

DICTIONARIES AND ALGORITHMS FOR SPARSITY CONSTRAINED IMAGE RECONSTRUCTION

by

Mathiruban Tharmalingam, B.Sc
University of Toronto, 2007

A thesis
presented to the University of Ryerson

in partial fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2013
© Mathiruban Tharmalingam, 2013

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

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

I further authorize Ryerson University to reproduce this thesis 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.

I understand that my thesis may be made electronically available to the public.

(Mathiruban Tharmalingam)

ABSTRACT

Dictionaries and Algorithms for Sparsity Constrained Image Reconstruction

Master of Applied Science

Electrical and Computer Engineering

Mathiruban Tharmalingam

Ryerson University, 2013

There has been a growing interest in the different types of dictionaries that can be used in image processing applications. We propose a hybrid dictionary composed of transform based atoms and additional nonlinear atoms generated using the polynomial, rectangular and exponential functions. The additional nonlinear atoms improve signal reconstruction quality for both transient and smooth signals. To further improve signal reconstruction quality, we optimize the hybrid dictionary using training samples from the signal. We also propose a signal coding algorithm that generates additional atoms by performing a circular shift on the provided dictionary prior to coding the signal.

We have evaluated the proposed methods against existing predefined dictionaries by visually examining the reconstructed images as well as evaluating the peak signal to noise ratio of the reconstructed signal. All methods proposed in this thesis improved signal reconstruction quality however; we require an in-depth cost analysis study to evaluate its limitations.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep gratitude to my research supervisor Dr.Kaamran Raahemifar, for his patient guidance, enthusiastic encouragement and useful critiques of this research work. I would also like to thank Dr. Raahemifar, for his advice and assistance in keeping my progress on schedule throughout my academic years at Ryerson.

I would like to thank Mathi for his input and supportive criticism towards the thesis and continued support in all my endeavors.

Finally, I wish to thank my parents, Manonmany and Tharmalingam, and my wife Kasthuri for their continued love, support and encouragement in all my endeavors.

TABLE OF CONTENTS

Author's Declaration	ii
Abstract.....	iii
Acknowledgements	iv
Table of Contents	v
List of Figures	ix
List of Tables.....	xi
List of Algorithms.....	xii
Abbreviation	xiii
CHAPTER 1 Introduction	1
1.1 Motivation.....	4
1.2 Research objective	6
1.3 Thesis Outline	9
CHAPTER 2 A Survey: from transforms to dictionaries	12
2.1 Discrete Cosine Transform (DCT).....	13
2.1.1 2-D DCT Transform.....	14
2.2 Wavelet Transform based dictionaries	16
2.2.1 The Haar wavelets	17
2.2.2 Gabor wavelet.....	18
2.2.3 Ricker Wavelet	19

2.3	The Benefits of a Dictionary	19
2.4	Image-based adaptive dictionaries	20
2.5	Other Dictionary Learning Algorithms.....	22
2.6	Chapter summary.....	22
CHAPTER 3 Sparse constrained signal coding algorithms.....		23
3.1	Matching pursuit (MP)	24
3.2	The orthogonal matching pursuit	25
3.3	Other under-determined system solvers.....	26
3.4	Chapter Summary	26
CHAPTER 4 The hybrid dictionary with nonlinear atoms		28
4.1	The limitations of the DCT and wavelet based dictionaries	29
4.2	Addition of nonlinear atoms	35
4.2.1	The 1-D polynomial dictionary atom generator	36
4.2.2	The 2-D polynomial dictionary atom generator	38
4.2.3	The rational atoms.....	38
4.2.4	The root function atoms.....	39
4.2.5	The boxcar and the shifted square wave function.....	40
4.2.6	The generic exponential and logarithmic atoms	41
4.3	The hybrid dictionary	42
4.4	Evaluation of the hybrid dictionary.....	43
4.4.1	Sparse constrained reconstruction of simple function	44

4.4.2	Sparse constrained image reconstruction using small patches.....	48
4.4.3	Sparse constrained image reconstruction using large patches	53
4.5	Hybrid dictionary robustness test.....	58
4.5.1	The Primate image.....	58
4.5.2	The bell pepper image.....	61
4.6	Limitations of the Hybrid dictionary.....	66
4.7	Chapter Summary	71
CHAPTER 5 Optimized Predefined Dictionary.....		73
5.1	Parameter optimization using Particle Swarm Optimization	74
5.2	The parameter screening fitness function.....	76
5.3	Optimized predefined dictionary results.....	79
5.4	Chapter summary.....	84
CHAPTER 6 Optimized Hybrid Dictionary.....		86
6.1	Multi-dictionary OMP algorithm	86
6.2	Optimized hybrid dictionary algorithm.....	88
6.3	Performance gain of the optimized hybrid dictionary	92
6.4	Chapter Summary	96
CHAPTER 7 Compression using Time-shifted OMP algorithm		97
7.1	Time-shifted OMP signal coding algorithm.....	97
7.2	Evaluation of the Time-shifted OMP algorithm	99

7.3	Chapter Summary	102
CHAPTER 8 Conclusion and future work.....		104
8.1	Conclusion.....	104
8.2	Future work.....	109
Bibliography		111

LIST OF FIGURES

Figure 2-1 Sample 8x8 pixel patches using 1D DCT Atoms.....	15
Figure 2-2 Sample 8x8 patches using 2D DCT Atoms	16
Figure 4-1 Normalized Ricker function with $\alpha = \pi/4$	31
Figure 4-2 Normalized Gabor function with $\alpha = \pi$ and $\theta = 0$	31
Figure 4-3 Normalized Gabor function with varying α	32
Figure 4-4 The discrete polynomial test signal	34
Figure 4-5 PSNR of the reconstructed harmonic test signal	46
Figure 4-6 PSNR of the reconstructed exponential test signal.....	47
Figure 4-7 PSNR of the reconstructed polynomial test signal.....	48
Figure 4-8 PSNR of the reconstructed Lena image	50
Figure 4-9 Reconstructed Lena test image	52
Figure 4-10 PSNR of the reconstruct Man test image with 16x16 pixel patches.....	54
Figure 4-11 Original 1024x1024 Man test image.....	55
Figure 4-12 Reconstructed Man test image using 13-terms per patch with a DCT dictionary	56
Figure 4-13 Reconstructed man test image using 13- terms per patch with hybrid dictionary	57
Figure 4-14 PSNR of the reconstructed primate image	59
Figure 4-15 Reconstructed primate images using 5 terms per patch with the DCT and hybrid dictionaries	60
Figure 4-16 PSNR of the reconstructed bell pepper image	62
Figure 4-17 Original bell peppers test image	63

Figure 4-18 Reconstruction using 5 terms per patch with a DCT dictionary.....	64
Figure 4-19 Reconstruction using 5-terms per patch with a hybrid dictionary	65
Figure 4-20 5-term bell pepper image reconstruction magnified	66
Figure 4-21 Signal Coding time for the Lena test image	70
Figure 4-22 Time required to reconstruct the encoded lena image	71
Figure 5-1 Predefined dictionary optimization feedback system	78
Figure 5-2 PSO optimized DCT dictionary vs. DCT dictionary in image reconstruction	80
Figure 5-3 PSO optimized Polynomial dictionary tested on the Lena image	81
Figure 5-4 Optimized Gabor dictionaries	83
Figure 6-1 Lena image reconstruction using the optimized hybrid dictionary	94
Figure 6-2 Primate image reconstruction using the OHD	95
Figure 7-1 PSNR of reconstructed Lena image using Time-shifted OMP algorithm	101
Figure 7-2 Lena image code time using the Time-shifted OMP algorithm	102

LIST OF TABLES

Table 4-1 Mean square error of the reconstructed polynomial test signal.....	34
Table 4-2 MSE of reconstructed polynomial test signal with hybrid dictionary	36
Table 4-3 Dictionary Generating Parameters for 64 data points	43
Table 5-1 Dictionary and tunable parameters.....	77
Table 6-1 Dictionary and number of atoms used for test.....	89
Table 6-2 OHD dictionary and parameters	93

LIST OF ALGORITHMS

Algorithm 2-1 Dictionary learning algorithm	20
Algorithm 3-1 Matching Pursuit pseudo algorithm	24
Algorithm 3-2 The orthogonal matching pursuit algorithm	25
Algorithm 5-1 Particle Swarm Optimization Algorithm	74
Algorithm 6-1 Pseudo Algorithm for Multi-dictionary OMP	87
Algorithm 7-1 Ad-hoc pseudo algorithm for the Time-shifted OMP	98

ABBREVIATION

1D	One-dimensional
2D	Two-dimensional
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
JPEG	Joint Photographic Experts Group
K-SVD	K – Singular Value decomposition
MOD	Method of Optimal Directions
MP	Matching Pursuit
MPEG	Moving Picture Experts Group
MSE	Mean Square Error
OHD	Optimized Hybrid Dictionary
OMP	Orthogonal Matching Pursuit
PSNR	Peak Signal to Noise Ratio
PSO	Particle Swarm Optimization
RGB	Red-Green-Blue color scheme
TsOMP	Time-shifted Orthogonal Matching Pursuit

CHAPTER 1

INTRODUCTION

Sparse representation modeling is the ability to describe a given data set as a combination of a few building blocks. Sparse representation assumes that the data set is inherently sparse. Fortunately, sparse representation is achievable in majority of the data found in the scientific fields using the appropriate building blocks. The most obvious application of sparse representation modeling is in compression. To successfully compress a signal such as an image, audio or any other arbitrary data set, we need to define a set of rules to capture the large energy portion of the signal with only a few building blocks. The Discrete cosine transform (DCT) is a good building block for natural images because most of the information in natural images is captured with a few coefficients. The realization of this property in natural images has led to the success of lossy compression standards like JPEG. Even today, the most widely used compression techniques are based on DCT such as JPEG for still images, MP3 for audio and MPEG4 for video[1,2]. One of the reasons behind the success of the DCT based techniques is the availability of fast algorithms to encode the signal [3,4].

One can view the collection of these building blocks as a dictionary, where each building block is an atom in the dictionary. Many smooth continuous signals can be sparsely represented using discrete Fourier transform (DFT) analysis. The DFT dictionary can be created by selecting distinct frequency component of the Fourier transform as the atoms of the dictionary. Given this DFT dictionary a continuous signal can be sparsely represented but we are unable to sparsely represent an impulse

function using the DFT dictionary. Similarly, if the atoms of the dictionary were the time shifted impulse function; we can represent transient signals but sparse representation of smooth continuous functions would provide poor results. Intuitively, this indicates a single transform-based dictionary is not capable of representing all signals in a sparse manner, thus better reconstruction quality can be achieved using a hybrid dictionary. We can add all the transform based atoms to form a hybrid dictionary which may be able to sparsely represent a variety of signals but this will increase the dictionary size and effectively increase the storage requirements and signal coding time. Therefore, it is important to select the atoms of the hybrid dictionary in such a way that the given signal can be represented in a sparse manner with a reasonably sized dictionary.

Currently, most of the image processing applications still use transform-based processing but there are benefits to dictionary based modeling. The most significant benefit is the separation of the dictionary designing task from the signal coding task[5]. Since the ideological change that occurred in the last few decades that separated the two tasks, there has been an influx of scholarly research on the different types of dictionaries that can be used in image processing applications. The two prominent types of dictionaries in image processing are the image-based adaptive dictionary and the predefined dictionaries generated through mathematical models. The image-based adaptive dictionaries are generated through sparse coding algorithms that process training samples and create a structured dictionary that is capable of sparsely representing the given training set and similar data outside of the training set. Two such sparse coding algorithms in literature are the method of optimal directions (MOD), and k- singular value decompositions (K-SVD) [6]. Both the image-based

adaptive dictionaries and the predefined dictionaries have found success in many image processing applications including compression, image de-noising, pattern recognition, feature extractions, background subtraction, image reconstructions, and image restoration. However each dictionary makes a compromise between robustness, computational cost, implementation complexity, and signal representation quality. The choice of the dictionary depends on the application and its constraints. For example, in real time signal processing application such as video playback the computation cost and signal reconstruction time may supersede the improvements gained in signal reconstruction quality and compression. However, the compression ratio is more important than the signal coding time when transferring data over a network because the computational cost can be easily absorbed by the sending and receiving nodes. The reduced file size will increase the transmission capacity and reduce the transmission time which is especially useful on wireless networks with limited bandwidth.

The objective in image compression is to compress the image. The successful dictionary based compression coding algorithm will use the given dictionary, either a predefined dictionary or image-based adaptive dictionary, and find an optimal sparse solution to the signal representation problem shown in equation (1.1). The sparsity is measured using L_0 -norm, which counts the number of non-zero coefficients in the representation vector. However, the dictionary that provides the sparsest solution may not necessarily be the best dictionary for compression because the cost of storing the dictionary must also be included. Thus the predefined dictionary is preferred for compression applications because the predefined dictionary can be stored more

efficiently than an image-based dictionaries and it is likely to have a better compression ratio.

$$Y \approx DX \tag{1.1}$$

Feature extraction, classification and pattern recognition applications require a dictionary that will highlight relevant features of the signal and diminish insignificant features. In these applications the complexity of the dictionary generation is generally not a concern and the image-based adaptive dictionaries may be the best choice.

The success of the application depends on selecting the best suited dictionary for the application which may be an image-based adaptive dictionary generated through sparse coding methods or a predefined dictionary generated using mathematical functions.

1.1 Motivation

Signal processing applications such as compression, image de-noising, and pattern recognition are embracing algorithms and methods using dictionaries in hopes of improving the performance of the application. This has led to the growing interest into the types of dictionaries that can be used for each application. There are two categories of dictionaries, the predefined dictionary and the signal-based adaptive dictionary. Currently, existing predefined dictionaries are primarily based on transforms such as the DCT, Haar, Gabor, Ricker and other wavelet dictionaries. The signal-based adaptive dictionary resulting from sparse coding algorithms have shown improved signal reconstruction quality when compared to the DCT dictionary [6] in image de-noising application. However, the signal-based adaptive dictionaries are

neither robust nor simple to create when compared to the predefined dictionaries. Due to the time consuming learning process, the computational complexity involved in creating the dictionary and the added storage requirements involved with representing the entire dictionary; the signal-based adaptive dictionaries should be the last resort where a suitable predefined dictionary is not available to provide the expected results. The advantage of a predefined dictionary is that unlike the signal-based dictionary, the entire predefined dictionary does not have to accompany the representation vector to reconstruct the signal. Therefore, it is effective in compression and data transmission applications since the predefined dictionary can be recreated at the receiving side with minimal information. This has contributed to the success of the predefined DCT dictionary in image compression. There is still a high demand to improve the compression ratio and image reconstruction quality in high resolution images. This demand has increased interest in the signal processing community on the type of dictionaries and algorithms that can be used to improve the compression performance by:

1. Increasing the signal reconstruction quality.
2. Improving the compression ratio.
3. Reducing the signal coding time.

In this thesis, we will examine predefined dictionaries and signal coding algorithms that can be used for image compression. The main objective of the thesis is proposing alternatives to existing techniques to improve compression and signal reconstruction quality at the expense of signal coding time.

1.2 Research objective

Signal compression can be achieved by solving the sparse constrained signal representation problem which can be stated as an optimization problem. Given any signal Y , and a dictionary D , the objective is to find a representation vector X with the minimal number of non-zero coefficients that minimizes the signal representation error. A suitable minimization problem that imposes a sparse constraint is shown in equation (1.2) for an approximation of the signal. The sparse constraint is achieved by minimizing the L0-norm of the representation vector subject to the additional constraint of minimizing the signal representation error. The L0-norm counts the number of non-zero coefficients in the vector. The number of non-zero coefficients is also referred to as the number of terms in this thesis. The sparse constraint can be relaxed by using the L1 or L2 norm instead of the L0 norm, which has been used in Lasso signal coding algorithms but in this thesis we will maintain the strict L0 norm constraint used in the OMP signal coding algorithm.

$$\min_x \|x\|_0 \quad \text{such that} \quad \|Y - DX\|_2 < \varepsilon \quad (1.2)$$

Given the optimization problem the following four objectives arise:

1. Find a dictionary that allows the signal to be sparsely represented.
2. Increase the signal representation quality by minimizing the error in the reconstructed signal.
3. Minimize the computational cost involved in finding the required dictionary.

4. Minimize the computational cost of finding the signal representation vector for given the dictionary.

Image-based adaptive dictionaries using sparse coding algorithms will satisfy the first two objectives since the sparse representation constraint is inherently included in the dictionary learning process. The image-based adaptive dictionary is expected to reproduce the signals within the training set and other similar signals outside the training set with exceptional reconstruction quality which would satisfy the second objective [7]. However, the cost of producing the image-based adaptive dictionary using sparse coding is the major setback and it may not be worth the time and resources required if the desired signal representation quality can be achieved with a predefined dictionary.

Thus we have proposed alternatives to the existing predefined dictionary and image-based adaptive dictionaries to alleviate some of the limitations of the existing techniques. The main contribution of the thesis can be summarized as follows:

1. The existing predefined dictionary such as the DCT dictionary is inefficient at representing transient signals while the Haar dictionary is inefficient at representing smooth signals. We propose a hybrid dictionary composed of the DCT atoms and additional nonlinear function such as the polynomial, boxcar, exponentials, and rational functions. The hybrid dictionary with the additional nonlinear atoms is able to reconstruct not only images but also harmonic and non-harmonic signals efficiently. The hybrid dictionary makes a good compromise between the number of atoms in the dictionary and its

ability to reconstruct a variety of signals. It is also able to make a good compromise between its ability in representing smooth signals and transient signals. The drawbacks of the hybrid dictionary are the additional storage cost to describe the dictionary, the additional computation cost to generate the dictionary and code the signal.

2. The second contribution of the thesis is the optimized predefined dictionary. We present an evolutionary algorithm to optimize the dictionary generator to maximize the peak signal to noise ratio on the training set. The optimized DCT dictionary has improved signal reconstruction quality compared to the existing over complete DCT dictionary generated through fixed phase shifts. The optimized dictionary is intended as an alternative to image-based adaptive dictionaries. The benefit of the optimized dictionary is that the entire dictionary is not required to reconstruct the image; instead we only require a few additional parameters to describe the dictionary. The optimization process increases the cost of generating the dictionary and should be used in applications that emphasize the reconstruction quality over the computational cost. The optimized dictionary does not incur additional cost to code the signal.
3. The third contribution is the optimized hybrid dictionary generated using the optimized predefined dictionaries along with a dictionary filtering algorithm. Heavy computation cost is incurred to produce the optimized hybrid dictionary but the optimization process significantly reduces the size of the dictionary. The optimized hybrid dictionary

improves the signal coding time because the signal coding time is proportional to the number of atoms in the dictionary. This alternative is proposed for applications that emphasize signal reconstruction quality over computational cost.

4. The forth contribution is the time-shifted orthogonal matching pursuit (TsOMP) signal coding algorithm. The TsOMP is inspired by the phase shifting methods used in the DCT dictionaries. Initially, we generated the time-shifted polynomial functions to form a hybrid dictionary. However, this requires a time-shifted implementation for all the different types of atoms in the hybrid dictionary such as the rational and exponential function atoms. To ease implementation and allow for greater flexibility, we generated the time-shifted atoms in the signal coding algorithm instead of the dictionary generation algorithm. During our experiments, the TsOMP algorithm has shown improvements in both hybrid dictionary and a simple image-based dictionary.

1.3 Thesis Outline

The thesis can be split into 3 sections; the first part will include a literature review on existing techniques and algorithms that can be used to solve the sparse constrained image reconstruction problem. The second part containing the major contributions of the research work will include the algorithms and methods to produce the hybrid dictionary and evaluate its effectiveness. Finally the third section

will conclude the thesis by suggesting future extension of this work. The detailed chapter overview of the thesis is as follows.

The first section is composed of 2 chapters. Chapter 2 will include background information on the influential research works that have contributed to the evolution from transform based signal processing to dictionary based signal processing including a survey on commonly used dictionaries such as the DCT dictionary, the wavelet based dictionaries, and image-based adaptive dictionaries. Chapter 3 will continue the literature review with a brief discussion of two prominent signal coding algorithms, the matching pursuit algorithm (MP) and the orthogonal matching pursuit (OMP) algorithm.

The second section is composed of 4 chapters containing the major contribution of the thesis. Chapter 4 will provide an intuitive reasoning for adding the additional nonlinear atoms to the DCT dictionary to form a hybrid dictionary. Chapter 4 will also contain experiments to evaluate the effectiveness of the hybrid dictionary. Chapter 5 will introduce the particle swarm optimization technique that will be used to optimize the predefined dictionaries. Chapter 6 will propose the hybrid dictionary optimization algorithm that can be used to reduce the size of the dictionary and improve signal coding time. The time-shifted OMP algorithm is presented in chapter 7. The time-shifted OMP algorithm can be used in conjunction with either the predefined or signal-based dictionaries to improve signal representation quality.

We have suggested both dictionaries and algorithms that improve compression and image reconstruction quality but all of our algorithms incur additional cost. Our proposed technique can be used in applications like medical image compression where the main concern is the quality of the image reconstruction and not the computational

cost of the algorithms. We have shown through image-reconstruction experiments that the proposed techniques are superior to existing methods because of the improved image reconstruction quality verified by the peak signal to noise ratio and through visual inspection of the reconstructed image. However we require an in-depth cost analysis study to determine the limitations of the proposed algorithms.

Section 3 contains chapter 8 that will present a summary of the presented work and extensions of the thesis.

CHAPTER 2

A SURVEY: FROM TRANSFORMS TO DICTIONARIES

There are two types of dictionaries, a predefined dictionary and the signal-based adaptive dictionary.

1. The predefined dictionaries use mathematical models or signal specific properties to construct the atoms of the dictionary. The atoms of the dictionary are usually generated through frequency analysis, or wavelet based transforms that extract the harmonic nature of signal. The frequency analysis based dictionaries usually lack the ability to sparsely represent asymptotic and transient signals.
2. Signal-based adaptive dictionaries are generated using training samples from the signal in conjunction with a sparse coding algorithm. The dictionaries are capable of sparsely representing the training data and similar samples outside of the training set. Examples of training data for the learning process are small patches of an image or a collection of images.

Many of the image processing applications still rely on predefined dictionaries. However in the recent years; the adaptive dictionaries have been gaining popularity in scholarly research. The following sections will provide a brief overview of some of the

predefined dictionaries based on transform analysis starting with the discrete cosine transform.

