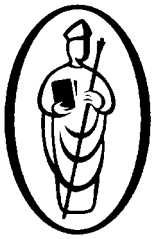
Frame Based Signal Representation and Compression

by

Kjersti Engan

Submitted in partial fulfillment
of the requirements for the degree of
DOKTOR INGENIØR



Departement of Electrical and Computer Engineering
Stavanger University College
Norway

2000

Abstract

The demand for efficient communication and data storage is continuously increasing and *signal representation* and *compression* are important factors in digital communication and storage systems.

This work deals with *Frame based* signal representation and compression. The emphasis is on the design of frames suited for efficient representation, or for low bit rate compression, of classes of signals.

Traditional signal decompositions such as transforms, wavelets, and filter banks, generate expansions using an analysis-synthesis setting. In this thesis we concentrate on the synthesis or *reconstruction* part of the signal expansion, having a system with no *explicit* analysis stage. We want to investigate the use of an *overcomplete* set of vectors, a *frame* or an overcomplete dictionary, for signal representations and allow *sparse* representations. Effective signal representations are desirable in many applications, where signal compression is one example. Others can be signal analysis for different purposes, reconstruction of signals from a limited observation set, feature extraction in pattern recognition and so forth.

The lack of an explicit analysis stage originates some questions on finding the optimal representation. Finding an optimal sparse representation from an overcomplete set of vectors is NP-complete, and suboptimal vector selection methods are more practical. We have used some existing methods like different variations of the Matching Pursuit (MP) [52] algorithm, and we developed a robust regularized FOCUSS to be able to use FOCUSS (FOCal Underdetermined System Solver [29]) under lossy conditions.

In this work we develop techniques for frame design, the Method of Optimal Directions (MOD), and propose methods by which such frames can successfully be used in frame based signal representation and in compression schemes. A Multi Frame Compression (MFC) scheme is presented and experiments with several signal classes show that the MFC scheme works well at low bit rates using MOD designed frames. Reconstruction experiments provides complimentary evidence of the good properties of the MOD algorithm.

Preface

This dissertation is submitted in partial fulfillment of the requirements for the doctoral degree of *doktor ingeniør* at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway. Professor John Håkon Husøy and Assistant Professor Sven Ole Aase at Stavanger University College (Høgskolen i Stavanger, HiS), Stavanger, Norway have been my supervisors.

The work, including the compulsory courses corresponding to full time studies in two semesters, as well as four months of lecturing, has taken place in the period of August 1996 to June 2000. Except for seven months from September 1998 to April 1999, spent at the Department of Electrical and Computer Engineering at the University of California San Diego (UCSD), USA, the research was carried out at the Electrical and Computer Engineering Department of HiS.

The work was funded by scholarships from the Norwegian Research Council (NFR).

Acknowledgments

First of all I would like to thank my supervisor, Professor John Håkon Husøy, for his inspiration and invaluable support, and my second supervisor, Assistant Professor Sven Ole Aase, for his enthusiasm and for working with me while developing the algorithm, MOD, which plays an important role in this thesis. I am very grateful to both of them for our fruitful 'brainstorming' meetings, which always brought up new ideas and inspiration.

I am grateful to Professor Bhaskar D. Rao for making my stay at the Department of Electrical and Computer Engineering at University of California San Diego (UCSD) possible, and along with Professor Kenneth Kreutz-Delgado for making my stay at UCSD most interesting. The work of Chapter 3 was done together with Bhaskar Rao and the Section 4.2 is a result of Kenneth Kreutz-Delgado's insight and our interesting discussions. The experimental section of Chapter 8 was also executed during my stay at UCSD.

My colleagues at Høgskolen i Stavanger have all contributed to a highly appreciated job atmosphere. Particularly I would like to thank Karl Skretting for providing me with some software for my image compression experiments and for helpful discussions. A special gratitude goes to Ranveig Nygaard with whom I've had many valuable discussions about the thesis work and about being a doctoral student in general. She has been a moral support an inspiration, and she is a very dear friend.

Thanks to my parents, Eli and Helge, and my sisters, Bente and Marianne, for always believing in me and supporting me. Finally, thanks to Kjell for encouragement and support.

Contents

Nomenclature

$\|\cdot\|$	l_2-norm
$\|\cdot\|_F$	Frobenius-norm
$\langle\cdot\rangle$	expectation of sthocastic variables
$\langle\cdot\rangle_M$	estimated expectation, i.e. mean, of a variable over a data set, M
$\langle a, b\rangle$	the inner product of a and b
$\mathbf{A}^+$	pseudoinvers
$\mathbf{B}$	basis
$\mathbf{b}_j$	basis vector
β	generalized variance in generalized Gaussian distribution
$\mathbf{C}$	vector quantizer codebook
$\mathbf{c}_i$	vector quantizer codebook vector no. i
$\mathbf{C}_g$	shape-gain vector quantizer *gain* codebook
$\mathbf{C_s}$	shape-gain vector quantizer *shape* codebook
$\mathcal{C}(\cdot)$	Cardinality of a set
D	distortion
$d(\mathbf{x}, \mathbf{y})$	distortion between $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{\Delta}$	adjustment matrix
Δ	quantizer step
$\boldsymbol{\delta}_j$	adjustment vector
$E^{(p)}(\cdot)$	$l_{p\leq 1}$ diversity measure
ϵ	error limit
$\mathbf{F}$	frame (synthesis)
$\mathbf{f}_j$	frame vector no. j
$\mathbf{F}_i$	frame no. i in a MFC scheme. Design for sparsity factor i
$\mathbf{F}_i^j$	frame no. i in a MFC scheme with variable sized frames. Design for sparsity factor i. Size $N \times jN$
$\Gamma(\cdot)$	standard gamma function
H	1) entropy 2) Hilbert space
K	number of frame vectors in a frame
$K(\cdot)$	curvature

L	no. of frames in MFC
λ	regularization parameter
M	1) number of signal vectors in a set 2) size of VQ codebook
m	no. of nonzero entries in $\breve{\mathbf{w}}$ (synthetic data)
N	vector size
$\mathbf{n}$	noise vector
p	parameter in diversity measures, FOCUSS and generalized Gaussian distribution (the 'same' parameter). p is used due to the diversity measures close connection to l_p norms.
$p(a)$	probability density function (pdf) of the random variable a
$p_a(\cdot)$	pdf of the random variable a, known pdf or with a different argument
$p(a,b)$	joint pdf of the random variables a and b
$p(a\|b)$	pdf of the random variable a, when the random variable b is given
$p(a;c)$	pdf of the random variable a, when the deterministic parameter c is known
Q	number of different quantizer step
$\mathbf{R}^N$	N dimensional space of real numbers
$\tilde{\mathbf{R}}_{ww}$	estimated auto-correlation matrix of $\mathbf{w}$
$\tilde{\mathbf{R}}_{rw}$	estimated cross-correlation matrix of $\mathbf{r}$ and $\mathbf{w}$, respectively
R	bit rate
$\mathcal{R}_i$	vector quantizer partition no. i
$\mathcal{R}_{i,j}$	shape-gain vector quantizer partition no. i,j
$\mathbf{r}$	residual vector
$\mathbf{r}_l$	residual vector no. l, corresponding to signal vector no. l, $\mathbf{x}_l$, from a set.
r	no. of nonzero entries in $\mathbf{w}$
$\bar{r}$	average no. of nonzero entries in a set, $\mathbf{w}_l$
σ^2	variance
T	threshold
$\mathcal{T}$	training set
$\mathbf{T}$	transform matrix (synthesis)
$\mathbf{t}_j$	transform vector (basis vector) no. j
$\mathbf{W}^M$	Coefficient vector set
$\mathbf{w}$	frame coefficient vector
$\mathbf{w}_l$	frame coefficient vector no. l, corresponding to signal vector no. l, $\mathbf{x}_l$, from a set.
$\mathbf{w}(n)$	frame coefficient vector at point n in time
$\mathbf{w}^{(k)}$	frame coefficient vector at iteration no. k
$\breve{\mathbf{w}}$	constructed coefficient vector used when making synthetic data

w_j	coefficient value corresponding to frame vector no. j, $\mathbf{f}_j$
$w_l(j)$	coefficient value corresponding to frame vector no. j, $\mathbf{f}_j$, for signal vector no. l, $\mathbf{x}_l$, from a set.
$w_j^{(k)}$	coefficient value corresponding to frame vector no. j, $\mathbf{f}_j$, at iteration no. k
X'	derivative of X
X''	second derivative of X
$\mathbf{X}^M$	signal vector set (observed data)
$\mathbf{X}$	2 dimensional signal block
$\mathbf{x}$	signal vector
$\mathbf{x}(n)$	signal vector at point n in time
$\hat{\mathbf{x}}$	approximated signal vector
$\mathbf{x}_l$	signal vector no. l (from a set)
$\check{\mathbf{x}}$	noise free, synthetic data vector
$\mathbf{Y}$	2 dimensional transform coefficient block
$\mathbf{y}$	transform coefficient vector
$\hat{\mathbf{y}}$	quantized transform coefficient vector
y_j	transform coefficient value corresponding to basis vector no. j, $\mathbf{t}_j$
$\hat{y}_j$	quantized transform coefficient value

List of abbreviations

AML	approximate maximum likelihood
BOB	beginning of block
DC	direct current
DCT	discrete cosine transform
DST	discrete slant transform
DWT	discrete Walsh transform
ECG	electrocardiogram
EOB	end of block
FOCUSS	focal underdetermined system solver
FOMP	fast orthogonal matching pursuit
GLA	generalized Lloyd algorithm
ICA	independent component analysis
iid	independent and identically distributed
JPEG	joint photographic experts group
KLT	Karhunen-Loève transform
LOT	lapped orthogonal transform
MAP	maximum a posteriori
MFC	multi frame compression
MITxxx	signal no. xxx from the MIT arrhytmia database of ECG signals
ML	maximum likelihood
MOD	method of optimal directions
MP	matching pursuit
MSE	mean squared error
MSVQ	multistage vector quantizer
ND	normalized distribution
OMP	orthogonal matching pursuit
ORMP	order recursive matching pursuit

PCA	principal component analysis
pdf	probability density function
PSNR	peak signal to noise ratio
RMSE	root mean squared error
SNR	signal to noise ratio
TLS	total least squares
VQ	vector quantizer

Chapter 1

Introduction

The demand for efficient communication and data storage is continuously increasing. One example is the enormous growth in Internet communication, where image and video signals play an important role. *Signal representation* and *compression* are important factors in digital communication and storage systems, and are the subjects studied in this thesis.

A *signal* can be a real world continuous signal that is sampled at some sample rate, like an ElectroCardioGram (ECG) signal, which is a monitoring of the electrical pulses the body makes during heart beats. Other examples are speech or audio signals, i.e. acoustic waves translated into electrical signals which can be translated back to sound through a speaker. For real world signals to become digital signals, two kinds of discretization is done. The signals are sampled at some sample rate so that we get a set of amplitude values representing the signal. The amplitude values also need discretization since a computer works with numbers of finite precision, thus the values are quantized so they can be represented by a finite number of bits. Another class of signals we use in this thesis are digital images. A natural image is represented by a finite number of pixels, e.g. 512×512, where a finite number of bits are used to represent the gray level image value at each pixel. Commonly, each pixel is represented as an 8 bit pattern in a gray tone image.

By *signal compression* we mean a bit-efficient representation of a signal. There are two distinct classes of compression methods: *lossless compression* and *lossy compression*. By lossless compression it is understood that the compressed signal is represented more efficiently than the original signal and that it can be reconstructed to *exactly* the same as the original signal. Lossy compression, on the other hand, gives a bit-efficient representation of an *approximation of the original signal*, and consequently has greater compression potential.

Both classes of compression methods are widely used, and often a compression scheme includes a combination, as is the case in this work.

For storing large amounts of data or transmitting data over a limited bandwidth channel signal compression will be highly beneficial. Since all physical storage media and bandwidths are limited, compression is widely used. Storage capacity and bandwidths are increasing with improved technology, but so is the demand for the amount of data to be stored or transmitted, and effective representation or compression will probably always be an issue.

Effective *signal representations*, alternative parameterizations of a signal, are desirable in many applications, where signal compression is one example. Others can be signal analysis for different purposes, reconstruction of signals from a limited observation set, feature extraction in pattern recognition and so forth.

The outline of this chapter is as follows: In the next section we define some terms frequently used in this thesis. The following section briefly describes a couple of signal representation and compression methods with relevance for our work, and motivates the use of frames for signal representation and compression. This is followed by a section briefly explaining how we regard statistical signal processing, and a section about previous work in the area of frame based representation and compression and frame design. The chapter is concluded by a section explaining the scope and contribution of this work.

1.1 Definitions

The title of this thesis is *Frame based signal representation and compression.* For now, frame vectors we use can be regarded as column vectors, each normalized to one, from an $N \times K$ matrix with $N \leq K$. The whole collection is called a *frame* or sometimes also a *dictionary.* The strict mathematical definition of frames along with a thorough explanation of how frames are used for representation purposes and frame based compression are presented in the next chapter.

Some terms, frequently used in this thesis, are defined below. Let $\mathbf{x}$ be a signal column vector of size N, each element being a signal sample. The signal vector is approximated:

$$\mathbf{x} \simeq \hat{\mathbf{x}} = \sum_j w_j \mathbf{f}_j, \tag{1.1}$$

where $\{\mathbf{f}_j\}$,$j = 1, \ldots, K$ are vectors constituting a frame, and the w_j's are coefficients. We have the following definitions:

- *Approximation*: $\hat{\mathbf{x}}$ is an approximation to $\mathbf{x}$ IF $\|\mathbf{x} - \hat{\mathbf{x}}\| < T$, where T is a threshold. We call the approximation error the residual: $\mathbf{r} = \hat{\mathbf{x}} - \mathbf{x}$.

- *Representation*: By signal representation we mean an alternative description of the signal vector which can be used to reconstruct the signal vector or an approximation to the signal vector. In Equation 1.1, if the frame is known the set of coefficients, $\{w_j\}$, $j = 1, \ldots, K$ is a signal representation. Note that: Exact as well as approximate representations of the form of Equation 1.1 are possible. Other times representation refers to a collection of bits that can be used for the reconstruction.

- *Compression*: After a signal vector is approximated, or described by a signal representation, the signal representation is typically coded by some coding technique. A compressed signal is the coded version of the signal representation. This is the collection of bits that can be used to reconstruct the signal or the approximated signal.

- *Sparsity*: Let the coefficient set $\{w_j\}$, $j = 1, \ldots, K$ constitute the K-dimensional vector $\mathbf{w}$, and let just a few of the K coefficients be different from zero. $\mathbf{w}$ is then said to be *sparse*. By sparsity we mean degree of sparseness.

1.2 Signal representation and compression methods

There exists many different methods for signal representation and compression, both lossless and lossy. This section briefly describes a couple of techniques with relevance for our work, and also motivates our work with frame based signal representation and compression.

1.2.1 Entropy coding

The goal of lossless coding is to reduce the average number of symbols sent while suffering no loss of fidelity. A classical example is the Morse code where short binary codewords are used for more probable letters and long codewords used for less probable letters. One such lossless coding scheme is *entropy coding*. The average amount of "information" per source symbol of a zero-memory source is called the *entropy* of the source, where "information" has a mathematical definition [2]. Entropy provides a lower bound to the average length of lossless codes, and good codes can perform close to this bound.

Therefore uniquely decodable variable length lossless codes are called *entropy codes* [26, 3].

The essence in entropy coding is utilizing a nonuniform probability density function (pdf) of the different symbols, and minimizing the average number of bits transmitted for each source symbol. The code words have different length, thus it is also called a variable length coder.

Entropy is a measure of the expected information in the outcome of a source, thus it is a measure of the variability in the probability of different source symbols. If the probability of the different source symbols are very different, the entropy is low and the possible average bit rate is low. If the symbols have equal probability, on the other hand, the same number of bits are used for each symbol, and nothing can be gained using entropy coding.

1.2.2 Transform coding

A transform coder decomposes a signal using an orthogonal basis and quantizes the decomposition coefficients [50, 26].

For an N dimensional signal vector $\mathbf{x}$, and a unitary transform matrix $\mathbf{T}$ of dimension $N \times N$ we have the analysis and synthesis equations[1]:

$$\begin{aligned} \mathbf{y} &= \mathbf{T}^T\mathbf{x} \\ \mathbf{x} &= \mathbf{T}\mathbf{y} = \sum_{j=1}^{N} y_j \mathbf{t}_j \simeq \sum_{j=1}^{N} \hat{y}_j \mathbf{t}_j = \mathbf{T}\hat{\mathbf{y}} = \hat{\mathbf{x}} \end{aligned}$$

where $y_j, j = 1, 2 \ldots N$ are the transform coefficients, and $\hat{y}_j, j = 1, 2 \ldots N$ are the quantized coefficients.

Transform based compression is a lossy compression technique, and the signal distortion is minimized by optimizing the quantization procedure, the basis (i.e. the transform matrix), and the bit allocation. The optimal basis for a signal depends on the statistics of the stochastic process that produced the signal. For high resolution quantization, the distortion-rate relationship $D(\bar{R})$ is optimized by using a basis which minimizes the average differential entropy [51]:

$$\bar{H} = \frac{1}{N} \sum_{j=1}^{N} H(y_j), \tag{1.2}$$

[1] For notational convenience we denote the *forward* matrix by $\mathbf{T}^T$ and the *inverse* or *reconstruction* matrix by $\mathbf{T}$.

where $y_j, j = 1, 2 \dots N$ are the transform coefficients, and $H(y_j)$ is the differential entropy associated with each coefficient. If the process is Gaussian then the coefficients y_j are Gaussian using any basis. Equation 1.2 is then minimized if the transform is given by the eigenvectors of the autocorrelation matrix of the stochastic process and it is called the Karhunen-Loève Transform (KLT) [26]. If the process is not Gaussian, or the high resolution assumption does not hold, the KLT need not be the optimal transform. It is then a nontrivial task to find the optimal transform even if the statistics are known [51]. In addition to these difficulties the signal is often non-stationary, and consequently no fixed transform will be optimal in all signal regions.

Potential for improvements

The limitations in the optimality of the KLT, the difficulties in finding optimal transforms, and the fact that for a non-stationary signal no fixed transform will be optimal are all factors that motivates the use of frames, or overcomplete dictionaries, for representation purposes. One reason for desiring an overcomplete dictionary is that it possesses greater robustness in the face of noise and other forms of degradation [57]. The reason more pertinent to our purposes is that an overcomplete dictionary will allow greater flexibility in matching the input signal with a sparse linear combination of frame vectors.

An orthogonal basis consist of N basis vectors of size N. A sparse coefficient vector is wanted for effective representation. Having more than N vectors to choose from when forming the sparse representation improves the flexibility. More vectors to choose from increases the probability of finding a small number of vectors whose linear combination match the signal vector well. Such a set of vectors is overcomplete, and it is no longer a basis but a *frame* [32]. The reconstruction can be written:

$$\mathbf{x} \simeq \hat{\mathbf{x}} = \mathbf{F}\mathbf{w} = \sum_{j=1}^{K} w_j \mathbf{f}_j, \tag{1.3}$$

where w_j and $\mathbf{f}_j$, $j = 1, \dots K$ are the coefficients and frame vectors respectively. Since a linearly dependent set of vectors is used, an expansion is no longer unique, and a unique "analysis frame" as in analysis transform does not exist. A good first step in a compression scheme is to use as few vectors as possible to obtain a good approximation for each signal vector, and subsequently quantizing the corresponding coefficients. Consequently it makes sense to apply a sparsity constraint when finding the coefficients of Equation 1.3. Finding the optimal frame vectors to use in such an approximation

is an NP-hard problem and requires extensive calculation [55]. Consequently suboptimal vector selection techniques are used.

In frame compression the bit budget depends upon the number of vectors used in the approximation; but also on the size of the frame. If the frame is large, more bits have to be used to identify which vectors are used in the approximation. On the other hand there are probably fewer vectors used than with a small frame implying fewer bits spent on the quantized representation of the coefficients. Consequently there is a trade-off between frame size and position information, and the number of vectors used in the approximation. These topics will be explored at length later in this thesis.

1.3 Statistical signal processing

In signal processing one can choose to deal with the known data as it is without considering the signals as statistical, or one can choose to use a statistical way of thinking. Using statistical signal processing, signals or variables can be regarded as stochastic or deterministic. A deterministic variable is a fixed parameter, known or unknown. A stochastic variable, or a random variable, can be modeled using a probability function. Data can be regarded as a realization of a stochastic or random process. However, a matter of discussion is whether a signal should *truly* be considered to be a realization of a random process or simply to be observed data that is treated using statistical methods.

In parts of this work we use a non-probabilistic way of thinking, and our algorithm for frame-vector design, entitled Method of Optimal Directions (MOD), is derived in this manner in Chapter 4.1. In other parts, especially Chapter 4.2, a probabilistic point of view is used. It is interesting to note that even though the viewpoints and technicalities in these sections are different, both viewpoints lead to the same design algorithm under a given set of assumptions.

1.4 Previous work

Goyal and Vetterli [30, 31, 32] have worked with frames or overcomplete expansions. They have done several experiments using different frames. The frames they have used are chosen rather than optimized. For example they propose the use of vectors on the N-dimensional spheres that maximize the minimum Euclidean norm between the vectors, or corners of the hypercube. Goodwin [28] use frames derived from a collection of damped sinusoids. The use of damped sinusoids for signal decompositions is motivated by the commonality

of damped oscillations in natural signals and the shortcoming of symmetric atoms for representing transient signal behavior. Berg and Mikhael use a frame that contains both the DCT (Discrete Cosine Transform) and the Haar transform vectors for compression of speech signals [6, 7] and images [54, 8]. DeBrunner et. al. construct lapped frames by combining several Lapped Orthogonal Transforms (LOT), like LOTDCT, LOT Discrete Walsh Transform (DWT), and LOT Discrete Slant Transform (DST), and these are successfully used in image representation simulations [13, 14]. The use of frames in compression schemes has been given some attention [31, 32, 28, 54, 15, 14, 56] whereas the problem of *frame design* in this context is largely unexplored. Some work in that area is done by Olshausen and Field [57], Lewicky and Sejnowski [49], and Lee et al. [47].

1.5 The scope and contributions of this thesis

Traditional signal decompositions such as transforms, wavelets, and filter banks, generate signal expansions in an analysis-synthesis setting. In this thesis we concentrate on the synthesis or *reconstruction* part of the signal expansion. We want to investigate the use of an *overcomplete* set of vectors, a *frame*, or an overcomplete dictionary, for signal representations with the objective of allowing *sparse* representations.

The focus of this thesis is on *sparse signal representation and compression* using a *frame based* approach. We start by defining frame based approximation and comparing it to the principles behind common compression schemes like transform coding, Vector Quantization (VQ), and filter banks. Approximation capabilities and compression results will be presented and compared with reference schemes.

We have tried to free ourselves from the analysis-synthesis paradigm by concentrating on the synthesis. This calls for a solution to the problem of finding the optimal coefficients, since they can no longer be obtained from an associated "analysis frame". We use existing methods like different variations of the Matching Pursuit (MP) [52] algorithm. We also want to use FOCUSS (FOCal Underdetermined System Solver [29]), but this algorithm gives an *exact* sparse representation. A robust regularized FOCUSS is developed to enable to use FOCUSS for an approximation, i.e. lossy representation, or for situations where the data is polluted by noise.

In this work we develop techniques for frame design and propose methods by which such frames can successfully be used in frame based signal representation and compression schemes.

Briefly summarized, the major contributions of this thesis are:

- Using training data we develop a method for frame design called Method of Optimal Directions (MOD).
- Frame design is regarded from a probabilistic point of view. This gives additional insight into, and an alternative justification of the MOD. Other possibilities in frame design are indicated but not fully investigated.
- Approximation capabilities for frames designed using MOD are investigated for ECG signals, speech signals and digital images.
- Compression experiments using *one* frame in the compression scheme is presented and compared to reference compression schemes for ECG signals and digital images.
- A Multi Frame Compression (MFC) scheme is developed to increase the representation flexibility and compression capability.
- The concept of using variable sized frames in the MFC scheme is introduced and a rationale for using variable sized frames is presented.
- Compression experiments on ECG signals using the MFC scheme with both fixed size and variable size frames are presented and shown to perform very well (1-4 dB better) compared to a reference compression scheme.
- Compression experiments on images using the MFC scheme with fixed size frames are presented and shown to perform up to 1 dB better than JPEG (Joint Photographic Experts Group) for very low bit rates (bit per pixel).
- Different methods for deciding the parameter in regularized FOCUSS is investigated, and a modified L-curve[2] approach is developed as a robust method for finding the regularization parameter.
- Some possible applications for frames and sparse representation, other than compression, are introduced, e.g. signal reconstruction and blind source separation. Experiments are done using a data set, MOD, and regularized FOCUSS to reconstruct the frame producing the data set. The reconstruction capability is shown to be very good.

[2]The L-curve was introduced by Hansen in [34] as a method for finding the parameter in a regularization problem.

The regularized FOCUSS is explained in Chapter 3. The last part of Chapter 2 discusses the vector selection algorithms we use in this work, and was followed naturally by the description of the robust regularized FOCUSS algorithm. The reader, however, may skip Chapter 3 without any loss in understanding of the frame design algorithm, the MOD, and the MFC scheme. The robust regularized FOCUSS is used as the vector selection algorithm in an experiment in Chapter 5 and in the reconstruction experiments of Chapter 8.

Chapter 2

Frames and overcomplete dictionaries

This chapter introduces the concept of frames, also called overcomplete dictionaries. It describes a frame and how it can be used for representation and compression purposes. Compression using frame techniques is compared to existing compression techniques like transform based compression, filter banks, wavelets, and different kinds of VQ techniques.

2.1 Bases and frames

If an N-dimensional vector space V contains a linearly independent set $\mathbf{B} = \{\mathbf{b}_i\}$ of N vectors, then $\mathbf{B}$ is called a *basis* for V, and it spans the space. Any vector, $\mathbf{v}$, in the set V can be expanded as a linear combination of the basis vectors:

$$\mathbf{v} = \sum_{j=1}^{N} \alpha_j \mathbf{b}_j, \tag{2.1}$$

where α_j is the coefficient corresponding to the vector $\mathbf{b}_j$. The expansion is unique because of the linear independence. If the set of vectors is orthogonal, that is $\mathbf{b}_i \perp \mathbf{b}_j$ when $i \neq j$, then $\mathbf{B}$ is called an *orthogonal basis* for V [44].

A vector can also be written as a linear combination of an overcomplete set of vectors. If the N-dimensional vector space V contains a set $\mathbf{F} = \{\mathbf{f}_j\}$ of K vectors where $K > N$, and $\mathbf{F}$ spans the space V, $\mathbf{F}$ is an overcomplete set. The vectors $\mathbf{f}_j$ are not linearly independent, and $\mathbf{F}$ is not a basis but a *frame*.

Any vector, $\mathbf{v}$, in the space V can be expanded as a linear combination of the frame vectors:

$$\mathbf{v} = \sum_{j=1}^{K} \alpha_j \mathbf{f}_j. \tag{2.2}$$

Because of the linear dependence of the frame vectors, the expansion is not unique.

The strict mathematical definition of a frame is as follows [67]: A family of elements $\{\varphi_j\}_{j \in K}$ in a Hilbert space H, where K is a countable index set, is called a *frame* if there exist an $A > 0$ and a $B < \infty$, such that for all x in H[1]:

$$A\|x\|^2 \leq \sum_j |\langle \varphi_j, x \rangle|^2 \leq B\|x\|^2. \tag{2.3}$$

A and B are called *frame bounds*. If $A = B$ the frame is said to be *tight*, and if all the frame vectors in a tight frame have unit norm, then A gives the redundancy ratio. This means that if, say, $A = 2$ there are twice as many vectors as needed to span the space. If $A = B = 1$ and all the vectors have unit norm, then the frame constitutes an orthonormal basis. The term *frame* thus covers both a basis and an overcomplete set of vectors. For a finite dimensional space, any finite set of vectors spanning the space forms a frame [32].

We use the term *frame* for a general linearly dependent set of vectors, typically overcomplete, which spans the space. Other terms, like dictionary or codebook have been used for similar sets, but these terms are often associated with vector quantization or classification.

