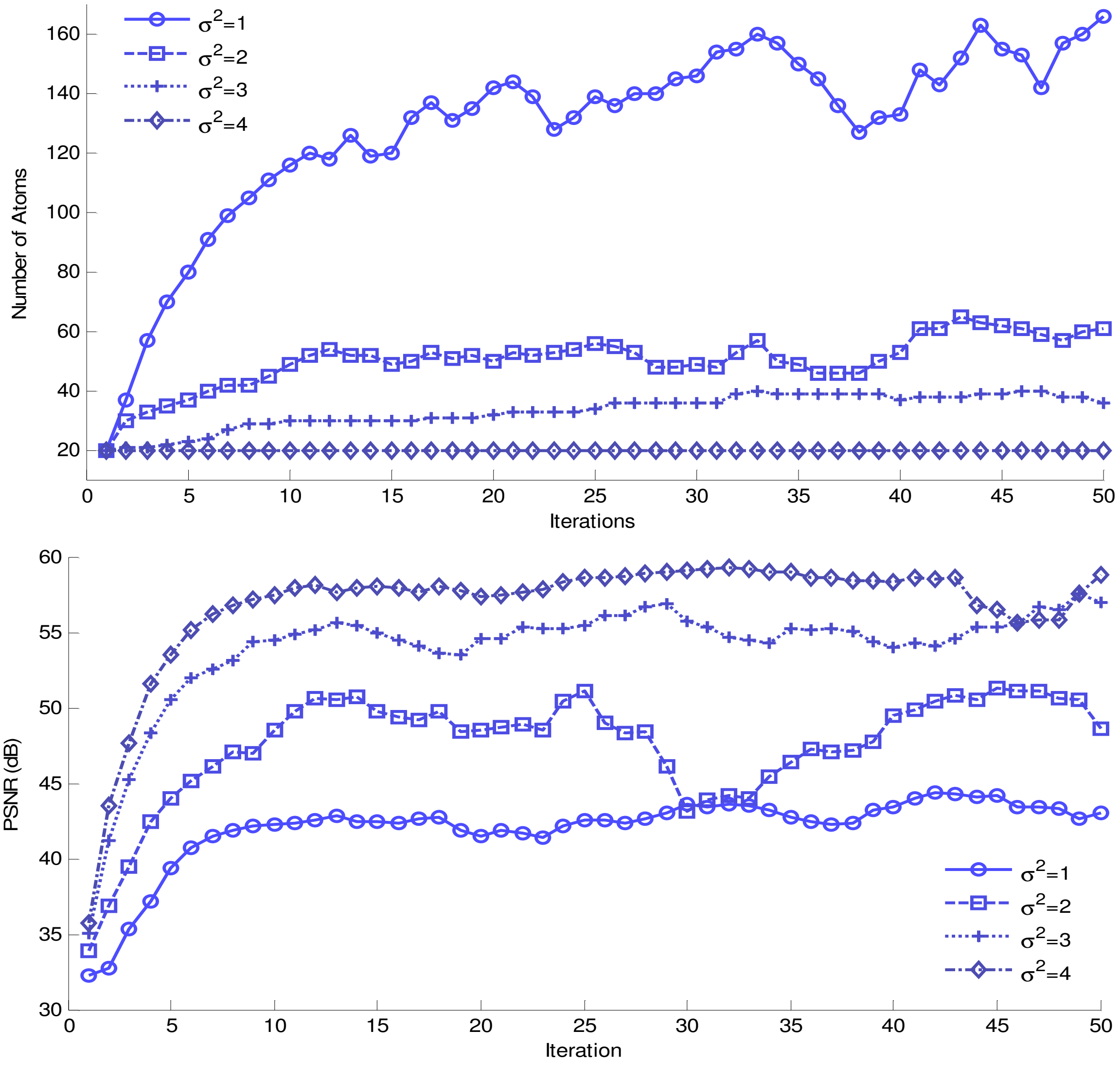


Figure 6.5. Demonstration of the relation between the number of atoms and the frequency domain information of input patches. A Gaussian low-pass filter with 15x15 window size and different sigma values is applied on the Lena image before learning the dictionary.

Chapter 7. Conclusion and Discussion

7.1. Discussion on Phase Included DCT (PI-DCT)

We presented a new analytical dictionary based on the DCT dictionary in which the atoms are created by changing both frequency and phase of the cosine function in vertical and horizontal directions. Using this definition, both phase and magnitude of components for all signal patches are extracted. In other words this dictionary provides an approximation of the Fourier transform. The conventional DCT dictionary only applies frequency variations to create atoms and phase information will be lost. The DCT dictionary is sensitive to phase shifts that result in blocking artifacts. Our proposed PI-DCT dictionary eliminates this problem and provides a more accurate signal representation. We have evaluated the accuracy of the proposed dictionary for different number of phase divisions. The results demonstrate that the new PI-DCT is a suitable dictionary to sparsely represent images as compared to other conventional analytical dictionaries. The obtained results show that having 2 phases including 0 and π are necessary to improve the result and the most efficient result is obtained using 3 phase divisions. Increasing the number of phases results in a larger dictionary and after 7 phase divisions, the enhancement achieved is negligible.