2.1 Discrete Cosine Transform (DCT)

The DCT is used in a wide variety of science and engineering applications, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) to spectral methods for the numerical solution of partial differential equations. The DCT dictionary is ideal because it can be easily implemented by collecting the discrete cosine functions of varying frequency. There are 4 common variation of the DCT transform of which DCT-II is the most prominent introduced by N.Ahmed, T.Natarajan and K.R.Rao [8]. The DCT-II is shown in equation (2.1), where the signal y has a finite length m and the maximum number of non-zero coefficients is limited by N . The DCT-II transform can be organized in the form of a dictionary matrix by selecting the cosine function as the column vectors. The dictionary is a matrix of size $R^{m \times N}$ where m is the number of rows determined by the size of the signal to be represented and N dictates the number of atoms in the dictionary.

$$y_k = \sum_{n=0}^N a_n \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right) \quad k = 1, 2, \dots, m \quad (2.1)$$

$$D_{DCT(k,n)} = \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right) \quad (2.2)$$

The DCT can now be restated as a linear combination of the DCT dictionary without any loss of information. The coefficients of the DCT transform can be grouped together in a column vector. Thus equation (2.1) and equation (2.3) are

identical. An over complete dictionary requires additional constraints such as minimizing the number of non-zero coefficients in the representation vector to provide a unique solution. The benefit of an over complete dictionary is the increased flexibility and improvements in the reconstructed signal quality. There are two ways to generate an over complete DCT dictionary the first method is to take small frequency steps and the second method is to perform a phase shift. A simple implementation of the phase shifted DCT is shown in equation (2.4), which allows for the frequency step size and phase step size to be selected independently.

$$Y = [y_1 \quad y_2 \quad \dots] = D_{DCT} [a_1 \quad a_2 \quad \dots] = D_{DCT} A \quad (2.3)$$

$$D_{PI-DCT(k, nN+n_p)} = \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k + \frac{2\pi}{Np}n_p\right) \text{ where } np = 0, 1, N_p - 1 \quad (2.4)$$

2.1.1 2-D DCT Transform

One dimensional DCT dictionary is useful for signals such as audio but still images are 2 dimensional and may be handled better with a 2D DCT dictionary. Since most image processing application usually works on small square patches, the 2D DCT dictionary may perform better. The typical size of a patch is an 8x8 pixel block. Intuitively, the one dimensional function is unable to capture the correlation along the two axes of the image patch as well as a two dimensional function. The atoms of a 2D DCT dictionary is generated using eq. (2.5). The 2D DCT atom needs to be reshaped into a column vector, just like the image patches, to format the signal and dictionary in the desired form shown in equation (2.3).

$$D_{PI-DCT} = \cos\left(\frac{\pi}{N_1}\left(n_1 + \frac{1}{2}\right)k_1\right)\cos\left(\frac{\pi}{N_2}\left(n_2 + \frac{1}{2}\right)k_2\right) \quad (2.5)$$

The standard 1D DCT dictionary atoms with 64 elements reshaped into an 8x8 patch is shown in Figure 2-1 and Figure 2-2 shows the 2D DCT dictionary atoms as 8x8 patches. The patches generated using the 2D DCT atom have slower transition than the corresponding 1D DCT atoms and it captures the correlation between the neighboring pixels better. The slow transitioning behavior may help the 2D DCT dictionary in reconstructing smooth continuous patches but it may hinder its ability to represent sharp transitions around the boundaries of the patches. The harmonic behavior of the DCT atoms may also cause poor signal reconstruction quality around each patch boundaries. Akin to the 1D DCT dictionary one can produce an over complete 2D DCT dictionary by applying the same principles.

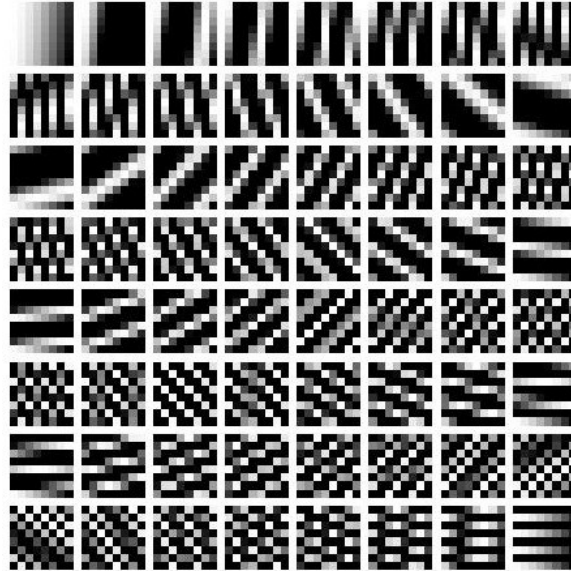


Figure 2-1 Sample 8x8 pixel patches using 1D DCT Atoms

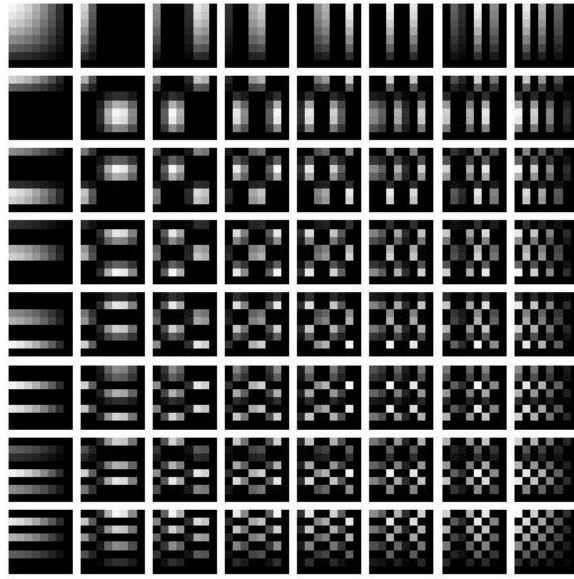


Figure 2-2 Sample 8x8 patches using 2D DCT Atoms

2.2 Wavelet Transform based dictionaries

Prior to 1930, signal processing applications primarily used Fourier analysis. However in 1909, Alfred Haar first introduced a sequence of rescaled discrete square wave functions now known as the Haar wavelet. The Haar sequence was later used in 1930 by Paul Levy, a physicist, to represent small complicated signals. Paul Levy's work showed that the Haar sequence is able to represent the Brownian motion more accurately than the Fourier series[5]. In 1984, the signal processing community had a breakthrough with the pioneering work done by Grossman and Morlet [9], in which they proposed to expand and create a family of functions by translation and dilation of a single function referred to as the mother wavelet. Additional work by Stephane Mallet, Yves Meyer and Ingrid Daubechies became the corner stone of wavelet

analysis. The basic theory of the wavelet analysis is to take an elementary wave function and create a family of functions by dilation and translation. The family of discrete wavelet functions is generated using eq. (2.6) for the selected elementary function ψ . The detailed discussion of the wavelet analysis, such as the prerequisites of the elementary function and discussion on the scaling functions is well beyond the scope of this thesis and the reader is directed to the literature [10-13]. However, choosing the right mother wavelet enables sparse representation of transient signals such as percussion sounds in audio and sharp transitions in images. Thus, this report will limit itself to some examples of wavelets used in image processing applications like the Haar, Gabor, and Ricker wavelets.

$$\psi_{a,b}(k) = \frac{1}{|a|^{m/2}} \psi\left(\frac{k\tau - b}{a}\right) \text{ where } \tau = \Delta t \quad (2.6)$$

2.2.1 The Haar wavelets

The Haar sequence was first introduced by Alfred Haar in 1909 and is the simplest of the wavelets. The elementary function is a square wave shown in equation (2.7). For practical implementation the general discrete Haar transform matrix i^{th} row is usually generated via equation (2.8) [14]. The Haar dictionary is produced by transposing the Haar transform matrix. There is limited literature on generating an over complete Haar dictionary, and the basic Haar dictionary may not be adequate for sparse representation of all signals. Thus, we had added sampled atoms using equation (2.7) to generate the over-complete Haar dictionary. The Haar wavelets have

shown promising results in image compression and image processing applications [15].

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & \text{else} \end{cases} \quad (2.7)$$

$$H_0 = \frac{1}{\sqrt{N}}$$

$$H_i(x) = \frac{1}{\sqrt{N}} \begin{cases} \sqrt[p]{2} & (q-1)/2^p \leq t < (q-0.5)/2^p \\ -\sqrt[p]{2} & (q-0.5)/2^p \leq t < (q+0.5)/2^p \\ 0 & \text{else} \end{cases} \quad (2.8)$$

For $x = (0, 1, 2, \dots, n-1)/N$, $p = \text{floor}(\log_2(k))$, $q = k \cdot 2^p + 1$

2.2.2 Gabor wavelet

The second wavelet of interest is the Gabor wavelet which has been successfully used in corner detection, pattern extraction, blob detection, facial recognition and various other image processing applications [16] but the ability of the Gabor wavelets in compression has not been fully explored. The simplified 1D Gabor wavelet is shown in eq. (2.9) where only the real part of the Gabor wavelet is used to simplify implementation. The DCT atoms can be easily produced using the Gabor wavelet by setting alpha to 0; however to verify the value of the Gabor atoms, the DCT atoms will not be included in the Gabor dictionary. The over complete

Gabor dictionary is generated by sampling the time-shifted Gabor functions at a specified sampling rate.

$$\psi(t) = \text{real}(e^{-a^2 t^2} e^{j2\pi ft}) = e^{-a^2 t^2} \cos(2\pi ft) \quad (2.9)$$

2.2.3 Ricker Wavelet

The Ricker wavelet uses the second derivative of the Gaussian function as the elementary function. The Ricker wavelet has had limited use in image processing applications, it has been used in blob detection and feature extraction. However, just like the Gabor wavelets the Ricker wavelets has not been used in compression. The mother wavelet function is shown in equation (2.10). The over-complete Ricker dictionary will be generated by sampling the time-shifted Ricker function at a sampling rate.

$$\psi(t) = \frac{1}{\sqrt{|a|}} (1 - t^2) e^{-at^2} \quad (2.10)$$

2.3 The Benefits of a Dictionary

The transition from transform based processing to dictionary based processing started in the 1990's. The work done by Mallet and Zhang in 1993 showed a novel scheme of sparse representation by using a subset of functions from a general over complete dictionary of functions [17]. The paper on Basis Pursuit by Chen, Donoho and Saunders proposed an algorithm that can code a signal using a given dictionary, effectively separating the task of designing the dictionary from the task of coding the

signal [18]. Since the two tasks were separated, many dictionaries and signal coding algorithms have been proposed. The initial dictionaries were inspired by the existing transforms like the DCT, and Haar transforms. Currently, one of the most popular research topics in image processing is the generation of image-based adaptive dictionaries. However, the predefined dictionaries are still widely used in many image processing applications and remain the benchmark for scholarly research due to their simplicity, efficiency and performance.

2.4 Image-based adaptive dictionaries

The pioneering work in the area of dictionary learning was provided by Olshausen and Field in 1996 [19], where the authors trained a dictionary, from a number of natural images, capable of sparsely representing small image patches from the training set. The advantage of image-based adaptive dictionary is its ability to provide a signal specific dictionary that can achieve excellent results. However, the dictionary learning process is a time consuming process and the new developments in the field are focusing on algorithms to produce a structured dictionary quicker. The dictionary learning process is an iterative process that can be decoupled into 2 parts. The first part of the dictionary learning process is the sparse coding of the signal using an initial dictionary and the second part is to update the dictionary using the sparse vector from the first step. The generic dictionary learning algorithm is shown in Algorithm 2-1.

Algorithm 2-1 Dictionary learning algorithm

1. Initialize a dictionary with N atoms
2. Use an algorithm to represent the signal over the given dictionary and

cost constraint.

3. Use an algorithm to update the dictionary using the signal representation vector from step 2.
4. If dictionary has not converged go to step 2.

The method of optimal directions (MOD) algorithm was introduced in 1999 by Engan et al. [20] and is among the earliest algorithms to solve the dictionary learning problem. The MOD algorithm solves the minimization problem in eq. (2.11) where x_i is the i^{th} column of X . The MOD algorithm specifies the dictionary update rule as the analytical solution to the quadratic problem given by $D = YX^+$ where X^+ is the Moore-Penrose pseudo-inverse of X . The MOD does not specify the signal coding algorithm. The MOD algorithm suffers from the high computational cost required for the matrix inversion used to update the dictionary. Since the introduction of the MOD algorithm there has been a variety of algorithms introduced to speed up the learning process. In 2005, M. Ahoran, M. Elad, and A. Bruckstein [21] presented the K-SVD algorithm which solves the same minimization problem as the MOD algorithm but reduces the computational cost by avoiding the matrix inversion by opting to update the dictionary one atom at a time. The K-SVD algorithm improves the speed of the learning process but it does not overcome all the drawback of the MOD algorithm. The high non-convexity nature of the dictionary learning problem can trap both the MOD, and K-SVD algorithms in local optimums and saddle points producing inefficient dictionaries. Also, the algorithms have been shown to be effective in small image patches but the effectiveness of adaptive dictionaries in large data sets have not been extensively studied yet.

$$\min_{D,X} \|Y - DX\|^2 \quad \text{subject to } \|x_i\|_0 \leq L \quad \forall i \quad (2.11)$$

2.5 Other Dictionary Learning Algorithms

There has been a growing interest in dictionary learning algorithms that can handle large data set. The online algorithm presented by J. Mairal, et al. in [22] used the L1-norm minimization for the sparse coding problem and it was capable of handling large data set. The report showed a dictionary that was learnt from a large 12Mp image for inpainting. Inpainting is the process of removing text from images.

The algorithm presented in [23] solves the dictionary learning problem using the linear gradient and adaptive gradient approach however the proposed algorithm suffers from some stability and convergence issues.

Currently existing learning algorithms take a long time to generate the dictionary and the resulting dictionaries are not robust which limits its ability in compression.

2.6 Chapter summary

Chapter 2 contained a literature review highlighting some of the influential work from the image processing community. It also described the transform based signal processing and the dictionary based signal processing techniques. The first section of the chapter was dedicated to the DCT, Haar, Gabor and Ricker transforms. The later sections of the chapter discussed the current research topics in the field of dictionary learning.

CHAPTER 3

SPARSE CONSTRAINED SIGNAL CODING ALGORITHMS

Dictionary based signal processing has decoupled the task of finding a dictionary from the signal coding task. Chapter 2 described some of the dictionaries that are currently being used in sparse image modeling. This chapter will focus on the sparse signal coding algorithms. Suppose a signal $y \in R^m$ has a sparse representation in the dictionary $D \in R^{m \times p}$, then it can be represented by $y \approx Dx$ where x has fewer than m nonzero coefficients. If the dictionary has more columns than rows ($p \gg m$), then the dictionary is over complete. An over complete dictionary contains additional atoms allowing for many solutions. The sparsity constraint reduces the number of possible solutions but it does not enforce a unique solution. The sparsity constrained signal representation problem is a combinatorial and NP-hard problem. In order to save time, we accept suboptimal solution and make a compromise between computational cost and solution quality. There is a high demand for fast, reliable, robust, and numerically efficient sparse signal coding algorithms. The two prominent approaches are the L1-norm minimization and the greedy algorithms based on L0-norm minimization. The L1-norm minimization based algorithms are generally slower and more robust compared to greedy algorithms.

3.1 Matching pursuit (MP)

In 1993 Mallat and Zhang proposed a greedy algorithm, known as the matching pursuit algorithm, capable of finding a representation vector for signals from a large over complete wavelet dictionary [17]. Following that paper in 1995, Mallat and Bergeaud successfully showed the matching pursuit algorithm was able to sparsely represent an image using a given dictionary [24]. The matching pursuit pseudo algorithm to solve the problem of finding L non-zero coefficients to represent the signal y , given a dictionary D with p atoms is shown in Algorithm 3-1.

Algorithm 3-1 Matching Pursuit pseudo algorithm

1. Initialization vector : $x[1:p] = 0$
2. Initialize residual error, $r = y$
3. While $\|x\|_0 < L$
 - a. $i = \max_i(\text{abs}(d_i^T r))$
 - b. $x[i] = x[i] + d_i^T r$
 - c. $r = r - (d_i^T r)d_i$
4. End while

The MP algorithm is a bottom up search that finds locally best solution at each stage for the minimization problem $\min_x \|y - Dx\|_2^2$ s.t. $\|x\|_0 < L$; however it does not guarantee an optimal solution. The algorithm may get trapped in a local optimal solutions and saddle points especially if the dictionary is an over complete non-orthogonal dictionary. The additional atoms in the dictionary increases the chances of

the algorithm getting trapped in local optimal solutions but the matching pursuit algorithm has a good compromise between signal coding time and solution quality.

3.2 The orthogonal matching pursuit

The orthogonal matching pursuit algorithm was presented by Krishnaprasad, Rezaifar and Pati in 1993 inspired by the matching pursuit algorithm [25]. The main contributions of the OMP algorithm are:

1. The residual and the coefficients are recalculated prior to the next iteration.
2. Each atom can only be selected once.

The OMP algorithm is an improvement to the MP algorithm since it runs faster, and consistently provides better results. The OMP pseudo algorithm is shown in Algorithm 3-2.

Algorithm 3-2 The orthogonal matching pursuit algorithm

1. Initialize vector : $x[1:p] = 0$
2. Initialize residual error : $r = y$
3. Initialize selected set $S = \{\text{Null set}\}$
4. Initialize unused set $S_{un} = \{i\}$ for all $i=1:p$
5. While $\|x\|_0 < L$
 - a. *find* i s.t $\min_i \|ad_i - r\|_2^2$ for all $i \in S_{un}$
 - b. Add i to the selected set: $S = S \cup \{i\}$

- c. Remove selected atoms from available set: $\mathcal{S}_{un} = \mathcal{S}^C$
 - d. Find best fit x from selected set: $x[i] = \min(y - D_{(S)}x)$
 - e. Find remaining error : $r = y - D_{(S)}x$
6. End while

3.3 Other under-determined system solvers

Other notable variations of the matching pursuit algorithm in the recent years are the stage-wise orthogonal matching pursuit (StOMP) [26] and regularized orthogonal matching pursuit (ROMP). The greedy pursuit algorithms build up the signal approximation one step at a time, selecting the locally optimal solution on each step. The matching pursuit and its variation is not the only type of algorithms that can be used to solve the signal coding problem. There are many algorithms in literature that use the L1-norm minimization such as the FOCUSS algorithm, to solve the sparse representation problem.

3.4 Chapter Summary

Chapter 3 introduced the MP and OMP signal coding algorithms that can be used to code a signal using a dictionary. The MP algorithm is among the first algorithms to offer a solution to the sparse coding problem. Both MP and OMP is a bottom up search algorithm that can offer an optimal solution provided the dictionary is orthogonal. The OMP algorithm has a well-balanced compromise between signal coding quality and execution time when dealing with large over complete dictionaries.

Since the introduction of the MP algorithm in 1993, many algorithms have been proposed to solve the sparsity constrained signal representation problem; however the OMP algorithm is still widely used in literature.

CHAPTER 4

THE HYBRID DICTIONARY WITH NONLINEAR ATOMS

Chapters 2 and 3 highlighted some key contribution to the signal processing community, from the early innovation in transform based signal processing to the current techniques based on dictionaries. The dictionary based signal processing is still in its infancy, and many scholars are still experimenting with different types of dictionaries. There have been two competing ideas for the dictionary generation: the adaptive dictionary and the predefined dictionary. One of the growing interests is to generate a hybrid dictionary that combines the benefits of both the predefined dictionaries and adaptive dictionaries. The signal-based adaptive dictionaries have outperformed predefined dictionaries when tested on the training samples but it has not been extensively studied in large problems. The undesirable traits of adaptive dictionaries are:

1. The high computational cost incurred in producing the dictionary.
2. The dictionary may not represent signals outside the training set with the same reconstruction quality. Existing adaptive dictionaries have not been extensively studied, thus we cannot make a claim on how robust they are.
3. The dictionaries are not flexible because they cannot easily expand to accommodate varying problem size. For example a dictionary that is

trained to produce 8x8 pixel patches cannot be used to represent 16x16 patches or 4x4 pixel patches without retraining the dictionary.

4. In order to recreate the signal the entire dictionary must accompany the sparse vectors. This increases the storage requirements and reduces the compression efficiency.

The benefits of a predefined dictionary are:

1. Simple mathematical functions to generate the dictionary, which can be encoded as an algorithm allowing the encoded signal to be reconstructed with only a few additional parameters to identify the type of dictionary and the signal representation vector.
2. Predefined dictionaries such as the DCT and Haar are robust.
3. Since the dictionaries are mathematical functions, they can be easily modified to accommodate varying problem size.

We propose a set of rules to generate a predefined hybrid dictionary that improves the signal representation quality when compared to a DCT and wavelet based dictionaries.

4.1 The limitations of the DCT and wavelet based dictionaries

The DCT dictionary is apt at sparse representation of harmonic signals. The Haar and the Gabor dictionaries are well suited for representing transient signals. The DCT and Haar functions are easily visualized thus the Ricker and Gabor function are shown in Figure 4-1 and Figure 4-2 respectfully. All of the dictionaries are composed of functions that have adjustable parameters and depending on the selected parameters the resulting dictionary atoms will change. This can be seen in Figure 4-3

showing the Gabor atoms generated by varying α without changing the frequency or region of interest. The performance of the dictionaries can be improved with the careful selection of these parameters. The Gabor function is unique because:

1. It is able to create a family of harmonic functions by setting α to 0 with a non-zero θ . This is the DCT dictionary atoms.
2. It can create a family of exponentials by setting θ to zero and setting a nonzero α .
3. It can create a growing harmonic or shrinking harmonic signal by selecting the appropriate region of interest, α and θ parameters.

One of the difficulties in using the Gabor dictionary is selecting the optimal parameters to generate the Gabor atoms. Since, the harmonic functions are already included with the DCT dictionary the Gabor dictionary will not include the DCT atoms. The Gabor and Ricker dictionaries are generated by selecting a range of values because well-established parameter set is not available for compression application in literature for these two transforms.