2.2 Signal expansion

A *signal expansion* is simply a weighted sum of vectors $\mathbf{f}_j$. This weighted sum may be identical to, or an approximation to a given signal vector $\mathbf{x}$. If the expansion is identical to $\mathbf{x}$ we can write

$$\mathbf{x} = \mathbf{F}\mathbf{w} = \sum_j w_j \mathbf{f}_j, \tag{2.4}$$

where $\mathbf{F}$ is a (possibly infinite) matrix with $\{\mathbf{f}_j\}$ as columns and $\mathbf{w}$ is the vector of expansion coefficients. Equation 2.4 can be interpreted as a *synthesis*

[1] $\langle a, b \rangle$ is the inner product of a and b.

formula in the sense that $\mathbf{x}$ is synthesized, or built up, from a library of expansion vectors using appropriately selected values for the expansion coefficients. For this reason $\mathbf{F}$ is sometimes referred to as a *waveform dictionary.* If the matrix $\mathbf{F}$ is invertible, a unique set of coefficients for the exact representation of any signal vector $\mathbf{x}$ can be obtained as[2]

$$\mathbf{w} = \mathbf{F}^{-1}\mathbf{x}, \tag{2.5}$$

and this is commonly referred to as the *analysis* equation in an analysis-synthesis setting. Depending on the dimensions of the matrices and vectors involved in Equations 2.4 and 2.5, which may extend to infinity, as well as the structure of the $\mathbf{F}$ matrix, the analysis-synthesis equations given above cover many important cases including transforms, filter banks, wavelets, and wavelet packets [1].

The main objective for using the analysis-synthesis framework in signal processing applications is to construct $\mathbf{F}$ such that the vector of coefficients, $\mathbf{w}$, is more attractive to work with than $\mathbf{x}$.

We concentrate on the *synthesis* or *reconstruction* part of a signal expansion. If we put no restrictions on the choice of the waveform dictionary $\mathbf{F}$, like invertibility, dimension, orthogonality etc., Equation 2.4 also describes frames. Let the synthesis describe an approximation, rather than an exact representation, of the signal vector $\mathbf{x}$. Given the coefficients $\{w_j\}$, the reconstructed signal vector $\hat{\mathbf{x}}$ is given by

$$\hat{\mathbf{x}} = \mathbf{F}\mathbf{w} = \sum_j w_j \mathbf{f}_j. \tag{2.6}$$

We let all the frame vectors, $\mathbf{f}_j$, be normalized to one, so that the frame vector represent a shape. The gain is set by the coefficient value.

2.3 Frames used for compression purposes

Let $\mathbf{F}$ denote an $N \times K$ matrix where $K \geq N$ and $\text{rank}(\mathbf{F}) = N$. The columns, $\{\mathbf{f}_j\}$, $j = 1, \ldots, K$, are normalized to one, and they constitute a frame. Let $\mathbf{x}_l$ be a real signal vector, $\mathbf{x}_l \in \mathbf{R}^N$, $\mathbf{x}_l$ can be represented or approximated as

$$\hat{\mathbf{x}}_l = \sum_{j=1}^{K} w_l(j)\mathbf{f}_j = \mathbf{F}\mathbf{w}_l, \tag{2.7}$$

[2]For notational convenience we denote the *forward* matrix by $\mathbf{F}^{-1}$ and the *inverse* or *reconstruction* matrix by $\mathbf{F}$.

where $w_l(j)$ is the coefficient corresponding to vector $\mathbf{f}_j$. In a good compression scheme, many of the $w_l(j)$'s will be zero, while the approximation of Equation 2.7 is good.

The corresponding error energy is

$$\|\mathbf{r}_l\|^2 = \|\mathbf{x}_l - \hat{\mathbf{x}}_l\|^2, \tag{2.8}$$

where $\|\cdot\|$ denotes the Euclidean norm in $\mathbf{R}^N$. For a set of M signal vectors, the mean squared error (MSE) can be calculated as

$$\text{MSE} = \frac{1}{NM}\sum_{l=1}^{M}\|\mathbf{r}_l\|^2. \tag{2.9}$$

2.3.1 Frame coding compared to transform coding

The main idea behind transform coding is to remove redundancy in the input vector, $\mathbf{x}$, by transforming it to a new vector, $\mathbf{y}$, with same dimension. The vector $\mathbf{y}$ contains the coefficients, and these are less correlated than the original samples, thus we have energy compaction and thereby hopefully $\mathbf{y}$ can be quantized more efficiently than $\mathbf{x}$. The lower dependency there is between a set of variables, the more efficient scalar coding becomes in the sense that there is less to be gained by using more complicated vector quantization algorithms [26].

A traditional transform coder use an $N \times N$ orthogonal transform. $\mathbf{x}$ is the signal vector of dimension N, $\mathbf{T}$ is the transform and $\mathbf{y}$ is the coefficient vector, also of dimension N. We have:

$$\mathbf{y} = \mathbf{T}^T\mathbf{x} \tag{2.10}$$

$$\mathbf{x} = \mathbf{T}\mathbf{y}, \tag{2.11}$$

$\mathbf{T}^{-1} = \mathbf{T}^T$ due to the orthogonality of $\mathbf{T}$. It is common to refer to Equation 2.10 as analysis and to Equation 2.11 as synthesis. After analysis, the coefficient vector is approximated or quantized in some way, for example by threshold and uniformly quantize each of the coefficients. This gives the quantized coefficient vector $\hat{\mathbf{y}}$, and the approximation error $\|\mathbf{y} - \hat{\mathbf{y}}\|$. The reconstructed signal vector using the quantized coefficients becomes:

$$\hat{\mathbf{x}} = \mathbf{T}\hat{\mathbf{y}} = \sum_{j=1}^{N}\hat{y}_j\mathbf{t}_j, \tag{2.12}$$

where $\hat{\mathbf{x}}$ is the reconstructed signal vector, and the $\mathbf{t}_j$'s are the columns of $\mathbf{T}$, and they are called *basis vectors* since $\mathbf{T}$ represents an orthogonal basis.

Let the signal vector to be coded, $\mathbf{x}$, be regarded as a stochastic variable. Then the overall distortion D is given by $\langle \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \rangle$ where $\langle \cdot \rangle$ is the expectation operator. For an orthogonal transform it can easily be shown that $\langle \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \rangle = \langle \|\mathbf{y} - \hat{\mathbf{y}}\|^2 \rangle$.

Comparing frame based compression with transform coding, the synthesis Equation 2.12 is seen to be very similar to Equation 2.7. The analysis part on the other hand does not have an equivalent in frame compression. When frames are used for compression, the focus is on the synthesis part, and we free ourself from the usual *analysis-synthesis* setting. If $K = N$ in Equation 2.7, and $\{\mathbf{f}_j\}$, $j = 1, \ldots, N$, spans the space, $\mathbf{F}$ constitute a transform. If the transform is orthogonal we have the synthesis equation for a traditional transform coder.

2.3.2 Frames compared to filter banks and wavelets

Figure 2.1 shows 4 different choices of $\mathbf{F}$, as in Equation 2.6, corresponding to traditional signal decompositions: Transform, frame, wavelet, and uniform FIR filter bank/LOT. In each case 3 identical blocks of the expansion vectors are shown. The dots signify nonzero entries of the dictionary matrix. The figure gives an illustration on the difference and similarities between transforms, frames, wavelets and filter banks.

The upper left part of Figure 2.1 corresponds to ordinary transform coding. The upper right part of the figure corresponds to frames as used in this thesis. The main focus of this work are design and use of such frames. From the figure we see that the frames are not overlapping, and therefore we can design a frame on a block based form as done in this work. The same way as the transform is expanded to an overcomplete frame, the two lower structures in the figure, wavelet and filter bank/LOT, can be expanded to overcomplete wavelets and filter banks, *lapped frames*.

In [1] Aase et. al. introduce a generalization of the frame design algorithm presented in Chapter 4. More general waveform dictionaries $\mathbf{F}$, e.g. with overlapping, is included. Using the generalized algorithm, traditional wavelets or filter banks can be initial waveform dictionaries in the design scheme. Frames as used in this thesis correspond to a non-unitary filter bank, not critically sampled, and without overlapping.

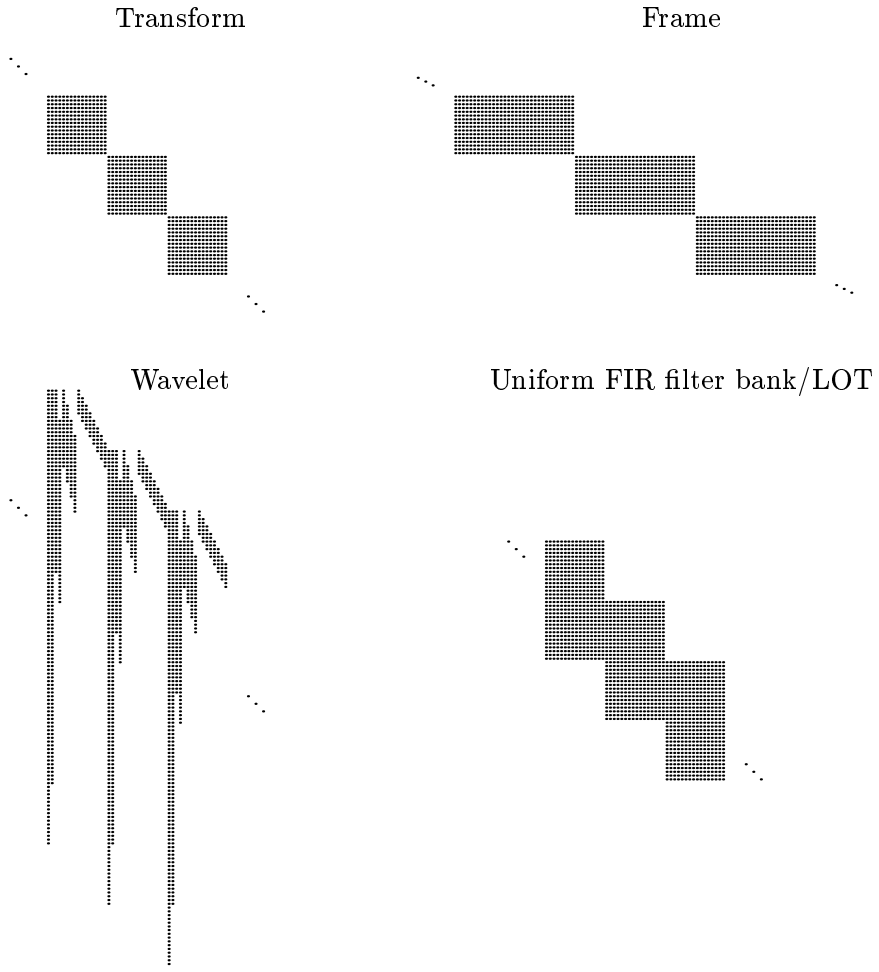


Figure 2.1: Waveform dictionaries corresponding to traditional decompositions. Starting with the transform dictionary, the dot columns within each square represent the transform vectors. The frame dictionary is similar but the number of frame vectors (K) is larger than the block length (N). In the filter bank/LOT case the vectors are twice as long as in the transform case, thus rendering a 50% overlap between adjacent blocks. A wavelet uses dyadic frequency partitioning, resulting in different time shifts for expansion vectors corresponding to different frequency bands. This is seen in the figure where the (large) vectors corresponding to the low frequency bands have longer shifts than the vectors corresponding to higher frequency bands.

2.3.3 Frames compared to vector quantization

Vector Quantization (VQ) is a generalization of scalar quantization to the quantization of a vector, but it's applications makes it far more than that. VQ is often used for compression, and that is the object of this discussion. In addition it can be used as a part of many digital processing tasks such as classification and recognition. A thorough study of VQ can be found in [26].

VQ

A vector quantizer designed for approximation of signal vectors of dimension N consists of a large codebook of vectors, all of dimension N, along with a vector selection strategy. The approximation of a specific input vector $\mathbf{x}_1$ is the vector from the codebook *closest to* $\mathbf{x}_1$, and the *index* to this vector from the codebook is used as the representation of that vector. The decoder uses the index in a table-lookup to find the right codebook vector representing $\mathbf{x}_1$. The definition of *closest to* is usually in the MSE sense:

$$\min_i \|\mathbf{x}_1 - \mathbf{c}_i\|, \tag{2.13}$$

where $\mathbf{c}_i,\ i = 1, 2 \ldots M$ are the codebook vectors.

We could picture the VQ technique as

$$\mathbf{x} \simeq \hat{\mathbf{x}} = \mathbf{C}\mathbf{g} \tag{2.14}$$

where $\mathbf{x}$ is the input vector, $\mathbf{C}$ is a $N \times M$ matrix containing all the codebook vectors as columns, and $\mathbf{g}$ is an indicator vector. $\mathbf{g}$ has *one* component equal to one, corresponding to the codebook vector closest to $\mathbf{x}$. All other entries in $\mathbf{g}$ are zero. The index of the nonzero component in $\mathbf{g}$ is used for representing $\mathbf{x}$.

Compared to the frame based approximation of Equation 2.7, we can see relations. In VQ, $M \gg N$ and the matrix (codebook) is indeed overcomplete. Lets look at an extreme case of frame based approximation, where $K \gg N$ and very few nonzero coefficients is needed. If we no longer let the frame vectors be normalized to one but have different magnitudes as well as shapes, the frame based system is similar to a VQ system. If only one nonzero coefficient is allowed in each approximation, and if the coefficient value is equal to one, we have a VQ system.

VQ is the ultimate solution to the quantization of signal vectors and no other existing coding technique can do better. This can be understood by the following theorem and proof from [26]:

Theorem: *For any given coding system that maps a signal vector into one of N binary words and reconstruct the approximate vector from the binary word, there exist a vector quantizer with codebook size N that gives exactly the same performance, i.e., for any input vector it produces the same reproduction as the given coding system.*

Proof: Enumerate the set of binary words produced by the coding system as indexes $1, 2, \ldots, N$. For the ith binary word, let the decoder output of the given coding system be the vector $\mathbf{c}_i$. Define the codebook $\mathbf{C}$ as the ordered set of code vectors $\mathbf{c}_i$. Then a VQ decoder achieves equivalent performance to the decoder of the given coding system and a VQ encoder can be defined to be identical to the encoder of the given coding system. □

This means that if we can find the optimal VQ for a given performance objective, no other coding system will be able to achieve a better performance. Unfortunately finding the optimal VQ is not straightforward.

Using codewords of fixed length for the representation of the indexes, the codebook can be optimized solely with respect to MSE. Allowing variable length coding, on the other hand, opens up for entropy coding. In this case the codebook should ideally be optimized with respect to both MSE and entropy, which is much more complicated. In VQ fixed codeword length is the by far most used technique, and the codebook design algorithms are optimizing with respect to MSE solely.

There are no known closed-form solutions to the problem of finding the optimal VQ. Iterative techniques for finding a local optimum are used. The Generalized Lloyd Algorithm (GLA) finds a local optimum by iteratively optimizing the encoder with fixed decoder and the decoder with fixed encoder. Given a training set $\mathcal{T}$, the GLA steps are as follows [26]:

1. Begin with an initial codebook $\mathbf{C}^{(1)}$. Set m=1.

2. Given a codebook $\mathbf{C}^{(m)} = \{\mathbf{c}_i\}$, partition the training set into cluster sets $\mathcal{R}_i$ using the Nearest Neighbor Condition:

$$\mathcal{R}_i = \{\mathbf{x} \in \mathcal{T} : d(\mathbf{x}, \mathbf{c}_i) \leq d(\mathbf{x}, \mathbf{c}_j);\ \text{all } j \neq i\},$$

   where $d(\mathbf{x}, \mathbf{c}_i)$ is a measure of the distortion between $\mathbf{x}$ and $\mathbf{c}_i$.

3. Using the Centroid Condition, compute the centroids for the cluster sets just found to obtain the new codebook, $\mathbf{C}^{(m+1)} = \{cent(\mathcal{R}_i)\}$.

$$cent(\mathcal{R}_i) = \frac{1}{\mathcal{C}(\mathcal{R}_i)} \sum_{j=1}^{\mathcal{C}(\mathcal{R}_i)} \mathbf{x}_j,$$

   where $\mathcal{C}(\mathcal{R}_i)$ is the cardinality of the set $\mathcal{R}_i$, that is, the number of elements in $\mathcal{R}_i$. If $\mathcal{C}(\mathcal{R}_i) = 0$ an alternate code vector assignment is made for that cell.

4. Compute the average distortion for $\mathbf{C}^{(m+1)}$. If it has changed by a small enough amount since the last iteration, stop. Otherwise set $m + 1 \rightarrow m$ and go to step 2.

Other disadvantages of VQ are the high search complexity for finding the best match from the codebook, and the storage demands of large codebooks. Especially the search complexity is a significant problem when the codebooks are large, and this has motivated the development of various constrained VQ techniques. In these techniques the optimality is traded in exchange for easier coding and/or smaller storage requirements. Two constrained VQ techniques with obvious relations to frame based coding is briefly explained in the following.

Shape-gain VQ

Shape-gain VQ is a technique that decomposes the problem of approximating and representing a vector into that of coding a scalar, the *gain*, and a normalized vector, the *shape*. The idea of shape-gain VQ is that the shape of a vector may recur with a wide variety of gain values. If this is true, it suggests that the probability distribution of the shape is approximately independent of the gain.

The gain of a vector $\mathbf{x}$ is $g = g(\mathbf{x}) = \|\mathbf{x}\|$, and the shape is $\mathbf{s} = \mathbf{s}(\mathbf{x}) = \frac{\mathbf{x}}{g(\mathbf{x})}$ defined for nonzero gain. This gives $\|\mathbf{s}\| = 1$.

Shape-gain VQ is described by three objects:

- the gain codebook $\mathbf{C}_g = \{g_i;\ i = 1, 2, \ldots, M_g\}$,
- the shape codebook $\mathbf{C}_\mathbf{s} = \{\mathbf{s}_j;\ j = 1, 2, \ldots, M_\mathbf{s}\}$,

- the partition that describes the encoder $\mathcal{R} = \{\mathcal{R}_{i,j} \in R^N;$ $i = 1, 2, \ldots, M_g;\ j = 1, 2, \ldots, M_{\mathbf{s}}\}$. If $\mathbf{x} \in \mathcal{R}_{i,j}$ the approximation is formed by the shape $\mathbf{s}_j$ and the gain g_i.

Using a shape-gain VQ to represent a signal vector, the shape vector closest to the shape of the input vector is selected first. The vector from the shape codebook with the largest inner product with the input vector is selected:

$$k = \arg\max_j (\mathbf{x}^T \mathbf{s}_j). \tag{2.15}$$

Using this shape vector, the gain from the gain codebook is chosen to be the one closest to the value of the inner product:

$$l = \arg\min_i (g_i - \mathbf{x}^T \mathbf{s}_k)^2. \tag{2.16}$$

k and l are the indexes representing the signal vector. If the shape and gain had been truly independent, they could have been designed independently. In practice they are *not* truly independent, however, and a common way of designing a shape-gain VQ, optimized with respect to MSE, is a variation of the GLA. This variation of the GLA is iterative and the main steps, using a training set $\mathcal{T}$, are as follows [26]:

1. Begin with an initial codebooks $\mathbf{C}_g^{(1)}$ and $\mathbf{C_s}^{(1)}$. Set m=1.

2. Given the codebooks $\mathbf{C}_g^{(m)}$, $\mathbf{C_s}^{(m)}$, find the optimal (minimum distortion) partition $\mathcal{R}[\mathbf{C}_g^{(m)}, \mathbf{C_s}^{(m)}]$ of $\mathcal{T}$.

3. Compute the average distortion: $D(\mathbf{C}_g^{(m)}, \mathbf{C_s}^{(m)}, \mathcal{R}[\mathbf{C}_g^{(m)}, \mathbf{C_s}^{(m)}])$. If it has changed by a small enough amount since the last iteration, stop. Else continue.

4. Compute the optimal shape codebook $\mathbf{C_s}^{(m+1)}$ using $\mathbf{C}_g^{(m)}$ and $\mathcal{R}[\mathbf{C}_g^{(m)}, \mathbf{C_s}^{(m)}]$ by finding the shape centroids.

5. Compute the optimal partition $\mathcal{R}[\mathbf{C}_g^{(m)}, \mathbf{C_s}^{(m+1)}]$.

6. Compute the optimal gain codebook $\mathbf{C}_g^{(m+1)}$ using $\mathbf{C_s}^{(m+1)}$ and $\mathcal{R}[\mathbf{C}_g^{(m)}, \mathbf{C_s}^{(m+1)}]$ by finding the gain centroids.

7. Set $m = m + 1$ and go to Step 2.

Comparing shape-gain VQ to frame based approximation we see that in the extreme case where only *one* frame vector is used in Equation 2.7 the methods are basically the same if we quantize the frame coefficient. In frame based approximation all the frame vectors are normalized, and thus correspond to the *shape*. The gain correspond to the coefficient value. Shape-gain VQ and frame based approximation may differ in the design of the codebooks. We have made no limitation in the frame based approximation in that the coefficients can only possess positive values. In other words, we let the sign be a part of the gain codebook. This gives a smaller shape codebook, but a larger gain codebook compared to the traditional shape-gain VQ.

Multistage VQ

MultiStage VQ (MSVQ) is also called cascaded VQ or residual VQ and has been widely used in speech coding. The basic idea is to divide the encoding task into several successive stages. The first stage performs a crude quantization of the input vector using a small codebook. The residual from this stage, that is the difference between the input vector and the reconstructed vector from the first codebook, is treated as the input vector to the second stage. Another relatively small codebook is used to quantize the residual, and this provides a second approximation vector and a new residual. A third quantizer may be used to approximate the second residual and so forth. The decoder adds the vectors from the different codebooks together to make an approximation of the input vector. This method gives significant reduction in storage requirements and search complexity compared to straightforward VQ.

Some 5-8 years ago it was common to do both the design of the codebooks, and the coding at the different stages independently of the other stages [26]. This was done as follows: A training set is used to design the first codebook. Using this codebook, a set of residuals is calculated by letting the training set be approximated by the codebook. This set of residuals is used to design the codebook of stage 2 and so forth. The encoding procedure is illustrated in Figure 2.2 a). The signal vector is first approximated using the first stage codebook. The residual is calculated and the second stage codebook is used to approximate the first stage residual and so forth, thus the coding strategy is *greedy*.

In recent years more sophisticated MSVQ methods have been used. Iterative sequential codebook design and simultaneous joint codebook design [69, 45] gives better codebooks, but are computationally more expensive. An M-L search procedure improves the coding strategy [45]. The strategy starts

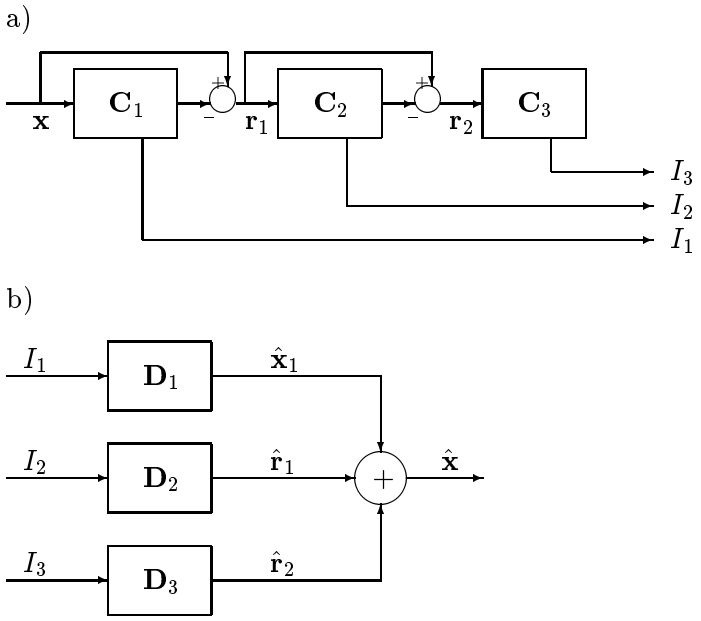


Figure 2.2: Illustration of traditional MSVQ. a) encoder b) decoder

with the first codebook. The M vectors achieving the lowest distortion are selected, and the M residual vectors are calculated. The second codebook is searched M times, once for each of the M residual vectors. From all the resulting distortions of each of the codebook vectors approximating each of the M residuals, the M paths achieving the overall lowest distortion are selected. The procedure continues for all the stages, and after the last stage, the path giving the lowest overall distortion is selected. Note that a given vector in any codebook may be the root of more than one of the M paths selected at any of the following levels.

The decoding part of the MSVQ has obvious similarity to signal approximation using frames. The decoder in Figure 2.2 b) approximate the signal vector by a sum of vectors. The coefficients in Equation 2.7 are the most obvious distinction between MSVQ and frame compression. If Equation 2.7 was describing an MSVQ system, it would correspond to the situation where each of the stages in the MSVQ was a shape-gain VQ. However, *the systems differ both in design and encoding procedures.* Let a signal be partitioned in blocks, where each signal block is treated as a signal vector. In a frame based system the number of frame vectors used in the sum in Equation 2.7, i.e. the number of vectors used in the approximation can vary with each signal block. This way

an approximation quality can be almost constant even if some of the signal block are more difficult to approximate than others. In an MSVQ based system the number of vectors is constant. There is a possibility of thresholding, but an approximation can never use more vectors than the number of stages, thus the MSVQ offers less flexibility in this sense.

A frame based system selects all the vectors from the same dictionary, or frame. An MSVQ in general consists of different codebooks at the different stages. Figure 2.3 a) illustrates a frame based system using m vectors in an approximation, and b) illustrates an MSVQ system.

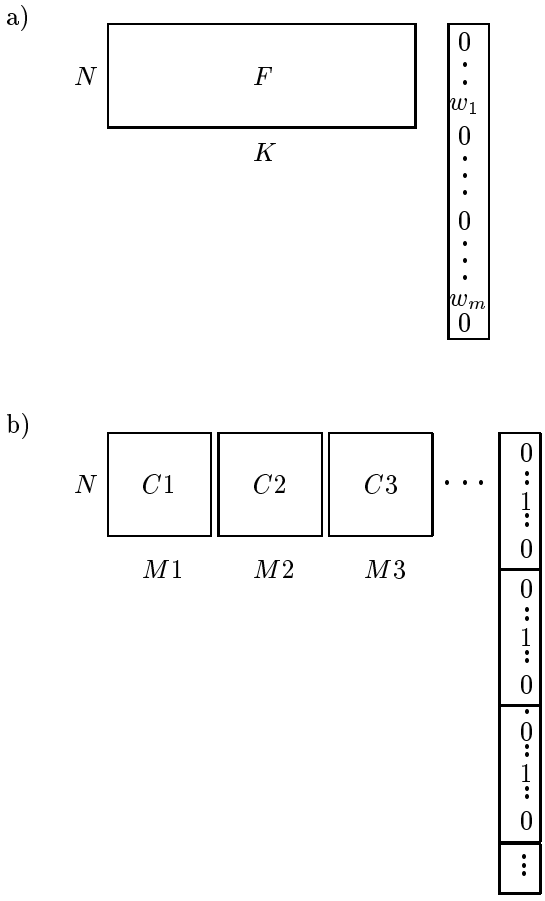


Figure 2.3: Illustration of compression systems: a) Frame, b) MSVQ.

We want to compare frame based coding with MSVQ in the following. Note

that even if the structure of the systems are similar, the design methods are different, and so are the encoding methods. The encoding is suboptimal for both the frame system and the MSVQ. Thus the systems are not the same, and will not produce exactly the same outputs, even if the systems have the exact same approximation possibilities. Let an MSVQ have m stages. To compare the MSVQ with a frame system, let the frame system use exactly m vectors in each approximation, as the MSVQ does, and let the m frame vector coefficients be quantized by a quantizer with L different scalar representation values. The Frame based system gives:

$$L^m \left(\begin{array}{c} K \\ m \end{array} \right) \tag{2.17}$$

different approximation possibilities. The number of bits needed when the coefficient positions and values are coded separately with fixed codeword length, i.e. when using no run-length or entropy coding, is:

$$\begin{aligned} m \log_2 L + \log_2 \left(\begin{array}{c} K \\ m \end{array} \right) &= \\ m \log_2 L + \sum_{i=1}^{K} \log_2 i - \sum_{i=1}^{m} \log_2 i - \sum_{i=1}^{K-m} \log_2 i &= \\ m \log_2 L + \sum_{i=(K-m+1)}^{K} \log_2 i - \sum_{i=1}^{m} \log_2 i & \end{aligned} \tag{2.18}$$

The MSVQ system gives:

$$\prod_{i=1}^{m} M_i \tag{2.19}$$

different approximation possibilities. The number of bits needed to code the indexes using a fixed codeword length, i.e. when using no run-length or entropy coding, is:

$$\sum_{i=1}^{m} \log_2 M_i. \tag{2.20}$$

Three different scenarios are investigated further:

Scenario 1: MSVQ with m equal codebooks

Let $\mathbf{C}_i = \mathbf{C}$ thus $M_i = M$, and let $M = KL$. Let $\mathbf{C}$ consists of the K frame vectors in $\mathbf{F}$ multiplied with all the L different coefficients allowed in the frame

based system. In this case these two systems have theoretically the exact same output possibilities. The MSVQ system needs:

$$m \log_2 M = m \log_2 KL = m \log_2 L + m \log_2 K \tag{2.21}$$

bits, whereas the number of bits needed for the frame system can be seen from Equation 2.18. Subtracting Equation 2.18 from Equation 2.21 we get:

$$m \log_2 K - \sum_{i=(K-m+1)}^{K} \log_2 i + \sum_{i=1}^{m} \log_2 i > 0 \tag{2.22}$$

except when $m = 1$ where they are equal, but that will never be the case for an MSVQ system. Thus the frame system requires fewer bits than the MSVQ system for all practical purposes, and the difference increases with increasing m. The systems have theoretically the same output possibilities, however both methods uses suboptimal vector selection techniques, so the compression result for one specific signal vector need not be exactly the same.