In this thesis, the number of phase divisions for all frequencies are the same. In the future work, it is possible to have more phase divisions for lower frequencies and fewer phases divisions for higher frequencies. This provides a better definition which improves the efficiency of the PI-DCT dictionary.

7.2. Discussion on Multi-Stage Orthogonal Matching Pursuit

We presented a new sparse coding algorithm that is based on the conventional orthogonal matching pursuit method. The conventional OMP calculates T_0 pseudo-inverse transforms to obtain T_0 non-zero coefficients which leads to a massive computations complexity. In the proposed method multiple atoms per each stage are selected, and therefore the number of pseudo-inverses needed to obtain T_0 non-zero coefficients are reduced. The proposed method is more efficient than the conventional OMP method. We compared the performance of the MS-OMP with the St-OMP method which is the most similar method to the proposed approach (MS-OMP). The evaluation results show that the St-OMP method is faster than the proposed method, however, the MS-OMP method provides much higher accuracy. The St-OMP method has no control on the number of entering atoms to the non-zero coefficients in the sparse vector per each stage whereas the MS-OMP determines the number of selected atoms as the output of the MP block per each stage. The proposed MS-OMP method has three parameters which provide a flexible reconstruction accuracy. The offered values for these parameters provide a trade of between the efficiency and accuracy of the sparse representation.

In future work, the MP block can be replaced with another selecting strategy based on the covariance matrix of atoms selected by the sorting block. This can result in a faster sparse coding. Moreover, it is possible to combine this approach with the Cholesky-OMP method to provide much faster sparse representation algorithm.

7.3. Discussion on Variable Length K-SVD

In this thesis, we proposed a novel enhancement on the conventional K-SVD dictionary learning method. The variable length K-SVD approach starts with only one atom in the feature space and applies a spreading strategy to create a sufficient number of atoms to precisely encompass all input patches in the feature space. This provides a framework to create a dictionary with an optimal number of atoms well-suited for each set of input images. The conventional K-SVD approach minimizes the reconstruction error of a fixed size dictionary and it lacks from a strong reasoning for the number of atoms in the dictionary.

The proposed dictionary learning approach iteratively updates and adds atoms until it converges to an optimal point. The number of added atoms depends on the level of details inside the training input patches. There is a relation between the frequency response of the training images and the number of atoms. We applied a low pass filter to reduce the band-width of the image frequency domain information. A stronger low-pass filter results in creation of fewer dictionary atoms.

The other problem of the conventional K-SVD dictionary learning is that it is highly sensitive to the selection of the initial dictionary. If the initial dictionary is unevenly distributed among the input patches in the feature space, some of patches are represented with more details in a dense area of atoms. But some others are poorly represented which are placed in a volume in the feature space where fewer number of atoms are placed. The proposed dictionary learning method (VLK-SVD) tries to insert atoms in the feature space to evenly cover the information.

The proposed method starts with a few atoms and spreads until convergence. In first iterations the dictionary contains fewer atoms and the dictionary learning approach performs faster. Therefore, the proposed dictionary learning method is faster than the conventional K-SVD method.

In future work, this method can be applied on different applications such as medical image representation and compress sensing, image classification and etc.

References

[1] R. Rubinstein, A. M. Bruckstein and M. Elad, "Dictionaries for Sparse Representation Modeling," *Proceedings of the IEEE,* vol. 98, no. 6, pp. 1045-1057, June 2010.

[2] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology,* vol. 160, no. 1, pp. 106-154, 1962.

[3] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing,* vol. 14, no. 12, pp. 2091-2106, 2005.

[4] U. Y. Desai, "Coding of Segmented Image Sequences," 1994.

[5] U. Y. Desai, "DCT AND WAVELET BASED REPRESENTATIONS OF ARBITRARILY SHAPED IMAGE SEGMENTS," in *International Conference on Image Processing, 1995.*, 1995.

[6] A. B. Watson, "Image Compression Using the Discrete Cosine Transform," *Mathematica Journal,* vol. 4, no. 1, pp. 81-88, 1994.

[7] N. Ahmed, T. Natarajan and R. K.R., "Discrete Cosine Transform," *IEEE Transactions on Computers,* Vols. C-23, no. 1, pp. 90-93, 1974.

[8] C. K. Choi, "Image Compression Using Wavelet Transform," University of Texas at Arlington,

1993.

[9] S. Mallat and Z. Zhang, "Mallat,S.G.; Zhifeng Zhang," *IEEE Transactions on Signal Processing,* vol. 41, no. 12, pp. 3397-3415, 1993.

[10] M. S. Lewicki and B. A. Olshausen, "Probabilistic framework for the adaptation and comparison of image codes," *Michael S. Lewicki; Bruno A. Olshausen,* vol. 16, no. 7, pp. 1587-1601, 1999.