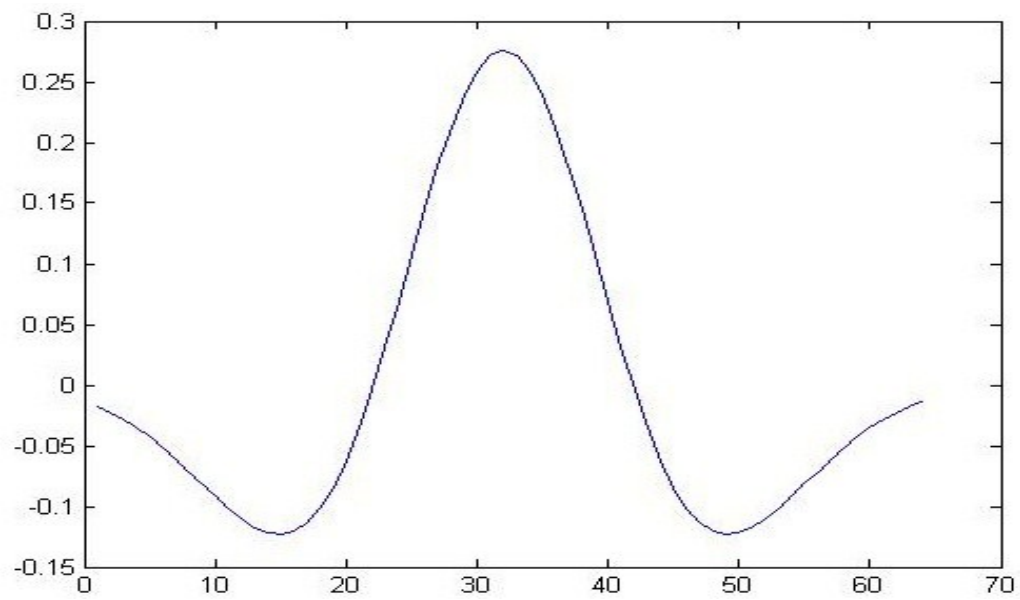


Figure 4-1 Normalized Ricker function with $\alpha = \pi/4$

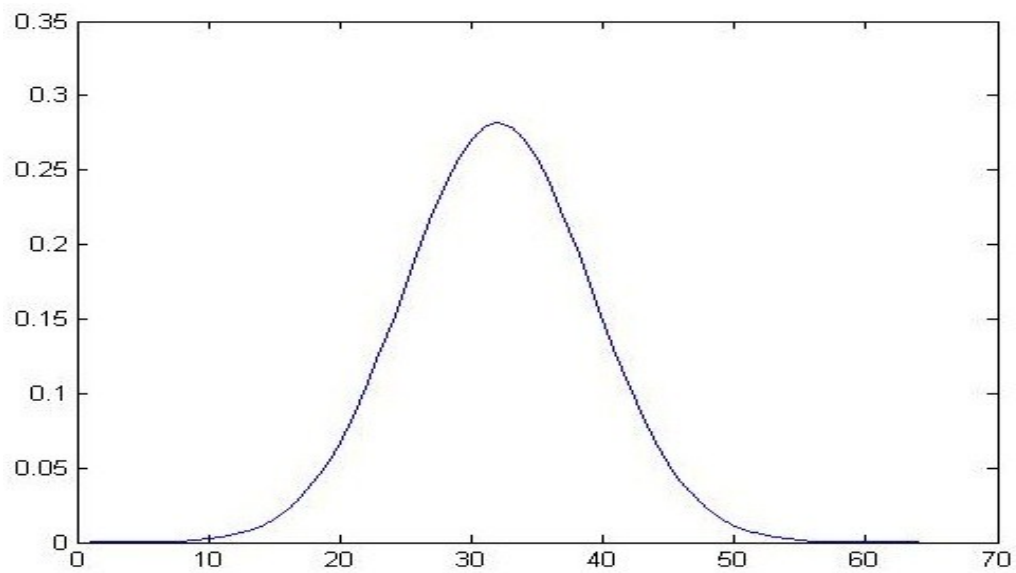


Figure 4-2 Normalized Gabor function with $\alpha = \pi$ and $\theta = 0$

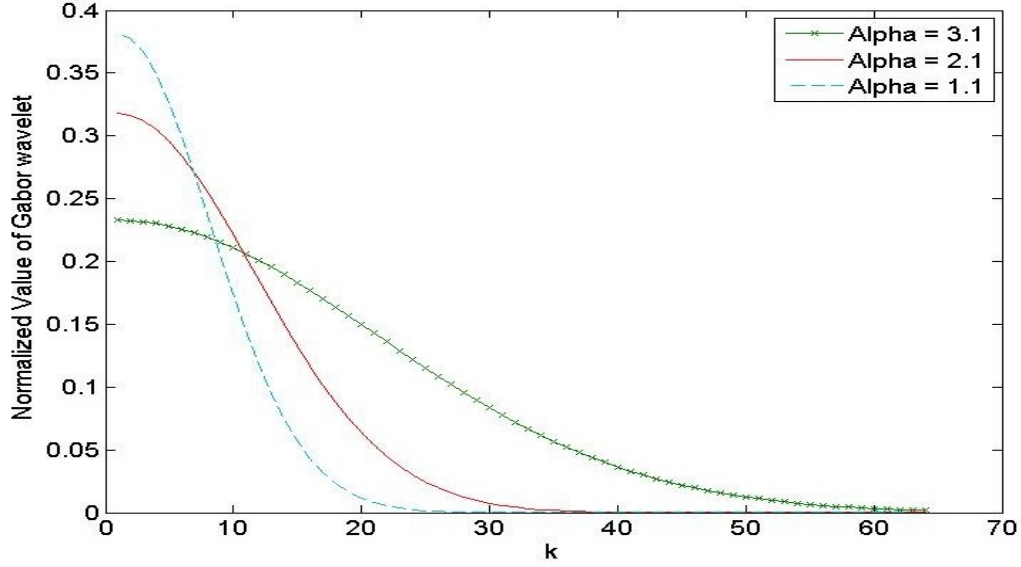


Figure 4-3 Normalized Gabor function with varying alpha

To illustrate the limitation of the existing dictionaries, consider reconstructing a non-harmonic signal such as a polynomial, exponential or logarithmic function using only harmonic functions such as the DCT atoms. Intuitively, the harmonic atoms will not be efficient. We can verify this limitation through an example. Let us consider the sparse representation of the discrete test signal

$$y[k] = 4 \left(-5 + \frac{10k}{64} \right)^3 + 5 \cos \left(\frac{50k}{64} \right) \text{ for the first 64 elements using only 4 non-zero}$$

coefficients in the representation vector. The discrete test signal is shown in Figure 4-4. The performance of the dictionaries is measured using the mean square error (MSE) of the reconstructed signal which is also shown in Table 4-1. The mean square error is calculated using equation (4.1). The MSE is a good indicator of the dictionary performance because it calculates the average error per data point. The Ricker

dictionary had a MSE of 325.04 while the DCT dictionary had an MSE of 10640, which is more than 3273%. The DCT dictionary performed poorly because it was unable to represent the quick transient behavior at the signal boundaries. The DCT dictionary is superior to the Haar dictionary at representing smooth harmonic signals but it is inefficient at representing transient signals. The Gabor and Ricker dictionaries lack a well-defined parameter set to create a structured dictionary for compression application. Both the Ricker and Gabor dictionary were generated using a wide range of parameters increasing the overall size of both dictionaries. The Gabor and Ricker dictionaries have more than 500 atoms while the DCT dictionary has 129 atoms making the Gabor and Ricker dictionaries approximately 4 times larger than the DCT dictionary. The increased dictionary size will increase computational cost effectively increasing the time required to generate the dictionary and code the signal. The additional computation cost is insignificant on a mid-range desktop computer, but it may become significant if implemented on processors with limited processing power such as Asics or microcontrollers.

$$MSE = \frac{1}{N} \sum_{i=0}^{N-1} (Y(i) - Y_{recon}(i))^2 \quad (4.1)$$

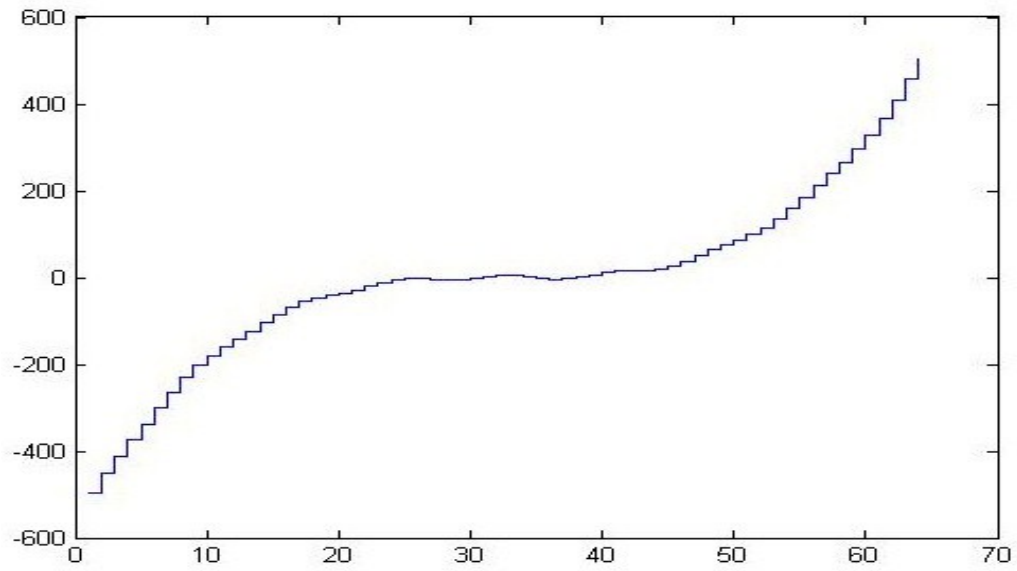


Figure 4-4 The discrete polynomial test signal

Table 4-1 Mean square error of the reconstructed polynomial test signal

Dictionary Type	Number of Atoms	MSE
Dct	129	10640
Gabor	513	5731.5
Haar	128	430.09
Ricker	557	325.04
Combined	1263	325.04

4.2 Addition of nonlinear atoms

The same test signal $y[k] = 4\left(-5 + \frac{10k}{64}\right)^3 + 5\cos\left(\frac{50k}{64}\right)$ can be sparsely represented by including a polynomial dictionary. The simplest polynomial atoms are in the form of t^e . The polynomial atoms can easily grow out of bounds for large values of t and this problem is alleviated by limiting the values of t between a reasonable range such as -2 and 2. The sparse representation test is repeated again but with the addition of the polynomial atoms to the existing dictionaries. Each dictionary has 10 new polynomial atoms generated using $t, t^2, t^3, \dots, t^{10}$, for equally spaced t between -1 and 1 to create 64 points. The test are shown in Table 4-2 highlighting the improvements in signal reconstruction quality especially for the DCT-polynomial hybrid dictionary which can reconstruct the signal with an MSE of 3.53. The DCT dictionary without the polynomial atoms had an MSE of 10640. The polynomial function is versatile because low order polynomials are able to describe many non-harmonic smooth signals, and the higher order polynomials can handle transient signals. The inclusion of the polynomial atoms to the DCT dictionary improves the DCT-polynomial hybrid dictionary's ability to handle smooth non-harmonic signals. The following sections will formulate an algorithm to generate a structured polynomial dictionary.

Table 4-2 MSE of reconstructed polynomial test signal with hybrid dictionary

Dictionary Type	Number of Atoms	MSE
DCT-Polynomial Hybrid	139	3.53
Gabor-Polynomial Hybrid	523	12.1
Haar-Polynomial Hybrid	139	11.26
Ricker-Polynomial Hybrid	567	12.16
Polynomial alone	10	11.70

4.2.1 The 1-D polynomial dictionary atom generator

The basic polynomial dictionary is composed of polynomial atoms without any translation or dilation, generated by the function t^c , where we are able to limit the maximum polynomial order and region of interest depending on the data set. This ensures that any polynomial function with a lower order can be represented using the basic polynomial dictionary. Once the basic polynomial functions are included, we propose to add a few dilated and translated polynomials generated using the generic polynomial generator shown in equation (4.2). The flexibility of the generic polynomial function generator adds additional storage and computation cost because to describe the polynomial dictionary, we now require each coefficient that was used to generate the polynomial dictionary. For example if there are 100 quadratic polynomials with 2 coefficients each in the dictionary, then we require a minimum of 200 coefficients to describe the dictionary. This inefficiency affects its ability in compression.

$$P_c(t) = t^c + \sum_{i=0}^{c-1} a_i t^i \quad (4.2)$$

We propose using a list of coefficients with a circular shifting algorithm to generate a family of polynomial atoms. The shifting algorithm will increase the cost of generating the dictionary but it allows for the inclusion of vital functions without the need of describing each polynomial in the dictionary. The storage requirements could be further reduced, if we generated the list of coefficients using an incremental step size scheme such as $a_i = a_0 + \Delta$. By using the fixed incremental step size scheme to generate the list of coefficients in conjunction with a shifting algorithm to generate the polynomial function; we can describe the entire polynomial dictionary with the following information:

1. The number of data points in the atom. This is required for any dictionary, thus it is not an additional requirement if more than one dictionary is added.
2. The starting coefficient value a_0 .
3. The coefficient step size Δ .
4. The coefficient final value.
5. The starting region of interest value t_0 .
6. The end region of interest value.
7. The maximum order of the polynomials to be generated.

The algorithm should create polynomials of the maximum order by selecting the coefficients from the list in sequential order. This will create a structured dictionary that can be used in sparse representation model of many smooth signals.

The polynomial atom generator is flexible allowing for vital functions to be included while minimizing the storage requirements to describe the dictionary.

4.2.2 The 2-D polynomial dictionary atom generator

The one dimensional polynomial function generator can be easily modified to generate 2D polynomial atoms which may be more efficient at representing images. Similar to the 1D polynomial, the basic 2D polynomial function generator will be in the form of $x^c y^d$ and to ease implementations x and y will have the same region of interest. Once the basic 2D polynomial functions are generated, more complex polynomials can be added using eq.(4.3). The 2D polynomials will be generated using a coefficient selector algorithm which would select the polynomial coefficients from a list of numbers. If the hybrid dictionary already contains 1D polynomials than the 2D polynomials can use the same set of parameters which would reduce storage requirements. If computation cost is a concern then either the 1D polynomial or 2D polynomial dictionary should be selected depending on the cost of the dictionary.

$$P(x, y) = \left(\sum_{i=0}^{\text{floor}(c/2)} a_i x^i \right) \left(\sum_{j=\text{floor}(c/2)}^c a_j y^{(j-\text{floor}(c/2))} \right) \quad (4.3)$$

4.2.3 The rational atoms

The inclusion of rational functions will aid the polynomial dictionary in sparsely representing asymptotic signals. All polynomials are rational functions but it is intentionally distinguished here because this separation allows for the rational dictionary to be defined and included without affecting the polynomial dictionary. If a

polynomial dictionary is included in the hybrid dictionary, the easiest rational functions is generated with the polynomial atoms using equation (4.4). Furthermore, the rational function in the generic form shown in eq. (4.5) can be created using the same strategies used to create the 2D polynomial atoms. The coefficients for the two polynomials are selected by the coefficient selector algorithm. The rational function has asymptotic behavior which is helpful in representing sharp transitions but it must be handled with care to avoid invalid numbers and infinite value. We resolve this issue by limiting the asymptotic points to a predefined maximum and minimum value making the function continuous.

$$R(t) = \frac{1}{t^c + \sum_{i=0}^{c-1} a_i t^i} \quad (4.4)$$

$$R(t) = \frac{t^c + \sum_{i=0}^{c-1} a_i t^i}{t^d + \sum_{i=0}^{d-1} a_i t^i} \quad (4.5)$$

4.2.4 The root function atoms

The hybrid dictionary composed of rational and polynomial functions will be able to handle a variety of signals with the inclusion of the square root and cubic root functions. The square root dictionary generator is shown in equation (4.6) where the absolute value of the polynomial or rational function is used to avoid handling complex numbers. The root function generates a slow moving signal that can help

represent smooth gradual signals. The cubic root function generator is shown in equation (4.7) and it does not need to take the absolute value of the rational function because we restricted all rational functions to be real valued.

$$S_2(t) = \sqrt{t^\ell + \sum_{i=0}^{\ell-1} a_i t^i} \quad (4.6)$$

$$S_3(t) = \sqrt[3]{t^\ell + \sum_{i=0}^{\ell-1} a_i t^i} \quad (4.7)$$

4.2.5 The boxcar and the shifted square wave function

The boxcar function is an extension of the delta function that can be used in representing transitions and small complicated signals where the rational functions are not applicable. Unlike the delta function which is nonzero at a single point, the boxcar function is non-zero within a region which enables it to efficiently represent transitions within a small region. There is a minor difference between the Haar wavelet and the step function. The step function shown in eq (4.8) can be generated by translation and dilation of the Haar wavelet in conjunction with the correct sampling rate but the generally used Haar wavelet matrix does not include all of these functions. Also separating the boxcar function from the Haar wavelet simplifies the implementation of the dictionaries. The basic boxcar dictionary of a specified width can be generated by performing N-circular shifts on the elementary boxcar function to produce the remaining atoms of the dictionary. An over complete Boxcar dictionary can be generated by combining multiple boxcar dictionaries with distinct widths together which is expected to improve its ability in representing transient

signals. However the boxcar dictionary will perform poorly when dealing with smooth functions and thus the boxcar dictionary is always used together with either the polynomial or DCT dictionary and it is not intended to be used alone.

The moving square wave dictionary is a dictionary with shifted square waves. The dictionary is created by performing N-circular shifts on the elementary square wave of fixed width. The moving square wave dictionary is generated just like the boxcar dictionary and will contain some atoms from the Haar dictionary. The inclusion of the moving square wave dictionary allows for specific Haar wavelet atoms to be included without including the entire Haar dictionary. Occasionally, both the boxcar and moving square wave dictionaries will be used together. The moving square wave dictionary is used to aid the polynomial and DCT dictionaries and it is not intended to be a standalone dictionary.

$$\psi(t) = \begin{cases} 1 & 0 < t < 1 \\ 0 & \text{else} \end{cases} \quad (4.8)$$

4.2.6 The generic exponential and logarithmic atoms

The generic exponential elements are created by taking the exponential value of any given function. Inspired by the Gabor wavelets that have been applied in image processing, we decided to produce a dictionary composed of generic exponential functions. The generic exponential elements will be generated using equation (4.9) for the polynomial dictionary atoms. Akin to the exponential dictionary atoms, the generic logarithmic dictionary atoms are generated using equation (4.10). The inclusion of both the exponential and logarithmic dictionary atoms may aid in

representing some complex signals. However adding the polynomial, rational, exponential and logarithmic functions to the hybrid dictionary will increase the size of the dictionary and increase the signal coding time. Therefore, the value of the exponential and logarithmic dictionary should be evaluated prior to inclusion.

$$\psi(t) = e^{f(t)} \quad (4.9)$$

$$\psi(t) = \log(f(t)) \quad (4.10)$$

4.3 The hybrid dictionary

The important contribution of our work is the hybrid dictionary for sparse constrained image reconstruction. The hybrid dictionary is created by including additional nonlinear functions such as the DCT, polynomial, rational, logarithmic, boxcar, exponential and the root functions. The inclusion of the additional functions increase the computational cost compared to a DCT dictionary but it is less expensive than the image-based adaptive dictionaries. The hybrid dictionary can select to include the proposed functions by providing the parameters to generate the atoms. To improve signal coding time, duplicate atoms are removed from the hybrid dictionary. The proposed hybrid dictionary can be recreated using a few additional parameters making it more efficient than the signal-based adaptive dictionary. The hybrid dictionary outperformed the predefined DCT and Haar dictionaries during our testing by providing better signal representation while using fewer non-zero coefficients in the representation vector.

4.4 Evaluation of the hybrid dictionary

This section will compare the hybrid dictionary against existing predefined dictionaries. Both harmonic and non-harmonic signals will be used to evaluate the dictionaries. The dictionaries will be simulated using the Matlab on a quad core 2.85 GHz computer with 16 GB of Ram operating at 1066 MHz. All algorithms are sequential and did not fully harness the parallel computing power that was available. The OMP signal coding algorithm was used to evaluate the dictionaries because it has a well-balanced compromise between execution time and solution quality when dealing with large dictionaries. The over complete dictionaries used to reconstruct the signals are generated using the parameters defined in Table 4-3.

Table 4-3 Dictionary Generating Parameters for 64 data points

Dictionary	Parameter	Number of Atoms
DCT	$N_p = 2$	129
Haar	N/A	64
Gabor	$f = \left\{ 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi \right\}$ $a = \{0.1, 0.6, 1.1, \dots, 4\}$	1281
Ricker	$a = \{-2, -1.5, -1, \dots, 2\}$	577
1D Polynomial	$a = \{-1, -0.5, \dots, 1\}$ Order = 2	204

Boxcar	$width = \{2, 3\}$	128
Hybrid	DCT : $N_p = 1$ Polynomial, boxcar, rational	600

The dictionaries will be evaluated using the peak signal to noise ratio (PSNR) of the reconstructed signal calculated using equation (4.11). The PSNR is the standard benchmark ratio used to measure the quality of the reconstructed signal in lossy compression methods. The original data to be compressed is the signal, and the noise is the error introduced by the sparse coding process. The PSNR is a good indicator of the human perception of the reconstructed signal. In general, a higher PSNR will indicate that the reconstructed signal has fewer errors but in some cases it may not lead to the best visual image. This can happen if the large error portions occur in patches that are not of interest. However the PSNR remains a good evaluation tool to compare the performance of the dictionaries since all dictionaries are reconstructing the same contents. The second measurement tool is the number of non-zero coefficients, referred to as the number of terms, used to represent the signal. The compression ratio is inversely proportional to the number of terms in the representation vector.

$$PSNR = 20\log_{10}(255) - 10\log_{10}(MSE) \quad (4.11)$$

4.4.1 Sparse constrained reconstruction of simple function

The three functions shown below are selected to evaluate the dictionaries; the first function is a harmonic function, while the last two are non-harmonic function.

1. $y_1(t) = \cos(9t)$
2. $y_2 = te^t$
3. $y_3 = t^2 \cos(5t) \cos(9t)$

All the test signals are confined between -5 and 5 and are sampled at equal spacing to create 64 data points. Figure 4-5 shows the PSNR of the first test signal a smooth harmonic function. As expected the DCT and the hybrid dictionary that contains some DCT atoms perform well. The first test signal reconstructed with 15 or more non-zero coefficients using the hybrid dictionary has a PSNR that is approximately 20 dB better than the DCT dictionary. The Haar, Ricker and Gabor dictionaries require 45 non-zero terms to reconstruct the test signal with a PSNR of 65 dB but the DCT and hybrid dictionary can reconstruct the test signal with equal quality using only 3 non-zero terms. When the peak signal to noise ratio is over 50db, the error is not readily distinguished by the naked eye. Therefore, the DCT dictionary outperforms the hybrid dictionary on this test signal because the DCT dictionary is smaller. However the limitation of the DCT dictionary is evident in Figure 4-6 showing the reconstruction of the second test signal, a non-harmonic smooth signal. The hybrid dictionary is able to achieve a PSNR of 60 dB using 20 non-zero terms and a maximum of 120 dB using 43 terms but the DCT dictionary is unable to keep up as it starts to plateau around 60 dB. The Ricker and Gabor dictionaries improve rapidly and perform better than the other dictionaries as expected because the test signal is highly correlated with the dictionary. The third test signal is a polynomial function and as expected the hybrid dictionary obtains the best PSNR as seen in Figure 4-7.

In conclusion, the DCT dictionary is capable of sparse representation of harmonic signals but is inferior compared to both Gabor, and Ricker dictionaries when representing exponentials. The Haar dictionary has difficulties representing smooth continuous functions. However the hybrid dictionary composed of the boxcar, DCT, and polynomial functions is able to sparsely represent both transient and smooth functions. These three test signals alone do not validate the hybrid dictionary, but it shows that the hybrid dictionary has a good composition of transient and smooth signals.