Scenario 2: MSVQ with different codebooks and a frame system with the same approximation possibilities

By letting the frame have the same approximation possibilities as the MSVQ we mean that the frame can produce exactly the same output as all the possible MSVQ outputs. This does not hold the other way around, since the frame turns out to have much more approximation possibilities.

Let $\mathbf{C}_j \neq \mathbf{C}_i,\ j \neq i$, that is let the codebooks at the different stages in the MSVQ be different. This is the common way to use an MSVQ. Let $M_i = KL,\ i = 1, \ldots, m$ for simplicity. If we want the frame based system to have the same approximation possibilities, the frame now need to be of size $N \times mK$. The MSVQ system still uses

$$m \log_2(KL) = m \log_2 L + m \log_2 K \tag{2.23}$$

bits per signal vector. The frame system needs

$$m \log_2 L + \sum_{i=(mK-m+1)}^{mK} \log_2 i - \sum_{i=1}^{m} \log_2 i \tag{2.24}$$

bits per signal vector. Subtracting Equation 2.24 from Equation 2.23 we get:

$$m \log_2 K - \sum_{i=(mK-m+1)}^{mK} \log_2 i + \sum_{i=1}^{m} \log_2 i < 0 \tag{2.25}$$

except when $m = 1$ where they are equal, thus for all practical purposes. The frame system seems to be more expensive, but on the other hand it turns out to have much more approximation possibilities than the MSVQ. The frame system has

$$L^m \left(\begin{array}{c} mK \\ m \end{array} \right) \tag{2.26}$$

distinct possible approximations whereas the MSVQ system has $\prod_{i=1}^{m} KL = (KL)^m$ distinct possible approximations.

$$\left(\begin{array}{c} mK \\ m \end{array} \right) > K^m \tag{2.27}$$

except when $m = 1$, and much greater for most practical purposes. Since the frame based system can be any combination of the mK different vectors with any of the L different values, an MSVQ system would have to have all these possible vectors in all of the codebooks to possess the same flexibility, thus all the codebooks would need to be $\mathbf{C} = [\mathbf{C}_1\ \mathbf{C}_2 \ldots \mathbf{C}_m]$.

Scenario 3: Exactly the same **number** *of possible approximations for MSVQ and frame system*

A third possible case is to let the frame based system and the MSVQ have exactly the same *number* of possible approximations, but not necessarily *same* approximations. From Equation 2.17 and Equation 2.19 we have:

$$L^m \left(\begin{array}{c} K \\ m \end{array} \right) = \prod_{i=1}^{m} M_i. \tag{2.28}$$

If $M_i = M$ we get:

$$M = L \sqrt[m]{\left(\begin{array}{c} K \\ m \end{array} \right)}. \tag{2.29}$$

The number of bits needed for the frame system is as in Equation 2.18. Combining Equation 2.20 with Equation 2.29 gives the number of bits needed for MSVQ:

$$\begin{aligned} m \log_2 (L \sqrt[m]{\left(\begin{array}{c} K \\ m \end{array} \right)}) \quad &= \\ m \log_2 L + \log_2 \left(\begin{array}{c} K \\ m \end{array} \right) & \end{aligned} \tag{2.30}$$

Comparing Equation 2.18 with Equation 2.30 we see that we need the exact same number of bits for the MSVQ system and the frame system when we have the same *number* of possible approximations instead of demanding the *same* possible approximations.

The bit comparison in this section are all done without any run-length or entropy coding, which can indeed change the situation.

2.4 Vector selection algorithms

The potential advantage in using a frame instead of an orthogonal transform is that we have more vectors to choose from and thus a better chance of finding a small number of vectors whose linear combination match the signal vector well. Since a linearly dependent set of vectors is used, an expansion is no longer unique. In a compression scheme the goal is to use as few vectors as possible to obtain a good approximation for each signal vector. Consequently it makes sense to apply a sparsity constraint to the coefficient set. Finding the optimal frame vectors to use in such an approximation is an NP-hard problem and requires extensive calculation [55]. A suboptimal technique is preferable in order to limit the computational complexity. There exist several different vector selection methods dealing with this problem. They can be grouped into sequential and parallel methods [60].

The sequential methods are greedy methods, selecting vectors one at a time. They start by choosing the frame/dictionary vector that match the signal vector best, building an approximation by iteratively selecting new vectors according to some criterion, like the best match to the residual. Matching Pursuit (MP) [52] and Orthogonal Matching Pursuit (OMP) [12] are examples of such greedy algorithms for choosing vectors from a frame.

In the parallel methods *all* the vectors of the frame/dictionary are initially selected, and processed. Vectors are eliminated until a requisite number remains. Basis Pursuit [11], and FOCUSS (FOCal Underdetermined System Solver) [29] are examples of parallel vector selection algorithms.

2.4.1 Matching Pursuit (MP) techniques

Greedy techniques for vector selection have been known for a long time. Mallat and Zhong reintroduced MP in 1993 [52], and their algorithm is closely related to algorithms used in statistics [25].

Let $\{\mathbf{f}_\gamma\}_{\gamma\in\Gamma}$, $\Gamma = \{1, 2 \dots K\}$ be a set of vectors constituting a frame, $\mathbf{F}$, with $K > N$ vectors, each of length N, and let $\{\mathbf{f}_\gamma\}_{\gamma\in\Gamma}$ span the space $\mathbf{R}^N$. A matching pursuit begins by projecting the signal vector $\mathbf{x}$ on a vector $\mathbf{f}_{\gamma_0}$, $\gamma_0 \in \Gamma$, and computing the residual $\mathbf{r}$ [50].

$$\mathbf{x} = (\mathbf{x}^T\mathbf{f}_{\gamma_0})\mathbf{f}_{\gamma_0} + \mathbf{r}, \tag{2.31}$$

$$\|\mathbf{x}\|^2 = |\mathbf{x}^T\mathbf{f}_{\gamma_0}|^2 + \|\mathbf{r}\|^2 \tag{2.32}$$

since $\mathbf{r}$ is orthogonal to $\mathbf{f}_{\gamma_0}$.
To minimize $\mathbf{r}$, $\mathbf{f}_{\gamma_0}$, $\gamma_0 \in \Gamma$ must be chosen such that $|\mathbf{x}^T\mathbf{f}_{\gamma_0}|$ is maximum:

$$|\mathbf{x}^T\mathbf{f}_{\gamma_0}| \geq \sup_{\gamma\in\Gamma} |\mathbf{x}^T\mathbf{f}_\gamma|. \tag{2.33}$$

This is iterated to form an approximation of $\mathbf{x}$. In iteration k the residual from iteration $k-1$ is projected on the frame vectors $\{\mathbf{f}_\gamma\}_{\gamma\in\Gamma}$. Let $\mathbf{r}_0 = \mathbf{x}$ and $\mathbf{r}_k$ the residual after the k'th iteration. The next vector $\mathbf{f}_{\gamma_k}$ chosen satisfies:

$$|\mathbf{r}_k^T\mathbf{f}_{\gamma_k}| \geq \sup_{\gamma\in\Gamma} |\mathbf{r}_k^T\mathbf{f}_\gamma|. \tag{2.34}$$

The approximation after iteration $k+1$ can be written:

$$\mathbf{x} = (\mathbf{x}^T\mathbf{f}_{\gamma_0})\mathbf{f}_{\gamma_0} + (\mathbf{r}_1^T\mathbf{f}_{\gamma_1})\mathbf{f}_{\gamma_1} + \dots (\mathbf{r}_k^T\mathbf{f}_{\gamma_k})\mathbf{f}_{\gamma_k} + \mathbf{r}_{k+1} \tag{2.35}$$

$$= \sum_{j=0}^{k} (\mathbf{r}_j^T\mathbf{f}_{\gamma_j})\mathbf{f}_{\gamma_j} + \mathbf{r}_{k+1}. \tag{2.36}$$

From Equation 2.36 we see that the coefficients used in the approximation of the vector $\mathbf{x}$ is the inner product between the residual at that stage in the iterations and the chosen frame vector. This is not changed later in the iterations. Since the frame vectors are not orthogonal, including a new vector in the approximation might change the optimal coefficient for the earlier chosen frame vectors. This motivates the different variations of MP algorithms. In the following the two algorithms we have used in the present work are explained.

Orthogonal Matching Pursuit (OMP)

The algorithm we refer to as OMP in this thesis is due to Davis [12].

The OMP differs from the MP by optimizing all the coefficient values after each iteration, thus the OMP gives a better approximation but is computationally more expensive. What is done in the OMP is that the residual, for each iteration, is orthogonalized to the space spanned by the previously chosen

vectors. The algorithm starts exactly like the MP algorithm. The difference occurs when the second frame vector has been selected. In MP the inner product between the residual and the frame vector is the coefficient value, and the coefficient value used with the previously chosen frame vector is kept. In OMP both these coefficients are optimized making the best possible approximation of $\mathbf{x}$ using just these two frame vectors. The optimization is done by finding the least squares solution, which is computed from the pseudo inverse for the overdetermined problem:

$$\tilde{\mathbf{w}} = (\tilde{\mathbf{F}}^T\tilde{\mathbf{F}})^{-1}\tilde{\mathbf{F}}^T\mathbf{x} = \tilde{\mathbf{F}}^+\mathbf{x}. \tag{2.37}$$

where $\tilde{\mathbf{F}}$ is a matrix consisting of the chosen frame vectors as columns, and $\tilde{\mathbf{w}}$ is a vector with the corresponding coefficients. Now the new residual is found:

$$\mathbf{r}_2 = \mathbf{x} - \tilde{w}_0\mathbf{f}_{\gamma_0} - \tilde{w}_1\mathbf{f}_{\gamma_1}. \tag{2.38}$$

The next vector is chosen as before:

$$|\mathbf{r}_k^T\mathbf{f}_{\gamma_k}| \geq \sup_{\gamma\in\Gamma} |\mathbf{r}_k^T\mathbf{f}_{\gamma}|. \tag{2.39}$$

The new coefficients is found by using Equation 2.37, and this is done iteratively until either the required number of vectors are selected or the residual is less than some limit. The approximation can be written:

$$\mathbf{x} = \sum_{j=0}^{k} \tilde{w}_j\mathbf{f}_{\gamma_j} + \mathbf{r}_{k+1}, \tag{2.40}$$

or if we let the sparse vector $\mathbf{w}$ consist of the $\tilde{w}_j$'s at the right positions and zero elsewhere:

$$\mathbf{x} = \mathbf{F}\mathbf{w} + \mathbf{r}. \tag{2.41}$$

In [12] the algorithm is based on orthogonalizing the selected vector with respect to the space that is spanned by the previous selected vectors. The coefficient used with this new orthogonalized vector is $(\mathbf{r}_k^T\mathbf{f}_{\gamma_k})$ and need not to be recalculated. This is equivalent to the explanation above.

Fast Orthogonal Matching Pursuit (FOMP)

The algorithm we refer to as FOMP was proposed by Garavi-Alkhansari and Huang [27]. It is *not* a fast version of Davis' OMP from the previous section,

but a fast version of a different variant of the MP algorithm. The literature is unfortunately not consistent in naming conventions used for different MP algorithms. We consistently refer to the various algorithms by the names given by their originators.

The difference between the OMP in [12] and the FOMP in [27] is that the OMP orthogonalize the residual with respect to the space spanned by the selected vector in each iteration, whereas the FOMP orthogonalize *the remaining vectors in the dictionary* with respect to the one just selected in each iteration. After the last iteration, the coefficients are recalculated using least squares method as in Equation 2.37. This method have been called Order Recursive Matching Pursuit (ORMP) in other references, and it gives better results than OMP. It is considerably more computational expensive than Davis' OMP, but Garavi-Alkhansari and Huang proposed a fast version in [27] and this is what we call FOMP.

2.4.2 FOCUSS

The FOCUSS algorithm is a *best basis*, or vector selection algorithm, developed by Gorodnitsky and Rao [29]. It was assumed that there is a perfect match between the data $\mathbf{x}$ and a linear combination of a few columns of $\mathbf{F}$:

$$\mathbf{x} = \mathbf{F}\mathbf{w}, \tag{2.42}$$

where $\mathbf{F}$ is a $N \times K$ matrix where $K \geq N$. As we see the vector selection problem consists of solving an *underdetermined* linear system of equations [60].

There are *many* solutions to the system of equations (2.42) and the best basis selection problem corresponds to identifying a few columns of the matrix $\mathbf{F}$ that best represent the data vector $\mathbf{x}$ [52]. This corresponds to finding a solution to (2.42) with few nonzero entries, i.e. a sparse solution.

FOCUSS, for **FOC**al **U**nderdetermined **S**ystem **S**olver [29] is a parallel vector selection algorithm. The FOCUSS method was motivated by the observation that if a sparse solution is desired then choosing a solution based on the smallest 2-norm is not appropriate. The minimum 2-norm criteria favors solutions with many small nonzero entries, a property that is contrary to the goal of sparsity [10, 29]. Consequently there is a need to consider the minimization of alternative measures that promote sparsity. In this context, of particular interest are diversity measures, functionals which measure the sparsity, and algorithms for minimizing them to obtain sparse solutions. A popular diversity

measure is the $\ell_{(p\leq 1)}$ diversity measure given by [62, 39],

$$E^{(p)}(\mathbf{w}) = \text{sgn}(p)\sum_{j=1}^{K}|w_j|^p, \quad p \leq 1. \tag{2.43}$$

Minimizing these measures, with the constraint given by Equation 2.42, naturally leads to the iterative algorithm FOCUSS. Let k be the iteration number. The iterations are as follows [29, 62, 39]:

$$\mathbf{w}^{(k+1)} = \mathbf{Q}^{(k+1)}\left(\mathbf{F}\mathbf{Q}^{(k+1)}\right)^{+}\mathbf{x}, \tag{2.44}$$

where $\mathbf{Q}^{(k+1)} = \text{diag}(|w_j^{(k)}|^{1-\frac{p}{2}})$. "+" denotes the pseudoinverse. Intuitively, the algorithm can be explained by noting that there is competition between the columns of $\mathbf{F}$ to represent $\mathbf{x}$. In each iteration, certain columns get emphasized while others are deemphasized. In the end a few columns survive to represent $\mathbf{x}$ providing a sparse solution. An initial solution, $\mathbf{w}^{(0)}$, is needed in the algorithm. If any of the columns are let out, that is any of the w_j's are zero, they can not get back in the competition, and the value will stay equal to zero. Therefore the initial vector $\mathbf{w}^{(0)}$ should not contain any zeros so that all possibilities are kept open. A good initial vector is the minimum norm solution to the Equation 2.42 since the minimum norm solution usually spreads the energy over all the coefficients.

Defining

$$\mathbf{q} \stackrel{\Delta}{=} (\mathbf{Q}^{(k+1)})^{-1}\mathbf{w}, \tag{2.45}$$

in each iteration of the FOCUSS algorithm the solution $\mathbf{w}^{(k+1)}$ is computed as $\mathbf{w}^{(k+1)} = \mathbf{Q}^{(k+1)}\mathbf{q}^{(k+1)}$, where

$$\mathbf{q}^{(k+1)} = \arg\min_{\mathbf{q}} \|\mathbf{q}\|^2 \text{ subject to } \mathbf{F}\mathbf{Q}^{(k+1)}\mathbf{q} = \mathbf{x}. \tag{2.46}$$

By doing this scaling transformation, the FOCUSS algorithm can be solved by a sequence of weighted minimum 2-norm problems. The FOCUSS algorithm can be summarized as:

$$\begin{aligned} \mathbf{Q}^{(k+1)} &= \text{diag}(|w_j^{(k)}|^{1-\frac{p}{2}}) \\ \mathbf{q}^{(k+1)} &= (\mathbf{F}^{(k+1)})^{+}\mathbf{w}, \text{ where } \mathbf{F}^{(k+1)} = \mathbf{F}\mathbf{Q}^{(k+1)} \\ \mathbf{w}^{(k+1)} &= \mathbf{Q}^{(k+1)}\mathbf{q}^{(k+1)} \end{aligned} \tag{2.47}$$

Note that the algorithms 2.47 and 2.44 are entirely equivalent because they are related by the scaling transformation of Equation 2.45.

More details on FOCUSS can be found in [29, 62]. The next chapter deals with a version of FOCUSS allowing noise in the data, or equivalently finding an *approximated* representation instead of an *exact* representation as done in FOCUSS. This is called regularized FOCUSS, and gives the following equation system: $\mathbf{x} = \mathbf{F}\mathbf{w} + \mathbf{n}$, or equivalent $\hat{\mathbf{x}} = \mathbf{F}\mathbf{w}$.

Chapter 3

Regularized FOCUSS

Our goal in the work presented in this chapter was to develop robust subset selection methods that have applications to signal representation and to find sparse solutions to linear inverse problems from noisy observations. This work can also be found in [61].

We wanted to make a version of the FOCUSS algorithm that could deal with noise in the data. This would make it possible to use it in situations where the data have a sparse structure, but is polluted with noise. It also makes it possible to use it in compression and frame design. To deal with noise in the data, basis selection procedures based on a Bayesian framework was considered. An algorithm based on the MAP estimation procedure was developed which lead to a regularized version of the FOCUSS algorithm. Some of the results in this chapter were published in [23].

3.1 Basis selection in the presence of noise

The derivation of FOCUSS [29, 62] did not explicitly account for noise in the data. It was assumed that there is a *perfect match* between the data $\mathbf{x}$ and a linear combination of a few columns of $\mathbf{F}$. In [29] reasonable modifications were made to the algorithm to deal with noise. Here we take a formal approach and extend the FOCUSS method to deal with noise in the measurements using a Bayesian framework. As we will see, the stochastic framework provides theoretical insights and assists in developing robust methods. For this discussion, we assume that the data vector $\mathbf{x}$ is the result of a true underlying sparse structure:

$$\mathbf{x} = \mathbf{F}\mathbf{w} + \mathbf{n}, \tag{3.1}$$

where $\mathbf{n}$ is a random additive noise vector. Furthermore, in this formulation $\mathbf{w}$ is also assumed to be a random vector independent of $\mathbf{n}$. $\mathbf{F}$ is assumed known. Under these assumptions, a Maximum A Posteriori (MAP) estimate of $\mathbf{w}$ can be obtained,

$$\begin{aligned} \mathbf{w}_{map} &= \arg\max_{\mathbf{w}} \ln p(\mathbf{w}|\mathbf{x}) \\ &= \arg\max_{\mathbf{w}} \left[\ln p(\mathbf{x}|\mathbf{w}) + \ln p(\mathbf{w})\right] \\ &= \arg\max_{\mathbf{w}} \left[\ln p_{\mathbf{n}}(\mathbf{x} - \mathbf{F}\mathbf{w}) + \ln p(\mathbf{w})\right]. \end{aligned}$$

The last equality is obtained because $p(\mathbf{x}|\mathbf{w}) = p(\mathbf{F}\mathbf{w} + \mathbf{n}|\mathbf{w})$, and $\mathbf{F}$ is assumed known. Then $\mathbf{n}$ is the only random variable since $\mathbf{w}$ is given. This can thereby be written as $p_{\mathbf{n}}(\mathbf{n})$, where $\mathbf{n} = \mathbf{x} - \mathbf{F}\mathbf{w}$.

This formulation is general with considerable flexibility. In order to proceed further, some assumptions about the noise $\mathbf{n}$ and the solution vector $\mathbf{w}$ have to be made. The distribution of $\mathbf{n}$ is not very critical to the approach except for analytical and computational tractability. We assume that $\mathbf{n}$ is a Gaussian random vector with independent and identical distributed (iid) elements[1], i.e.:

$$p_{\mathbf{n}}(\mathbf{n}) = c_1 e^{-\frac{\|\mathbf{n}\|^2}{2\sigma^2}}. \tag{3.2}$$

The distribution of $\mathbf{w}$ is quite important for the generation of sparse solutions. For this purpose, the elements w_j are assumed to be iid random variables with generalized Gaussian distribution. The probability density function of the generalized Gaussian distribution family is defined as [58, 68]:

$$f(w; p, \beta) = \frac{p}{2\sqrt[p]{2}\beta\Gamma(\frac{1}{p})} e^{-\frac{|w|^p}{2\beta^p}}, \quad p > 0 \tag{3.3}$$

where $\Gamma(\cdot)$ is the standard gamma function. If $p = 2$ and $\beta = 1$ this is the standard normal distribution. p controls the shape, and β is a generalized variance. If unit variance, $\sigma^2 = 1$, is wanted then β becomes a function of p, and only one parameter can be varied:

$$\beta^2 = 2^{-\frac{2}{p}} \frac{\Gamma(\frac{1}{p})}{\Gamma(\frac{3}{p})} \tag{3.4}$$

Figure 3.1 shows a plot of the pdf, $f(w; p)$, for different p's when $\sigma^2 = 1$. From the figure it can be seen that the pdf moves towards a uniform distribution

[1] More general Gaussian distributions can be also easily dealt with.

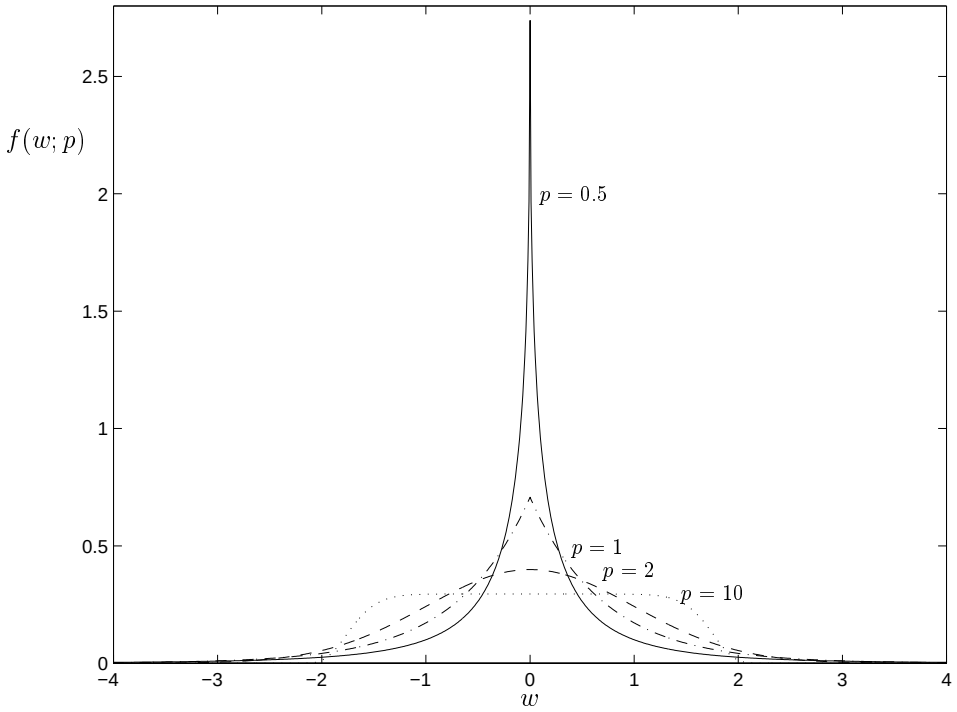


Figure 3.1: pdf for the generalized Gaussian distribution with different p, $\sigma^2 = 1$. dotted: $p = 10$, dashed: $p = 2$ (standard normal distribution), dash-dot: $p = 1$, solid: $p = 0.5$

.

as $p \to \infty$, and towards a very peaky distribution as $p \to 0$. A vector $\mathbf{w}$ with dimension K, where the elements are generalized Gaussian and independent, has the following pdf:

$$p_{\mathbf{w}}(\mathbf{w}) = \left(\frac{p}{2\sqrt[p]{2}\beta\Gamma(\frac{1}{p})}\right)^K e^{-\frac{1}{2\beta^p}\mathrm{sgn}(p)\sum_{j=1}^{K}|w_j|^p} \tag{3.5}$$

To be consistent with the $l_{(p\leq 1)}$ diversity measure of Equation 2.43 where $p \leq 1$, the $\mathrm{sgn}(p)$ is added to allow for $p < 0$.
Substituting these densities in the expression for the MAP estimate results in

$$\begin{aligned} \mathbf{w}_{map} &= \arg\min_{\mathbf{w}} J(\mathbf{w}), \\ \text{where } J(\mathbf{w}) &= \left[\|\mathbf{F}\mathbf{w} - \mathbf{x}\|^2 + \mathrm{sgn}(p)\frac{\sigma^2}{\beta^p}\sum_{j=1}^{K}|w_j|^p\right]. \end{aligned}$$

Note that $p = 2$ gives rise to the standard regularized least squares problem. For $p \leq 1$ it can be shown that the minima of $J(\mathbf{w})$ are sparse. Following the factored-gradient approach in [62, 39], an iterative algorithm can be derived to minimize $J(\mathbf{w})$ which has the form[2] [63]:

$$\mathbf{w}^{(k+1)} = \mathbf{Q}^{(k+1)} \left(\mathbf{F}^{(k+1)T} \mathbf{F}^{(k+1)} + \lambda \mathbf{I} \right)^{-1} \mathbf{F}^{(k+1)T} \mathbf{x}, \tag{3.6}$$

where $\mathbf{F}^{(k+1)} = \mathbf{F}\mathbf{Q}^{(k+1)}$ with $\mathbf{Q}^{(k+1)} = \text{diag}(|w_j^{(k)}|^{1-\frac{p}{2}})$ and $\lambda = \text{sgn}(p)\frac{\sigma^2}{\beta^p}$. Using the fact that

$$\mathbf{F}^{(k+1)T} \left(\mathbf{F}^{(k+1)} \mathbf{F}^{(k+1)T} + \lambda \mathbf{I} \right) = \left(\mathbf{F}^{(k+1)T} \mathbf{F}^{(k+1)} + \lambda \mathbf{I} \right) \mathbf{F}^{(k+1)T}, \tag{3.7}$$

algorithm (3.6) can be expressed as

$$\mathbf{w}^{(k+1)} = \mathbf{Q}^{(k+1)} \mathbf{F}^{(k+1)T} \left(\mathbf{F}^{(k+1)} \mathbf{F}^{(k+1)T} + \lambda \mathbf{I} \right)^{-1} \mathbf{x}. \tag{3.8}$$

This is the iteration in the regularized FOCUSS algorithm. When the noise level is reduced, $\sigma \to 0$, then $\lambda \to 0$ and the algorithm reduces to the original FOCUSS algorithm (2.44). The algorithm (3.8) has an interesting interpretation as Tikhonov regularization applied to (2.46). This can be readily seen by rewriting (3.8) as a solution to a regularized least squares problem. Then we have $\mathbf{w}^{(k+1)} = \mathbf{Q}^{(k+1)} \mathbf{q}^{(k+1)}$, where

$$\mathbf{q}^{(k+1)} = \arg\min_{\mathbf{q}} \|\mathbf{F}\mathbf{Q}^{(k+1)}\mathbf{q} - \mathbf{x}\|^2 + \lambda \|\mathbf{q}\|^2 \tag{3.9}$$

Interestingly, this results in an algorithm identical to that suggested in [29]. In [29], the algorithm was arrived at as a way to make the 2-norm minimization problem of (2.46) more robust to noise. This derivation provides formal support to the approach. Convergence results can be found in [61].

3.2 The regularization parameter

The quality of the sparse solution obtained via the regularized FOCUSS is governed by the choice of λ, and there remains the problem of determining a proper value for λ. Determining a proper value for λ is an important problem and has implications to other subset selection methods as well as to other regularization problems. Sparsity adds an interesting twist to this classical

[2] When the elements of $\mathbf{F}$ and $\mathbf{x}$ are complex, the transpose operation has to be replaced by the Hermitian transpose

problem, and in the subset selection context, there appears to be no practical reason to limit the choice of λ to a fixed value for all the iterations. A value that is dependent on the iteration may be more appropriate. With this in mind we suggest and study three approaches motivated by three different scenarios. The first approach is motivated by the desire to ensure a certain quality of representation and exploits the availability of some information on the perturbations. The second is motivated by the need to ensure a certain degree of sparsity on the solution as would be required in applications like compression. The third is induced by the desire to produce stable sparse solutions without the need for much prior information.