[11] R. Mersereau and T. Speake, "The processing of periodically sampled multidimensional signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 31, no. 1, pp. 188-194, 1983.

[12] R. H. Bamberger and M. J. T. Smith, "A filter bank for the directional decomposition of images: theory and design," *Bamberger,R.H.; Smith,M.J.T.,* vol. 40, no. 4, pp. 882-893, 1992.

[13] S. I. Park, M. J. T. Smith and M. R. Mersereau, "Improved structures of maximally decimated directional filter Banks for spatial image analysis," *IEEE Transactions on Image Processing,* vol. 13, no. 11, pp. 1424-1431, 2004.

[14] C. d. Amaral and A. Luiz, "Geometrical representation, processing, and coding of visual information," Urbana-Champaign, 2007.

[15] A. K. Jain, "A Fast Karhunen-Loeve Transform for Digital Restoration of Images Degraded by White and Colored Noise," *IEEE Transactions on Computers,* Vols. C-26, no. 6, pp. 560-571, June 1977.

[16] I. T. Jolliffe, Principal component analysis, ebrary, Inc, 2002.

[17] A. d'Aspremont, L. E. Ghaoui, M. I. Jordan and G. R. G. Lanckriet, "A Direct Formulation for

Sparse PCA Using Semidefinite Programming," *SIAM Review,* vol. 49, no. 3, p. 434, 2007.

[18] V. Deepu, S. Madhvanath and A. G. Ramakrishnan, "Principal component analysis for online handwritten character recognition," *Proceedings of the 17th International Conference on Pattern Recognition,* vol. 2, pp. 327-330, 2004.

[19] P. J. B. Hancock, A. M. Burton and V. Bruce, "Face processing: Human perception and principal components analysis," *Memory & cognition,* vol. 24, no. 1, pp. 26-40, 1996.

[20] D. Mishra, R. Dash, A. K. Rath and M. Acharya, "Feature Selection in Gene Expression Data Using Principal Component Analysis and Rough Set Theory," *Biomedical and Life Sciences,* pp. 91-100, 2011.

[21] H. Zou, T. Hastie and R. Tibshirani, "Sparse Principal Component Analysis," *Journal of Computational and Graphical Statistics,* vol. 15, no. 2, pp. 265-286, 2006.

[22] R. Hamdan, F. Heitz and L. Thoraval, "A low complexity approximation of probabilistic appearance models," *Pattern Recognition,* vol. 36, no. 5, pp. 1107-1118, 2003.

[23] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society.Series B (Methodological),* vol. 58, no. 1, pp. 267-288, 1996.

[24] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* vol. 67, no. 2, pp. 301-320, 2005.

[25] I. T. Jolliffe, N. T. Trendafilov and M. Uddin, "A Modified Principal Component Technique Based on the LASSO," *Journal of Computational and Graphical Statistics,* vol. 12, no. 3, pp. 531-547, 2003.

[26] J. H. Friedman and W. Stuetzle, "Projection Pursuit Regression," *Journal of the American Statistical Association,* vol. 76, no. 376, pp. 817-823, 1981.

[27] S. G. Mallat and Z. Zhifeng, "Matching Pursuits With Time-Frequency Dictionaries," *IEEE Transactions on Signal Processing,* vol. 41, no. 12, pp. 3397-3415, 1993.

[28] Y. C. Pati, R. Rezaiifar and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers,* vol. 1, pp. 40-44, 1993.

[29] R. Rubinstein, M. Zibulevsky and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," Technical Report - CS Technion, 2008.

[30] D. Donoho, Y. Tsaig, I. Drori and J. Starck, "Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory,* vol. 58, no. 2, pp. 1094-1121, February 2012.

[31] C. Shaobing and D. Donoho, "Basis pursuit," *Conference Record of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers,,* vol. 1, pp. 41-44, 1994.

[32] S. S. Chen, L. D. Donoho and M. A. Saunders, "Atomic Decomposition by Basis Pursuit," *SIAM Review,* vol. 43, no. 1, pp. 129-159, 2001.

[33] F. A. Potra and S. J. Wright, "Interior-point methods," *Journal of Computational and Applied Mathematics,* vol. 124, no. 1, pp. 281-302, 2000.

[34] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural computation,* vol. 12, no. 2, pp. 337-365, 2000.

[35] M. Aharon, M. Elad and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Transactions on Signal Processing,* vol. 54, no. 12, pp. 4311-4322, 2006.

[36] K. Engan, S. O. Aase and J. H. Husoy, "Method of optimal directions for frame design," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999.

[37] K. Engan, B. D. Rao and K. Kreutz Delgado, "Frame design using focuss with method of optimal directions (mod)," *Norwegian Signal Processing Symposium,* pp. 65-69, 1999.

[38] H. Lee, A. Battle, R. Raina and A. Y. Ng, "Efficient Sparse Coding Algorithms," in *Advances in Neural Information Processing Systems 19*, 2007.