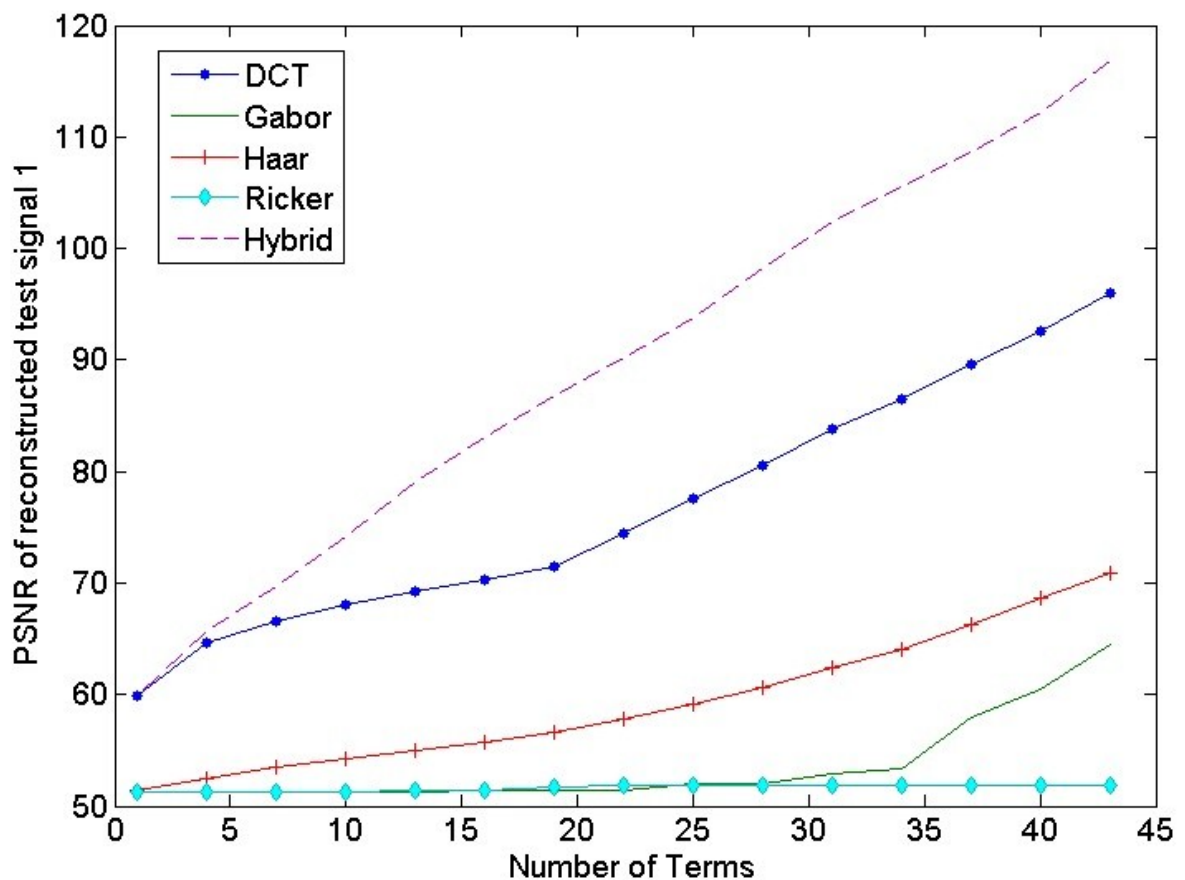


Figure 4-5 PSNR of the reconstructed harmonic test signal

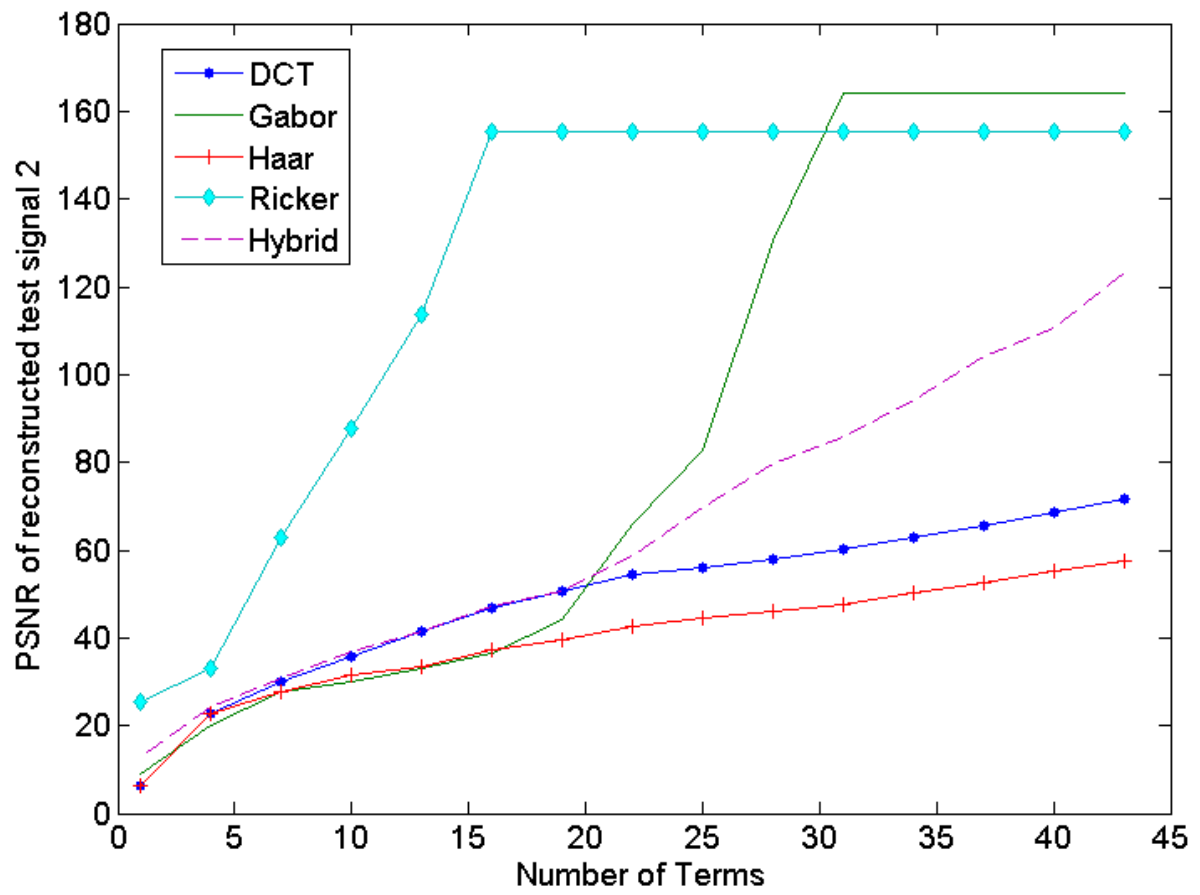


Figure 4-6 PSNR of the reconstructed exponential test signal

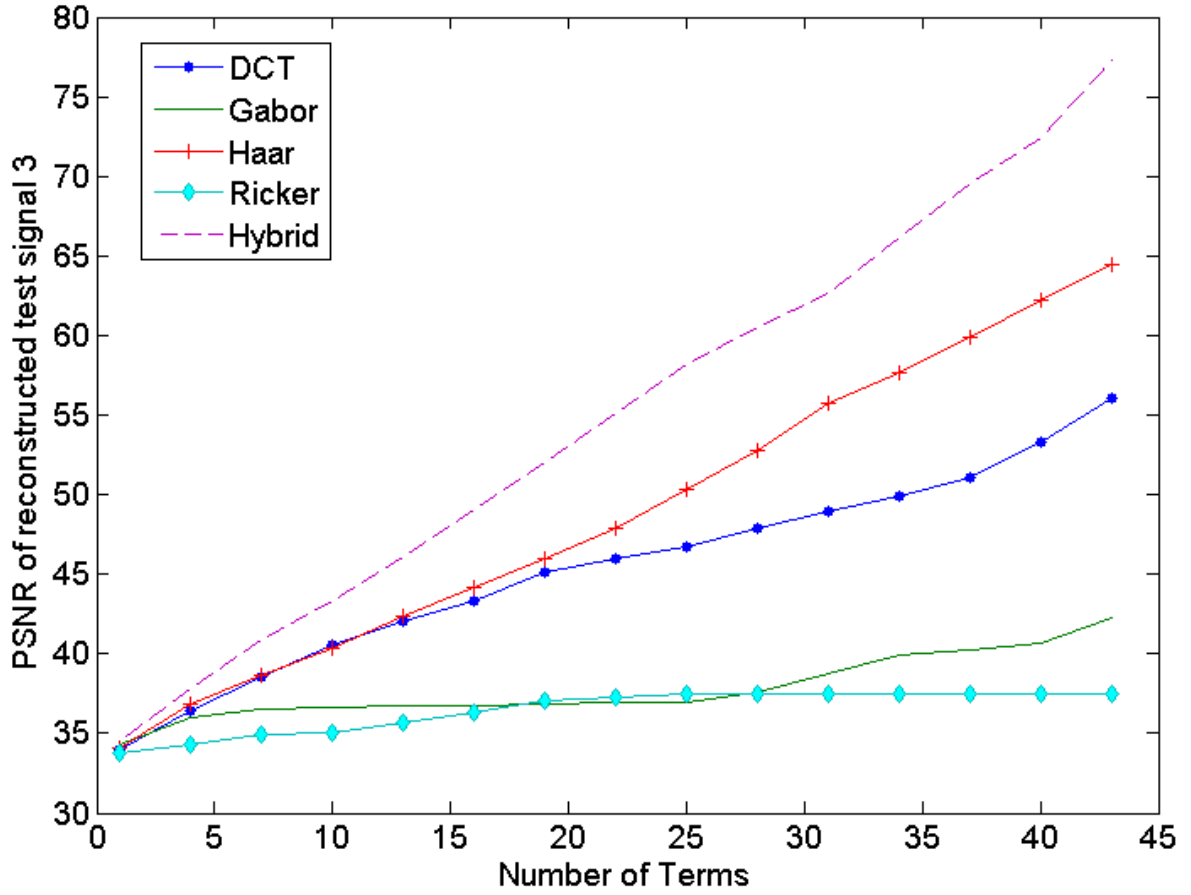


Figure 4-7 PSNR of the reconstructed polynomial test signal

4.4.2 Sparse constrained image reconstruction using small patches

Inspired by the success of the hybrid dictionary on the tests with simple discrete signals, the hybrid dictionary was tested on images. The first test image is the Lena 512x512 pixel image. The test image is broken into smaller 8x8 pixel patches and each patch will contain 64 integers to represent the 8-bit grey tone pixel data. To use the OMP algorithm, each test patch matrix is reshaped into a column vector and

sequentially aligned to form the signal matrix. The sparse representation problem of the image is now in the form $Y = Dx$, and the OMP signal coding algorithm can be applied. The image can be reconstructed by reshaping the column vectors into image patches and then aligning the patches sequentially.

The peak signal to noise ratio for the reconstructed Lena image is shown in Figure 4-8. The DCT and the hybrid dictionary obtain the best image reconstruction when using fewer than 15 non-zero coefficient per patch. The Haar dictionary performs better than the DCT dictionary when using more than 15 terms per patch by achieving a PSNR difference of at least 1 dB and a maximum of 7 dB when reconstructing the image using 43 terms. The Ricker and Gabor dictionaries plateau early and the reconstructed image quality does not improve as the number of non-zero coefficients in the representation vector is increased. The Haar, DCT and Hybrid dictionaries provide better PSNR as the number of non-zero terms used in the reconstruction vector is increased. The PSNR of the reconstructed Lena image using the hybrid dictionary is 2dB better than the image reconstructed using the DCT dictionary with 4 non-zero coefficients per patch and the PSNR difference between the two dictionaries is 18 dB when using 43 non-zero coefficients per patch. A PSNR of 35-50 dB is considered good quality image reconstruction. The hybrid dictionary obtains a PSNR of 35 dB using only 10 non-zero terms per patch but the DCT dictionary requires approximately 22 non-zero terms per patch. The DCT dictionary requires twice as many coefficients to achieve the same image reconstruction quality.

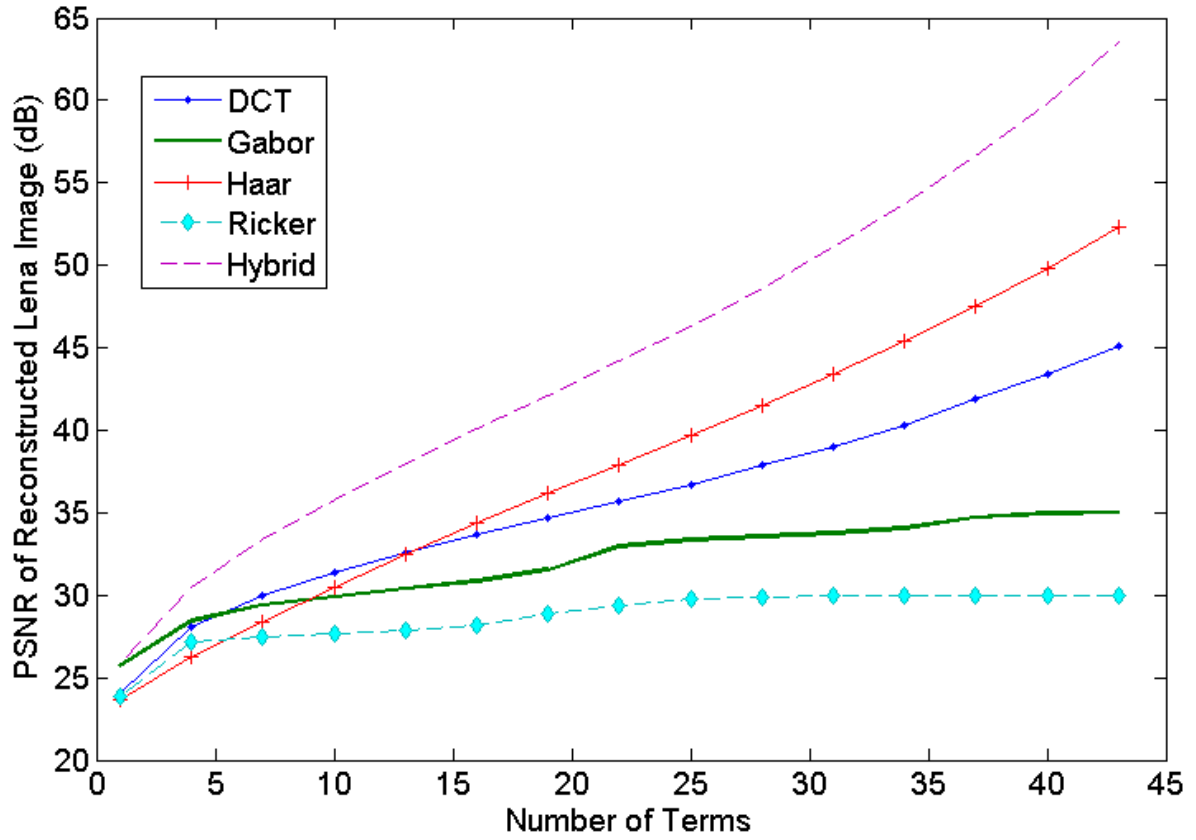


Figure 4-8 PSNR of the reconstructed Lena image

The Lena image is reconstructed for visual verification of the PSNR plot. The reconstructed Lena image using the DCT and hybrid dictionaries are shown in Figure 4-9 using 3 non-zero terms and 5 non-zero terms per patch. The difference in quality is easily evident in the 3-term reconstruction and slightly evident in the 5 term reconstruction. The image reconstructed using the DCT dictionary with 3-terms per patch has more pixilation around the shoulder, chin and retina area compared to the image reconstructed using the hybrid dictionary with 3 non-zero terms per patch.

The 5-term DCT dictionary based reconstruction is slightly lighter and has more pixilation in the background.



(a) Original Image



(a) 3-term DCT reconstruction



(c) 3-term Hybrid reconstruction



(d) 5-term DCT reconstruction



(e) 5-term hybrid reconstruction

Figure 4-9 Reconstructed Lena test image

4.4.3 Sparse constrained image reconstruction using large patches

This following test will measure the impact of increasing the patch size. The 512x512 grey-tone Lena image was segmented into 8x8 pixel patches for a total of 4096 patches with 64 integers representing the pixel data per patch. The hybrid dictionary obtained an image reconstruction with a PSNR of 40 dB using only 13 terms per patch. However, we may achieve better compression by using larger patches because it reduces the total number of patches that need to be encoded. This section will evaluate the effectiveness of the hybrid dictionary on images with larger patch size. This section also demonstrates the benefit of the hybrid dictionary's ability to accommodate varying problem size easily. Figure 4-10 shows the PSNR of the reconstructed image using the DCT and hybrid dictionaries. The hybrid dictionary is the only dictionary that is able to produce a reconstructed image with a PSNR of 40 dB while using fewer than 62 non-zero coefficients. The difference in reconstruction quality is easily evident when comparing the images that are reconstructed using 13 non-zero coefficients per patch. The reconstructed image using the DCT and hybrid dictionary is shown in Figure 4-12 and Figure 4-13 respectively. The image reconstructed using the DCT dictionary has pixilation within the patch that is transferring from black to light gray. The patches around the shoulder are a good example of this scenario.

Dividing the image into larger patch size did not improve the compression ratio. We were able to reconstruct the Lena image with a PSNR of 40 dB using 13 non-zero terms in the representation vector when the image was divided into 8x8 pixel patches which equates to an equivalent of 52 non-zero terms for an image

divided into 16x16 pixel patch. The larger patches have too much transition causing our linear signal coding model to fail. All remaining tests will have the images divided into 8x8 pixel patches. This experiment showed that the hybrid dictionary maintained its performance in larger problems.

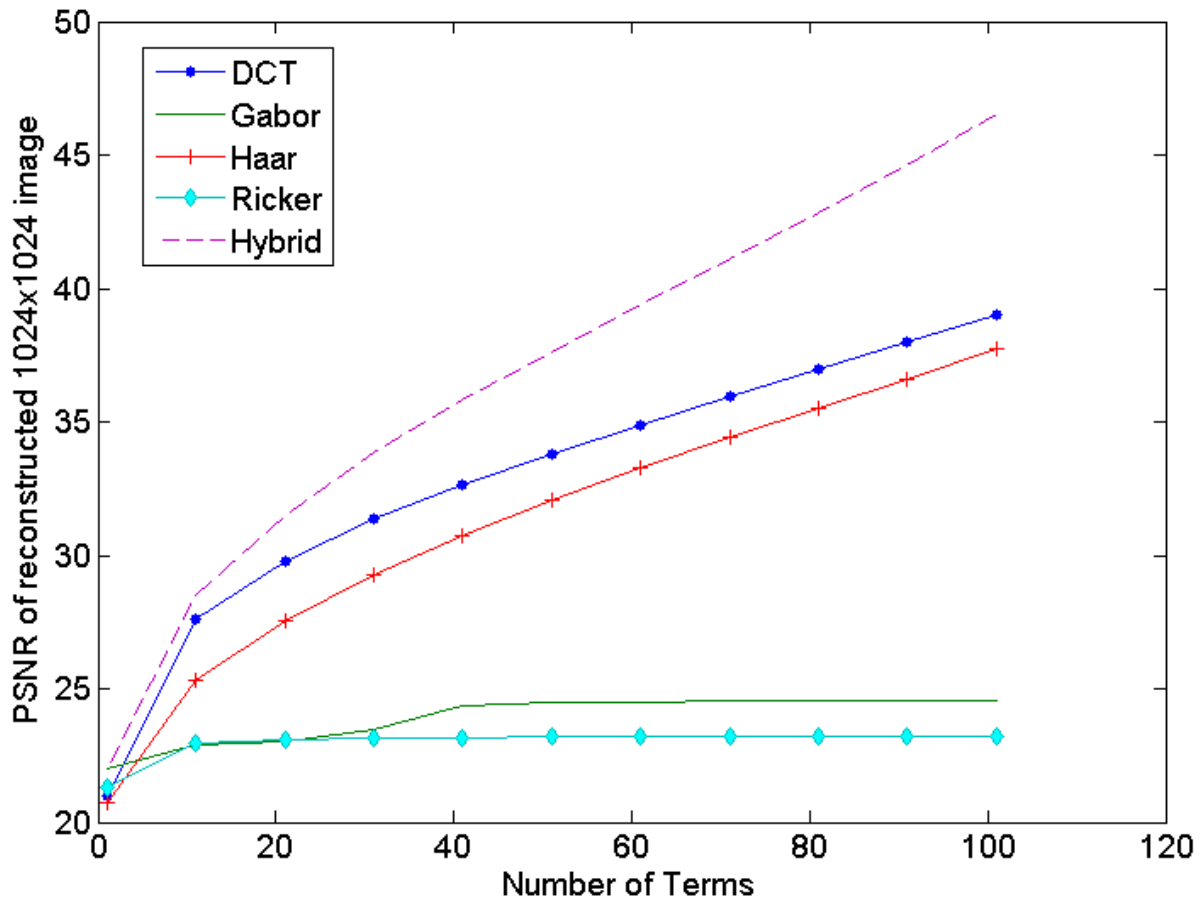


Figure 4-10 PSNR of the reconstruct Man test image with 16x16 pixel patches



Figure 4-11 Original 1024x1024 Man test image



Figure 4-12 Reconstructed Man test image using 13-terms per patch with a DCT dictionary



Figure 4-13 Reconstructed man test image using 13- terms per patch with hybrid dictionary

4.5 Hybrid dictionary robustness test

Two additional test images are selected to test the robustness of the hybrid dictionary. The first image is a grey-tone image of a primate and the second is an RGB image of bell peppers. Both images are 512x512 pixel image and are divided into 8x8 pixel patches.

4.5.1 The Primate image

The PSNR of the primate image reconstruction indicates a similar pattern. The hybrid dictionary performs better than the DCT and Haar dictionaries and the Ricker and Gabor dictionaries stagnate early and little improvements are seen as the number of non-zero terms in the representation vector is increased. The Haar, DCT and hybrid dictionaries all improve the image reconstruction quality as the number of non-zero terms per patch is increased. The hybrid dictionary is able to reconstruct the primate image with a PSNR of 35 dB using 19 non-zero terms per patch while it takes the DCT dictionary 32 terms per patch and the Haar dictionary 37 terms per patch to reconstruct the primate image with a PSNR of 35 dB. The PSNR plot for the reconstructed primate image is shown in Figure 4-14.

Also the original primate image along with the 5-term per patch reconstructions using both the DCT and the hybrid dictionaries are shown in Figure 4-15. One can easily notice the quality difference in the reconstructed image when examining the retina of the primate. The image reconstructed using the DCT dictionary has more pixilation while the hybrid dictionary produces a sharper image. Also the reconstructed primate image using the hybrid dictionary has a closer

resemblance to the original image because it contains more details in the face with defined hair lines and a closer color pallet match.