3.2.1 Quality of fit criterion / discrepancy principle

A potentially useful approach is to try to seek a sparse solution that assures a certain quality in the nature of the representation, i.e. $\|\mathbf{F}\mathbf{w} - \mathbf{x}\| \leq \epsilon$. This is called the *discrepancy principle* [35]. Algorithmically this reduces to solving the optimization problem

$$\min_{\mathbf{w}} E^{(p)}(\mathbf{w}) \text{ subject to } \|\mathbf{F}\mathbf{w} - \mathbf{x}\| \leq \epsilon. \tag{3.10}$$

Assuming that the inequality constraint is active, which is usually true, and following the approach used to derive the regularized solution, an iterative algorithm can be derived which at each iteration computes $\mathbf{w}^{(k+1)} = \mathbf{Q}^{(k+1)}\mathbf{q}^{(k+1)}$, where

$$\mathbf{q}^{(k+1)} = \arg\min_{\mathbf{q}} \|\mathbf{q}\|^2 \text{ subject to } \|\mathbf{F}\mathbf{w} - \mathbf{x}\| \leq \epsilon. \tag{3.11}$$

An algorithm for computing $\mathbf{q}^{(k+1)}$ is given in [59]. The convergence of the algorithm to a sparse solution can be shown because it is possible to show that in each iteration $\|\mathbf{q}^{(k+1)}\|^2 \leq \sum_{j=1}^{K} |w_j^k|^p$, and the following lemma from [61]:

Lemma 1 *in each iteration of the regularized FOCUSS algorithm (3.8), if* $\|\mathbf{q}^{(k+1)}\|^2 \leq \sum_{j=1}^{K} |w_j^{(k)}|^p$, *then the algorithm converges and the stable fixed points are sparse solutions.*

The proof follows readily from the convergence proof of FOCUSS presented in [62].

3.2.2 Sparsity criterion

Another option is to choose λ so that the solution produced has a predetermined number of nonzero entries r. Note that upon convergence the rank of $\mathbf{FQ}^{(k+1)}$ is equal to r, i.e. $\lim_{k\to\infty} \text{rank}(\mathbf{FQ}^{(k+1)}) = r$. So a desirable approach would be to use a sequence λ_k to satisfy this limiting rank property, while providing the best fit possible. A reliable procedure for doing this is not yet available. One practical approach is to use a sequential basis selection method like the OMP to select r columns, and to determine a value for the error ϵ in the representation. This ϵ can be the basis of FOCUSS along the lines suggested in section 3.2.1. If the procedure returns more columns than desired, one can either prune the selected subset or go with OMP solution whichever is better.

3.2.3 Modified L-curve criterion

In this approach, the regularizing parameter is found by striking a compromise between minimizing the norm of the solution vector, $\|\mathbf{q}\|^2$, versus the error in the representation, $\|\mathbf{FQ}^{(k+1)}\mathbf{q} - \mathbf{x}\|^2$. In this context, this choice also translates into controlling the sparse nature of the solution, so that a trade off between quality of fit and sparsity is done. The use of such an approach was first suggested in [29]. The L-curve was introduced by Hansen in [34] as a method for finding the parameter λ in the regularization problem:

$$\min_{\mathbf{w}}\{\|\mathbf{Fw} - \mathbf{x}\|^2 + \lambda\|\mathbf{w}\|^2\}. \tag{3.12}$$

When using regularized FOCUSS, as described in section 3.1, the regularization problem can be written as:

$$\min_{\mathbf{q}}\{\|\mathbf{FQ}^{(k+1)}\mathbf{q} - \mathbf{x}\|^2 + \lambda\|\mathbf{q}\|^2\}. \tag{3.13}$$

If λ is varied from 0 to ∞, $\|\mathbf{q}\|^2$, a measure of sparsity, decreases monotonically from $\|(\mathbf{FQ}^{(k+1)})^+\mathbf{x}\|^2$ to zero and $\|\mathbf{FQ}^{(k+1)}\mathbf{q} - \mathbf{x}\|^2$, a measure of the approximation error, increases monotonically. The theory of the L-curve poses that a plot of $\|\mathbf{q}\|^2$ versus $\|\mathbf{FQ}^{(k+1)}\mathbf{q} - \mathbf{x}\|^2$ for different λ will be shaped as an L and that a good λ is the one corresponding to the corner in the L. Further more it is suggested [34, 35, 33] that the corner of the L-shaped curve can be found by finding the maximum curvature. The plot of $\|\mathbf{q}\|^2$ versus $\|\mathbf{FQ}^{(k+1)}\mathbf{q} - \mathbf{x}\|^2$ can be shown to be convex [35], and the maximum curvature will be at a trade

off point between sparsity and accuracy. The curvature can be computed by means of the formula:

$$K(\lambda) = \frac{X'(\lambda)Y''(\lambda) - X''(\lambda)Y'(\lambda)}{\{[X'(\lambda)]^2 + [Y'(\lambda)]^2\}^{3/2}}, \tag{3.14}$$

where $X(\lambda) = \|\mathbf{F}\mathbf{Q}^{(k+1)}\mathbf{q} - \mathbf{x}\|^2$ and $Y(\lambda) = \|\mathbf{q}\|^2$. This way the computations are done in the linear scale. In [35, 33] the curvature computations are done in the log-log scale, that is $X(\lambda) = log\{\|\mathbf{F}\mathbf{Q}^{(k+1)}\mathbf{q} - \mathbf{x}\|^2\}$, $Y(\lambda) = log\{\|\mathbf{q}\|^2\}$. The reasons for doing this is somehow unclear, but in [35] there are some arguments for the corner to be more distinct in the log-log scale. A problem pointed out by Reginska in [64] is that the L-curve in the log-log scale is no longer convex in general. In [48] a linear scale L-curve is used, and in [24] both linear and log-log scale L-curves are mentioned. In fact experiments have shown that the log-log curve often has several corners and finding the maximum curvature in this scale does not necessarily correspond to a λ with a good trade off between sparsity and accuracy.

Experiments show that using the log-log approach is not good for this application of the L-curve method. The algorithm ends up emphasizing quality of fit too much, and sparsity to little. Linear scale experiments show a potential for the regularized FOCUSS to perform better than greedy algorithms like the OMP, but for some data vectors it fails completely. The variance in the error is large, which indicates that the method is not very robust. The L-curve approach fails because the data will *not* produce an L-curve in each iteration of the regularized FOCUSS algorithm.

To improve the robustness, we propose a method using a combination of the discrepancy principle and the L-curve method, linear scale. We call this the *modified L-curve method*. When using the basic L-curve to decide λ there is no direct control on *how many* vectors to select, or *a limit* on the error. The thought is simply that the L-curve is able to find the best way of minimizing *both* these terms, or finding the best trade off between accuracy and sparsity. In the proposed modified L-curve method, we have to know something about the variance of the noise, or alternatively something about the target SNR after doing an approximation. From this knowledge an upper and an lower target on the residual norm, $\epsilon^2 = \|\mathbf{F}\mathbf{w} - \mathbf{x}\|^2$ can be made. Then for every iteration in regularized FOCUSS the upper and lower target on ϵ^2 is used to find an upper and a lower limit for λ, $\{\lambda_{min}, \lambda_{max}\}$. The λ corresponding to the maximum curvature in the linear scale, λ_c, is also calculated in every regularized FOCUSS iteration. λ_c is then compared with the limits. If $\lambda_c < \lambda_{min}$ then λ_{min} is used, if $\lambda_c > \lambda_{max}$ then λ_{max} is used, else λ_c is used. This ensures that the λ will always be acceptable even if there is no distinct L-corner.

3.3 The regularization parameter - experiments and results

Numerous experiments are conducted on synthetic data to understand the reliability of the methods proposed above. Experiments are done using an 20×30 matrix, $\mathbf{F}$, with random entries chosen from a normal distribution with mean zero and variance one. The columns in $\mathbf{F}$ are normalized. The noise free data vector is obtained as a linear combination of m randomly picked vectors from $\mathbf{F}$ where the coefficients are Gaussian random variables with zero mean and unit variance. The constructed coefficient vector is denoted $\check{\mathbf{w}}$. Two different values for m is used: 4, and 7. That means that $\check{\mathbf{w}}$ has 4, and 7 nonzero values respectively. The noise free data vector is normalized $\Rightarrow \check{\mathbf{x}}$. The noisy data vector, $\mathbf{x}$, is $\check{\mathbf{x}} + \mathbf{n}$ where $\mathbf{n}$ is a noise vector with Gaussian random entries with zero mean and variance depending on the Signal to Noise Ratio (SNR) in the experiment. Mathematically, the synthetic data can be described as

$$\begin{aligned} \mathbf{F}\frac{\check{\mathbf{w}}}{\|\mathbf{F}\check{\mathbf{w}}\|} &= \check{\mathbf{x}} \\ \mathbf{x} &= \check{\mathbf{x}} + \mathbf{n}. \end{aligned}$$

Each experiment is done with at least 100 different data vectors; $\mathbf{x}_l$ $l = 1, 2 \ldots M$, $M \geq 100$.
In the experiments we know the frame, $\mathbf{F}$, and the noisy data, $\mathbf{x}_l$, $l = 1, 2 \ldots M$, and we use the different versions of regularized FOCUSS and OMP to find a coefficient vector $\mathbf{w}$.
Several factors have been studied to evaluate the experiments. There are two types of error:

$$\epsilon_1^2 = \|\mathbf{F}\mathbf{w} - \check{\mathbf{x}}\|^2 \tag{3.15}$$

$$\epsilon^2 = \|\mathbf{F}\mathbf{w} - \mathbf{x}\|^2 \tag{3.16}$$

ϵ_1^2 from Equation 3.15 is the error of the reconstructed signal compared to the original signal *before* noise was added, and ϵ^2 from Equation 3.16 is the error of the reconstructed signal compared to the original signal *with* noise. If one is trying to find the underlying function of a known sparse process, then the first error measure is the most informative. If one is trying to represent a signal in the best possible manner without knowing the underlying generating function (e.g. compression), then the latter will be the most informative.
Three different experiments were done: The quality of fit criterion described in section 3.2.1, the sparsity criterion described in section 3.2.2, and the modified L-curve criterion described in section 3.2.3. Experiments are done using OMP on the same data sets for comparison.

3.3.1 Test 1 and test 2 - Discrepancy principle and sparsity criterion

The discrepancy principle (test 1) and sparsity criterion (test 2) are tested on the same data set, and are therefore evaluated together.

In the experiment of the discrepancy principle we assume that we know something about the variance of the noise. This way we can set the bound on the norm of the error as a function of the noise variance. Let the variance of $n_i, i = 1, 2 \dots N$ be σ^2. Then $\langle \|\mathbf{n}\|^2 \rangle = N\sigma^2$, and the error bound is set to $CN\sigma^2$, where C is a selected constant. When using this approach the number, r, of vectors from the $\mathbf{F}$ matrix selected to approximate a data vector $\mathbf{x}$ will vary for different data vectors. To be able to compare the result using regularized FOCUSS with result using OMP we have to either fix the error and compare the number of vectors used, or fix the number of vectors used and compare the error for each trial. Since it is not possible to fix the error at an exact level, we choose to fix the number of selected vectors, r, using regularized FOCUSS and OMP for the same data vector, and compare the error. Using regularized FOCUSS, there is an upper bound on the norm of the error, the r is not controlled directly. Thus for every data-vector, $\mathbf{x}_l$, the regularized FOCUSS algorithm runs first, and the r_l is found. Then the OMP can be run for the same data vector with the restriction that it has to pick exactly r_l vectors. OMP is a greedy algorithm, and can easily be stopped at any r or with an upper bound on the error.

In the experiment of the sparsity criterion the number of vectors to be selected to approximate the data is fixed. That means that the number of nonzero entries in $\mathbf{w}$ is fixed. In this experiment the goal is to find the best possible approximation in terms of minimal MSE using a linear combination of r columns from the $\mathbf{F}$ matrix. We use the same r as the m that where used when producing the synthetic data, assuming that this factor is known. Unfortunately it is not trivial to control the r when regularized FOCUSS is used as vector selection algorithm. The way it is done in this experiment is as follows: For a data-vector $\mathbf{x}_l$ OMP runs first finding an approximation using r vectors. The ϵ^2 from Equation 3.16 is calculated and used as an input for the upper bound when running regularized FOCUSS as in the discrepancy principle. Let r_{focuss} be the number of vectors that regularized FOCUSS uses. If $r_{focuss} > r$ it is pruned down to r by using OMP to select r of the r_{focuss} vectors. If $r_{focuss} < r$ extra vectors are added using OMP until r vectors are selected. This way we always use r vectors in each approximation.

Explanation to the tables: 1 FOCUSS and 2 FOCUSS means test 1 and 2 using the regularized FOCUSS approach. p is a factor in the regularized

Test	p	C	$\bar{r}$	$\bar{r}_m$	mean ϵ_1^2	mean ϵ^2	% ϵ_1^2	% ϵ^2
1 FOCUSS	0	0.8	9.3	5.86	0.0080	0.0044	57	24
1 OMP			9.3	5.58	0.0123	0.0074	29	62
2 FOCUSS	0		7	5.47	0.0150	0.0135	38	30
2 OMP			7	5.29	0.0179	0.0162	35	43
1 FOCUSS	0	1.2	8.35	5.60	0.0091	0.0066	46	20
1 OMP			8.35	5.68	0.0131	0.0095	40	66
2 FOCUSS	0		7	5.61	0.0135	0.0118	38	25
2 OMP			7	5.59	0.0143	0.0126	39	52
1 FOCUSS	0.5	1.2	6.7	5.44	0.0096	0.0098	30	25
1 OMP			6.7	5.37	0.0169	0.0144	33	38
2 FOCUSS	0.5		7	5.67	0.0098	0.0086	35	27
2 OMP			7	5.57	0.0124	0.0104	37	45
1 FOCUSS	0.8	1.5	7.88	5.84	0.0080	0.0063	45	23
1 OMP			7.88	5.58	0.0125	0.0094	41	63
2 FOCUSS	0.8		7	5.70	0.0126	0.0114	33	24
2 OMP			7	5.40	0.0163	0.0146	26	35

Table 3.1: Experiments done on test1 and test2 with the same data set. SNR= 20 dB, $m = 7$

FOCUSS algorithm[3], $\bar{r}$ means the average number of vectors selected per data vector, $\bar{r}_m$ means the average number of selected vectors which is identical with vectors used to construct $\hat{\mathbf{x}}$, % ϵ_1^2/ϵ^2 means the percentage of the trials where regularized FOCUSS/OMP performs better in terms of ϵ_1^2/ϵ^2. The reason why e.g. % ϵ^2 regularized FOCUSS and % ϵ^2 OMP does not add to 1 is that they perform exactly the same for some of the trials. For the experiments with test 1, C means the selected constant in the error bound as explained.

From Table 3.1 it can bee seen that the mean of both ϵ_1^2 and ϵ^2 is less for regularized FOCUSS than OMP in all the experiments. For low SNR in Table 3.2 this is still the case for the mean of ϵ_1^2 but no longer for ϵ^2. For high SNR, but with $m = 4$, seen in Table 3.3 the mean of both ϵ_1^2 and ϵ^2 is less for regularized FOCUSS than for OMP in most of the experiments. With one exception the mean number of correct selected vectors, $\bar{r}_m$, is larger for regularized FOCUSS than OMP for both SNR's when using $m = 7$, but this is more variable when $m = 4$. Most of these results is in favor of the regularized FOCUSS algorithm. The reason for the $\bar{r}_m$ to be better for regularized FOCUSS when using $m = 7$

[3]described in Section 3.1 (from the $l_{p\leq 1}$ diversity measure)

Test	p	C	$\bar{r}$	$\bar{r}_m$	mean ϵ_1^2	mean ϵ^2	% ϵ_1^2	% ϵ^2
1 FOCUSS	0	0.8	6.97	4.21	0.0782	0.0496	55	22
1 OMP			6.97	4.10	0.0840	0.0427	32	65
2 FOCUSS	0		7	4.26	0.0839	0.0470	56	32
2 OMP			7	4.13	0.0881	0.0428	34	58
1 FOCUSS	0.5	1	5.25	3.99	0.0844	0.0739	38	50
1 OMP			5.25	3.78	0.0909	0.0700	31	38
2 FOCUSS	0.5		7	4.46	0.0785	0.0444	15	26
2 OMP			7	4.30	0.0824	0.0408	54	62
1 FOCUSS	0.8	1	6.39	4.28	0.0791	0.0587	41	19
1 OMP			6.39	4.09	0.0841	0.0513	41	63
2 FOCUSS	0.8		7	4.43	0.0798	0.0452	39	29
2 OMP			7	4.18	0.0829	0.0449	32	42

Table 3.2: Experiments done on test1 and test2 with the same data set. SNR= 10 dB, $m = 7$

Test	p	C	$\bar{r}$	$\bar{r}_m$	mean ϵ_1^2	mean ϵ^2	% ϵ_1^2	% ϵ^2
1 FOCUSS	0	1	3.06	2.79	0.0265	0.0323	8	5
1 OMP			3.06	2.82	0.0279	0.0334	10	13
2 FOCUSS	0		4	3.54	0.0060	0.0101	16	6
2 OMP			4	3.58	0.0084	0.0122	18	28
1 FOCUSS	0.5	1	5.02	3.53	0.0057	0.0069	29	8
1 OMP			5.02	3.67	0.0071	0.0076	20	41
2 FOCUSS	0.5		4	3.51	0.0086	0.0117	13	7
2 OMP			4	3.64	0.0079	0.0115	20	26
1 FOCUSS	0.8	1	5.45	3.57	0.0051	0.0055	69	9
1 OMP			5.45	3.58	0.0102	0.0082	15	75
2 FOCUSS	0.8		4	3.56	0.0048	0.0085	13	4
2 OMP			4	3.52	0.0103	0.0136	7	16

Table 3.3: Experiments done on test1 and test2 with the same data set. SNR= 20 dB, $m = 4$

than when using $m = 4$ is that the greedy algorithm, OMP, works well when only a few vectors are to be selected. The sub-optimality becomes greater if more vectors are selected. Regularized FOCUSS is also suboptimal, but as it is a parallel algorithm it will have a greater probability of working better when many vectors need to be selected.

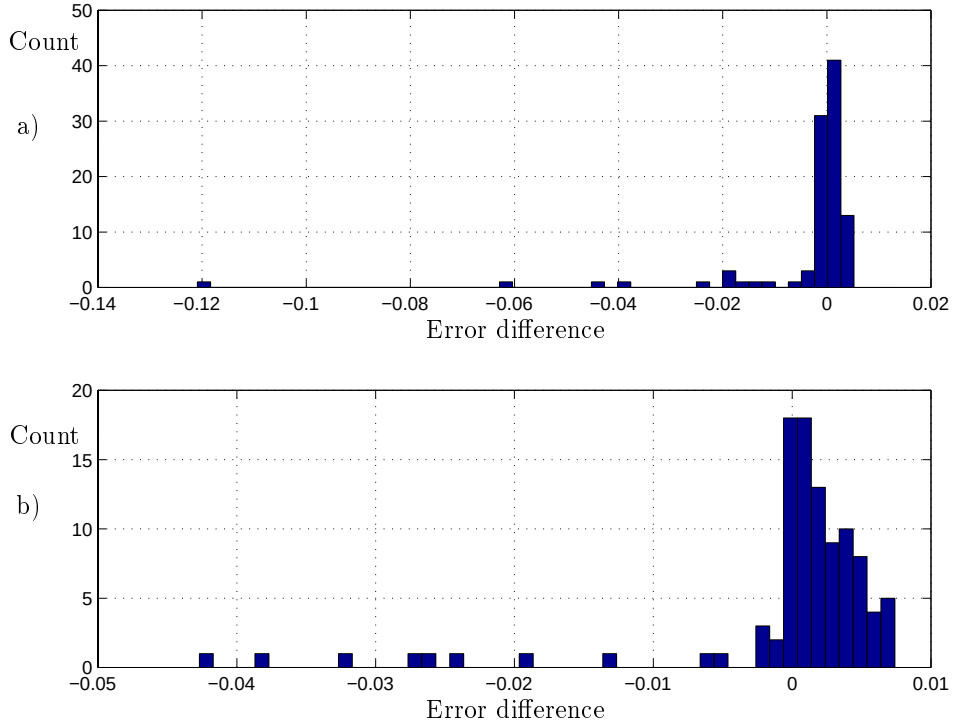


Figure 3.2: Histogram of $\epsilon^2_{focuss} - \epsilon^2_{omp}$ for test 1 with SNR=20 dB, $m = 7$, p=0. a) $C = 0.8$ b) $C = 1$

Figure 3.2 shows a histogram of $\epsilon^2_{focuss} - \epsilon^2_{omp}$ for test 1 with SNR=20 dB, $m = 7$, p=0, and $C = 0.8$ and $C = 1$ in a) and b) respectively. As is seen from the Tables 3.1, 3.2, and 3.3 the results of % ϵ_1^2 and % ϵ^2 seems to be in favor of the OMP in many of the experiments, but still the mean values of ϵ_1^2 and ϵ^2 are in favor of the regularized FOCUSS. This is because when OMP performs better it only performs marginally better, but when regularized FOCUSS performs better it sometimes performs significantly better. This is seen by the skew histograms in Figure 3.2.

3.3.2 Test 3 - modified L-curve method

The modified L-curve method requires some knowledge of the noise level, or a target on the approximation SNR. In particular, the largest (ϵ^2_{max}) and smallest (ϵ^2_{min}) error in the approximation are required. This is used to find $\lambda_{max}, \lambda_{min}$, as described in section 3.2.3. In each regularized FOCUSS iteration $\lambda_{max}, \lambda_{min}$, and λ_c is found.

The noise vector $\mathbf{n}$ has Gaussian random entries with variance $\sigma_n^2 = \frac{\text{SNR}}{N}$, and the SNR level is 10 or 20 dB. $\|\mathbf{n}\|^2$ has a chi-squared distribution and is used to find the limits. The limits ϵ^2_{min} and ϵ^2_{max} are chosen as $P(\|\mathbf{n}\|^2 \geq \epsilon^2_{min}) = P(\|\mathbf{n}\|^2 \leq \epsilon^2_{max}) = T$, where T is a chosen threshold. For these experiments a threshold of 0.1 was used and this gives $\epsilon^2_{min} = 0.0062$ and $\epsilon^2_{max} = 0.0142$ for SNR = 20 dB, and 10 times as much for 10 dB.

If the true SNR of the data is unknown, targets for the SNR can be used to decide the error limits. If the wanted SNR is approximately X dB, an upper error limit can be set using $X - \Delta_1$ dB as an SNR target, and a lower limit using $X + \Delta_2$ dB.

$$\epsilon^2_{upper} = 10^{-(X-\Delta_1)/10}\|\mathbf{x}\|^2 \tag{3.17}$$

$$\epsilon^2_{lower} = 10^{-(X+\Delta_2)/10}\|\mathbf{x}\|^2 \tag{3.18}$$

For every data vector, $\mathbf{x}$, an ϵ^2_{upper} and ϵ^2_{lower} is calculated using Equation 3.17, and 3.18 before the regularized FOCUSS iterations start.

For each data vector in the experiment, regularized FOCUSS runs first and r_l is found, then OMP runs on the same data vector and stops after selecting exact r_l frame vectors. The errors are then compared.

In Table 3.4 results from the modified L-curve method are showed with SNR's on 10 and 20 dB.

Table 3.5 shows experiments where the true SNR for the generated data is 20 dB, but assumed to be unknown. A lower target is set to 15 dB and a higher to 25 dB pretending not to know anything about the noise but wanting the approximation to have an SNR between 15 and 25 dB. The results are in favor of the regularized FOCUSS when compared to OMP. The achieved SNR can be calculated from the mean ϵ^2. For $p = 0$ the SNR_{focuss} is 16.8 dB and for $p = 0.5$ it is 17.6 dB. The results has a lower SNR than the true SNR, but the number of selected vectors is approximately 5.5 when $m = 7$ was used to generate the data.

Figure 3.3 a) is a plot of $\bar{r}_m$ for the different data vectors, where $\bar{r}_m$ is between 3 and 11. From b) it is seen that the the variance in the error is small and that

Test	SNR	p	$\bar{r}$	$\bar{r}_m$	mean ϵ_1^2	mean ϵ^2	% ϵ_1^2	% ϵ^2
FOCUSS	20 dB	0	7.04	5.35	0.0097	0.0103	53	42
OMP			7.04	5.05	0.0192	0.0176	47	51
FOCUSS	20 dB	0.5	6.86	5.32	0.0102	0.0094	45	36
OMP			6.86	5.05	0.0200	0.0178	55	55
FOCUSS	20 dB	0.8	10.69	5.97	0.0083	0.0036	74	26
OMP			10.69	5.68	0.0117	0.0049	26	74
FOCUSS	10 dB	0	4.08	3.46	0.1171	0.1152	52	39
OMP			4.08	3.06	0.1283	0.1186	39	48
FOCUSS	10 dB	0.5	4.34	3.58	0.0991	0.0938	59	41
OMP			4.34	3.22	0.1087	0.1168	34	46
FOCUSS	10 dB	0.8	8.38	4.57	0.0824	0.0393	76	21
OMP			8.38	4.14	0.0939	0.0295	24	77

Table 3.4: Experiments done on the modified L-curve method. $m = 7$, SNR on 10 and 20 dB.

Test	p	$\bar{r}$	$\bar{r}_m$	mean ϵ_1^2	mean ϵ^2	% ϵ_1^2	% ϵ^2
FOCUSS	0	5.48	4.69	0.0204	0.0211	44	42
OMP		5.48	4.67	0.0224	0.0231	50	46
FOCUSS	0.5	5.51	4.93	0.0163	0.0174	50	38
OMP		5.51	4.71	0.0218	0.0215	45	41

Table 3.5: Experiments done on the modified L-curve method. $m = 7$, SNR target between 15 and 25 dB. True SNR for generated data is 20 dB

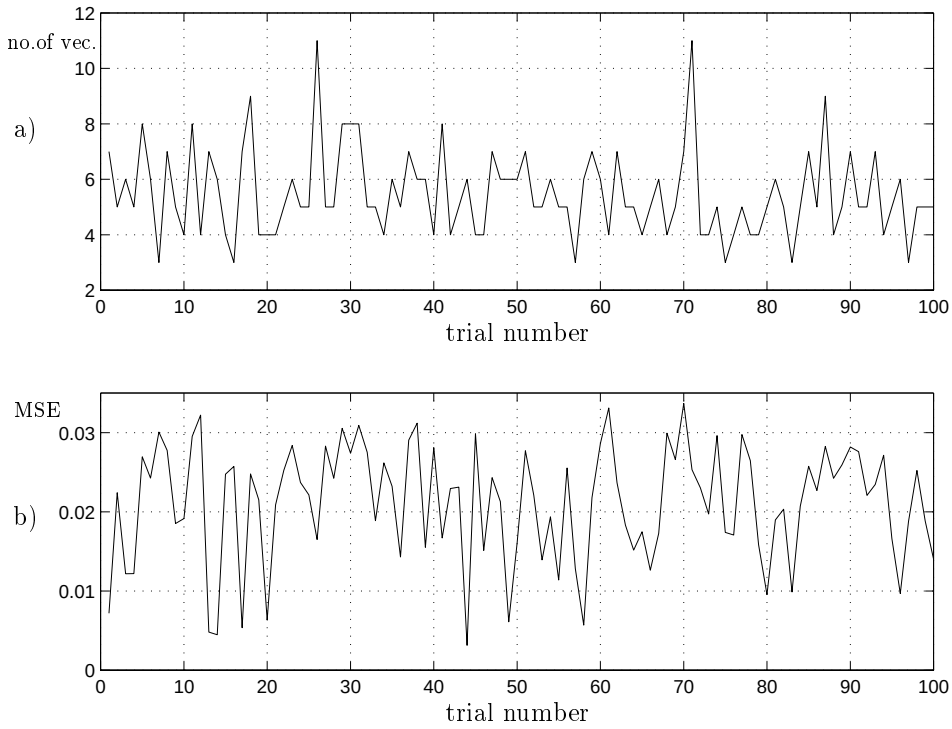


Figure 3.3: Modified L-curve, true SNR for generated data is 20 dB, SNR target between 15 and 25 dB. a) number of selected vectors in each trial, b) ϵ^2_{focuss} for each trial

means that the variance in the approximation quality for the different trails are small. In this experiment the achieved SNR for each trail varies between 15 and 25 dB, which corresponds to the predetermined limits on the SNR. Comparing a) and b) in Figure 3.3 it can bee seen that the error is in general not smaller for the trails where $\bar{r}_m$ is large. This, and the small variance in the error, indicates that the method combining the target SNR with the linear scale L-curve is working well.

In summary, the original L-curve scheme exerts no strict controls over the approximation quality, and this often results in the regularization parameter improperly choosing between quality of approximation and sparsity, leading to an unreliable procedure. Our proposed scheme remedies this by the requirement of a target SNR, and procedures for determining the target SNR are presented. The target SNR enables setting limits on the SNR desired of the approximations, and then letting the L-curve algorithm find a good trade

off between sparsity and quality of fit within the controlled limits ensures robustness. In the context of compression, the possibility of controlling bounds for the error, while obtaining the minimum bit rate at that error level can be a very desirable property.

Regularized FOCUSS in the rest of this thesis uses the *modified L-curve method* to find the regularization parameters.

Chapter 4

Frame design

In this chapter the problem of *frame design* is addressed. We presented an algorithm for frame design using a training set in [16]. In Section 4.1 we present a significantly improved version of the frame design algorithm which we call the Method of Optimal Directions (MOD). MOD was first presented in [18]. In Section 4.2 we discuss frame design from a probabilistic point of view and establish that some of these approaches gives the same solution as the MOD.

4.1 Method of Optimized Directions, MOD

The iterative algorithm designed to optimize frames is inspired by the GLA described in Section 2.3.3. Each iteration in the GLA consists of two parts. First the optimal classification for the training set is found using a given codebook. In the context of VQ, classification corresponds to finding the best vector in the codebook representing the training vector. Secondly a better codebook is found using the existing classification and training set. It follows that the new codebook is guaranteed to be no worse than the previous, and the GLA will eventually find at least a local optimum.

The frame design problem is tackled the same way by dividing each iteration in the training algorithm into two parts. Let $\mathbf{F}$ be the $N \times K$ frame, $\mathbf{x}_l,\ l = 1, 2 \ldots M$ the training set, and $\mathbf{w}_l,\ l = 1, 2 \ldots M$ the set of coefficients found when computing approximations for the vectors in the training set. The iteration can be summarized as:

1. $\mathbf{F}$ and $\mathbf{x}_l,\ l = 1, 2 \ldots M$ are given. Find $\mathbf{w}_l,\ l = 1, 2 \ldots M$ by using a vector selection algorithm.

2. $\mathbf{w}_l$ and $\mathbf{x}_l,\ l = 1, 2 \ldots M$ are given. Find the best possible $\mathbf{F}$, and normalize the frame vectors.

As in a shape-gain VQ, the frame vectors are normalized, thus they represent shape. The corresponding coefficients represent the gain. Finding approximations for the training vectors in a frame based system is done by using a suboptimal vector selection algorithm. Thus, as opposed to the GLA, there is no guarantee for the next frame to be better than the previous after an iteration. Finding the best possible frame when $\mathbf{w}_l$ and $\mathbf{x}_l,\ l = 1, 2 \ldots M$ are known is also a much more complicated task than using the centroid conditions as done in the GLA. To solve this problem, we propose an algorithm to find the optimal $\mathbf{F}$ in terms of MSE, when $\mathbf{w}_l$ and $\mathbf{x}_l,\ l = 1, 2 \ldots M$ are known, or estimated. Let $\mathbf{x}_l$ be approximated using a vector selection algorithm:

$$\hat{\mathbf{x}}_l = \sum_{j=1}^{K} w_l(j)\mathbf{f}_j = \mathbf{F}\mathbf{w}_l \tag{4.1}$$

where $w_l(j)$ is the coefficient corresponding to vector $\mathbf{f}_j$. The coefficient vector $\mathbf{w}_l$ is sparse, i.e. only a few of the $w_l(j)$'s are different from zero. The residual is:

$$\mathbf{r}_l = \mathbf{x}_l - \hat{\mathbf{x}}_l, \tag{4.2}$$

The idea is now to adjust all frame vectors in such a manner that the total MSE, given by

$$\sum_l \|\mathbf{r}_l\|^2, \tag{4.3}$$

becomes as small as possible. Denote by $\boldsymbol{\delta}_j$ the adjustment of frame vector $\mathbf{f}_j$:

$$\tilde{\mathbf{f}}_j = \mathbf{f}_j + \boldsymbol{\delta}_j, \quad j = 1, 2 \ldots K. \tag{4.4}$$

In the following we show how to find the optimal adjustment vectors $\boldsymbol{\delta}_j,\ j = 1, 2, \ldots K$. Since we find the optimal directions in Equation 4.4, we call the frame design algorithm *the Method of Optimal Directions (MOD)*. The new residual for a training vector $\mathbf{x}_l$ is:

$$\tilde{\mathbf{r}}_l = \mathbf{r}_l - \sum_{j=1}^{K} w_l(j)\boldsymbol{\delta}_j, \tag{4.5}$$

where $w_l(j)$ is the existing coefficient corresponding to the approximation of training vector $\mathbf{x}_l$. A reduction of the total MSE over all training vectors is desired:

$$\sum_l \|\tilde{\mathbf{r}}_l\|^2 \leq \sum_l \|\mathbf{r}_l\|^2. \tag{4.6}$$

The resulting MSE after adjusting the frame vectors is investigated:

$$\begin{aligned} \sum_l \|\tilde{\mathbf{r}}_l\|^2 &= \sum_l \|\mathbf{r}_l - \sum_{j=1}^{K} w_l(j)\boldsymbol{\delta}_j\|^2 & (4.7)\\ &= \sum_l (\mathbf{r}_l - \sum_{j=1}^{K} w_l(j)\boldsymbol{\delta}_j)^T(\mathbf{r}_l - \sum_{j=1}^{K} w_l(j)\boldsymbol{\delta}_j) & (4.8) \end{aligned}$$

$$= \sum_l \|\mathbf{r}_l\|^2 - 2\sum_l \sum_{j=1}^{K} w_l(j)\boldsymbol{\delta}_j^T\mathbf{r}_l + \sum_l \sum_{j=1}^{K}\sum_{k=1}^{K} w_l(j)w_l(k)\boldsymbol{\delta}_j^T\boldsymbol{\delta}_k. \tag{4.9}$$

If Equation 4.6 is satisfied, then:

$$\sum_{j=1}^{K}\sum_{k=1}^{K} a_{jk}\boldsymbol{\delta}_j^T\boldsymbol{\delta}_k - 2\sum_{j=1}^{K}\boldsymbol{\delta}_j^T\mathbf{b}_j \leq 0 \tag{4.10}$$

where

$$\begin{aligned} a_{jk} &= \sum_{l=1}^{M} w_l(j)w_l(k) & (4.11)\\ \mathbf{b}_j &= \sum_{l=1}^{M} w_l(j)\mathbf{r}_l. & (4.12) \end{aligned}$$

We want to find the minimum of $\sum_l \|\tilde{\mathbf{r}}_l\|^2$, and this is equivalent to finding the minimum of the left side of Equation 4.10:

$$\frac{\partial}{\partial\delta_q(p)}\left(\sum_{j=1}^{K}\sum_{k=1}^{K}\sum_{i=1}^{N} a_{jk}\delta_j(i)\delta_k(i) - 2\sum_{j=1}^{K}\sum_{i=1}^{N}\delta_j(i)b_j(i)\right) = 0, \tag{4.13}$$

where $q = 1, 2 \ldots K$, and $p = 1, 2 \ldots N$. After some manipulations we get:

$$\sum_{j=1}^{K} a_{jq}\delta_j(p) - b_q(p) = 0. \tag{4.14}$$

This can be written in matrix form. Let

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1K} \\ a_{21} & a_{22} & & \\ \vdots & & \ddots & \\ a_{K1} & & & a_{KK} \end{bmatrix} = \sum_{l=1}^{M} \mathbf{w}_l \mathbf{w}_l^T = M\tilde{\mathbf{R}}_{ww} \tag{4.15}$$

$$\begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \dots \mathbf{b}_K \end{bmatrix} = \sum_{l=1}^{M} \mathbf{r}_l \mathbf{w}_l^T = M\tilde{\mathbf{R}}_{rw} \tag{4.16}$$

where $\tilde{\mathbf{R}}_{ww}$ and $\tilde{\mathbf{R}}_{rw}$ are the estimated auto-correlation matrix of $\mathbf{w}$, and the estimated cross-correlation matrix of $\mathbf{r}$ and $\mathbf{w}$, respectively. Equation 4.14 becomes:

$$M\tilde{\mathbf{R}}_{ww}\boldsymbol{\Delta}^T = M\tilde{\mathbf{R}}_{rw}^T. \tag{4.17}$$

where:

$$\boldsymbol{\Delta} = \begin{bmatrix} \boldsymbol{\delta}_1 & \dots & \boldsymbol{\delta}_K \end{bmatrix}. \tag{4.18}$$

The $\boldsymbol{\Delta}$ matrix contains the optimal adjustment vectors. According to Equation 4.11, $\tilde{\mathbf{R}}_{ww}$ is symmetric.

Assuming $\tilde{\mathbf{R}}_{ww}$ to have full rank, we get:

$$\boldsymbol{\Delta} = \tilde{\mathbf{R}}_{rw}\tilde{\mathbf{R}}_{ww}^{-1}. \tag{4.19}$$

The new frame can consequently be written:

$$\tilde{\mathbf{F}} = \mathbf{F} + \tilde{\mathbf{R}}_{rw}\tilde{\mathbf{R}}_{ww}^{-1} \tag{4.20}$$

and this is shown in Appendix A.1 to be equivalent to:

$$\tilde{\mathbf{F}} = \tilde{\mathbf{R}}_{xw}\tilde{\mathbf{R}}_{ww}^{-1}, \tag{4.21}$$

where $\tilde{\mathbf{R}}_{xw}$ is the estimated cross-correlation matrix between the signal vectors $\mathbf{x}_l$ and the coefficient vectors $\mathbf{w}_l$. $\sum_l \|\tilde{\mathbf{r}}_l\|^2$ can not be less than 0, thus we know that the problem has a minimum solution. Since Equation 4.19 has only one solution when $\tilde{\mathbf{R}}_{ww}$ is full rank, this is the minimum solution.

For each iteration, if the frame is adjusted according to Equation 4.20 this gives the optimal improvement in MSE for the existing vector selection and corresponding coefficients.

We have now focused on part 2 in the training algorithm. If an optimal selection algorithm had been used in the part 1, the new frame would always be better than, or as good as, the previous one, with respect to MSE. Selection algorithms for frames are not optimal, so there is no way to guarantee a better frame when using a practical selection algorithm, but test results show that this scheme works remarkably well and produces frames that are well suited for a given class of input data. Let $\mathbf{F}^{(k)}$ be the frame after k iterations. In summary, the algorithm for frame design works as follows:

1. Begin with an initial frame $\mathbf{F}^{(0)}$ of size $N \times K$, Assign counter variable $k = 1$.
2. A vector selection algorithm is used to find an approximation for each training vector, and all the residuals are calculated.
3. The frame is adjusted according to Equation 4.20. The frame vectors are then normalized to unit length $\Rightarrow \mathbf{F}^{(k)}$.
4. A vector selection algorithm is used to find the new approximations and residuals.
   If (stop-criterion = FALSE) $\Rightarrow k = k + 1$, go to step 3, else terminate.

Several stop-criteria can be used; for example maximum number of iterations or almost constant MSE. The convergence properties are not yet fully understood. Due to the lack of guarantee for the new frame to be better than, or as good as, the previous, the algorithm should allow the MSE to grow for several iterations without terminating the training. This can be seen from training results in Chapter 5.

4.2 Frame design from a probabilistic point of view

The objective of this section is to get some more insight into the problem of frame design by looking at it from a probabilistic point of view. A further study of this topic can be found in [40, 41, 42, 43].

In Section 4.1 the frame design algorithm Method of Optimal Directions was developed using training data without the use of statistics. In this section we will consider frame design from a probabilistic point of view, and show the interesting fact that the same algorithm, the MOD, can result using a probabilistic way of thinking. This gives additional insight and understanding of the MOD algorithm, and also opens for other frame design possibilities. There are

several possible approaches. One is to assume that the frame is deterministic but unknown, another is to assume that the frame is stochastic. Both these approaches are investigated.

We start with a signal model:

$$\mathbf{x} = \mathbf{F}\mathbf{w} + \mathbf{n} = \sum_{j=1}^{K} w_j \mathbf{f}_j + \mathbf{n}, \tag{4.22}$$

where as before $\mathbf{F}$ is a frame: an $N \times K$ matrix where $K \geq N$ and $\text{rank}(\mathbf{F}) = N$. $\mathbf{x}$ is the observed (known) data vector, $\mathbf{x} \in \mathbf{R}^N$ and $\mathbf{X}^M = (\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_M)$ is an observation set. In this section we assume that the data are constructed from a structure as in Equation 4.22. When constructing a observation set, the frame $\mathbf{F}$ is assumed to be constant, either if it is a constant deterministic (but unknown) parameter or if it is *one* realization of a stochastic process (still unknown). Different realizations of the vectors $\mathbf{w}$ and $\mathbf{n}$ gives different observations $\mathbf{x}$. $\mathbf{n}$ is assumed to be a random additive noise vector with the pdf $p_{\mathbf{n}}(\mathbf{n})$. Furthermore, $\mathbf{w}$ is also assumed to be a random vector independent of $\mathbf{n}$. $\mathbf{W}^M = (\mathbf{w}_1, \mathbf{w}_2 \dots \mathbf{w}_M)$ is the true coefficient set that made the observation set together with the frame and the additive noise. $\hat{\mathbf{W}}^M = (\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2 \dots \hat{\mathbf{w}}_M)$ is the estimated coefficient set, estimated from the observation set.

4.2.1 Deterministic but unknown frame

If $\mathbf{F}$ is deterministic but unknown, it would be desirable to find the Maximum Likelihood (ML) estimate, since this is known to hold good qualities. The ML estimate is defined as:

$$\hat{\mathbf{F}}_{ml}(\mathbf{x}) = \arg\max_{\mathbf{F}} p(\mathbf{X}^M; \mathbf{F}). \tag{4.23}$$

The ML estimate is an estimate of an unknown but deterministic parameter. The ML estimate for a parameter is the estimate that makes the *given* value of the observation set the *most likely value*.

If the observation data are assumed to be iid, the ML estimate becomes:

$$\hat{\mathbf{F}}_{ml}(\mathbf{x}) = \arg\max_{\mathbf{F}} \prod_{j=1}^{M} p(\mathbf{x}_j; \mathbf{F}),$$

where:

$$p(\mathbf{x}_j; \mathbf{F}) = \int p(\mathbf{x}_j, \mathbf{w}; \mathbf{F}) d\mathbf{w} = \int p(\mathbf{x}_j | \mathbf{w}; \mathbf{F}) \cdot p_{\mathbf{w}}(\mathbf{w}) d\mathbf{w} \tag{4.24}$$

$p(\mathbf{x}_j|\mathbf{w};\mathbf{F}) = p(\mathbf{F}\mathbf{w}+\mathbf{n}_j|\mathbf{w};\mathbf{F})$ where $\mathbf{n}_j$ is the only random variable since $\mathbf{w}$ is given. This can thereby be written as $p_{\mathbf{n}}(\mathbf{n}_j)$, where $\mathbf{n}_j = \mathbf{x}_j - \mathbf{F}\mathbf{w}$. Thus Equation 4.24 becomes:

$$p(\mathbf{x}_j;\mathbf{F}) = \int p_{\mathbf{n}}(\mathbf{x}_j - \mathbf{F}\mathbf{w}) \cdot p_{\mathbf{w}}(\mathbf{w}) d\mathbf{w}. \tag{4.25}$$

The integral of Equation 4.25 is in general hard to solve. Different approaches have been proposed to deal with this difficulty. In [49] an assumption on the probability of the coefficients, $p_{\mathbf{w}}(\mathbf{w})$, is made so that the integral can be solved analytically. The assumption made is that the coefficients are independent and follow a Laplacian distribution, the latter to ensure sparseness of the coefficients. The results were not too convincing, and the authors claim that the assumptions on the $p_{\mathbf{w}}(\mathbf{w})$ probably can be partially responsible.

In [57] Olshausen and Field make the assumption that $p_{\mathbf{n}}(\mathbf{x}_j - \mathbf{F}\mathbf{w}) \cdot p_{\mathbf{w}}(\mathbf{w})$ has a fairly tightly peaked maximum in $\mathbf{w}$-space, thus the integral of Equation 4.25 can be approximated by evaluating the argument of the integral only at its maximum. This way they split up the problem in two to make it easier to solve. This has similarities to the GLA way of thinking, where a complicated problem is divided in two to make it easier to solve by iterating, but it can only guarantee local minima. We do something similar here, and call the estimate an Approximate Maximum Likelihood (AML). Assuming a current estimate $\hat{\mathbf{F}}$, for $\mathbf{F}$:

$$\hat{\mathbf{w}}(\hat{\mathbf{F}}) = \arg\max_{\mathbf{w}} p(\mathbf{x}_j, \mathbf{w}; \hat{\mathbf{F}}), \tag{4.26}$$

gives a estimation for the set $\mathbf{W}^M$, $\hat{\mathbf{W}}^M$. This matrix is now used as if it were known data:

$$\begin{aligned}
\hat{\mathbf{F}}_{aml}(\mathbf{x}) &= \arg\max_{\mathbf{F}} \prod_{j=1}^{M} p(\mathbf{x}_j;\mathbf{F},\hat{\mathbf{w}}_j) \\
&= \arg\max_{\mathbf{F}} \prod_{j=1}^{M} p_{\mathbf{n}}(\mathbf{x}_j - \mathbf{F}\hat{\mathbf{w}}_j) \\
&= \arg\min_{\mathbf{F}} \langle \|\mathbf{x} - \mathbf{F}\hat{\mathbf{w}}\|^2 \rangle_M,
\end{aligned} \tag{4.27}$$

where $\langle\cdot\rangle$ means expectation in the case of statistical variables, and mean (estimated expectation) in the case of a data set. The last step is due to the assumption that $p_{\mathbf{n}}(\mathbf{n})$ is Gaussian with iid elements and zeros mean:

$$p_{\mathbf{n}}(\mathbf{n}) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} e^{-\sum \frac{n_i^2}{2\sigma^2}} = Ce^{-\frac{\|\mathbf{n}\|^2}{2\sigma^2}}, \tag{4.28}$$

and by use of the equality :

$$\arg\max_a(g(a)) = \arg\min_a(-\ln g(a)). \tag{4.29}$$

The optimization problem of Equation 4.27 can be solved by taking the derivative with respect to $\mathbf{F}$ equal to zero:

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{F}}\langle\|\mathbf{x}-\mathbf{F}\hat{\mathbf{w}}\|^2\rangle_M &= \\
\langle\frac{\partial}{\partial \mathbf{F}}\|\mathbf{F}\hat{\mathbf{w}}\|^2 - 2\frac{\partial}{\partial \mathbf{F}}\mathbf{x}^T\mathbf{F}\hat{\mathbf{w}}\rangle_M &= \\
\langle 2\mathbf{F}\hat{\mathbf{w}}\hat{\mathbf{w}}^T - 2\mathbf{x}\hat{\mathbf{w}}^T\rangle_M &= \\
2\mathbf{F}\tilde{\mathbf{R}}_{\hat{w}\hat{w}} - 2\tilde{\mathbf{R}}_{x\hat{w}} &= 0
\end{aligned}$$

This gives the following solution for $\mathbf{F}$:

$$\hat{\mathbf{F}}_{aml} = \tilde{\mathbf{R}}_{x\hat{w}}\tilde{\mathbf{R}}^{-1}_{\hat{w}\hat{w}}, \tag{4.30}$$

where $\tilde{\mathbf{R}}_{\hat{w}\hat{w}}$ is the estimated auto-correlation matrix for $\hat{\mathbf{w}}$ and $\mathbf{R}_{x\hat{w}}$ is the estimated cross-correlation matrix between the signal vector $\mathbf{x}$ and the coefficient vector $\hat{\mathbf{w}}$:

$$\begin{aligned}
\tilde{\mathbf{R}}_{\hat{w}\hat{w}} &= \frac{1}{M}\sum_{l=1}^{M}\hat{\mathbf{w}}_l\hat{\mathbf{w}}_l^T \\
\tilde{\mathbf{R}}_{x\hat{w}} &= \frac{1}{M}\sum_{l=1}^{M}\mathbf{x}_l\hat{\mathbf{w}}_l^T.
\end{aligned}$$

Equation 4.30 can be seen to be the exact same solution as Equation 4.21, thus the AML estimate is the MOD. The goodness of the AML estimate, or the MOD, is of course dependent of how good the estimate of the coefficient vector set, $\hat{\mathbf{W}}^M$, is. It turns out that performing the optimization required in Equation 4.26 is nontrivial. It is the same NP complete vector selection problem as before, and can be estimated by using a vector selection technique like MP, OMP, FOCUSS etc.

These equations, which are identical to the MOD equations, were developed using some assumptions. The noise vector is assumed to have iid elements that are Gaussian distributed with zero mean. The other assumption is that the approximation done by dividing the problem in two and iterate is a fairly good approximation, but this kind of iterative algorithm only guarantee a local optimum.

4.2.2 Stochastic frame

In this section, $\mathbf{F}$ is considered to be a random variable. Both $\mathbf{w}$ and $\mathbf{n}$ are still considered random variables, and $\mathbf{X}^M$ is the observed data set. Let the elements in the set $\mathbf{W}^M$ be mutually independent, and assume that the set $\mathbf{W}^M$ is independent of the frame $\mathbf{F}$. The latter assumption may seem unreasonable, especially if we think in terms of $\mathbf{Fw}$ being a representation of $\mathbf{x}$, like we do in many cases like compression. On the other hand, if we assume that $\mathbf{x}$ is produced from a true underlying sparse structure of the form $\mathbf{Fw}+\mathbf{n}$, like we do here, it may be very reasonable to assume independence between $\mathbf{F}$ and $\mathbf{w}$.

It is now desirable to find the Maximum A Posteriori (MAP) estimate of both the frame $\mathbf{F}$ and the coefficient vector set $\mathbf{W}^M$:

$$\{\hat{\mathbf{F}}_{map}, \hat{\mathbf{W}}^M_{map}\} = \arg\max_{\mathbf{F},\mathbf{W}^M} p(\mathbf{F}, \mathbf{W}^M|\mathbf{X}^M).$$

$$\begin{aligned} p(\mathbf{F}, \mathbf{W}^M|\mathbf{X}^M) &= \frac{1}{p(\mathbf{X}^M)} p(\mathbf{X}^M|\mathbf{F}, \mathbf{W}^M) p(\mathbf{F}, \mathbf{W}^M) \\ &= \frac{1}{p(\mathbf{X}^M)} p(\mathbf{X}^M|\mathbf{F}, \mathbf{W}^M) p(\mathbf{F}) p(\mathbf{W}^M) \end{aligned} \tag{4.31}$$

since $\mathbf{F}$ and $\mathbf{W}^M$ are assumed independent. By use of Equation 4.29 and the fact that the denominator in Equation 4.31 is not dependent on $\mathbf{F}$ and $\mathbf{W}^M$, the MAP becomes:

$$\begin{aligned} \{\hat{\mathbf{F}}_{map}, \hat{\mathbf{W}}^M_{map}\} &= \arg\max_{\mathbf{F},\mathbf{W}^M} p(\mathbf{X}^M|\mathbf{F}, \mathbf{W}^M) p(\mathbf{F}) p(\mathbf{W}^M) \\ &= \arg\min_{\mathbf{F},\mathbf{W}^M} \{-\ln p(\mathbf{X}^M|\mathbf{F}, \mathbf{W}^M) - \ln p(\mathbf{F}) - \ln p(\mathbf{W}^M)\} \\ &= \arg\min_{\mathbf{F},\mathbf{W}^M} \{-\ln p_{\mathbf{n}}(\mathbf{X}^M - \mathbf{FW}^M) - \ln p(\mathbf{F}) - \ln p(\mathbf{W}^M)\} \end{aligned}$$

The noise vector, $\mathbf{n}$ is assumed to have iid elements with normal distribution and zero mean. By the use of Equation 4.28 the estimates can be written:

$$\{\hat{\mathbf{F}}_{map}, \hat{\mathbf{W}}^M_{map}\} = \arg\min_{\mathbf{F},\mathbf{W}^M} \langle \|\mathbf{x} - \mathbf{Fw}\|^2 - \beta \cdot \ln p(\mathbf{F}) - \beta \cdot \ln p(\mathbf{w}) \rangle_M. \tag{4.32}$$

Some assumptions have to be made to be able to continue. Let $\mathbf{F}$ be bounded, so that $\|\mathbf{F}\|$ is constant. $\mathbf{F}$ is assumed to be uniformly distributed on the $\mathbf{R}^{N\times K}$ space, within the limits caused by the bounding of $\mathbf{F}$. Then $p(\mathbf{F})$ is some kind of constant, not dependent on $\mathbf{F}$ or $\mathbf{W}$, and in this case it can be

removed from Equation 4.32. Let $(-\beta \ln p(\mathbf{w})) = \lambda f(\mathbf{w})$, the MAP estimates can be written as:

$$\{\hat{\mathbf{F}}_{map}, \hat{\mathbf{W}}_{map}\} = \arg \min_{\mathbf{F} \in \mathcal{F}, \mathbf{W}_M} \langle \|\mathbf{x} - \mathbf{F}\mathbf{w}\|^2 + \lambda f(\mathbf{w}) \rangle_M \tag{4.33}$$

This is a very hard problem to solve.

4.2.3 A less stringent estimation approach

In this section the problem of finding the MAP estimate is split up so that it gets easier to solve. The only thing that is known is the observed set of data vectors, $\mathbf{X}^M$, and from this it is desired to find both a Frame, $\mathbf{F}$, and a set of coefficient vectors, $\mathbf{W}^M$. It is desirable to find a solution that has a small error or noise, this means that

$$\langle \|\mathbf{n}\|^2 \rangle = \langle \|\mathbf{x} - \mathbf{F}\mathbf{w}\|^2 \rangle_M,$$

should be small. At the same time the coefficient vectors, $\mathbf{w}_j$, should be sparse so that the representation is as efficient as possible. The sparsity measure:

$$d(\mathbf{w}) = \text{sgn}(p) \sum_{j=1}^{K} |w_j|^p, \quad p \leq 1, \tag{4.34}$$

discussed in Chapter 3 is a good indicator of the sparsity, and is used here.

The optimization problem that needs to be solved can be written:

$$\arg \min_{\mathbf{F}, \mathbf{W}^M} \langle \|\mathbf{x} - \mathbf{F}\mathbf{w}\|^2 + \lambda d(\mathbf{w}) \rangle_M \tag{4.35}$$

which is similar to the problem of the MAP estimates in Equation 4.33. This is a very hard problem to solve, and to make it easier it is split up:

$$\arg \min_{\mathbf{F}} \langle \arg \min_{\mathbf{W}^M} \langle \|\mathbf{x} - \mathbf{F}\mathbf{w}\|^2 + \lambda d(\mathbf{w}) \rangle_M \rangle_M. \tag{4.36}$$

In practice this means that first the best possible coefficient set, $\mathbf{W}^M$, is found using an estimated $\mathbf{F}$ as a known parameter, and second the best estimate of $\mathbf{F}$ is found using the set $\mathbf{W}^M$ as known parameters. Thereby an iterative algorithm results:

1. Let $\mathbf{F}$ be a known parameter

$$\arg \min_{\mathbf{W}^M} \langle \|\mathbf{x} - \mathbf{F}\mathbf{w}\|^2 + \lambda d(\mathbf{w}) \rangle_M, \tag{4.37}$$

2. Let $\mathbf{W}_M$ be known parameters

$$\arg\min_{\mathbf{F}} \langle \|\mathbf{x} - \mathbf{F}\mathbf{w}\|^2 + \lambda d(\mathbf{w}) \rangle_M = \arg\min_{\mathbf{F}} \langle \|\mathbf{x} - \mathbf{F}\mathbf{w}\|^2 \rangle_M \qquad (4.38)$$

The two steps in the algorithm are now investigated closer. Starting with step 1, the optimization problem is to find the best coefficient vector so that the error is minimized *and* the sparsity is maximized. In general this is the NP-complete problem discussed in Chapter 2.4 which the different vector selection algorithms give suboptimal solutions for. It can be shown, however, that if some assumptions are fulfilled, FOCUSS will give the true optimal solution to this problem. If the elements in the coefficient vector $\mathbf{w}$ are independent and generalized Gaussian distributed, the logarithm of the distribution of the coefficient vector is equal to the sparsity measure from Equation 4.34, thus Equation 4.33 and Equation 4.35 are equivalent. In Chapter 3.1 it was shown how FOCUSS gives the optimal solution to Equation 4.37 when the elements in the coefficient vector are assumed iid and have a generalized Gaussian distribution.

If the coefficients do not have this distribution, the optimization problem of Equation 4.37 is hard to solve, and a vector selection algorithm like FOCUSS or MP techniques can be used to give a good suboptimal solution.

Equation 4.38 in step 2 can be solved by taking the derivative with respect to $\mathbf{F}$ equal to zero. This is equivalent to what is done with Equation 4.27 in Section 4.2.1, and the results are the same as shown in Equation 4.30. And again, this is equivalent to the MOD algorithm.

Chapter 5

Approximation using frames

As described in Section 2.3, frames can be used for compression purposes. The idea is to approximate a signal vector using a fixed frame and a *sparse* coefficient vector, i.e. a sparse representation. This chapter deals with frame design using the MOD algorithm presented in Chapter 4, and the approximation capabilities for the frames.

The objective was to investigate the approximation capabilities of frames designed using MOD for different classes of signals. In order to achieve this objective we devised a number of experiments presented in Table 5.