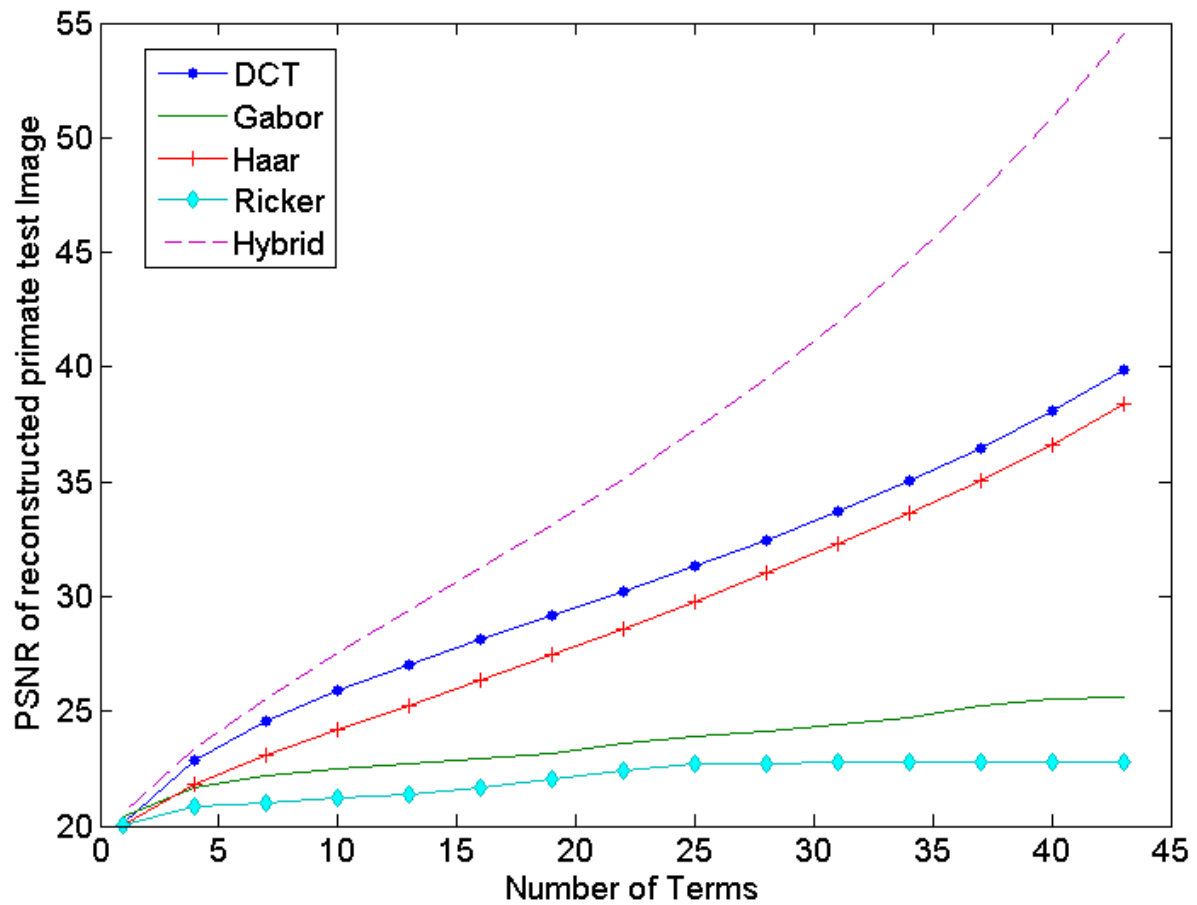
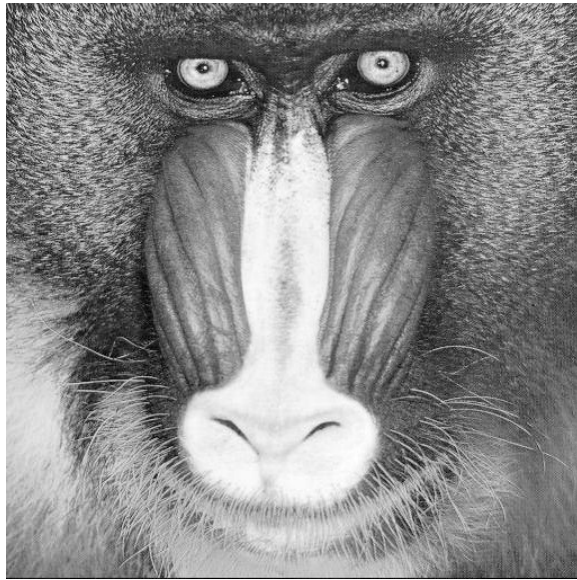
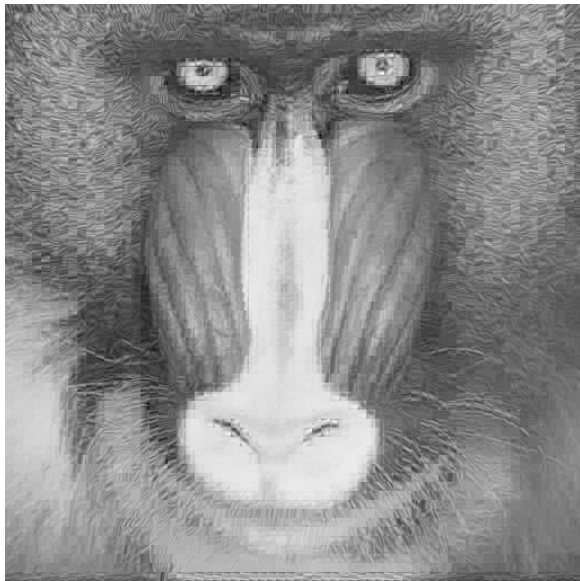


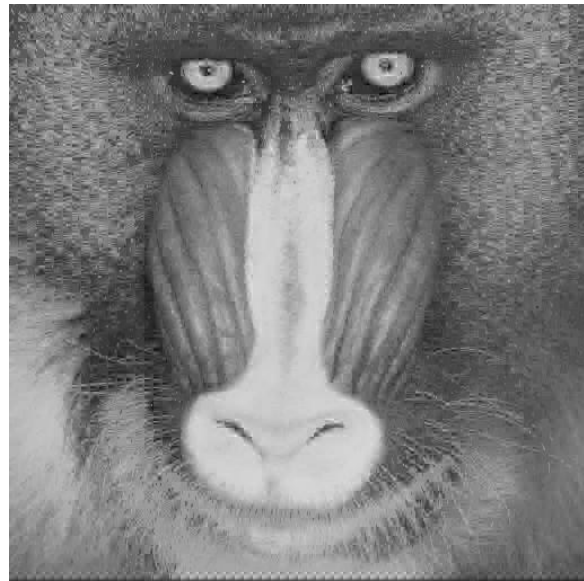
Figure 4-14 PSNR of the reconstructed primate image



(a) Original test image of the primate



(b) 5-term DCT reconstruction



(c) 5-term Hybrid reconstruction

Figure 4-15 Reconstructed primate images using 5 terms per patch with the DCT and hybrid dictionaries

4.5.2 The bell pepper image

The bell pepper test image is a 512x512 pixel RGB color image. This image tests the dictionaries ability to represent sharp transitions. The original image is shown in Figure 4-17. The image reconstructed using 5 non-zero terms per patch with the DCT dictionary is shown in Figure 4-18 and the image reconstructed using 5 non-zero terms per patch with the hybrid dictionary is shown in Figure 4-19. The image reconstructed with the DCT dictionary is inferior to the image reconstructed with the hybrid dictionary, especially in the region where the bell pepper creates a shadow and the colors are transitioning quickly. The difference is clearly evident when examining the magnified segments of the reconstructed image shown in Figure 4-20. The PSNR of the reconstructed image also indicates a similar pattern showing the hybrid dictionary being able to reconstruct the image with a PSNR of 35 dB using only 12 non-zero terms per patch but the DCT and Haar dictionaries both require 20 non-zero term per patch. The PSNR of the reconstructed bell pepper image for the varying degree of sparsity is shown in Figure 4-16.

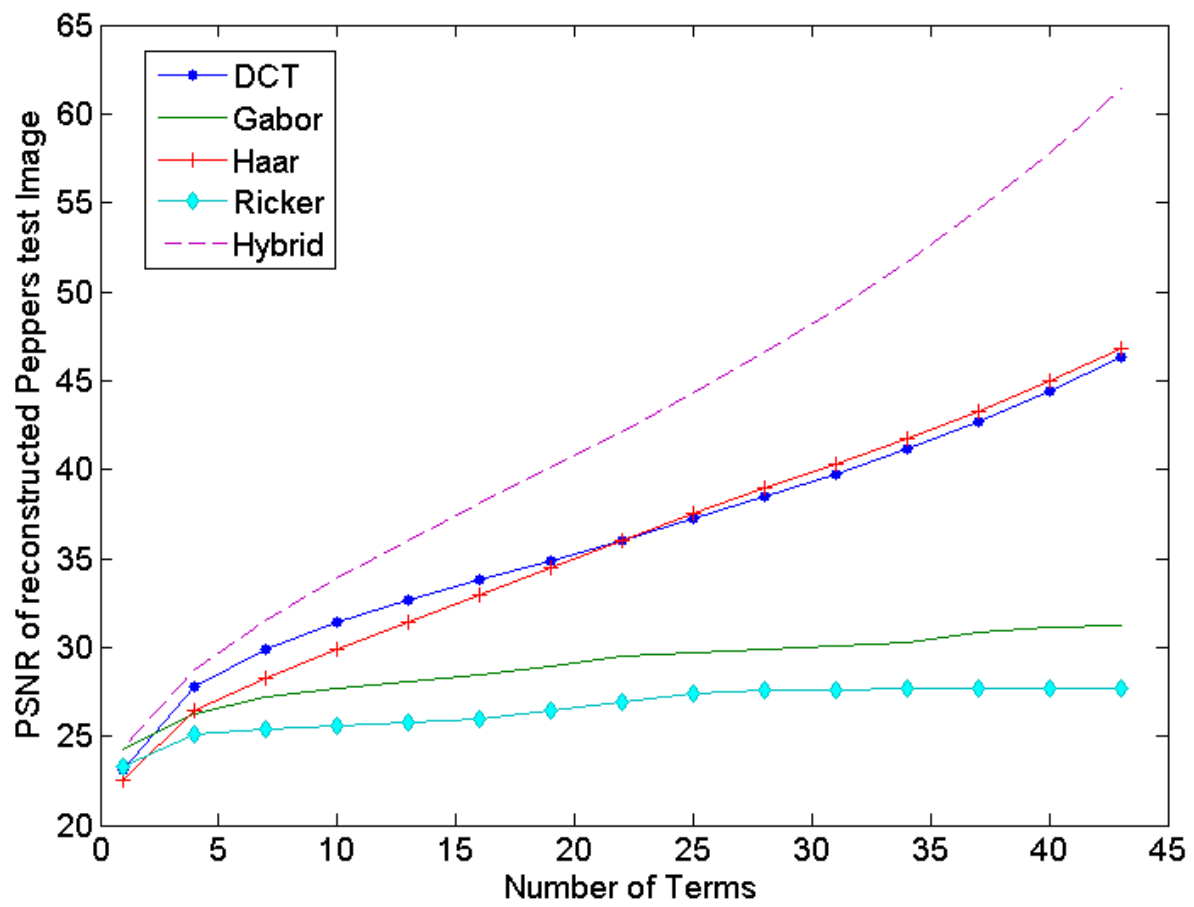


Figure 4-16 PSNR of the reconstructed bell pepper image



Figure 4-17 Original bell peppers test image



Figure 4-18 Reconstruction using 5 terms per patch with a DCT dictionary



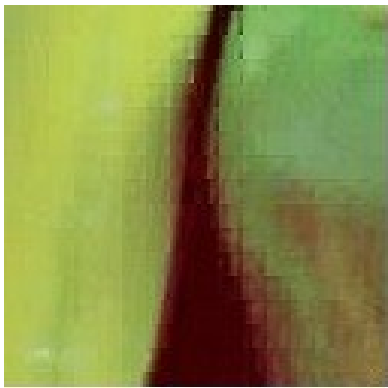
Figure 4-19 Reconstruction using 5-terms per patch with a hybrid dictionary



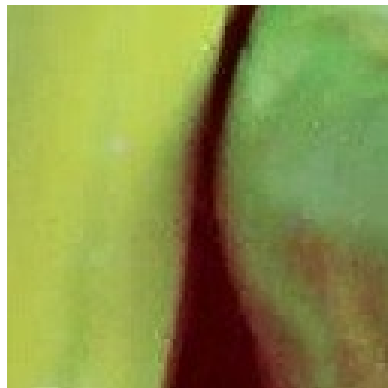
(a) 5-term DCT reconstruction



(b) 5-term Hybrid reconstruction



(c) 5-term DCT reconstruction



(d) 5-term Hybrid reconstruction

Figure 4-20 5-term bell pepper image reconstruction magnified

4.6 Limitations of the Hybrid dictionary

The hybrid dictionary is able to reconstruct a variety of test signals and images with better reconstruction quality while using fewer non-zero terms per patch compared to both the DCT and Haar dictionaries. The performance gain is due to the inclusion of additional nonlinear functions such as the polynomial, boxcar, rational, and exponential function. The inclusion of these functions comes with additional

computation cost. The success of the discrete cosine transform is due to the availability of fast algorithms to code the signal. The DCT can also be implemented using hardware because of the well-defined atoms used in the DCT which frees up computing power for other tasks. This is one of the reasons why the DCT-based JPEG compression standard is widely used on application-specific circuitry such as digital cameras, DVD players, mobile phones, and other applications requiring fast response with low computing power. The DCT is a cost effective solution since the cost of additional storage is less than the cost of processing power. The hybrid dictionary will not be a suitable solution for hardware based application such as cameras because of the difficulty involved with computing the polynomial, exponential, and rational functions using limited computing power. The hybrid dictionary is very flexible and does not have a well defined set of parameters like the DCT dictionary which will make it harder to implement on Asic, micro-controllers and other low-end microprocessors that are used in portable devices.

However, there are applications where the hybrid dictionary can be used such as archival applications where the emphasis is on compression ratio and not on compression time. The hybrid dictionary can also be used in high quality image compression applications such as medical imaging where the reconstruction quality is more important than the speed and computational cost of the compression algorithm. Another application is in transmission over wireless and other slow networks. The reduced file size allows the file to be transmitted quicker through the network reducing the required bandwidth and increasing the transmission capacity effectively reducing cost. The encoding and decoding of the signal can be easily offloaded to the computers with reasonable computing power.

A simple cost analysis is performed in this section to validate the hybrid dictionary. There are costs associated with generating the dictionary, coding the signal, and decoding the signal. We are going to evaluate the cost by measuring the average time required to complete each task. Both dictionaries are tested on the same test computer, a Quad core 2.85 Ghz computer with 16 Gb of RAM running at 1066 Mhz. The DCT dictionary was generated in 80 microseconds with an error tolerance of 10 microseconds and the hybrid dictionary took 29 milliseconds with an error tolerance of 4 milliseconds. The time to generate the hybrid dictionary is at worst 33 milliseconds longer than the time needed to create the DCT dictionary.

The hybrid dictionary incurs additional computation cost in the signal coding algorithm if it is required to encode the image using the same number of coefficients per patch. For example, the hybrid dictionary requires approximately 176 seconds to code the Lena image using 46 non-zero coefficients while the DCT dictionary only requires 125 seconds. The hybrid dictionary is computationally more expensive when using the same number of terms but the hybrid dictionary does not require 46 terms to reconstruct the Lena image. The hybrid dictionary achieved a PSNR of 35 dB using 10 non-zero coefficients per patch equating to approximately 12 seconds to encode the image. The DCT dictionary achieved a PSNR of 35 dB for the Lena image using 15 non-zero coefficients per patch which requires approximately 18 seconds to encode. The hybrid dictionary is approximately 6 seconds faster than the DCT dictionary in coding an image of equal quality. The signal coding time for both the DCT and hybrid dictionary is shown for varying degree of sparsity in Figure 4-21

The Hybrid dictionary incurs an additional cost decoding the encoded signal. Shown in Figure 4-22 is the time required to reconstruct the image using the DCT

and hybrid dictionaries assuming the dictionary is already generated. If the dictionary is not generated the reconstruction cost will be increased by the cost of generating the dictionary. Once the dictionary is generated, the cost of reconstructing the signal is insignificant since there is a maximum difference of 10 milliseconds. The total cost to encode a 512x512 image with the hybrid dictionary is 12.03 seconds and approximately 0.04 seconds to reconstruct the signal. The DCT dictionary can encode and decode the same image with equal quality in approximately 18 seconds. The DCT dictionary generation and coding algorithm was not optimized for speed but there are optimized techniques available in literature for image-encoding with DCT. The hybrid dictionary was not optimized and we strongly believe there are opportunities for performance improvements in the process. Thus, the hybrid dictionary is faster than the DCT dictionary if there is enough computing power to generate the additional functions. The hybrid dictionary requires additional run time memory but it is ignored because it is insignificant on our testing equipment.

The hybrid dictionary requires more parameters to completely define the dictionary and thus it incurs additional storage cost. Assuming each dictionary parameter and coefficient is encoded using 16 bits and the dictionary index is encoded using 12 bits, then the cost of encoding the image is: the Storage Cost in bits = (Number of parameters to define the dictionary)*16 + (Number of terms per patch)*(Number of patches per image)*28. The hybrid dictionary has a total of 13 parameters: 8 parameters to define the polynomial atoms, 1 parameter to define the DCT atoms, 2 parameters to define the rectangle atoms, 1 parameter to indicate what type of dictionaries are included and another to indicate the number of data points in each atom. The DCT dictionary has only 2 parameters one to indicate the number of

data points in each atom and the second to indicate the number of additional phase shifts. To encode the Lena test image and achieve a PSNR of 30 dB, the hybrid dictionary requires only 4 non-zero terms and the corresponding storage cost is 458,960 bits. The DCT dictionary requires 8 non-zero coefficient per path and the storage cost is 917,536 bits. The original image is encoded with an 8 bit integer representing a grey-tone image for a total of 2,097,152 bits. The DCT dictionary reduces the file size by 56.25% while the hybrid dictionary obtains a 78.12% compression. The additional cost to define the hybrid dictionary is overcome by its ability to reconstruct the signal using fewer non-zero coefficients per patch.

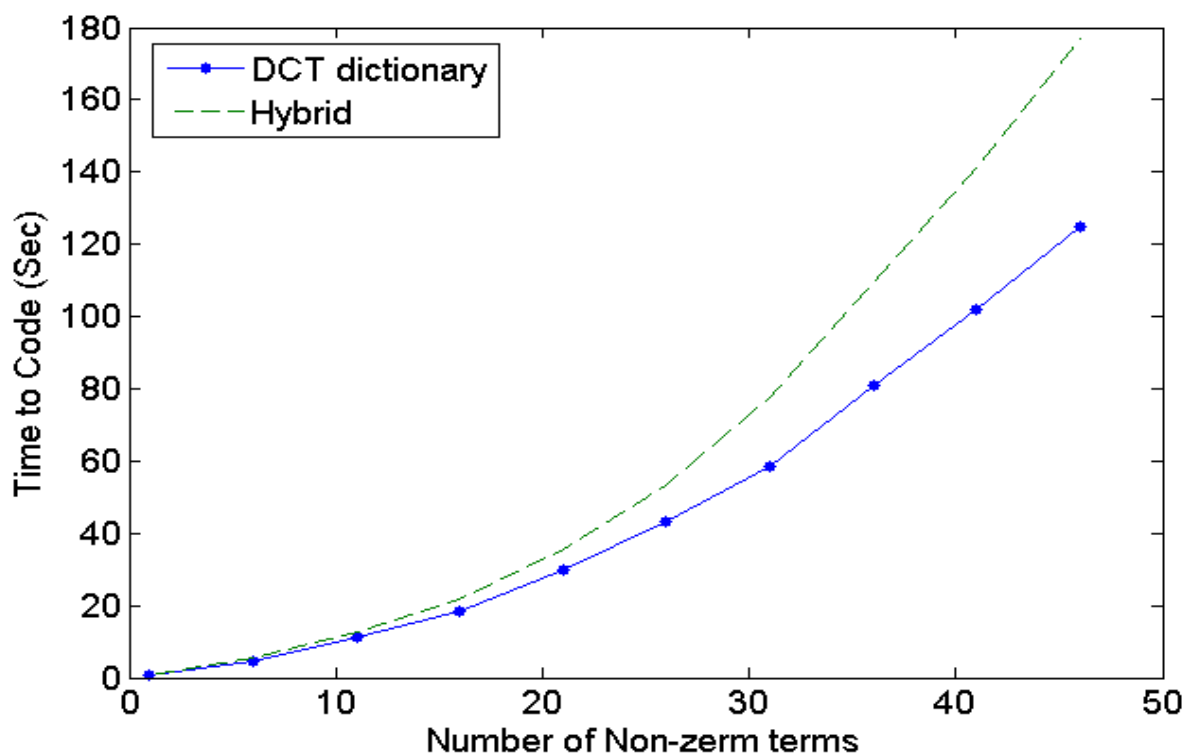


Figure 4-21 Signal Coding time for the Lena test image

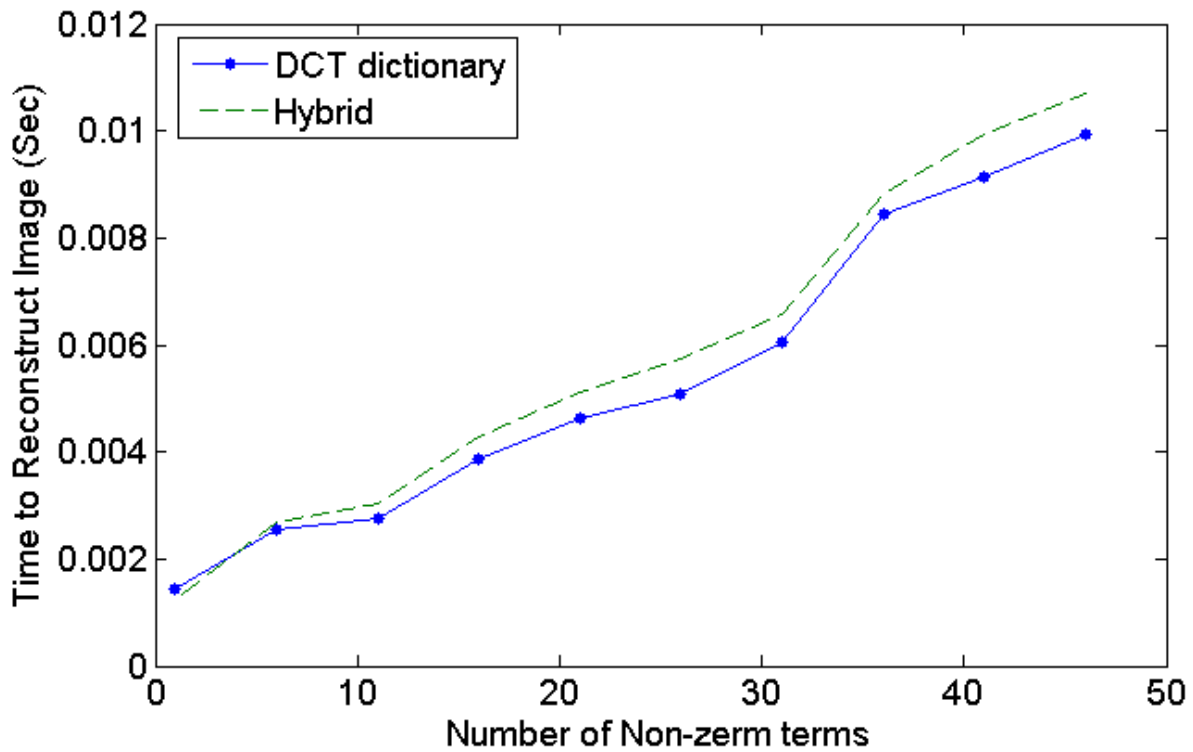


Figure 4-22 Time required to reconstruct the encoded lena image

4.7 Chapter Summary

Chapter 4 introduced the hybrid dictionary as a composition of the DCT, polynomial functions, rational functions, boxcar functions, root functions and a generic exponential function set. Furthermore the hybrid dictionary was tested against the DCT, Haar, Gabor and Ricker dictionaries on various test signals and images. The Lena, primate, and bell pepper images were 512x512 pixel image segmented into 8x8 pixel patches. The 1024x1024 Man test image was divided into 16x16 pixel patches. The hybrid dictionary produced better image reconstruction

compared to the DCT, Haar, Gabor and Ricker dictionaries because it was able to reconstruct the images with a PSNR of 35 dB or more using fewer number of non-zero coefficients in the representation vector. Furthermore several images were reconstructed for visual verifications.