No.	Initial frame	Signal class	Frame size	Vector selection method	Approach to ensure sparsity
1	ad hoc	ECG	16×41	FOMP	Sparsity criterion
	ad hoc	Speech	16×32	FOMP	Sparsity criterion
2	training vec.	ECG	32×64	OMP	Sparsity criterion
	training vec.	Speech	32×64	OMP	Sparsity criterion
3	training vec.	ECG	32×64	OMP	MSE limit
	training vec.	ECG	32×64	FOCUSS[1]	SNR target
4	training vec.	images[2]	64×64	FOMP	sparsity criterion
	training vec.	images	64×128	FOMP	sparsity criterion
5	training vec.	images	64×128	FOMP	MSE limit

Table 5.1: Different experiments where frames are designed and the approximation capabilities tested and compared to DCT.

[1] Regularized FOCUSS with the regularization parameter decided by the modified L-curve method, as described in Chapter 3.

[2] 8×8 image blocks are formed into image vectors with dimension 64.

The frame design algorithm is applied to ECG, speech signals and images. Training results from the experiments are shown, and the approximation capabilities of the trained frames are demonstrated on test signals and compared to the approximation capability of the DCT.

The first experiment uses frames designed in an ad hoc manner as initial frames. This requires some knowledge about the signal. Using the MOD as frame design algorithm we can only guarantee a local optimum, thus the initial frame will influence on the resulting frame. When having some knowledge about the signal we can choose a good initial frame. In the latter experiments normalized vectors from the training set are used to constitute an initial frame. This way no prior knowledge is needed, and the algorithm constructs an initial frame from the training set.

Different ECG signals were used in training as well as testing. For diagnostic purposes it can sometimes be necessary to continuously record the heart beat of a person during a long time period (weeks). In situations like this it would be natural to train the compression scheme for that person before using it. In other applications a more general system that can be used on different persons is needed. On other signal classes similar issues may occur. Therefore we have done experiments covering both these situations. The image experiments are trained on a set of images and tested on images *not* included in the training set.

In experiment no. 1 the block size is set to 16. This is because here the frame consists of ad hoc made frame vectors, reflecting typical signal segments, and a large block size may be less practical. In experiments 2 and 3 a block size of 32 is used because longer block size gives better results since there is a correlation between signal samples. In experiment 4 and 5 image blocks of 8×8 are used. This is a typical block size for image coding, and a reasonable size for frame based approximation.

In experiment 1, 4, and 5 FOMP is used as the vector selection algorithm. In experiment 2 OMP is used, and in experiment 3 both OMP and regularized FOCUSS are used as vector selection algorithms. Our frame design algorithm can be used in conjunction with any vector selection algorithm, and experiment 2 and 3 might get slightly better results using FOMP.

The approach to ensure sparsity used in experiment 1, 2 and 4 is to impose a sparsity criterion allowing a predetermined number of nonzero entries in the coefficient vector. In 3 however we wanted to use the regularized FOCUSS described in Chapter 3. The regularized FOCUSS requires a target on the overall SNR. We do a similar experiment with OMP for comparison, and use a limit on MSE. In 5 an MSE_{limit} limits the number of iterations, and FOMP

is the vector selection algorithm. Using an MSE_{limit} instead of a sparsity criterion allows flexibility in the number of nonzero entries from signal vector to signal vector and provides a more stable MSE throughout the signal.

Regularized FOCUSS is a parallel vector selection algorithms and very computationally expensive. The practical signal vector sizes using such algorithms are therefore limited, and this makes regularized FOCUSS impractical to use for image representation.

In our training and testing experiments we use:

$$\text{ND} = \frac{\text{RMSE}}{\sigma_x} = \frac{\sqrt{\text{MSE}}}{\sigma_x} \tag{5.1}$$

as the normalized distortion (ND) measure, where σ_x is the power of the signal calculated over the entire signal used in the training or testing experiment, and RMSE is the Root Mean Squared Error.

5.1 Approximation capabilities for ECG and speech signals

Representation and compression of ECG signals play an important role in this thesis. ECG signals are used for monitoring patients, but also for diagnostic purposes. Medical doctors may also be able to follow a disease development having access to ECG records from different stages of the disease. In this case it might be useful to have records of ECG signals from different stages in a disease or at different points in a person's life. Maybe in the future everybody will have a piece of ECG recording in an archive for comparison purposes in case of diseases later in life.

Long term collection of ECG data can be a subject when diagnosing patients with irregular heart rhythms [65]. In cases like this the patient might need to wear the recording and storage device 24 hour a day for several days. A typical ECG signal is sampled at 360 Hz with 12 bits per sample. That gives 356 Mbit in 24 hours. Since the size of such a carry-on device has to be small, the need for efficient representation is obvious. While transporting a patient to a hospital it may be an advantage if a wireless transfer of ECG signals from the patient to the hospital is in operation. Diagnosing could be executed before the patient arrives. The transfer of ECG records from a previously used hospital or an archive to a treating hospital is another example of ECG transmission.

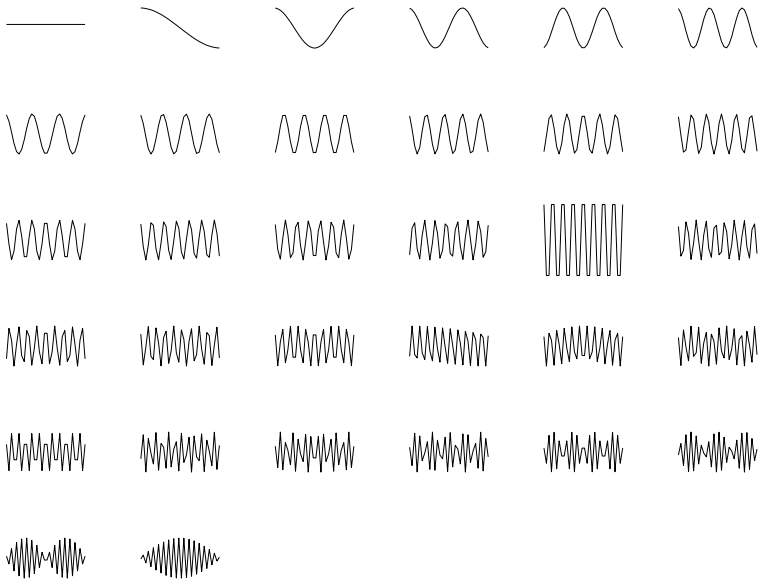


Figure 5.1: Basis vectors of the 32×32 DCT transform.

In these transmission and storage examples compression is useful. Dealing with medical signals the question arises of whether the quality after lossy compression is acceptable. Our guess is that for some purposes good quality lossy compression can be used without problems whereas for other purposes it might not be good enough. We have not investigated this any further.

This section concerns approximation capabilities for ECG and Speech signals when using frames designed with the MOD algorithm. The experiments 1,2, and 3 from Table 5 are presented here with figures and results from the frame design, and also test results on approximation capabilities.

Our test experiments are compared to approximation capabilities for signal representations using the DCT. The basis vectors in the DCT of dimension 32 are plotted in Figure 5.1.

5.1.1 ECG and speech signals used

The ECG signals used are signals from the MIT arrhythmia database [53]. The records are represented with 12 bits per sample, and the sampling frequency is 360 Hz. The ECG signals used for training are MIT100, 0:00 to 5:00 minutes, and MIT207, 6:00 to 11:00 minutes, called MIT100_{train} and MIT207_{train}

respectively. Training is also done on a mixed signal, MIT_{mix}. MIT_{mix} is constructed from the following signal segments: MIT100 0:00 to 2:00, MIT103 03:25 to 05:25, MIT113 0:00 to 2:00, MIT207 06:00 to 08:00, and MIT217 0:00 to 2:00, thus MIT_{mix} is 10 minutes of data from 5 different patients. The ECG signals used for testing are MIT100, 5:30 to 10:30 minutes, MIT113, 0:00 to 0:30 minutes, and MIT207, 12:00 to 17:00 minutes, called MIT100_{test}, MIT113_{test}, and MIT207_{test} respectively.

The speech signals used are recorded at 16 kHz in a room without echo, and down-sampled to 8 kHz. The training set, Speech_{train}, consists of 8.75 seconds of speech data. Another 8.75 seconds segment of speech data is used for testing, Speech_{test}.

Small samples of some of the signals used are given in Figure 5.2. a) shows 20000 samples of Speech_{train}. The following plots show 2000 samples of b) MIT100_{train}, c) MIT207_{train}, d) MIT101, e) MIT103, and f) MIT217.

5.1.2 Experiment no. 1 - Ad hoc designed initial frames

The vector selection algorithm used in these experiments is FOMP, described in Section 2.4.1.

Experiments on improving ad hoc designed frames was done on both ECG and speech signals, with a block size of $N = 16$. Constructing a frame by using segments of a typical signal in combination with DCT basis vectors was shown in [15] to work quite well on ECG signals. In [32] the possibility of adapting the frame by augmenting it with samples from the source is mentioned, but not tried. In the experiment using ECG signals as the training set, the initial frame is composed of DCT vectors in addition to vectors constructed using typical QRS complexes (heartbeats in a normal sinus rhythm). The frame is almost the same as the ad hoc based frame presented in [15], and it consists of the 7 first DCT vectors and 34 ad hoc vectors made to match QRS complexes in typical ECG signals, i.e. the initial frame has size 16×41.

In [6] encouraging results are obtained using a frame with DCT and Haar vectors for speech signals. We therefore use this frame as initial frame in the speech signal experiment. With the chosen block size of 16, we have 32 frame vectors. Since both the DCT and Haar transform contains a Direct Current (DC) vector, one of them is replaced with a normalized random vector.

Figure 5.3 shows how the normalized distortion develops as we iterate for the training experiments with ad hoc designed frames as the initial frames. The training signals used where Speech_{train} and MIT100_{train}.

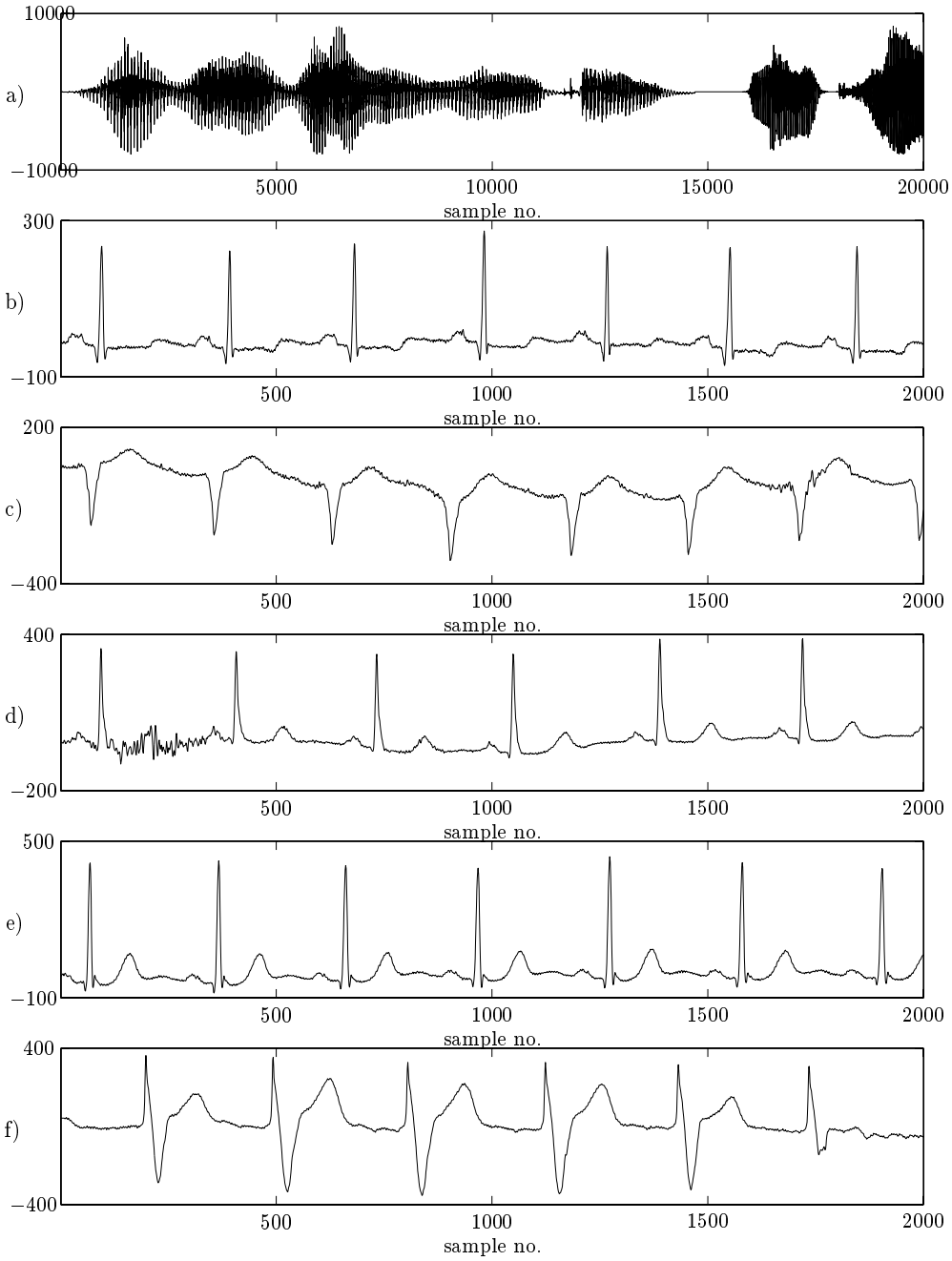


Figure 5.2: a) Segment from the speech signal used in training experiments. Segments of different ECG signals are showed in: b) MIT100$_{train}$, c) MIT207$_{train}$, d) MIT101, e) MIT103, and f) MIT217

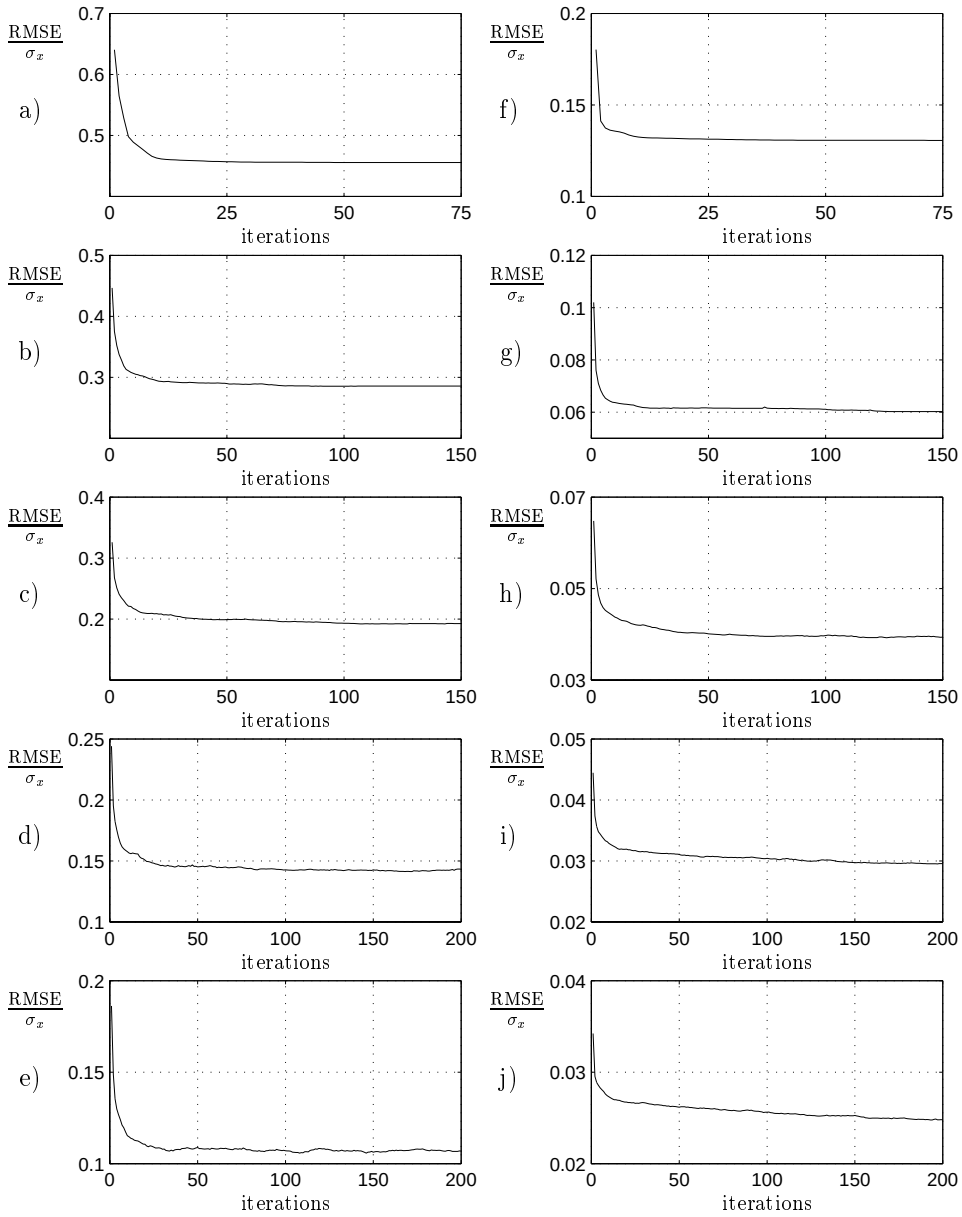


Figure 5.3: The normalized distortion is plotted as a function of training iterations. In a), b), c), d), and e) Speech$_{train}$, and 1,2,3,4, and 5 frame vectors are used in each approximation, respectively. Initial frame: DCT *and* Haar. In f), g), h), i), and j) MIT100$_{train}$, and 1,2,3,4, and 5 frame vectors are used in each approximation, respectively. Initial frame: 7 DCT vectors *and* ad hoc designed vectors.

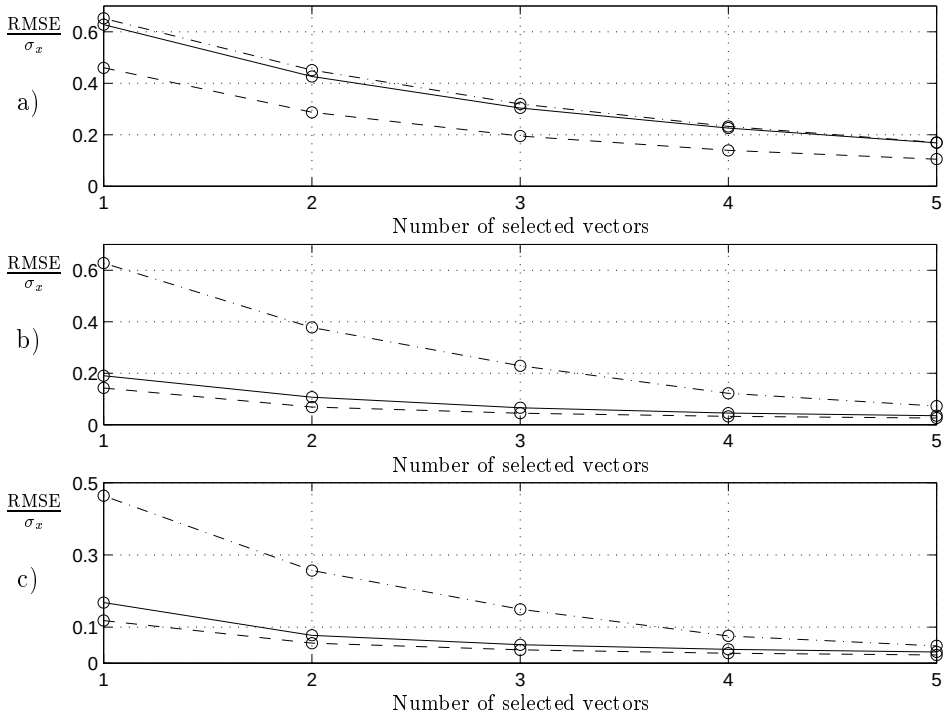


Figure 5.4: Testing approximation capabilities of MOD designed frames. The normalized distortion is plotted as a function of different numbers of vectors in an approximation. **Dash-dot**: DCT, **solid**: initial frame, **dashed**: optimized frames. a) Speech$_{test}$, b) MIT100$_{test}$, c) MIT113$_{test}$.

These frame sets are tested on test signals. The frames optimized for ECG signals are tested on MIT100$_{test}$ and MIT113$_{test}$. The frames optimized for speech signals are tested on Speech$_{test}$.

In Figure 5.4 the test results using the optimized frames and the initial frames are shown together with test results using DCT. The comparison in Figure 5.4 shows that the improvement using the optimized frames, with respect to the normalized distortion, is significant both compared to the DCT and the initial ad hoc designed frames.

For the experiment with speech signal the reduction in normalized distortion using the optimized frame compared to the initial frame when using 1,2, 3, 4, and 5 vectors in the approximation are 26.7%, 32.6%, 35.9%, 38.2%, and 37.8%. The initial frame was the ad hoc designed frame made from both DCT and Haar transform used by other authors, [6], with good results. For the

ECG experiments the improvement is largest when using MIT100_{test}. This is not surprising since the MIT100_{test} is data from the same patient as the training set. The MIT113_{test} is also a sinus rhythm, but for another patient. The good results when using few frame vectors in each approximation indicate that this technique will perform well at low bit-rates. Tables with normalized distortion values from the tests can be found in Appendix B.

Some prior knowledge about the signal is required to use an ad hoc designed frame as the initial frame. Using normalized training vectors to constitute an initial frame is easier and requires no prior knowledge. A test was done with initial frames of the same sizes as in the experiment with the ad hoc designed initial frames, but with normalized training vectors as the frame vectors.

No. of vectors in the approximations	ND after terminated training			
	Speech_{train}		MIT100_{train}	
	ad hoc 16×32	training vectors 16×32	ad hoc 16×41	training vectors 16×41
1	0.4555	0.4357	0.1306	0.1534
2	0.2857	0.2809	0.0602	0.0688
3	0.1925	0.1990	0.0394	0.0425
4	0.1431	0.1554	0.0296	0.0330
5	0.1069	0.1229	0.0248	0.0261

Table 5.2: ND after terminated training for training experiments on Speech_{train} and MIT100_{train} with ad hoc based initial frames and initial frames constructed of normalized training vectors

The ND values after terminated training for both experiments are compared in Table 5.2. The difference in the ND after terminated training for the experiments with ad hoc designed initial frames and the experiments with initial frames constructed of normalized training vectors is relatively small. For simplicity and practical reasons we therefore use normalized training vectors to constitute initial frames in the rest of the experiments in this thesis.

5.1.3 Experiment no. 2 - Initial frame from training set

The vector selection algorithm used in these experiments is OMP as described in Section 2.4.1. The frame size is $N \times K$, where $N = 32$ and $K = 64$. Normalized versions of the first signal vectors in the training sets are used as the initial frame vectors.

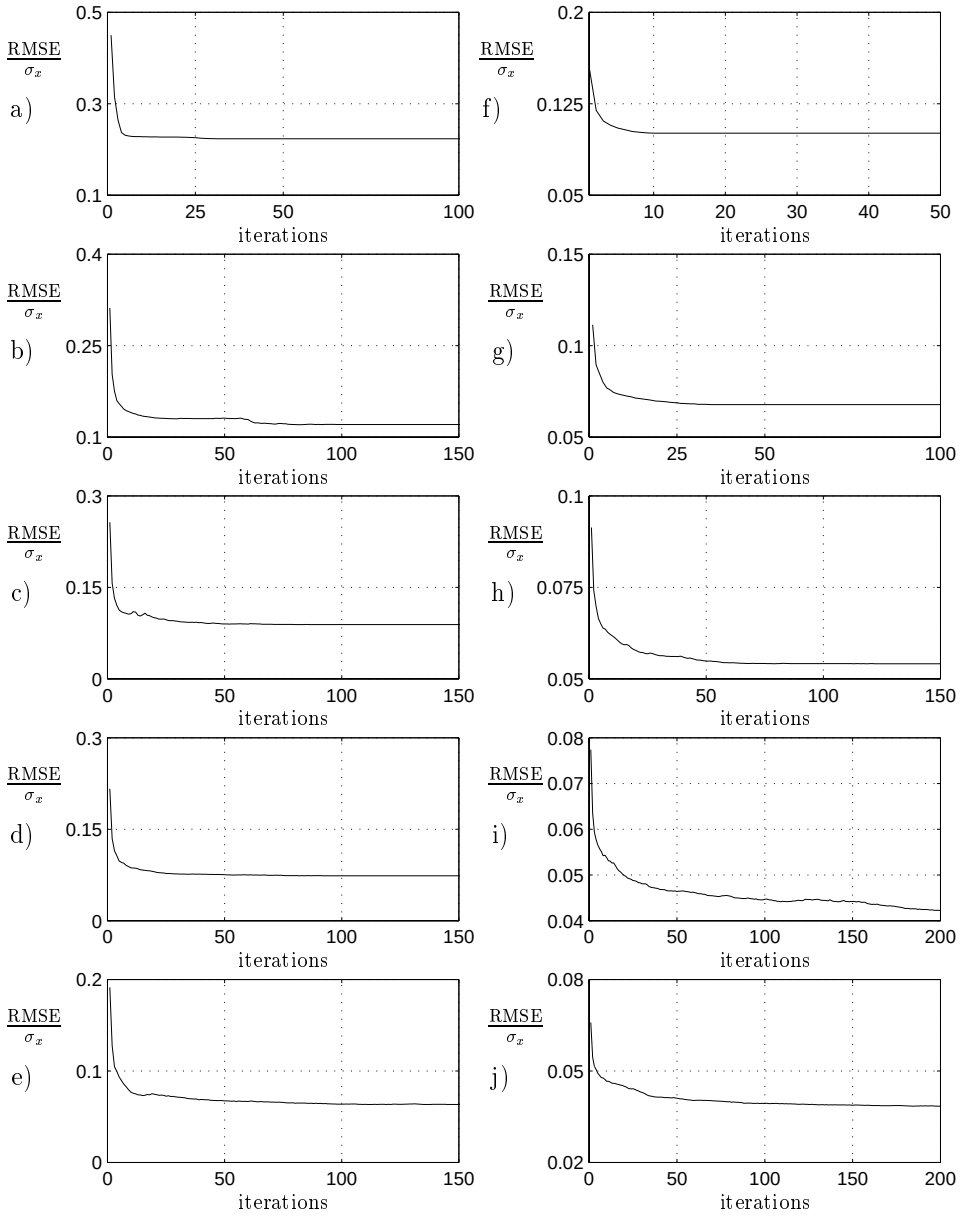


Figure 5.5: Normalized distortion is plotted as a function of training iterations where 1,2,3,4, and 5 frame vectors are used in each approximation. a), b), c), d), and e) MIT100$_{train}$. f), g), h), i), and j) MIT207$_{train}$.

Training was done using MIT100$_{train}$ and MIT207$_{train}$ shown in Figure 5.5. Training was also done using MIT$_{mix}$ as the training signal, and the initial

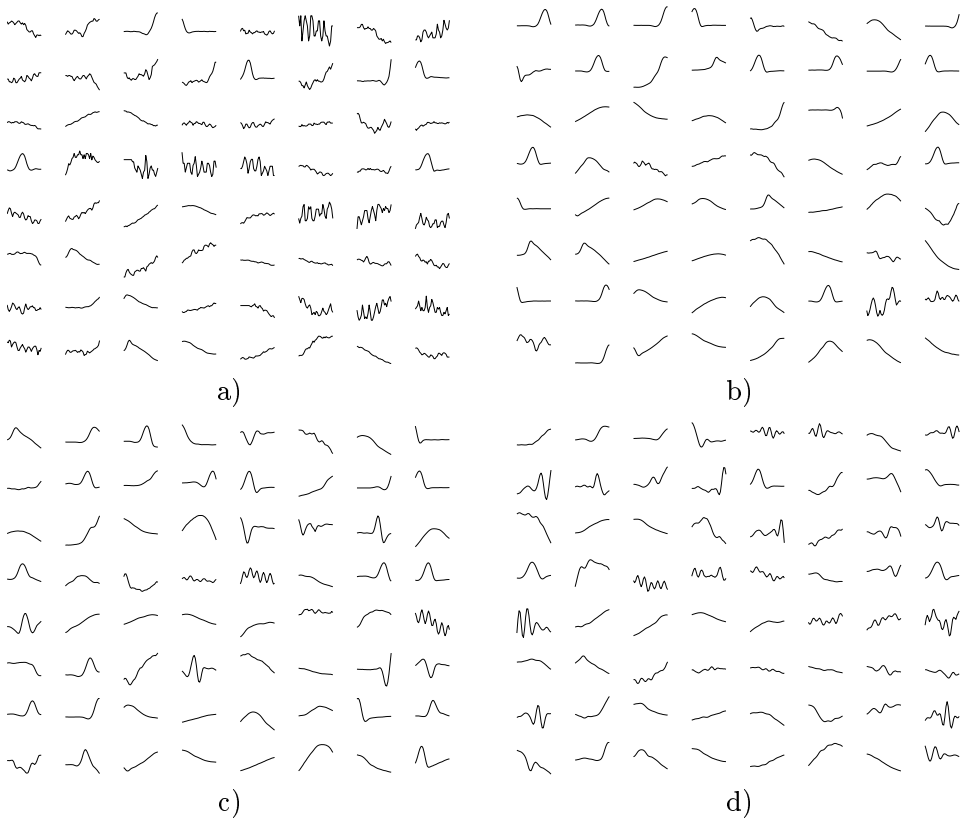


Figure 5.6: Some frames from experiments when training on mixed ECG signal MIT$_{mix}$. a) Initial frame (training vectors), b) Frame trained for using 1 vector/block, c) Frame trained for using 3 vector/block, d) Frame trained for using 7 vector/block

frame was constructed using all the five MIT signals that the MIT$_{mix}$ was composed from. The initial frame had normalized versions of the 13 first training vectors from the MIT100, MIT103, MIT113, and MIT207 training signals, and 12 from the MIT217 training signal.

In Figure 5.6 some of the frames trained in the experiment with MIT$_{mix}$ is shown. a) shows the initial frame used in the training. In b) the frame that results after training when just allowing one frame vector to be used in each approximation is shown. c) shows the frame resulting after allowing three

frame vector to be used in each approximation, and d) shows the resulting frame after allowing seven frame vectors to be used in each approximation. The frame vectors are all normalized, thus the plots show the shapes. The MIT_{mix} consists of segments from five different ECG signals, and it can be seen that the frame vectors reflects shapes that can be found in typically ECG signals. Different variations of shapes that are similar to QRS segments can easily be seen. Naturally this can particularly be seen on the frame vectors shown in b), trained for using just one frame vector in each block. The frame shown in c) has more vectors with higher frequencies, and even more so in the frame plotted in d). This indicates that the frames are well trained for use on ECG signals and should give better energy packing than an ordinary transform like the DCT which is not optimal for this kind of signal.

Training was done using Speech_{train} as the training signal, and in Figure 5.7 a) the shape of the initial frame vectors ares plotted. Figure 5.7 b), c), and d) shows the frame vector after training using Speech_{train}. The frames are trained for using 1, 3, and 7 vectors in each approximation, respectively. The shape of the frame vectors trained for speech can be seen to have shapes that correspond to different typical segments in speech, specially the frame vectors where the frames are designed to use few vectors in each approximation.

The initial frames and the optimized frames trained on Speech_{train}, MIT100_{train}, and MIT207_{train} are tested on Speech_{test}, MIT100_{test}, and MIT207_{test}. The results of the initial and optimized frames are shown in Figure 5.8 a), b), and c). As can be seen from the figure, the improvement after using the MOD to design the frames is substantial. For the speech signal, the improvement of the normalized distortion is 25, 39, 49, 54, 58, and 61 % when choosing 1, 2, 3, 4, 5, and 6 vectors in each approximation respectively. For the MIT100_{test} the improvement in normalized distortion is 47, 63, 68, 71, 72, and 71 %, and for the MIT207_{test} the improvement is 40, 47, 51, 53, 53, and 52 % when choosing 1, 2, 3, 4, 5, and 6 vectors respectively. For comparison a test using Speech_{test}, MIT100_{test}, and MIT207_{test} using DCT with the same number of vectors in each approximation, is presented as well, and the MOD designed frames have significantly better approximation capabilities than the DCT in these three examples. The frames trained on the MIT_{mix} signal is tested on both MIT100_{test} and MIT207_{test}. For comparison a test using MIT100_{test} and MIT207_{test} on the initial frame, and using DCT with the same number of coefficients, is presented as well. Figure 5.8 d), and e) shows the results from these tests, and the trained frames work obviously better than the initial frame and the DCT for both the test signals. Tables with normalized distortion values from the tests can be found in Appendix B.

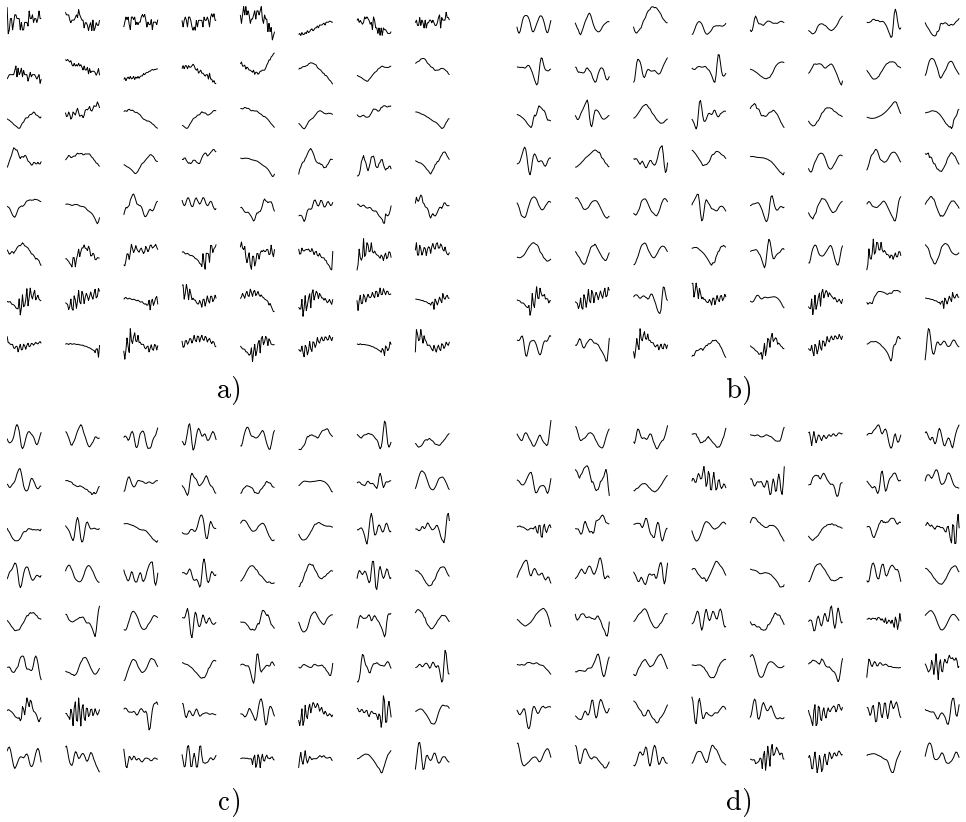


Figure 5.7: Some frames from experiments when training on the speech training signal Speech_{train}. a) Initial frame (training vectors), b) Frame trained for using 1 vector/block, c) Frame trained for using 3 vector/block, d) Frame trained for using 7 vector/block

5.1.4 Experiment no. 3 - Target on SNR/limit on MSE

In the experiments done so far, a sparsity criterion decides how many vectors to choose for each approximation, and this is held constant throughout each experiment. In this experiment we use a target on the SNR or a limit on the MSE for each block instead of fixing the number of vectors to use in each approximation. A block length of $N = 32$ is used in this experiment. Using an $\text{SNR}_{target}/\text{MSE}_{limit}$ instead of a sparsity criterion allows flexibility in the number of nonzero entries from signal vector to signal vector and provides a more stable MSE throughout the signal. As opposed to a sparsity criterion, the MSE can *not* be held constant since the number of vectors used in an

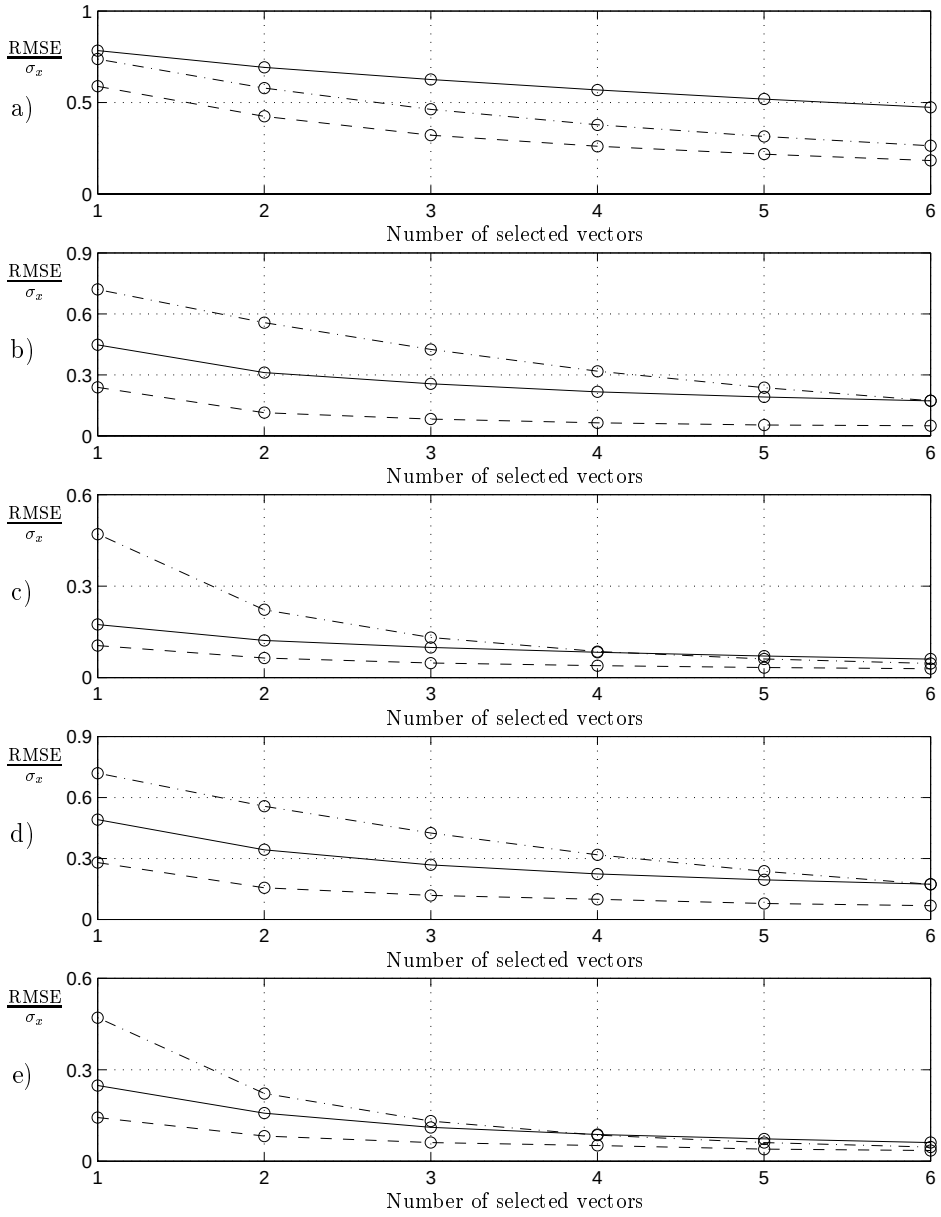


Figure 5.8: Normalized distortion is plotted as a function of the number of selected vectors in the approximations. **Solid**: initial frame, **dashed**: optimized frames, **dash-dot**: DCT. a) Speech$_{test}$, b) MIT100$_{test}$, c) MIT207$_{test}$, d) and e) are trained on MIT$_{mix}$ and tested on: d) MIT100$_{test}$, e) MIT207$_{test}$.

approximation is a discrete value.

When the OMP is used as vector selection algorithm an MSE_{limit} means that for each signal vector the OMP continues to select new vectors until the MSE requirement is satisfied.

The regularized FOCUSS works as described in Chapter 3 and need a target SNR. The target SNR is less stringent than the MSE_{limit} we use with the OMP because the SNR_{target} in regularized FOCUSS indicates the total quality level we want to end up with, but it does *not* impose a stringent limit on a block to block basis as is done with OMP. The FOCUSS algorithm is a parallel algorithm and can not be controlled the same way as greedy algorithms like OMP.

Now the number of vectors used in an approximation is *not* fixed. Since we use the same SNR_{target} or MSE_{limit} throughout the training, the normalized distortion does not change as much as in the training were the number of selected vectors in an approximation is fixed. In these training experiments the change in the *average number of selected vectors* as a function of the iterations, is the crucial factor. We denote this factor $\bar{r}$. The change in the normalized distortion is still interesting however since it can not be held constant, and both these variables are shown as functions of iterations in the figures in this experiment.

To normalize the MSE_{limit} we use

$$\text{ND}_{limit} = \frac{\sqrt{\text{MSE}_{limit}}}{\sigma_x} \tag{5.2}$$

as a normalized distortion limit. Experiments are done with MIT100_{train} and MIT_{mix} as training signals, and with different MSE_{limit} using OMP as the vector selection algorithm. $\text{MSE}_{limit} = 40$ and $\text{MSE}_{limit} = 70$ are used and this gives ND_{limit} at 0.180 and 0.238 when MIT100_{train} is used as training signal, and ND_{limit} at 0.075 and 0.100 when MIT_{mix} is used as the training signal.

The results are shown in Figure 5.9. The average number of vectors used in each approximation, $\bar{r}$, is plotted as a function of iterations, and so is the normalized distortion. The normalized distortion will always be less than the ND_{limit}, and in our experiments it is significantly less than the limit. Investigating the distortion block by block, however, we find that some block's have a distortion very close to the limit while others are much better so that the total mean distortion turns out to be significantly less than the limit. The reason for the normalized distortion to decrease with the iterations is that the

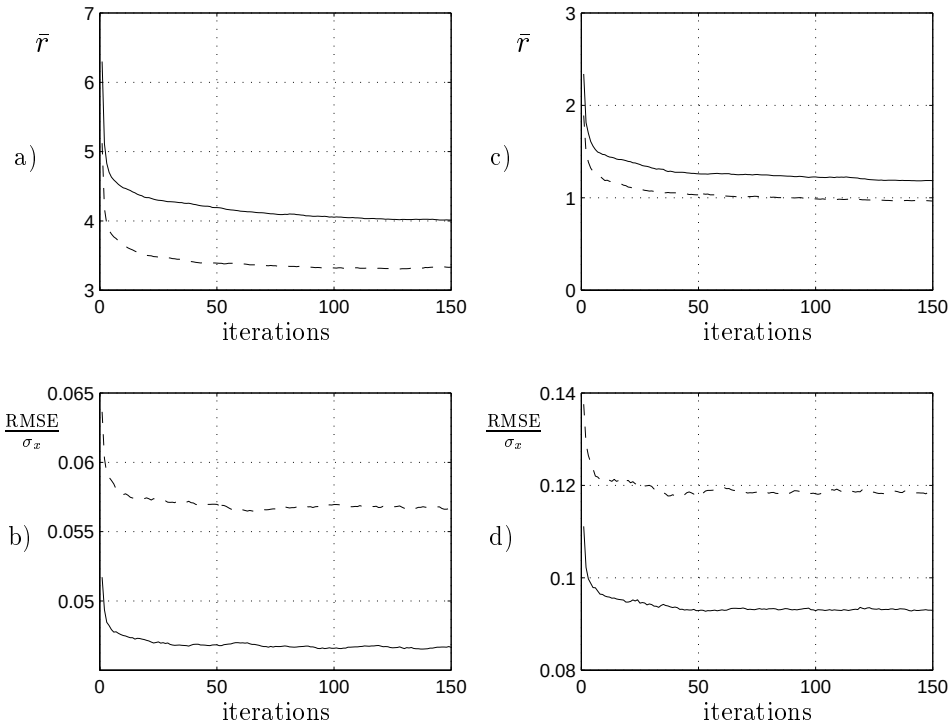


Figure 5.9: Training using OMP and ND_{limit}. The average number of vectors used in the approximations, $\bar{r}$, and the normalized distortion are plotted as a function of training iterations. a) and b) MIT_{mix} is the training signal. **Solid**: $\text{ND}_{limit} = 0.075$ and **dashed**: 0.100. c) and d) MIT100_{train} is the training signal. **Solid**: $\text{ND}_{limit} = 0.180$ and **dashed**: 0.238.

frame vectors get better matched to the training set as the iterations proceeds so that for signal blocks were the same $\bar{r}$ is used as in an earlier iteration, the distortion will decrease.

Figure 5.10 shows training plots of an experiment using regularized FOCUSS, explained in Section 3, as the vector selection algorithm. In this training the target SNR in the regularized FOCUSS is set to 20 dB. In terms of normalized distortion this gives an $\text{ND}_{target} = 0.1$. Another training experiment with target SNR at 10 dB was tried, which gives an $\text{ND}_{target} = 0.316$. The latter choice of target SNR caused some problems. In this case the matrix $\tilde{\mathbf{R}}_{ww}$ in the MOD algorithm ended up *not* having full rank. This means that after approximating the whole training set, there were still some frame vectors that were not used at all. This never occurred in the experiments using Matching

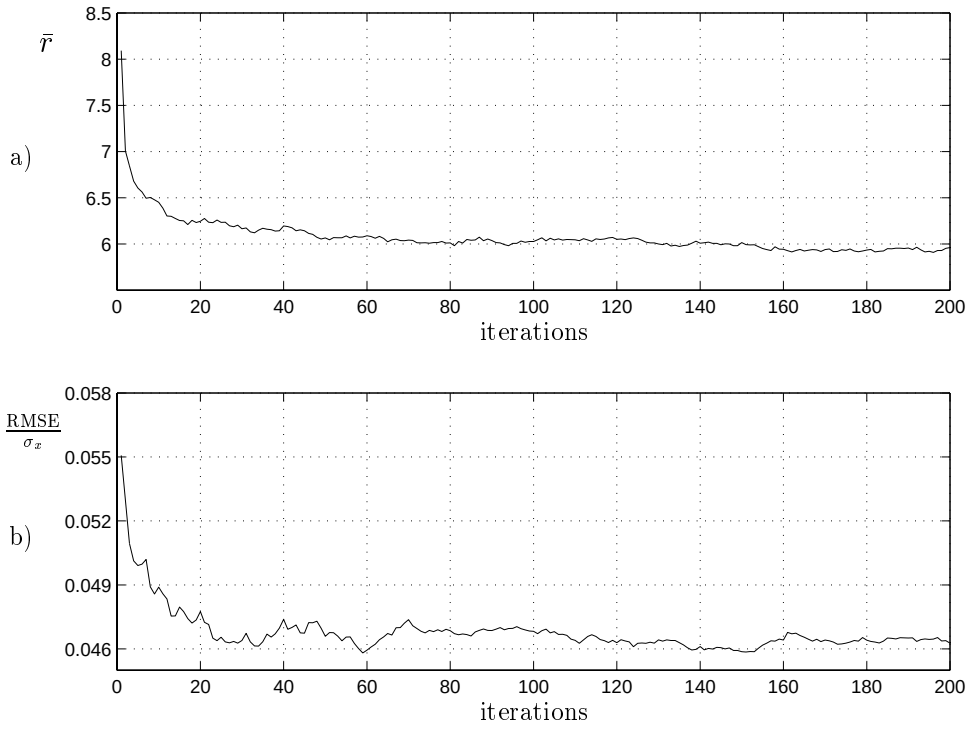


Figure 5.10: Training using regularized FOCUSS, target SNR 20 dB equivalent $\text{ND}_{target} = 0.1$. The average number of vectors used in the approximations, $\bar{r}$, and the normalized distortion is plotted as a function of training iterations. a) Average number of vectors used in the approximations b) normalized distortion for training on MIT100_{train}.

Pursuit techniques. Maybe a larger training set would have reduced this risk, but the training set was already reasonably big.

Instead of enlarging the training set, we dealt with the problem as follows: After going through all the training vectors, the approximations are investigated. If any of the frame vectors never occur in the approximations, the vector is removed from the frame. One option is to let the frame shrink. On the other hand if it is desirable to keep the frame size constant, the removed frame vector can be replaced by another vector. Replacing the removed vector with a random vector was tried, but the random vector often ended up never being used either. It turned out that a better approach was to replace the vector with a vector that we knew was going to be used at least once. Thus we replaced it by the training vector that had been using *the most* frame vectors in its

approximation in the previous iteration. The existing frame obviously lacked a good fit for this particular training vector since it had to use many frame vectors in the approximation. Letting a normalized version of this training vector be a part of the frame of course reduces the number of vectors needed in the approximation for that particular training vector (to one), and it might reduce the number of vectors required in other approximations as well. This training is shown in Figure 5.11.

We pick the frame with minimum MSE and average number of vectors used during training, and it is displayed in Figure 5.12. Comparing Figure 5.12 with Figure 5.6 we see that the frame trained using regularized FOCUSS seems to reflect typical QRS complexes, like in Figure 5.6 b) where one frame vector is used in each approximation. The frame trained using regularized FOCUSS also contains vectors with higher frequencies, corresponding to Figure 5.6 d) where seven frame vectors is used in each approximation. This makes sense since the frame trained using regularized FOCUSS uses one frame vector in some approximations, and more in others.

The frames trained using OMP with MSE_{limit} and the frames trained using regularized FOCUSS are tested on the test signal MIT100_{test}. The approximation capability test is presented in Figure 5.13. For comparison a test using DCT with different MSE_{limit}'s is presented as well (dash-dotted curve). Frames trained using MIT_{mix}, OMP, and different MSE_{limit}'s are tested and plotted in the dashed curve with o's. The dashed curve represents test results on frames trained using MIT100_{train}, OMP, and different MSE_{limit}'s. We use the same MSE_{limit} in testing as we did in training. This means that the ND_{limit} might be different in training and testing because the signal variance might be somewhat different. We do not expect dramatic differences, however, because the training and test data are form the same class of data.

We chose to use a constant MSE_{limit} instead of a constant ND_{limit} (or constant *local* SNR) because we do not want a more accurate representation in parts of the signal with low variance, like between two heart beats, than in the heart beat itself which contains much more important information. Using a constant MSE on a block to block basis is also the optimal, in a rate-distortion sense, in terms of the global MSE (or equivalent global SNR). In perceptual speech and audio coders a larger MSE is tolerated in signal regions with larger local SNR to explore irrelevance in the human auditory system. Note however that a perceptual coder in general have a lower SNR than a source coder of equivalent rate, but higher *perceptual* quality.

The solid curve with stars shows test results on the frame trained using regularized FOCUSS with target SNR at 20 dB. Different thresholding is done.

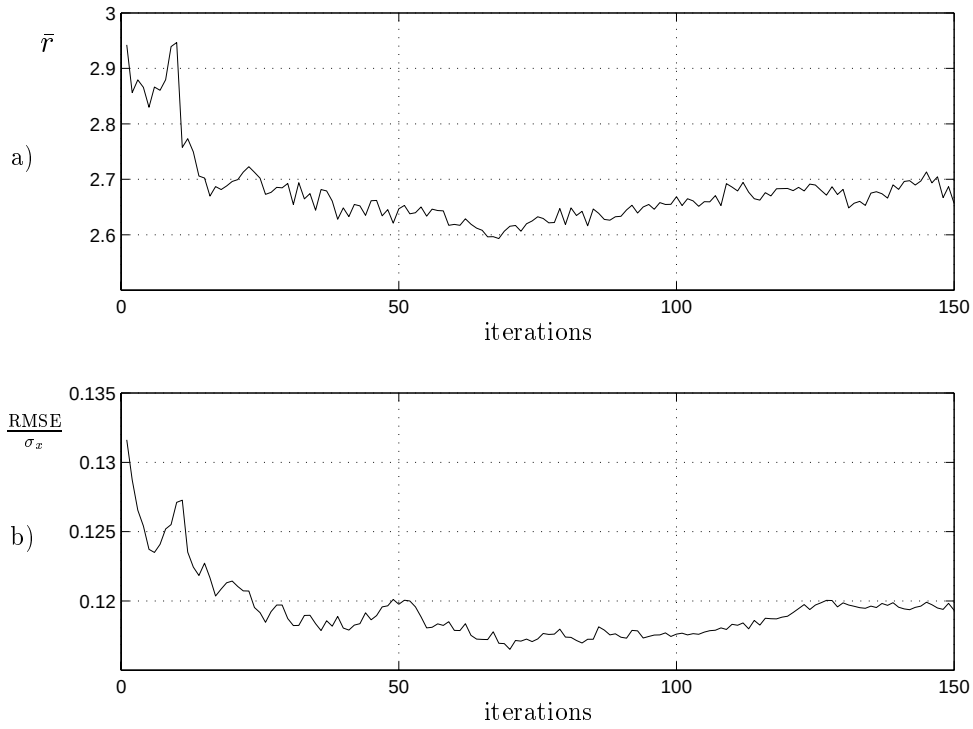


Figure 5.11: Training using regularized FOCUSS, target SNR 10 dB equivalent $\text{ND}_{target} = 0.316$. The average number of vectors used in the approximations, $\bar{r}$, and the normalized distortion is plotted as a function of training iterations. a) Average number of vectors used in the approximations b) normalized distortion for training on MIT100_{train}.

The solid curve shows test results on the frame trained using regularized FOCUSS with target SNR at 10 dB, also here with different thresholding.

From Figure 5.13 we see that the OMP and MSE_{limit} approach performs much better than the DCT with different MSE_{limit}'s at the low $\bar{r}$'s that we concentrate on. MIT100_{train} of course performs better than the frame trained on MIT_{mix}, but they both outperforms the DCT. The frame trained on the regularized FOCUSS with target SNR at 20 dB seems to perform poorer than the DCT. The frame trained with target SNR at 10 dB, on the other hand, performs better than the DCT at low $\bar{r}$'s (with thresholding).

The test results indicate very good energy packing and approximation capability in the optimized frames for low average number of vectors in the approximations, $\bar{r}$. The frames designed in this section have been optimized

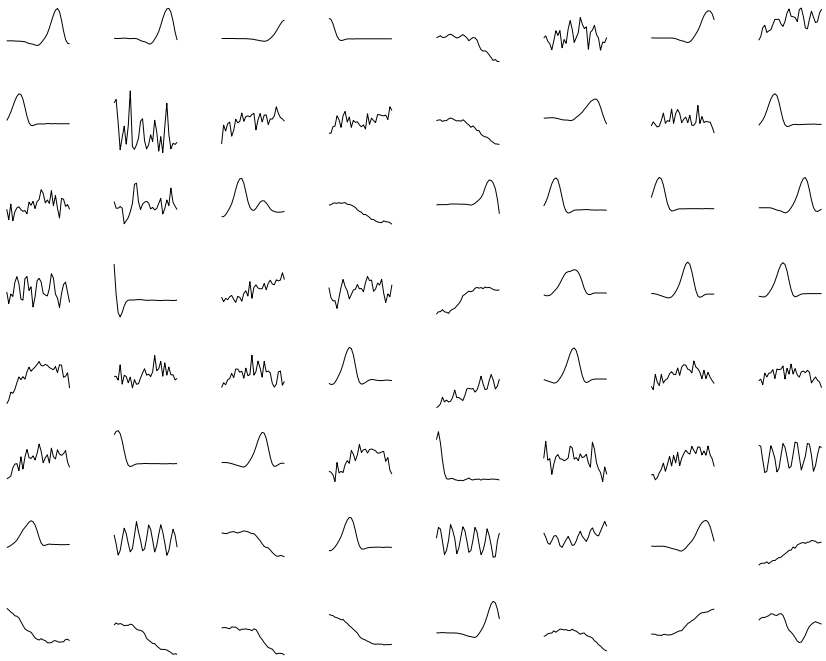


Figure 5.12: Frame vectors resulting after training on MIT100_{train} using regularized FOCUSS, target SNR 10 dB.

for the type of signal they were trained on, and this can be seen by the way typical signal segments is reflected in the frame vectors. The basis vectors in the DCT of dimension 32 can be seen in Figure 5.1 for comparison, and they are not similar to signal segments from the different signals used in the section. This explains some of the reasons for the frames to have more effective energy packing than a DCT when used on signals from the same signal class as the training signals.