The hybrid dictionary is a predefined dictionary since all the atoms in the dictionary is generated using mathematical models without feedback from the signals. The proposed dictionary is novel due to the inclusion of additional nonlinear atoms that allowed the dictionary to sparsely represent both harmonic and non-harmonic smooth signals as well as transient signals and images. However the hybrid dictionary has its limitations. The hybrid dictionary is applicable in applications such as large database archival and medical imaging applications where the compression ratio and reconstruction quality outweigh the cost of computing power and run-time memory. The ability to implement the DCT in hardware gives the DCT based compression techniques a major advantage over the hybrid dictionary in real-time application with limited computing power such as the digital cameras.

CHAPTER 5

OPTIMIZED PREDEFINED DICTIONARY

In chapter 4 we proposed a hybrid dictionary composed of cosine, polynomial, boxcar, rational, and exponential functions. The hybrid dictionary was able to provide significant improvements in reconstruction quality using additional computing resources. During our testing, both the DCT and the Haar dictionary showed improvements as the number of coefficients used in the representation vector increased but the Ricker and Gabor dictionaries stagnated. The DCT and Haar dictionaries have a well-established set of parameters to generate the atoms but the Ricker and Gabor dictionary parameters were arbitrarily selected. Could the parameters that generate the dictionary be optimized? Have we been using the optimal parameters for the dictionaries? In this chapter we propose an evolutionary algorithm to optimize the dictionary generator parameters to examine if improvements can be made.

The optimized dictionary differs from the signal-based adaptive dictionaries because the signal-based adaptive dictionaries use the training samples to learn the atoms. The learnt atoms are components of the signal and the atoms cannot be represented using a simple mathematical function without some kind of curve fitting algorithm. The optimization process is different because it does not learn new atoms from the training samples instead it optimizes the dictionary parameters to find the best fit for the training samples. The optimization process allows the optimized

dictionary to maintain the benefits of a predefined dictionary such as low storage cost while increasing its ability to represent complex signals.

5.1 Parameter optimization using Particle Swarm Optimization

We propose an evolutionary algorithm to find the parameters for the dictionary generators. Traditional optimization algorithms like the gradient descent make assumptions about the fitness function which restricts the type of fitness functions that can be optimized. Evolutionary algorithms such as the particle swarm optimization (PSO) method have the following benefits that are ideal for solving the dictionary generator parameter optimization problem:

1. The algorithms make little assumption about the fitness function.
2. The fitness function does not have to be continuous, or differentiable.
3. The implementation is easy with low computational cost.
4. The algorithm on average provides a good quality solution.

The PSO algorithm was first introduced by James Kennedy, and Russell Eberhart in 1995 IEEE convention [27]. The particle Swarm optimization algorithm is an evolutionary computation technique based on the motion of an individual in a swarm. The first PSO algorithm was used to model the movements of a flock of birds searching for food. The PSO algorithm has been used to solve many engineering problems including optimization of PID controller gains for complicated nonlinear systems [28-31]. The PSO algorithm is shown in Algorithm 5-1.

Algorithm 5-1 Particle Swarm Optimization Algorithm

- | |
|--|
| 1. Initialize a population with particles. |
|--|

2. While exit condition not met
 - a. Update the swarm constriction factor using equation (5.1)
 - b. For each particle
 - i. Update particle velocity $V_i(k)$ using equation (5.2) .
 - ii. Ensure ($V_{\min} < V_i < V_{\max}$)
 - iii. Update particle state $X_i(k)$ using equation (5.3).
 - iv. Ensure ($X_{\min} < X_i < X_{\max}$)
 - v. Current fitness = $F(\text{particle state})$
 - vi. If Current fitness is better than particle best fitness:
 1. Set particle best state = current state
 2. Set particle best fitness = Current fitness
 - vii. If Current fitness is better than Global best fitness:
 1. Set global best state = current state
 2. Set global best fitness = Current fitness.
 - c. End for
3. End loop

Many successful variations for the PSO algorithm have been suggested since its introduction in 1995 but the prominent variation is the constriction factor PSO [32]. The constriction factor allows the particles to aggressively search the solution space at the early stage of the algorithm and as the algorithm evolves the particles start to search locally around its current state. The swarm's constriction factor is calculated using equation (5.1) where the minimum and maximum value is predetermined at the start of the algorithm and the suggested values in literature are 0.4 to 0.9. Each

particle's $(k+1)^{\text{th}}$ velocity is calculated using equation (5.2) where $U(0,1)$ indicates random numbers from a uniform distribution and $C1$ and $C2$ are predetermined constants typically set to 2.05. Also, since the problem is multi-dimensional the \otimes indicates an element by element multiplication. The particle's state is calculated using equation (5.3). The PSO is a recursive algorithm and it does not guarantee an optimal solution. Also due to the random nature of the particle's movements within the solution space, the final solution is not exactly the same each time the algorithm is run. However, the solutions are within a close range and on average the algorithm finds a good solution.

$$w_k = \left(w_{\max} - \frac{w_{\max} - w_{\min}}{Iteration_{\max}} k \right) \quad (5.1)$$

$$\vec{V}_{k+1} = w_k \vec{V}_k + C_1 U(0,1) \otimes (\vec{P}_{gbest} - \vec{X}_k) + C_2 U(0,1) \otimes (\vec{P}_{Best} - \vec{X}_k) \quad (5.2)$$

$$\vec{X}_{k+1} = \vec{X}_k + \vec{V}_{k+1} \quad (5.3)$$

5.2 The parameter screening fitness function

Prior to suggesting the fitness function, the dictionaries and the parameters that can be optimized for each dictionary along with the expected solution space is presented in Table 5-1. The Gabor, Ricker and Polynomial dictionaries all have multiple parameters that can be optimized. Attempting to optimize all parameters is a time consuming process. Thus a compromise is made by optimizing a limited set of parameters at any given time. For example, the polynomial dictionary generator has 2 sets of adjustable parameters the coefficients of the polynomial, and the region of

interest. The PSO algorithm is used to optimize either the region of interest with a default list of polynomial coefficients or vise-versa. Similarly for the Gabor and Ricker dictionaries, only one of the parameters will be optimized at any given time. The standard Haar dictionary presented in this thesis does not have any adjustable parameters and thus cannot be optimized. The boxcar and moving square wave dictionaries have limited adjustable parameters. It is possible to optimize the width of the pulses however the intention of adding the boxcar and the moving square wave atoms is to aid the hybrid dictionary and it is not intended to be a standalone dictionary.

Table 5-1 Dictionary and tunable parameters

Dictionary	Tunable Parameters	Solution Space
DCT $DCT = \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k + 2\pi p_1\right)$	Phase = p1	p1 = [0, 1]
Ricker Wavelet $\psi(t) = \frac{1}{\sqrt{ a }}(1 - t^2)e^{-at^2}$	Scaling factor = a Region = t	a = [-5, 5] t = [-5, 5]
Boxcar & moving square wave	Width of impulse	W = [1, size]
Polynomial, Rational, Roots $P_c(t) = t^c + \sum_{i=0}^{c-1} a_i t^i$	Coefficients = ai Region = t Order = c	ai = [-1, 1] Region = [-5, 5] C = [1, 5]
Gabor	Scaling factor = a	a = [0, 5]

$Gabor = e^{-a^2 t^2} \cos(2\pi ft)$	Frequency = f	f= [0, 100]
	Region = t	t= [-3, 3]

The proposed optimization feedback system is shown in Figure 5-1. The PSO algorithm should optimize the parameters to generate a dictionary that will maximize the PSNR of the test signal. The fitness function will include the dictionary generator using the parameters proposed by the PSO algorithm. The OMP algorithm will use the generated dictionary to code the test signal and the resulting PSNR of the reconstructed test signal is fed back to the PSO algorithm to be maximized.

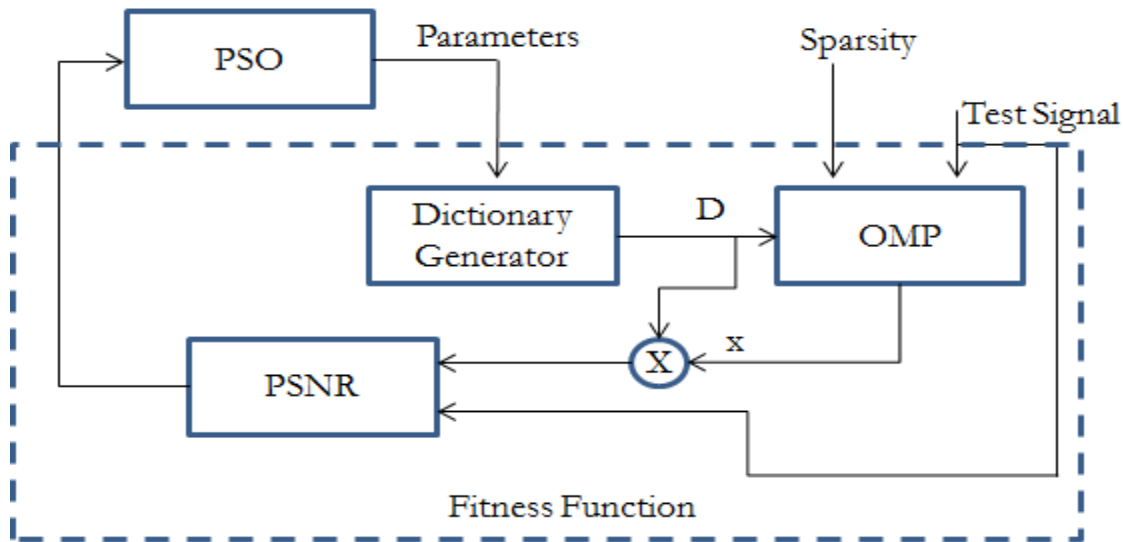


Figure 5-1 Predefined dictionary optimization feedback system

5.3 Optimized predefined dictionary results

The preliminary testing clearly shows that the dictionaries were not optimized and we can gain significant improvements by optimizing each dictionary using the evolutionary algorithm. The optimization of the dictionary generator was done by randomly selecting 1% of the image as a training sample and optimizing the parameters of the dictionary generator to achieve the highest PSNR for the training set. The PSO algorithm ran with 15 particles for a maximum of 500 iterations or 5 minutes to limit the optimization time. The optimized results improve with the number of samples in the training set until about 10% for the Lena test image. The PSO optimization algorithm does not provide better results when the training samples is greater than 25% of the entire image unless the number of iterations are significantly increased to approximately 10 000 counts or more. The additional computation cost did not justify the performance gain.

In Figure 5-2, the PSNR of the Lena test image using a DCT dictionary with 129 atoms generated using the phase shifting method presented in Chapter 2 is shown against the PSO optimized DCT dictionary and the hybrid dictionary. Both, the DCT dictionary and the PSO optimized DCT dictionary is the same size and have similar storage requirements. The optimization process took on average 3 minutes on the test computer. The average solution to the optimization problem was found to be $p_1 = 0$ and $p_2 = 0.302$. The optimized DCT dictionary provides better PSNR compared to the standard DCT dictionary. The hybrid dictionary is still better than the optimized DCT dictionary. Once the optimized parameters are found, the dictionary does not incur any additional cost during the signal encoding and decoding

process. The hybrid dictionary is able to encode the Lena image using 10 non-zero terms per patch and obtain a reconstructed image with a PSNR of 35 dB. The optimized DCT dictionary requires 11 non-zero coefficients per patch to obtain an image of equal quality. The optimized DCT achieves a 72.3 % compression while the standard DCT dictionary only achieved 52.6% compression. The initial computation cost can be easily absorbed even on low end desktop computers but the implementation of the optimization technique is too expensive on low end processors such as ASICs. The optimized DCT dictionary does not have the added signal coding cost incurred by the hybrid dictionary.

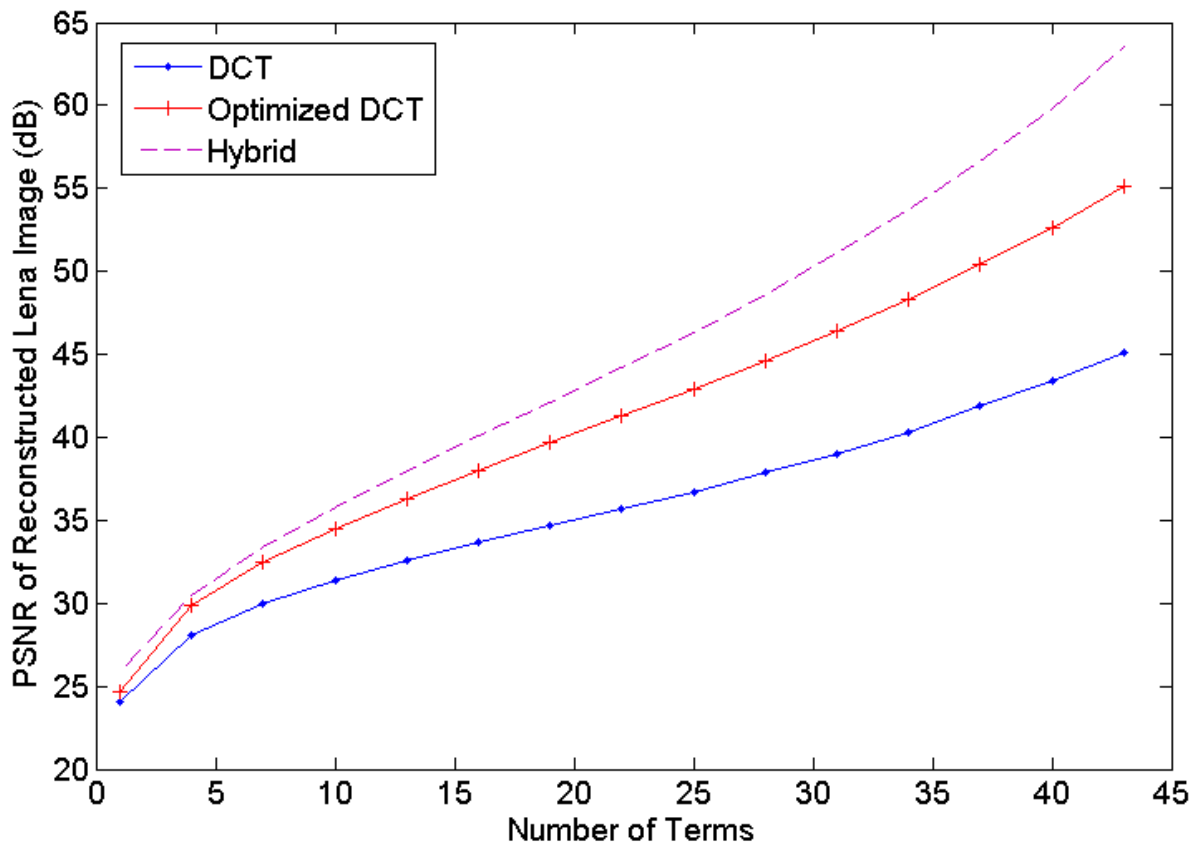


Figure 5-2 PSO optimized DCT dictionary vs. DCT dictionary in image reconstruction

Figure 5-3 shows the PSNR of the reconstructed Lena test image using a PSO optimized polynomial dictionary against the standard polynomial dictionary generated using the algorithm presented in Section 4.2. There are a few tunable parameters that can be optimized for the polynomial dictionary generator such as the region of interest, the coefficients of the polynomial and the highest order of the polynomial. Optimizing the region of interest for the polynomial dictionary generator provided the greatest improvements. The standard region of interest was between -1 and 1 and the optimized region of interest for the Lena test image was between -1.5 and 1.5.

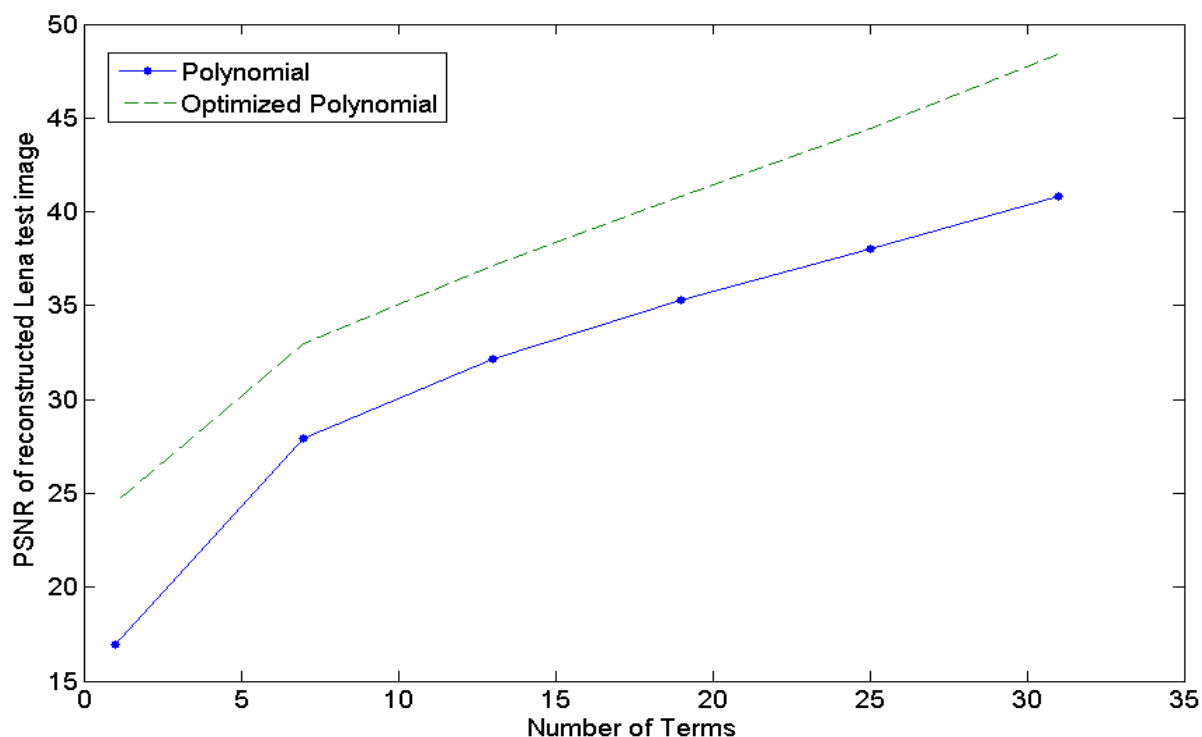


Figure 5-3 PSO optimized Polynomial dictionary tested on the Lena image

The PSNR of the reconstructed Lena image using the PSO-optimized Gabor dictionary is shown in Figure 5-4. In Figure 5-4, the default Gabor dictionary is compared with 3 PSO-optimized Gabor dictionaries. The Gabor dictionary generator has 3 parameters, and each parameter was optimized while keeping the default values for the other 2 parameters. The standard Gabor dictionary had 4 scaling factors, and 2 frequency values to generate 512 atoms. The optimized region of interest does not change the size of the dictionary. Frequency and scaling factor optimization process only selects 2 values thus the optimized scaling factor dictionary is half the size of the standard Gabor dictionary. The PSO optimized Gabor dictionaries are able to represent the signal better using fewer non-zero coefficients. However, the PSO-optimized Gabor dictionaries still plateau without significant improvements if the number of non-zero coefficients in the representation vector is increased.

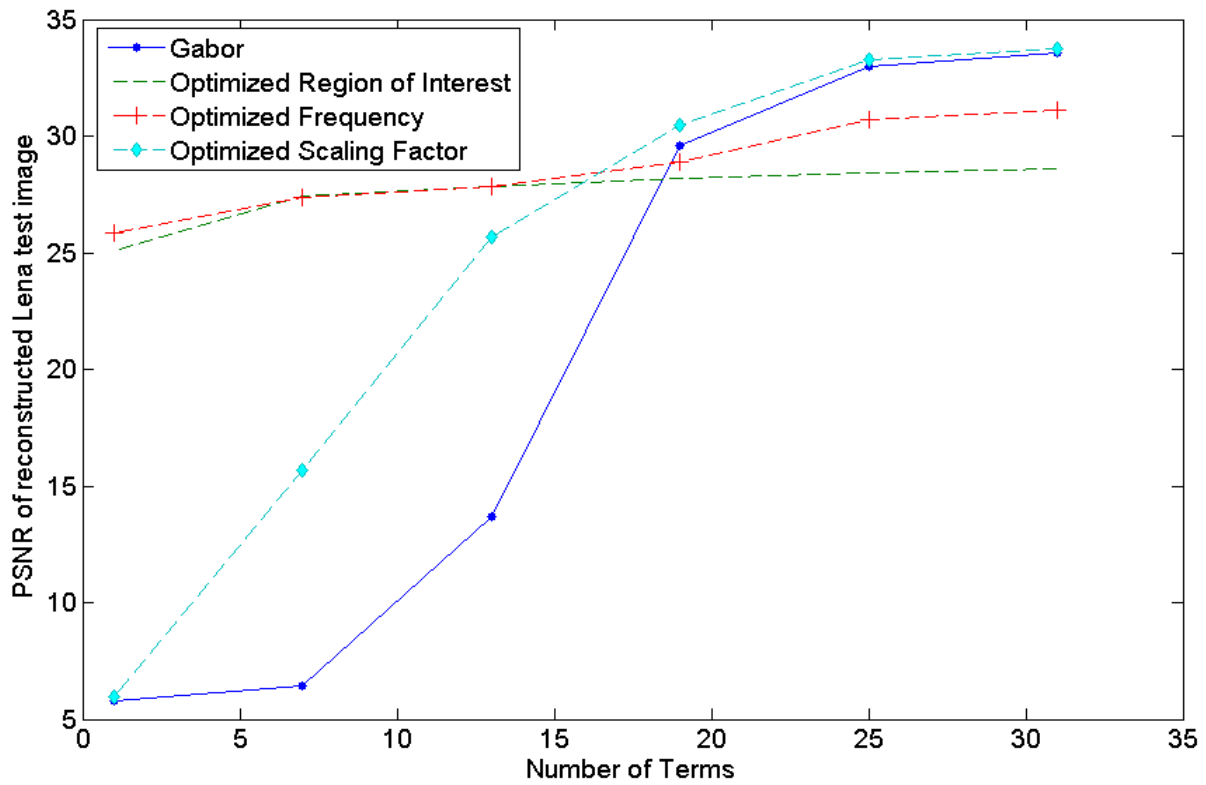


Figure 5-4 Optimized Gabor dictionaries

We have demonstrated that the DCT, polynomial and Gabor dictionaries can be optimized to improve the signal reconstruction quality. However optimization of some dictionaries may not produce significant improvements which will indicate one of the following scenarios has occurred:

1. The dictionary is already optimized
2. The PSO algorithm is stuck in a local optimal solutions or saddle points.
3. The dictionary is not well suited for the current signal.

We can reduce the computational cost of the PSO optimization algorithm in the event the dictionary is not suited for the signal by terminating the process prior to the exit condition. We can monitor the status of the global best state and if there is no change in the global best state for more than the preset number of consecutive iterations, the PSO algorithm is considered to be stuck in a saddle point and would be terminated. This additional check to see if the algorithm is stuck in a local optimal solution or saddle points on average will reduce the overall computation cost.

5.4 Chapter summary

We proposed to optimize the predefined dictionary in hopes of improving the signal reconstruction quality. The optimization process could be a time consuming process and thus we limited our algorithm to maximum of 5 minutes. We also limited the number of samples from the image to reduce the computation cost. Using more samples increases the computation cost per algorithm iteration, reducing the number of iterations and compromising the solution quality. During our testing, we have found 1- 5% of the patches to be sufficient. The optimized DCT and polynomial dictionaries performed better than the corresponding dictionaries generated using arbitrarily parameters. The Gabor dictionary showed improvements when using fewer non-zero coefficients but it did not sustain the improvements as the number of nonzero terms in the representation vector increased. The predefined dictionary optimization is an effective solution because the additional computation cost is only incurred in the dictionary optimization process; it does not incur any additional cost when coding or reconstructing the signal. The optimized predefined dictionary

contains all the other benefits of the predefined dictionary such as low storage cost, low signal coding and signal reconstruction cost.

The PSO algorithm has its own drawbacks because the exact same solution is not repeatable. However, the dictionary optimization problem is clearly a challenging multi-modal problem, and the PSO algorithm attempts to select one of these depending on the random evolution of the algorithm. The PSO algorithm may get stuck between local optimum solutions and end up providing sub-optimal solutions. However, all optimization algorithms have similar limitations and compromises. We believe the PSO algorithm makes a good compromise between computation cost and algorithm run time and average solution quality.

CHAPTER 6

OPTIMIZED HYBRID DICTIONARY

In chapter 5 we proposed to optimize the predefined dictionary parameters to improve signal reconstruction quality. In this chapter, we propose to filter the dictionary types that are included in the hybrid dictionary, effectively reducing the cost of coding the signal. We anticipate that the optimized hybrid dictionary will be robust, which will allow us to absorb the initial cost of optimizing the hybrid dictionary. To ease implementation we consider the hybrid dictionary as a list of smaller dictionaries instead of a single dictionary. Once we view the hybrid dictionary as a list of dictionaries, we can simply add or remove the smaller dictionaries from the list by adding and removing the corresponding parameters. We also suggest simple modifications to the OMP algorithm to enable the algorithm to better handle a list of dictionaries.

6.1 Multi-dictionary OMP algorithm

The added benefit of viewing the hybrid dictionary as a list of dictionaries is it allows us to determine the signal composition. The underlying mathematical model of the signal can be seen by the number of atoms selected from each dictionary. We can use this knowledge to optimize and filter the hybrid dictionary.

The pseudo code of the modified OMP algorithm, referred to as multi-dictionary OMP, to handle multiple dictionaries is shown in Algorithm 6-1. The

multi-dictionary OMP algorithm generates a list of atoms with the maximum projection of the signal residual from each dictionary type then it selects the atom from the generated list that minimizes the representation error. The OMP algorithm only uses the projection of the residual onto the dictionary as a basis to select the best fit atom; this will provide an optimal solution if the atoms are orthogonal but since the hybrid dictionary is not orthogonal minimizing the error again aids in picking the best fit atom. Just like the OMP algorithm, the multi-dictionary OMP algorithm does not guarantee an optimal solution. The multi-dictionary OMP algorithms computation cost increases with the number of dictionaries in the list and the number of atoms in each dictionary.

Algorithm 6-1 Pseudo Algorithm for Multi-dictionary OMP

1. Given Dictionary $D_list = \{D_1, D_2, D_3, \dots, D_{dm}\}$
2. Initialize residual error : $r = y$
3. Initialize selected dictionary set $S = \{\text{Null set}\}$
4. Initialize selected Atoms set $AS = \{\text{Null set}\}$
5. Initialize selected dictionary $D_{sel} = \{\text{Null Set}\}$
6. Initialize used Atoms $D_{used} \{1, 2, \dots, m\} = \{\text{Null set}\}$
7. While $\|x\|_0 < L$
 - a. For $j = 1: dm$
 - i. $D = D_list\{j\}$
 - ii. $UsedAtomList = D_{used}\{j\}$
 - iii. Find i s.t $\min_i \|ad_i - r\|_2^2 \quad \forall d_i \notin UsedAtomList$

- iv. Error (j) = $\|ad_i - r\|^2$
- v. SelectedAtom(j) = d_i ;
- b. end
- c. find dictionary: $i \leftarrow \min_i(\text{error}(i))$ for $i=1,..dm$
- d. Add i to the selected dictionary list: $S = S \cup \{i\}$
- e. Add to atom list : $AS = AS \cup \{\text{SelectedAtom}(i)\}$
- f. Add selected atom: $D_{sel} = (D_{sel} \cup D_i(:, \text{SelectedAtom}(i)))$
- g. Add atom to used set:

$$\text{Dused } \{i\} = \{ \text{Dused}\{i\} \cup \text{SelectedAtom}(i) \}$$
- h. Find best x: $x = \min(y - (D_{sel})x)$
- i. Find remaining error : $r = y - (D_{sel})x$
- 8. end while
- 9. Provide sparse matrix for each dictionary that is used using S, AS.

6.2 Optimized hybrid dictionary algorithm

The optimization objective is to find a smaller dictionary that achieves similar signal representation quality while using fewer than the preset maximum number of nonzero terms in the representation vector. The hybrid dictionary can be improved by optimizing the atoms that are best suited to represent the signal. The optimization process could also remove redundant dictionaries which will improve the speed of the signal coding algorithm. The efficiency ratio of the dictionary is defined in equation

(5.4) which gives a rough estimate of the redundancy in each dictionary. The efficiency ratio indicates the percentage of the dictionary that is used by the multi-dictionary OMP algorithm compared to the number of atoms in the dictionary. The objective is to maximize the efficiency.

$$\eta_{Di} = \frac{T_{Di}}{\left(\frac{N_{Di}}{\sum_{i=1}^{dm} N_{Di}} \sum_{i=1}^{dm} T_{Di} \right)} \quad (5.4)$$

where T_{Di} = Number of terms used from Dictionary i

N_{Di} = Number of atoms in Dictionary i

We will demonstrate the hybrid dictionary optimization process by optimizing the dictionary for the test signal $y(t) = 4t^3 + 5\cos(5t)$. The test signal is a polynomial signal with some oscillatory behavior. To test if the efficiency ratio can be used to select the appropriate type of dictionaries; all the dictionaries shown in Table 6-1 are included in the list. The premise is to include the dictionaries with high efficiency ratio. The signal is reconstructed using 16 terms per patch which is about 25% of maximum number of terms per patch.

Table 6-1 Dictionary and number of atoms used for test

Dictionary Type	Total Atoms (N_{Di})	Terms Used (T_{Di})	Efficiency (η)
Dct	129	4	5.21

Gabor	513	0	0
Haar	64	0	0
Ricker	557	0	0
Basic Polynomial	10	3	50.06
Linear	11	0	0
Quadratic	110	0	0
Cubic	990	0	0
boxcar	195	8	444.5
Square waves	192	1	0.86
2D DCT	65	0	0
Logarithmic	64	0	0
Exponential	64	0	0
Rational	64	0	0

The efficiency ratio for each dictionary is also shown in Table 6-1. The results are positive because even though the polynomial dictionary only used 3 atoms to represent the signal it is the second most efficient dictionary in the list. The filtering process correctly filtered out the Logarithmic, exponential, rational, Gabor, linear, quadratic and the Ricker dictionaries. The filtering processes also correctly choose the 1D DCT dictionary instead of the 2D DCT dictionary since the test signal is one dimensional. The efficiency ratio can be used as a measure by the dictionary optimization algorithm to automatically expand the influential dictionaries while reducing the size of the ineffective dictionaries. However, the dictionary filtering algorithm needs to be careful in eliminating dictionaries since the filtering process failed to select the cubic

functions even though the dominant part of the test signal is a cubic polynomial. The cubic dictionary may have been excluded because the dictionary is not optimized. Prior to eliminating the dictionaries the optimization algorithm should optimize the parameters of the dictionary generator to find the best fit for the signals. The predefined dictionary optimization process using the evolutionary algorithm was demonstrated in the previous chapter.

The efficiency ratio alone does not provide enough information to remove a dictionary because even an inefficient dictionary may contain valuable atoms. The peak signal to noise ratio gained by the inclusion of the dictionary is a good indicator of the dictionaries worth. The Dictionary gain ratio is calculated using equation (5.5) using the following parameters: the $PSNR_{Di+}$ is the PSNR achieved with the dictionary, the $PSNR_{Di-}$ is the PSNR achieved without the dictionary. The dictionary gain per atom ratio is shown in equation (5.6) where T_{Di} is the total number of atoms in the dictionary. The dictionary gain ratio in conjunction with the dictionary gain per atom ratio could be used to remove a dictionary. Deciding to remove a dictionary based only on the dictionary gain ratio or gain per atom ratio would not produce the optimal solution because a large dictionary can be incorrectly removed while providing the best results. If a dictionary PSNR gain ratio and PSNR gain per atom ratio falls below a set minimum threshold then the dictionary should be removed.

$$PSNRGain_{Di} = PSNR_{Di+} - PSNR_{Di-} \quad (5.5)$$

$$GainPerAtom_{Di} = \frac{PSNR_{Di+} - PSNR_{Di-}}{T_{Di}} \quad (5.6)$$

The objective of the hybrid optimization algorithm is to reduce the size of the hybrid dictionary. The steps to generate the optimized hybrid dictionary are:

1. Optimize all dictionaries in the list.
2. Use the multi-dictionary OMP algorithm to code the signal with the optimized dictionaries.
3. Calculate the efficiency ratio for all the dictionaries in the list.
4. For any dictionary that has an efficiency ratio below the set threshold
 - a. Calculate the dictionary PSNR gain ratio and PSNR gain per atom ratio.
If both dictionary gain ratios are below the preset minimum threshold, remove the dictionary.

The resulting optimized hybrid dictionary (OHD) will attempt to maintain the signal representation quality while reducing the size and storage requirements of the hybrid dictionary. The smaller dictionary will speed up both the encoding and decoding process which may make up for the added cost of the optimization process. The OHD may not be worth the additional cost for natural images that can be represented with DCT, Wavelet and the proposed hybrid dictionary but the OHD is an alternative to signal-based dictionaries for complex signals.

6.3 Performance gain of the optimized hybrid dictionary

The hybrid dictionary optimization algorithm is optimized on the Lena test image. We compare the optimized hybrid dictionary with the DCT dictionary. The OHD algorithm ran for 45 minutes optimizing each dictionary and removing inefficient dictionaries. The final OHD composition had a DCT, polynomial, square

wave and boxcar dictionaries. The OHD dictionary had a total of 449 atoms generated with the parameters shown in Table 6-2.

The DCT dictionary (using $N_p = 2$) had 129 atoms and was the smallest dictionary. The hybrid dictionary composed of the DCT, polynomial, Gabor, Haar, Ricker and boxcar dictionaries had a total of 1411 atoms.

Table 6-2 OHD dictionary and parameters

Dictionary Type	Parameter	Number of Atoms
DCT	$p1 = 0.2816, 0.9532$	128
Polynomial	Power = 3 $T = [-1.14, -0.78]$	64
Square wave	Width = [1, 2]	128
Boxcar	Width = [1, 2, 64]	129

The PSNR of the reconstructed Lena test image is shown in Figure 6-1, showing the OHD performing on par with the hybrid dictionary. The proposed hybrid dictionary had the best signal reconstruction when using fewer than 25 non-zero coefficients per patch but the difference in PSNR is less than 0.5 dB. The hybrid dictionary is more than 3 times as large as the OHD. The OHD dictionary also has less storage cost than the hybrid dictionary and it will also be quicker at coding and decoding the signal because the cost of coding the signal is proportional to the number of atoms in the dictionary as seen in our cost analysis in chapter 4. The OHD will be beneficial if it is robust. Thus, the OHD optimized for the Lena image is tested on the primate image and the results are shown in Figure 6-2. The OHD

performs slightly better than the hybrid dictionary that included all the smaller dictionaries on the primate image indicating the OHD is robust.

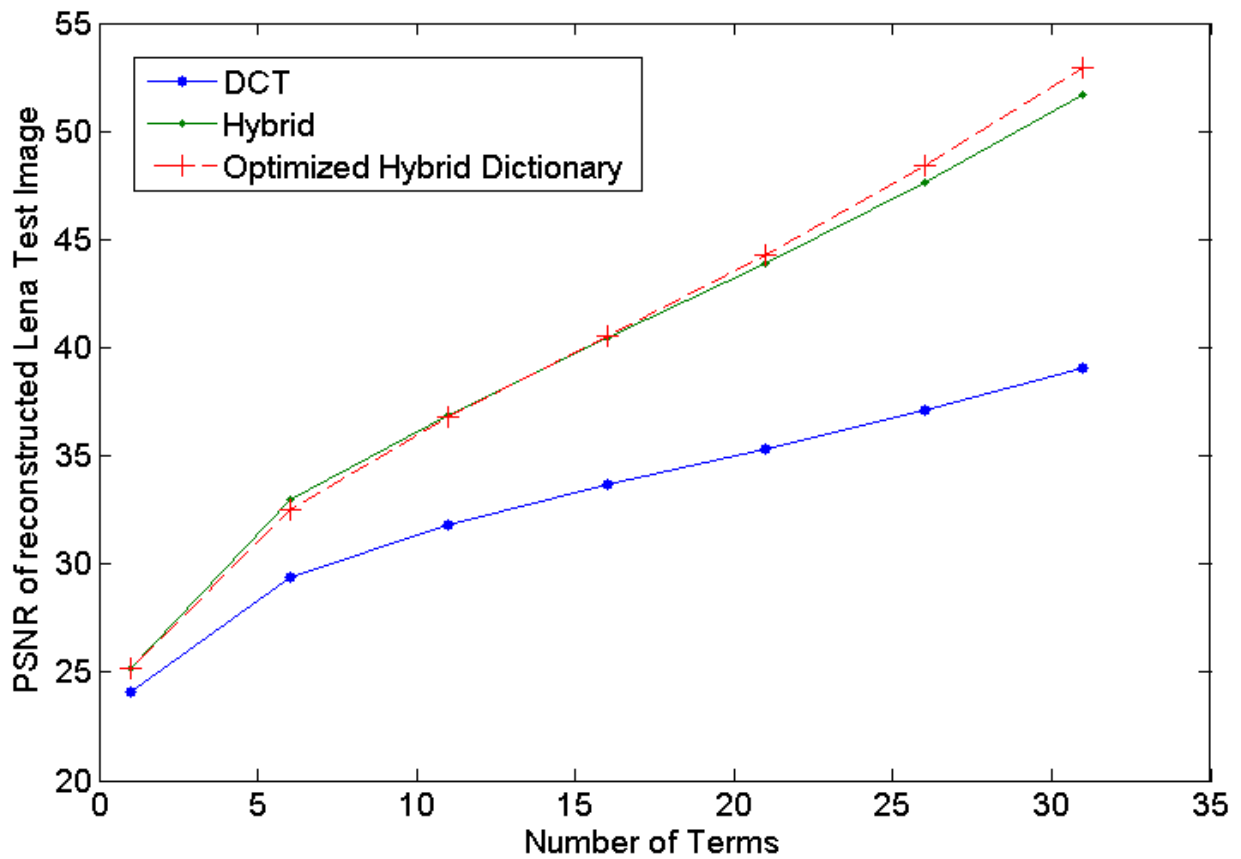


Figure 6-1 Lena image reconstruction using the optimized hybrid dictionary

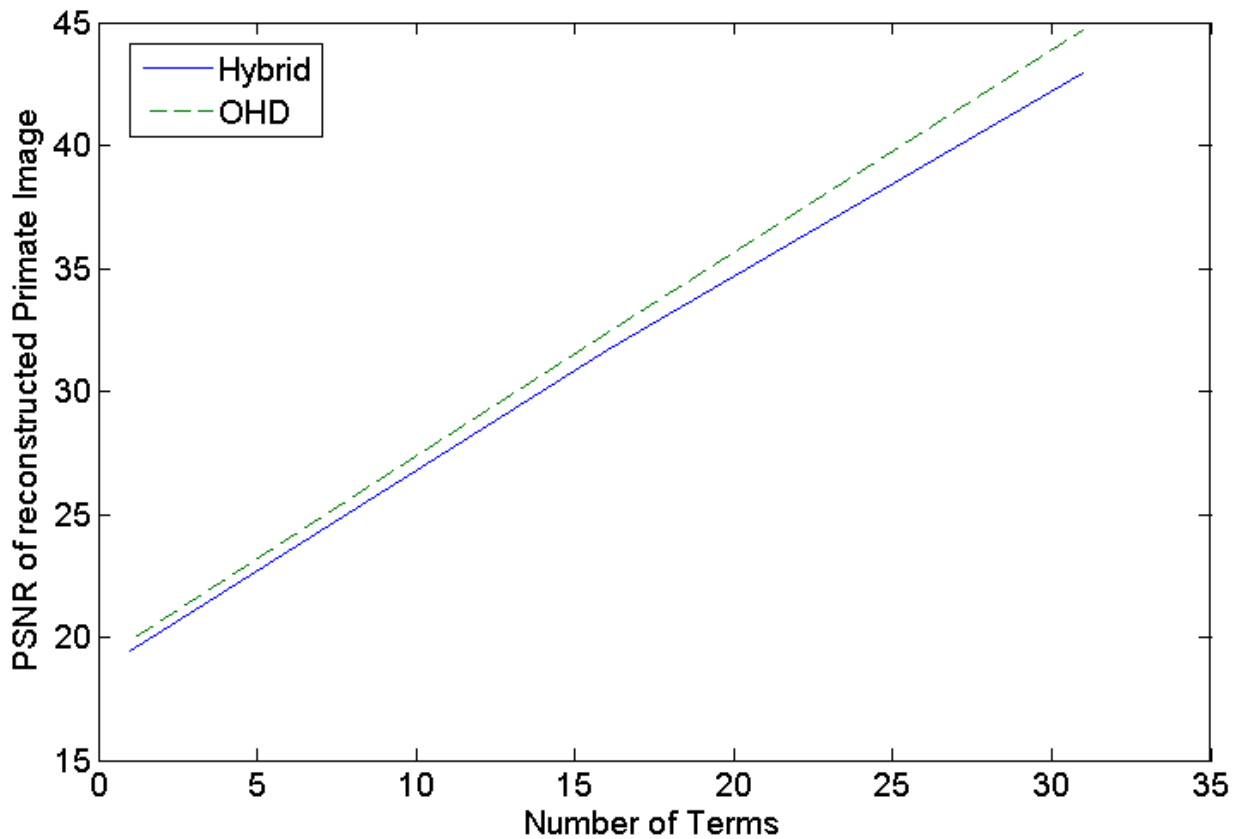


Figure 6-2 Primate image reconstruction using the OHD

The drawback of the optimized hybrid dictionary is the time consuming optimization process. However, there are many ways to improve the optimization process. One way to speed up the optimization process is to add the dictionaries to the selected dictionary list instead of starting with all of the dictionaries and removing inefficient dictionaries from the list. The bottom up build algorithm could use the same dictionary ratios to add the dictionaries one at a time imitating the OMP algorithm. With some improvements to the optimization process the OHD would be

an alternative to large over complete dictionaries and signal-based adaptive dictionaries.

6.4 Chapter Summary

This chapter proposed an algorithm to generate an optimized hybrid dictionary. To evaluate each dictionaries performance, the efficiency ratio and dictionary PSNR gain ratios were introduced. The efficiency ratio measured the redundancy of the dictionary. The dictionaries that had an efficiency ratio below the preset threshold were further evaluated to measure if the cost of including the dictionary was more than the gain provided by the dictionary. If the cost incurred was greater than the provided gain, the dictionary was removed from the hybrid dictionary list. The proposed optimization algorithm generated an optimized hybrid dictionary which was 3 times smaller than the arbitrarily selected hybrid dictionary and it was still able to provide equivalent reconstruction quality. We acknowledge the additional computation cost incurred in the optimization process and have suggested methods to improve the process.

CHAPTER 7

COMPRESSION USING TIME-SHIFTED OMP ALGORITHM

The DCT dictionary contains atoms that are phases shifted discrete cosine functions. Wavelet analysis generates additional atoms using dilation and translation of the elementary function prior to coding the signal. The hybrid dictionary does not have an equivalent time-shifting property. Thus we propose an algorithm to implement the time-shifting property for the hybrid dictionary but instead of incorporating the time-shift into the dictionary generation phase we propose to include the time-shift in the signal coding stage.

7.1 Time-shifted OMP signal coding algorithm

The time-shifted OMP algorithm is an ad-hoc method shown in Algorithm 7-1 that generates a list of time-shifted dictionaries by performing N -circular shifts to create N time-shifted variations of the dictionary before signal coding. The multi-dictionary OMP algorithm is used to code the signal using the dictionary list and the OMP algorithm will provide a sparse matrix for each time shifted dictionary that is used.

We can reconstruct the signal by reversing the steps. The first step is to create the time shifted version of the dictionary that is used in the encoded signal. Then use the corresponding sparse matrix to reconstruct the signal as a linear combination of the time shifted dictionaries. This method may improve the speed of the signal

reconstruction algorithm since only the time shifted dictionaries that were used need to be recreated and this can be done on demand. Also, with the proper data coding and parsing techniques, the time shifted index and sparse matrix can be encoded as efficiently as a single large dictionary. Intuitively this modification would be most beneficial for signal-based adaptive dictionaries as it reduces the amount of data that is required to recreate the dictionary while providing the time-shifted versions of the learnt atoms which have not been considered before. The time-shifted OMP signal coding algorithm will improve the signal reconstruction quality for existing predefined dictionaries such as the hybrid dictionary and the optimized hybrid dictionary since many of the atoms of the hybrid dictionary do not have an implementation for the time-shift property.

Algorithm 7-1 Ad-hoc pseudo algorithm for the Time-shifted OMP

1. Given Dictionary D
2. N = number of elements in dictionary atom
3. Initialize $D_masterList = \{\text{Null}\}$
4. for $j = 1:N$
 - a. $D_masterList = \{ D_masterList \cup \text{circular shift}(D, j) \}$
5. End
6. Multi-Dictionary OMP ($D_masterList, y$)

7.2 Evaluation of the Time-shifted OMP algorithm

We test the time-shifted OMP algorithm on the existing dictionaries such as the DCT, optimized hybrid dictionary as well as a simple image-based dictionary. We generate the simple image-based dictionary with 5 atoms. The first atom is found by taking the average of the signal matrix column, and the rest of the atoms are the boxcar functions with distinct widths. The image-based dictionary can be defined using 68 parameters, where the first 64 parameters are used to describe the image-based atom; one parameter is used to identify the number of elements per atom, another 2 parameters to identify the boxcar atoms and finally a parameter to indicate the type of atoms available. The DCT dictionary requires only 2 parameters and the optimized hybrid dictionary requires 11 parameters to be fully defined.

We will evaluate the signal coding algorithm by coding each dictionary using both the OMP algorithm and the time-shifted OMP algorithm. Shown in Figure 7-1 is the PSNR of the reconstructed Lena test image. The hybrid dictionary image reconstruction encoded using the time-shifted OMP algorithm achieves has a PSNR of 32.25 dB which is 5.9% better than the image encoded using the OMP algorithm. The time-shifted OMP algorithm also improves the signal reconstruction quality of the DCT dictionary, the PSNR for the image reconstructed using 4 non-zero coefficients is 30.18 dB which is 7.4% better than the image encoded using the DCT dictionary with the OMP algorithm. The image-based dictionary only had 5 atoms, and the OMP algorithm is limited to using a maximum of 5 non-zero coefficients per patch however the TsOMP algorithm is able to use more than 5 nonzero coefficients per patch improving the signal representation quality. The image-based dictionary

using the time-shifted OMP algorithm is able to reconstruct the Lena image with 11% better PSNR than the reconstructed image that is encoded with the OMP algorithm in conjunction with the image-based dictionary. The time-shifted OMP algorithm for the image-based adaptive dictionary had significant improvements as it is able to reconstruct the Lena image with a PSNR of 31 dB using only 4 non-zero coefficients per patch corresponding to a 68.14% compression. The OMP algorithm is unable to obtain the same quality of reconstruction on the simple image-based dictionary. The image encoded using the DCT dictionary with the OMP algorithm requires 9 non-zero coefficients per patch to reconstruct the image with a PSNR of 29.8 dB which would be approximately 59% of the original size. The image coding is also faster for the combination of the image-based dictionary with the time-shifted OMP algorithm since it only requires 27.12 seconds compared to 30.2 seconds to encode the image using the OMP algorithm and the DCT dictionary. However for larger dictionaries such as the DCT and the hybrid dictionary the computation cost of the time-shifted algorithm is very expensive which is shown in Figure 7-2 and may outweigh the benefits. The time-shifted OMP algorithm can be used for small dictionaries such as the image-based dictionary or in applications where the computational time is not a concern.

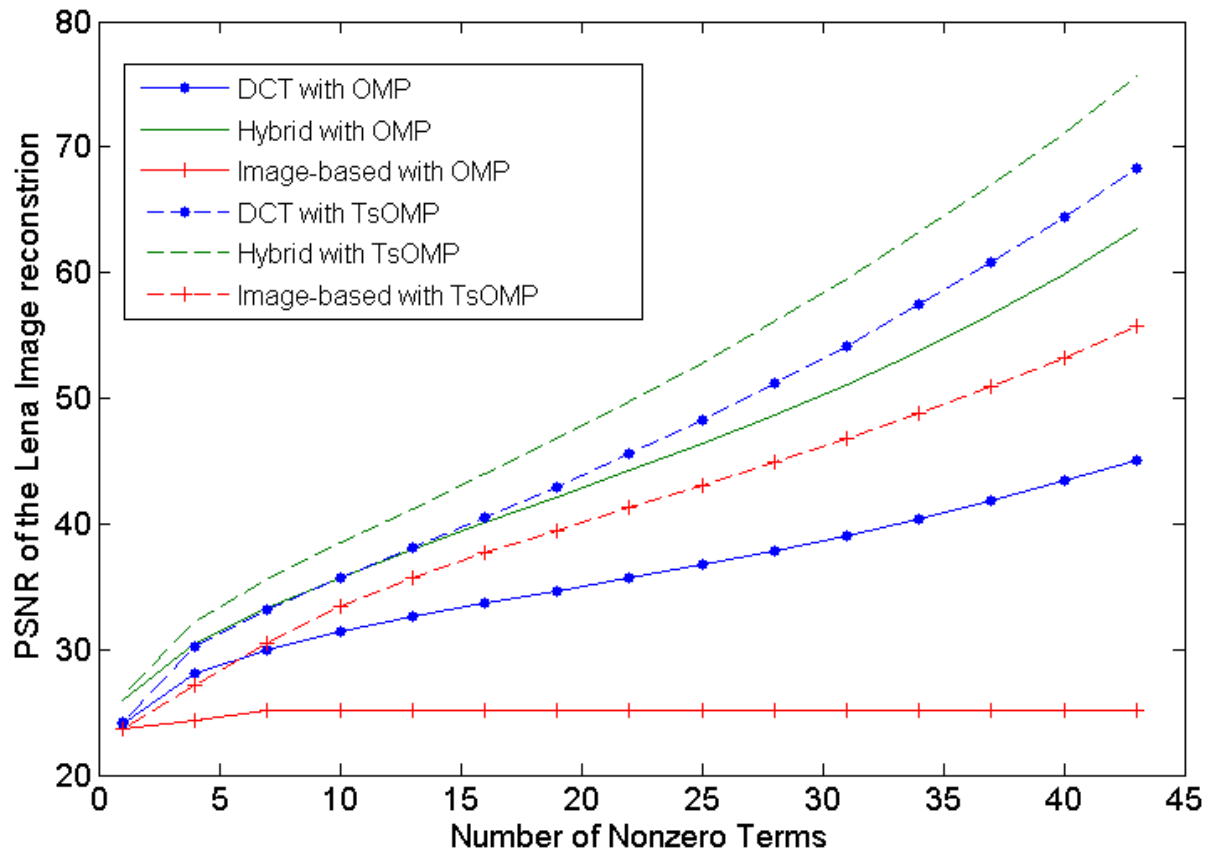


Figure 7-1 PSNR of reconstructed Lena image using Time-shifted OMP algorithm

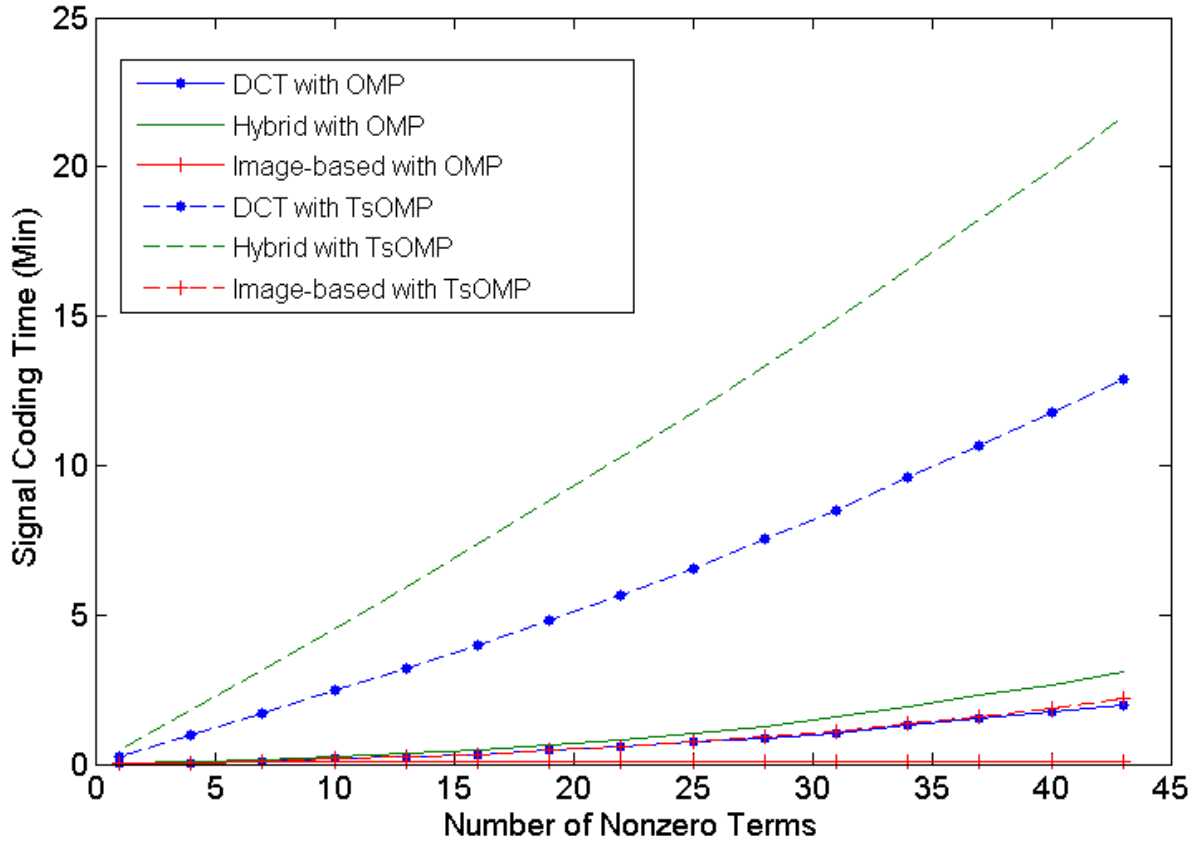


Figure 7-2 Lena image code time using the Time-shifted OMP algorithm

7.3 Chapter Summary

This chapter presented a signal coding algorithm to improve sparse signal coding by utilizing the time shifted variation of the dictionary. The time-shifted OMP algorithm generates $(N-1)$ additional dictionaries using $(N-1)$ -circular shift of the given dictionary. Then the OMP coding process selects the time-shifted dictionary and the corresponding sparse vector that best represent the signal. The time-shifted OMP algorithm improves the quality of the reconstructed signal by 5- 10% for the DCT,

and hybrid dictionary. The image-based dictionaries will benefit the most from utilizing the time-shifted OMP algorithm because the image-based dictionaries are in general much smaller than the predefined dictionaries and the signal coding time will not outweigh the improvements in the signal reconstruction quality provided by the time-shifted OMP algorithm. The TsOMP algorithm can also be used on existing dictionaries such as the DCT and the hybrid dictionary at the price of additional signal coding time.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

In compression technology, the large energy components need to be represented using a few atoms. The existing dictionaries such as the DCT dictionary, and wavelet based dictionaries are unable to provide the same signal reconstruction quality as signal-based dictionaries generated using sparse coding techniques. However the signal-based dictionaries have additional storage requirements and it has not been extensively studied on large problems. The ideas, methods and algorithms proposed in this thesis attempt to solve some of the limitations of the predefined dictionaries. The major contributions of the thesis are summarized below:

1. The primary contribution of the thesis is the hybrid dictionary, generated by including additional nonlinear functions such as polynomial, boxcar, exponentials, and rational functions. The inclusion of the additional nonlinear atoms allowed the dictionary to reconstruct not only images but also harmonic and non-harmonic test signals efficiently. The hybrid dictionary was also robust and can be completely described using a few parameters. The hybrid dictionary is larger than the DCT and Haar dictionaries that were used to compare the hybrid dictionary's performance. The signal coding time is proportional to the size of the

dictionary. The larger hybrid dictionary required more time to encode the signal with the same sparsity level. However, the inclusion of the additional nonlinear functions allowed the signal to be represented more sparsely, thus a signal with equal quality measured by the PSNR of the reconstructed signal can be encoded faster using the hybrid dictionary than the DCT dictionary. The hybrid dictionary could be used in many applications that emphasis the signal reconstruction quality more than the signal coding time. A few examples of applications that can benefit from the hybrid dictionary are listed below:

- a. Archival applications where large data set need to be stored.
- b. Medical imaging applications place high importance to the quality of the reconstructed image.
- c. Data transmission applications. The reduced file size will increase the transmission capacity and reduce transmission time over limited resources. The signal coding and decoding is easily offloaded to the nodes.
- d. Portable devices with limited storage capacity.

The hybrid dictionary is not well suited for real-time applications or applications running on limited resources. Some of the limitations of the hybrid dictionary are:

- a. The dictionary requires additional time to encode the signal and generate the dictionary.

- b. The dictionary lacks well-defined set of parameters to easily generate the functions set.
 - c. Implementation is difficult in resource limited applications such as the digital cameras which use ASICs to reduce cost.
- 2. The second idea presented in the thesis was the optimized predefined dictionary, where we proposed using the PSO algorithm to optimize the predefined dictionary generator to improve signal reconstruction quality. We were able to optimize the DCT, polynomial and Gabor dictionaries using a small sample from the image. The optimized DCT and polynomial dictionaries provided significant improvements compared to the dictionaries that were generated using arbitrary parameters. The optimization algorithm requires a training sample and additional time to optimize the dictionary. This algorithm cannot be used on resource-limited applications due to the added cost of implementing the optimization algorithm and the time required to optimize and code the signal. The additional cost of optimizing the dictionary is incurred once and there is no additional cost when encoding and decoding the signal using the optimized dictionary. Thus, the optimized dictionary is a good alternative to image-based adaptive dictionaries because it contains benefits of the predefined dictionary. The optimized dictionary can be used to encode medical images such as X-rays and MRI images because the initial training cost can be absorbed by the large volume of images that need to be coded.

3. The third contribution is the proposal of the optimized hybrid dictionary. The optimized hybrid dictionary is an extension of the optimization algorithm used to optimize the predefined dictionaries. We used a dictionary filtering algorithm to reduce the size of the hybrid dictionary. The optimized hybrid dictionary was able to provide comparable signal reconstruction quality with fewer atoms in the dictionary. The optimized hybrid dictionary had better signal reconstruction compared to the optimized DCT dictionary and slightly inferior reconstruction quality compared to the hybrid dictionary. Since the optimization process is a one-time cost for similar images, it can be applied successfully in place of the hybrid dictionary.
4. We also proposed a simple signal-based dictionary with additional predefined dictionary atoms. This idea was not fully developed in this thesis nor has it been extensively explored in other scholarly papers. This signal-based dictionary will contain a few signal-based atoms which will enable it to better represent the specific signal while the predefined dictionary atoms will represent signals that are not of the same type. This dictionary type makes a good compromise on the cost of generating the dictionary and the signal reconstruction quality.
5. The forth contribution was the introduction of the time-shifted OMP algorithm used to code the signal using a given dictionary. The time-shifted OMP signal coding algorithm performs a time shift on the provided dictionary and uses the additional atoms to encode the signal. The TsOMP algorithm is innovative as it proposes a time-shift for

image-based dictionaries which effectively adds additional atoms without the additional storage cost. The TsOMP algorithm execution time increases exponentially with the number of non-zero coefficients used in the signal representation vector and the size of the dictionary. The TsOMP algorithm can be used to encode the images or other large data in an automated batch mode. The batch mode allows the algorithm to run during off-peak time and minimize the usage of valuable resources. The time-shifted OMP algorithm is not applicable for real-time applications.

All compression techniques make a compromise between the compression ratio, signal reconstruction quality and the computation cost of the algorithm. We have suggested algorithms that improve compression by increasing the computation cost. Our proposed technique can be used in applications like medical image compression where the main concern is the quality of the image reconstruction and not the computational cost of the algorithm.

We have shown through image-reconstruction experiments that the proposed techniques are able to produce better quality reconstruction compared to the existing methods seen by visual inspection of the reconstructed image and the peak signal to noise ratio of reconstructed image. However we require further study to identify if there are any ways to optimize the proposed methods and algorithms prior to understanding the limitations of the proposed techniques.

8.2 Future work

Three dictionaries were presented in this thesis. The predefined hybrid dictionary and the optimized predefined hybrid dictionaries were extensively evaluated. The third signal-based dictionary with predefined atoms was introduced and used by the time-shifted OMP algorithm. The hybrid dictionaries were compared to the DCT dictionary. Since the DCT dictionary is successful in many image processing application, future work could evaluate the effectiveness of the hybrid dictionaries in other image processing applications such as facial recognition, feature extraction, super resolution image compression.

The signal-based dictionary with predefined atoms is an interesting idea worth dwelling into. The signal-based atoms can be used to aid the predefined dictionary atoms. This dictionary will have a few signal-based atoms which will reduce the cost of generating the dictionary and the dictionary will be robust. The few signal-based atoms will allow the dictionary to adapt to the signal being represented.

The optimized hybrid dictionary used the particle swarm optimization to optimize the parameters of the dictionary generator; future work can experiment with other optimization algorithms such as the genetic algorithm to improve the dictionary optimization process. In the algorithm presented in the thesis each dictionary was optimized without the knowledge of the other dictionaries. Also, we removed dictionaries sequentially based on their efficiency ratios. It might be faster if the optimization algorithm actually built upon the dictionaries by including dictionaries that is needed instead of removing dictionaries that are not needed. Also in the current method the dictionary that is currently being optimized may conflict with

existing dictionaries since the optimization algorithm does not know about the existence of the pre-selected dictionaries. Future work can adjust the fitness function and the optimization parameters of the dictionary generator to represent the residual of the signal as each dictionary is added. This way the newly added atoms will aid the existing dictionaries and remove potential conflicts. Future work can also improve signal coding time using the TsOMP and the multi-dictionary algorithms.

BIBLIOGRAPHY

- [1] Munenori Oizumi, "Preprocessing method for DCT-based image-compression," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 3, pp. 1021-1026, August 2006.
- [2] XiHong Zhou, "Research on DCT -based Image Compression Quality," in *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference*, 2011, pp. 1490-1494.
- [3] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transforms," in *IEEE transactions on signal processing*, vol. 40, 1992, pp. 2174-2193.
- [4] K.R. Rao, D.N. Kim, and J.J Hwang, *Fast Fourier Transform - Algorithms and Applications*, 1st ed. Newyork: Springer, 2001.
- [5] 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.
- [6] Micheal Elad and Michal Aharon, "Image Denoising via sparse and redundant representation over learned dictionary," in *IEEE Transaction on Image Processing*, 2006, pp. 3736-3745.
- [7] Ivana Tosoic and Pascal Frossard, "Dictionary Learning," *IEEE Signal Processing Magazine*, pp. 27-38, March 2011.
- [8] N. Ahmed, T. Natarajan, and K.R. Rao, "Discrete Cosine Transform," *IEEE Transactions on Computer*, pp. 90-93, January 1974.

- [9] J. Morlet and A. Grossman, "Decomposition of Hardy functions into square integrable wavelets of constant shape," *Society for Industrial and Applied Mathematics*, vol. 15, no. 4, pp. 723-736, July 1984.
- [10] I. Daubechies, "The wavelet transform time-freueency localization and signal analysis," *IEEE Transaction on Information theory*, vol. 36, no. 5, pp. 961-1005, 1990.
- [11] I. Daubechies, A. Grossmann, and Y. Meyer, "Painless nonorthogonal expansions," *American Institute of Physics*, vol. 27, no. 5, pp. 1271-1283, May 1986.
- [12] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transaction on pattern analysis and machine intelligence*, vol. 11, no. 7, pp. 674-693, July 1989.
- [13] Ingrid Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 910-996, 1988.
- [14] Yang Qiang, "Image Denoising Based on Haar Wavelet Transform," in *International Conference on Electronics and Optoelectronics (ICEOE)*, 2011, pp. V3 129-132.
- [15] Kamrul Hasan Talukder and Koichi Harada, "Haar Wavelet Based Approach for Image Compression and Quality Assessment of Compressed Image," *International Journal of Applied Mathematics - Online Publication*, pp. 1-8, February 2007.
- [16] Wing-Pong Choi, Siu-Hong Tse, Kwok-Wai Wong, and Kin-Man Lam, "Simplified Gabor wavelets for human face recognition," *Pattern Recognition Society*, no. 41, pp. 1180-1198, July 2007.

- [17] Stephane G Mallat and Zhifeng Zhang, "Matching Pursuit with Time-Frequency Dictionaries," in *IEEE Transactions on signal Processing*, 1993, pp. 3397-3415.
- [18] S.S Chen, D.L Donoho, and M.A Saunders, "Atomic Decomposition by Basis Pursuit," *Journal on Scientific Computing*, vol. 20, no. 1, pp. 33-61, May 1998.
- [19] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6853, pp. 607-609, June 1996.
- [20] K. Engan, S.O. Aase, and J Hakon Husoy, "Method of optimal directions for frame design," *IEEE International Conference on Aoustics, Speech, and Signal Processing*, vol. 5, pp. 2443-2446, 1999.
- [21] Michal Aharon, Michael Elad, and Alfred Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Transactions on Signal Processing*, vol. 54, pp. 4311-4322, November 2006.
- [22] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online Dictionary Learning for Sparse Coding," Institute For Mathematics and its Applications, Minneapolis, Technical Document 2009.
- [23] Boris Mailhé and Mark D. Plumbley, "Dictionary Learning with Large Step Gradient Descent for Sparse Representations," School of Electronic Engineering and Computer Science, London, Technical Paper 2012.
- [24] F. Bergeaud and S., Mallat, "Matching Pursuit of Images," in *International Conference on Image Processing*, 1995, pp. 53-56.

- [25] Y.C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *1993 Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40-44.
- [26] D.L. Donoho, Y. Tsaig, I. Drori, and J-L Starck, "Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit," in *IEEE Transactions on Information Theory*, 212, pp. 1094-1121.
- [27] James Kennedy and Russel Eberhart, "Particle Swarm Optimization," in *Proceedings of IEEE international Conference on Neural Networks*, 1995, pp. 1942-1948.
- [28] M.R. Dastranj, M. Moghaddas, S.S. Afghu, and M. Rouhani, "PID control of inverted pendulum using particle swarm optimization (PSO) algorithm," in *IEEE 3rd International Conference on Communication Software and Networks*, 2011, pp. 575-578.
- [29] Abolfazl Jalivand, Ali Kimiyaghlam, Ahmad Ashouri, and Mahdavi Meisam, "Advanced Particle Swarm Optimization-based PID controller parameters tuning," in *Proceedings of the 12th IEEE International Multitopic Conference*, 2008, pp. 429-435.
- [30] Xu-zhou Li, Fei Yu, and You-bo Wang, "PSO Algorithm based online self-tuning of PID Controller," in *International Conference on Computational Intelligence and Security*, 2007, pp. 128-132.
- [31] M. Tharmalingam and K. Raahemifar, "Strategic initialization of a hybrid particle swarm optimization-simulated annealing algorithm (HPSOSA) for PID

- controller design for a nonlinear system," in *25th IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, Montreal, 2012, pp. 1-4.
- [32] Riccardo Poli, James Kennedy, and Tim Blackwell, "Particle swarm optimization: An overview," *Swarm Intell*, pp. 33-57, 2007, Particle Swarm Optimization an overview.
- [33] Dongyun Wang and Lipin Liu, "Improved Particle Swarm Optimization and its Application," in *The 3rd International Conference on Innovative Computing Information*, 2008, pp. 1-5.
- [34] Seokbeop Kwon, Byonghyo Shim, and Jian Wang, "Generalized Orthogonal Matching Pursuit," School of Information and Communication, Seoul, Korea, PhD Thesis 2011.
- [35] Honglin Wu and Shu Wang, "Adaptive Sparsity Matching Pursuit Algorithm for Sparse Reconstruction," *IEEE Signal Processing Letters*, vol. 19, no. 8, pp. 471-474, August 2012.