5.2 Approximation capabilities for images

The vector selection algorithm used in these experiments is FOMP as described in Section 2.4.1. In the experiments we use image blocks of 8×8 pixels, $\mathbf{X}$. The frame based approximation capability experiments are compared to approximation capability using the DCT. The two dimensional DCT possesses the separability property, i.e. the 2-D DCT can be obtained in two steps by successive application of the 1-D DCT. Doing DCT on a image block we first

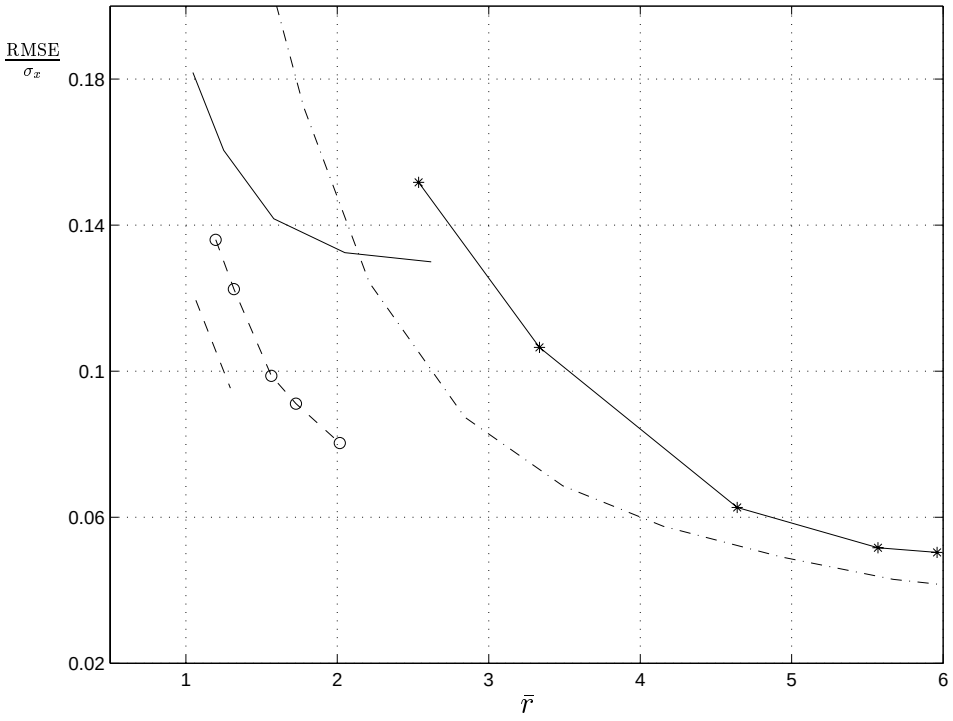


Figure 5.13: Approximation capability test on MIT100$_{test}$. Normalized distortion is plotted as a function of average numbers of vectors in the approximations. **dash-dot**: DCT with different MSE$_{limit}$'s, **solid**: Regularized FOCUSS with target SNR at 10 dB and different thresholding, with *****: Regularized FOCUSS with target SNR at 20 dB and different thresholding, **dash**: frame trained with OMP and different MSE$_{limit}$'s on MIT100$_{train}$, with **o**: frame trained with OMP and different MSE$_{limit}$'s on MIT$_{mix}$

do 1-D DCT over the rows, and then 1-D DCT over the columns of the result. In matrix formulation this can be written as:

$$\mathbf{Y} = \mathbf{T}\mathbf{X}\mathbf{T}^T \tag{5.3}$$

where $\mathbf{X}$ is the 8×8 image block, $\mathbf{T}$ is the 8×8 DCT (synthesis) and $\mathbf{Y}$ is the 8×8 block of transform coefficients, or the transform image.

Traditional image coders like JPEG use the separable DCT, which gives a dictionary of size 64×64. The 8×8 basis images of the separable DCT are shown in Figure 5.14. The separability property puts restrictions on the

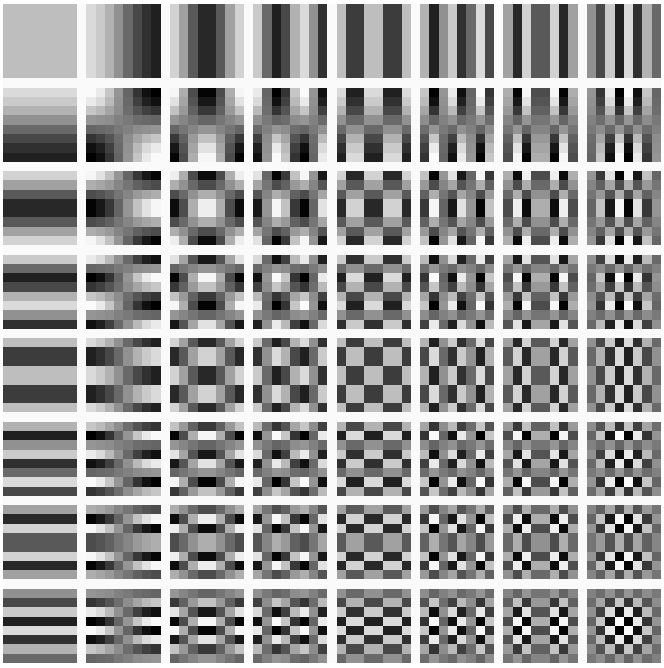


Figure 5.14: Basis images of the separable DCT transform

2D transform. Training a frame to be separable is probably hard, and since it would limit the flexibility of the frame, due to the restrictions, we do the frame based image approximation in a non-separable manner. This means that the $8 \times 8 = 64$ pixels from the image block are formed by lexicographically ordering

of the rows of $\mathbf{X}$ into a 64×1 vector, $\mathbf{x}$:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & & \\ \vdots & & \ddots & \\ x_{N1} & & & x_{NN} \end{bmatrix}, \tag{5.4}$$

$$\mathbf{x} = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1N} \\ x_{21} \\ \vdots \\ x_{NN} \end{bmatrix} \tag{5.5}$$

The synthesis equation becomes:

$$\mathbf{x} = \mathbf{F}\mathbf{w} + \mathbf{r}, \tag{5.6}$$

where $\mathbf{F}$ is the $N \times K$ frame, $N = 64$. $\mathbf{w}$ is the $K \times 1$ coefficient vector and $\mathbf{r}$ is the residual vector. In our experiments $K = iN$ where i varies.

A frame vector with dimension 64 is formed as a frame image by taking the first 8 elements as the first row, the next 8 elements as the second row an so forth.

5.2.1 Images used

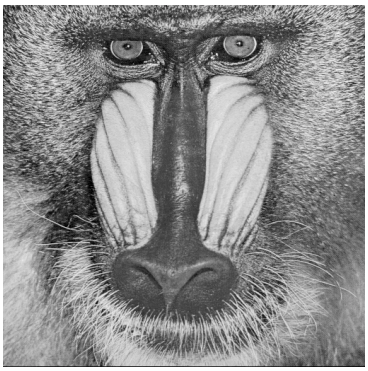
The images used for testing and training are all 512×512 green component images, taken from composite RGB color images, originally represented with 8 bit per pixel. Six training images are used in all the training experiments. The training images are Sailboat, Baboon, Barbara, Paglady, Bridge and Pepper, and they are shown in Figure 5.15. The images used for testing are Lena and Jet, shown in Figure 5.16 and 5.17.

5.2.2 Experiment no. 4 - Sparsity criterion, images

In this experiment training and testing is done on frames of size $N \times 2N = 64 \times 128$ and also on frames of size $N \times N = 64 \times 64$, i.e. *not* overcomplete. The latter test is done to show that MOD gives improved approximation capability

a) Boat

b) Baboon



c) Barbara

d) Paglady

e) Bridge

f) Pepper

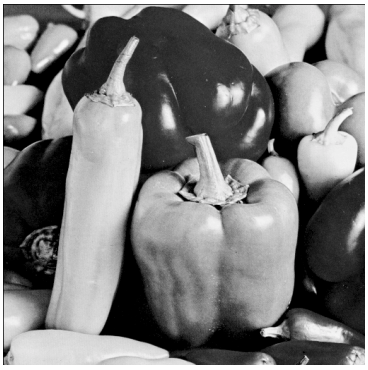


Figure 5.15: Images in the training set

Figure 5.16: The original of the test image Lena

Figure 5.17: The original of the test image Jet

over the DCT even if the frame is not overcomplete, and that the approximation capability is improved even more by using overcomplete frames. In [38] experiments show significant improvement potential in the approximation capabilities of known $N \times N$ transforms (KLT,DCT) after training using MOD and a training set.

The initial frames are made of normalized vectors from the training set. We have learned from experiments that if the initial frame consists of training vectors, the actual choice of training vectors does not effect the training result much as long as all distinct classes or images are represented. Thus, the initial frame is built of image blocks randomly picked according to a uniform distribution over the training set.

The training is done by fixing the number of vectors that is allowed in each approximation for a specific frame, i.e. the representation has a sparsity criteria.

Figure 5.18 shows some training plots when training on images. The normalized distortion in these experiments can be seen to drop with 17 to 35 % during training. The distortion drops rapidly in the beginning of the training, and after 50-100 iterations the improvement is less significant.

Some of the frames that result after training can be seen in Figure 5.19, Figure 5.20, Figure 5.21, and Figure 5.22. The frame vectors are shown as frame images since the frame vectors are used to approximate image blocks. A frame vector with dimension 64 is formed as a frame image by taking the first 8 elements as the first row, the next 8 elements as the second row an so forth. The frame images in the figures are ordered. The most frequently used frame image is placed in the upper left corner. The frame images are placed row by row and the one in the lower right corner is the least frequently used frame image. For comparison with the frame images, see the 8×8 basis images of the separable DCT shown in Figure 5.14.

The trained frame images reflect typical image features, as the frames in Section 5.1.1 reflects typical features of the ECG and the speech signal. Looking at Figure 5.14, the basis images of the DCT do not reflect typical image features, but it does have a DC block, and it has blocks with horizontal and vertical edges. Note that the trained frame images also show edges that are diagonal since the approximation is non-separable.

Figure 5.23 shows an approximation capability experiment of the test images Lena and Jet. The sizes of the frames in the experiment are 64×64 and 64×128. The approximation capabilities are compared to the separable DCT transform on 8×8 image blocks, whose basis images can be seen in Figure 5.14.

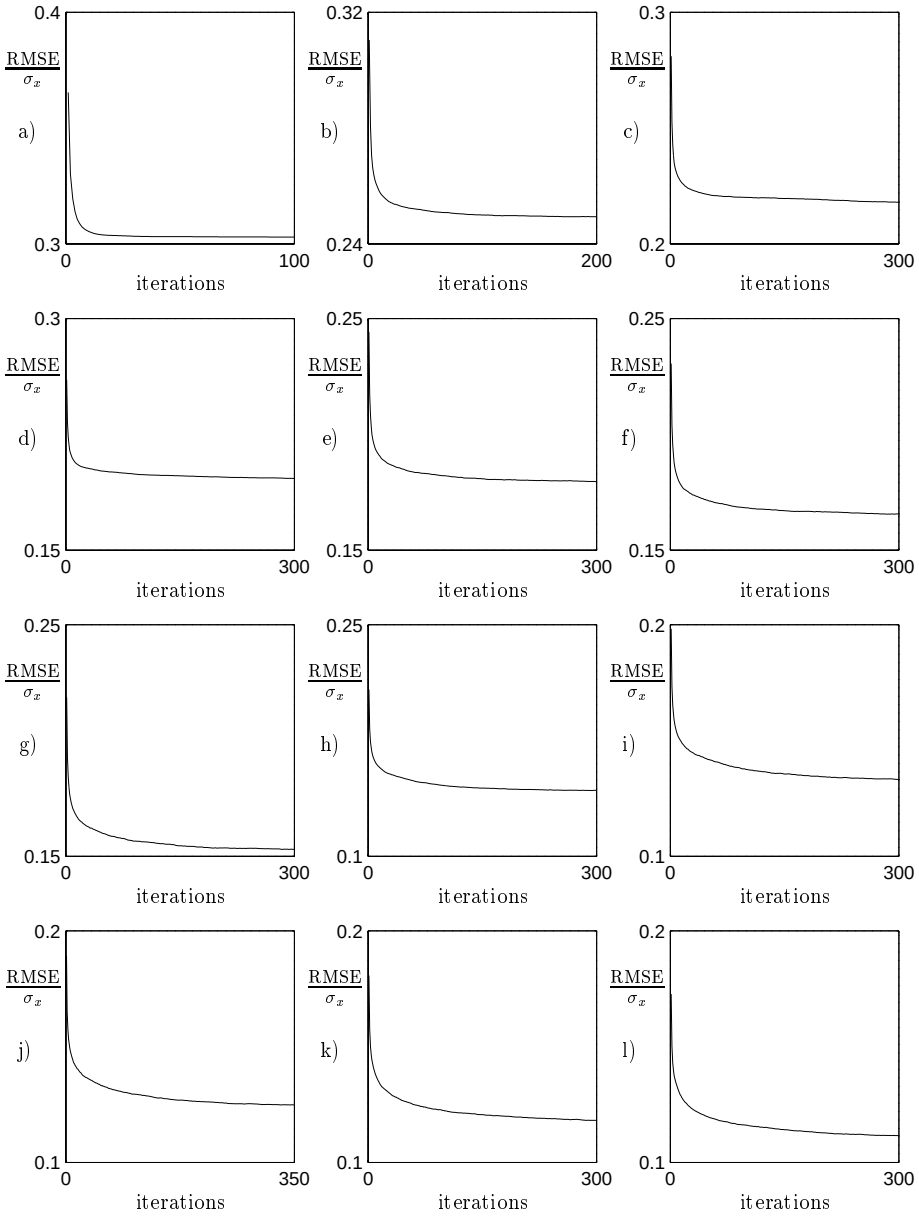


Figure 5.18: The normalized distortion is plotted as a function of training iterations for training on images. a) 1, b) 2, c) 3, d) 4, e) 5, f) 6, g) 7, h) 8, i) 9, j) 10, k) 11, and l) 12 frame vectors are used in each approximation.

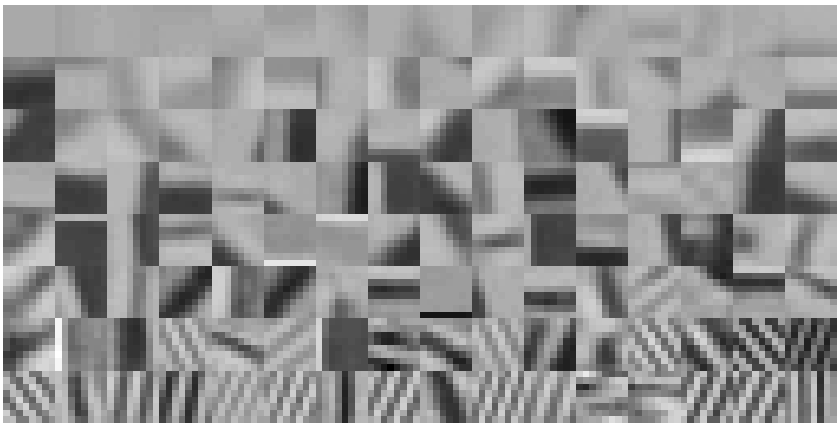


Figure 5.19: Frame after training, 1 frame vector selected in each block

Figure 5.20: Frame after training, 3 frame vector selected in each block

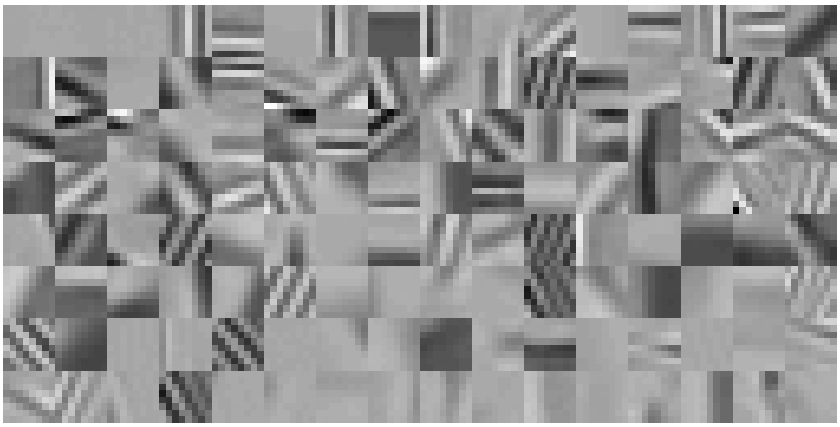


Figure 5.21: Frame after training, 5 frame vector selected in each block

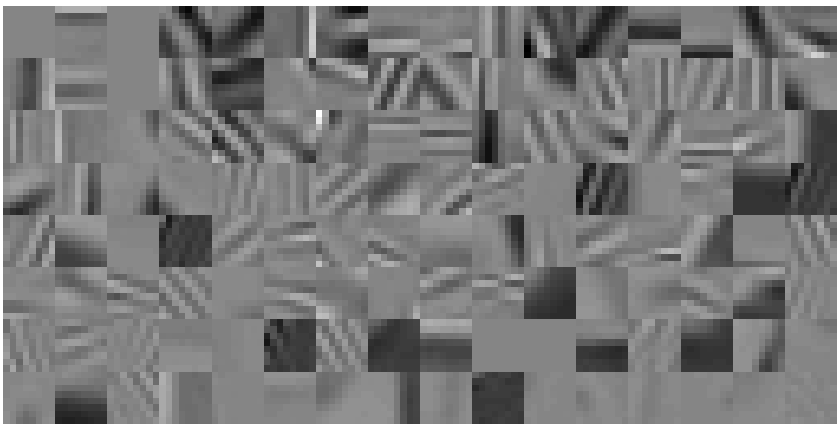


Figure 5.22: Frame after training, 7 frame vector selected in each block

From Figure 5.23 it can bee seen that the approximation capabilities of the MOD designed $N \times N$ frames are significantly better than that of the DCT transform, and that the overcomplete frames have even better approximation capabilities than the $N \times N$ frames.

No. of vectors in the approximation	Lena				
	DCT $N \times N$	Frames $N \times 2N$	Red. %	Frames $N \times N$	Red %
1	0.320	0.219	31.7	0.227	29.2
2	0.237	0.166	29.8	0.177	25.5
3	0.194	0.140	28.0	0.151	22.3
4	0.167	0.122	26.9	0.130	22.0
5	0.147	0.110	25.6	0.120	18.9
6	0.132	0.100	24.4	0.110	17.1
7	0.120	0.091	24.4	0.100	16.5
8	0.111	0.085	23.5	0.093	15.7
9	0.102	0.078	23.2	0.089	13.2
10	0.095	0.074	22.2	0.083	12.3
11	0.089	0.070	21.5	0.079	10.8
12	0.083	0.065	21.3	0.075	9.7

Table 5.3: Normalized distortion after test on the image Lena. Red. % is reduction in the normalized distortion relative to the DCT test.

The normalized distortions for the tests are printed in the Tables 5.3, and 5.4, together with the reductions in % of the normalized distortions in the frame tests compared to the DCT tests. The tables show that the reduction in the normalized distortion is between 9% and 32% in the tests. In the first test, using overcomplete frames, it is obvious that there is a potential for distortion reduction since there are more vectors to chose from when making an approximation compared to an ordinary orthogonal transform. In the second experiment where the frames are of size $N \times N$ there are not more vectors to chose from, but still the improvement potential is proven to be significant. This illustrates the great potential of frames, not necessarily overcomplete, optimized for a class of data using the MOD. The $N \times N$ frames, or transforms, are *not* orthogonal as the traditional transforms like the DCT.

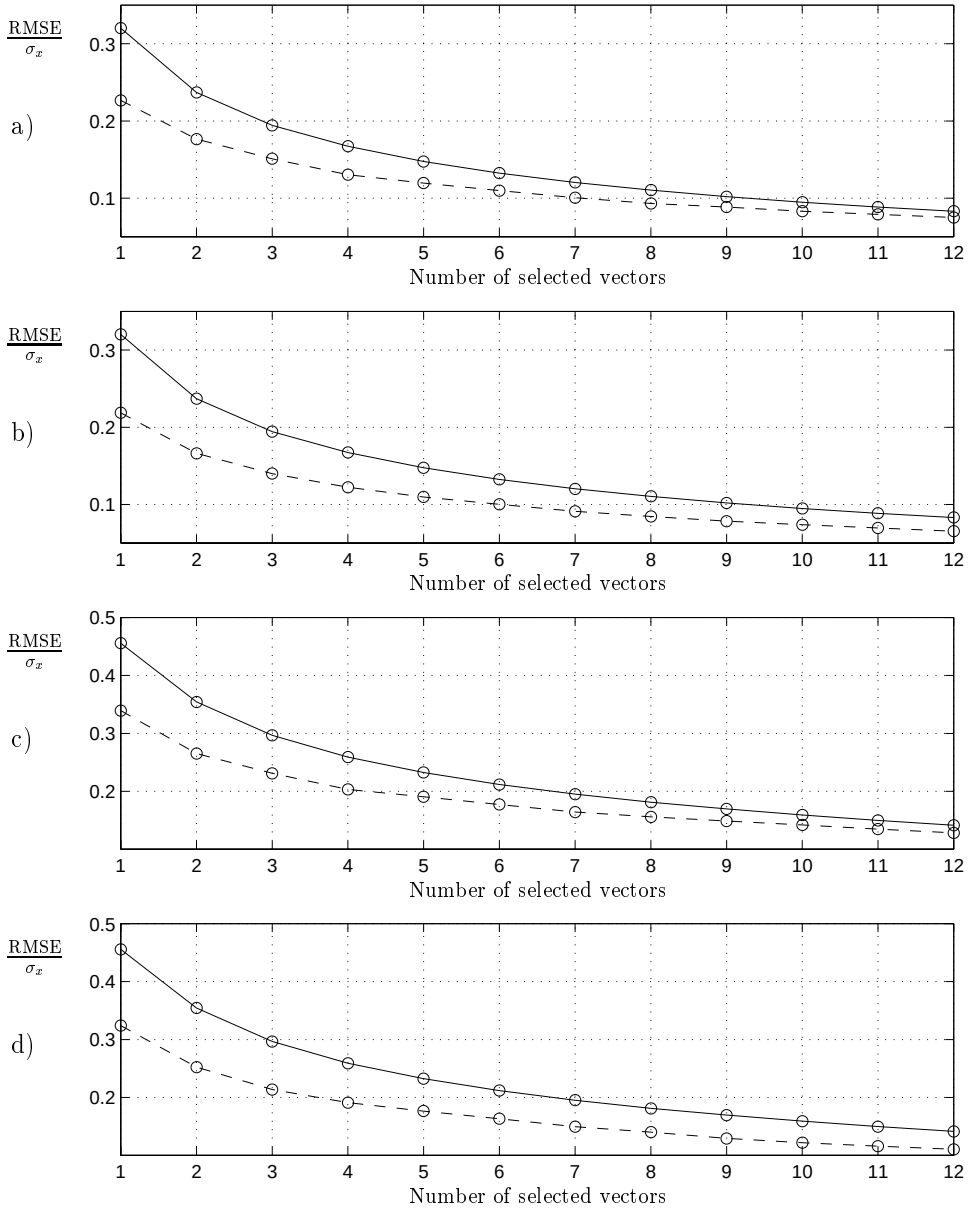


Figure 5.23: Approximation capabilities of frames trained on training images are tested on the test images. The normalized distortion is plotted as a function of different numbers of vectors in an approximation. **Solid**: DCT, **dashed**: Optimized frames. a) Lena size $N \times N$, b) Lena size $N \times 2N$, c) Jet size $N \times N$, d) Jet size $N \times 2N$.

No. of vectors in the approximation	Jet				
	DCT $N \times N$	Frames $N \times 2N$	Red. %	Frames $N \times N$	Red %
1	0.456	0.324	28.9	0.340	25.5
2	0.354	0.252	28.8	0.265	25.1
3	0.297	0.214	28.0	0.231	22.1
4	0.259	0.191	26.2	0.203	21.6
5	0.233	0.176	24.2	0.190	18.1
6	0.212	0.163	23.0	0.177	16.3
7	0.195	0.149	23.4	0.164	16.0
8	0.181	0.140	22.7	0.156	14.1
9	0.170	0.129	23.7	0.149	12.4
10	0.159	0.122	23.5	0.142	10.8
11	0.150	0.116	22.5	0.135	10.0
12	0.141	0.110	21.9	0.128	9.2

Table 5.4: Normalized distortion after test on the image Jet. Red. % is reduction in the normalized distortion relative to the DCT test.

5.2.3 Experiment no. 5 - Limit on MSE

In this experiment training and testing are done on frames of size $N \times 2N = 64 \times 128$. The initial frames are made of normalized vectors from the training set as described in Experiment no. 4, thus we have 128 initial frame images, representing all the images from the training set. Instead of using a sparsity criterion, an MSE_{limit} is used on a block to block basis as the representation requirement. FOMP is the vector selecting algorithm in both training and testing. Training was carried out with different MSE_{limit}'s.

Four frames were trained with MSE_{limit} at 40, 70, 100, and 150, and this corresponds to ND_{limit} at 0.11, 0.14, 0.17, and 0.20 respectively. The frames were tested on Lena and Jet with the same MSE_{limit} as designed for, and with thresholding at different levels. For the test image Lena, the MSE_{limit}'s corresponds to ND_{limit}'s at 0.12, 0.16, 0.19, and 0.23, and for test image Jet, they correspond to 0.29, 0.38, 0.45, and 0.55.

As in the ECG experiment in Section 5.1.4, we keep the MSE_{limit} constant during the training, and we use the same MSE_{limit} in testing as done in training. Keeping an MSE_{limit} constant for an image (training or testing), and not a relative measure as the ND_{limit}, is best in a rate-distortion sense. A constant

ND_{limit} would demand increased accuracy at image parts with low local variance, like in a background in an image, and this is not desired. Since all the images are limited between 0 and 255 peak values, the same argument can be used for using the same absolute measure, MSE_{limit}, on the test image as the training images instead of a relative measure like ND_{limit}. An image, or part of an image, with very low variance does usually not contain much important information and we use the same MSE_{limit} as in the more important images or part of the images.

The results can be seen in Figure 5.24. A DCT test is depicted for comparison. The DCT test is done by imposing different MSE_{limit}'s on a block to block basis, to make the comparison as fair as possible. Figure 5.24 a) shows the results after testing on Lena, and in b) the results after testing on Jet are plotted.

Comparing Figure 5.24 with Figure 5.23 indicates that the frames trained with an MSE_{limit} performs better in terms of approximation capabilities in these tests.

5.3 Discussion

The results presented in this chapter demonstrate significant benefits associated with optimizing frames for a given class of input data. The approximation capabilities for the optimized frames are shown to be very good for one dimensional signals such as ECG and speech, as well as for two dimensional signals, i.e. digital images. Our experiments demonstrate an MSE improvement decreasing with the number of vectors used in each approximation as can be seen in Table 5.3 and Table 5.4 as well as in the Figures 5.4 and 5.8. This is intuitively right since when using many vectors in each approximation it is possible to get a good approximation with a lot of different frames or transforms. From Figure 5.6, Figure 5.7, Figure 5.19, Figure 5.20, Figure 5.21, and Figure 5.22 it can be seen that the frames trained for different number of vectors used in each approximation exhibit different characteristics. These results motivate the use of *several* MOD-designed frames in a complete compression scheme. This is investigated and tested in Chapter 7.

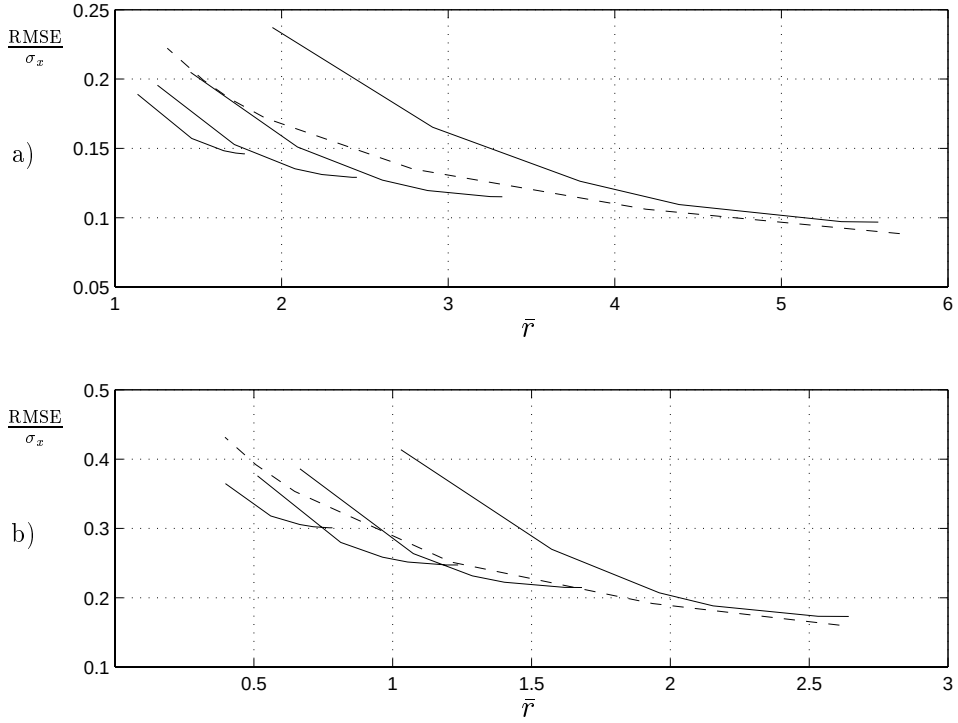


Figure 5.24: Approximation capabilities of frames trained on training images are tested on the test image a) Lena, ND_{limit}'s at 0.12, 0.16, 0.19, and 0.23, and b) Jet, ND_{limit}'s at 0.29, 0.38, 0.45, and 0.55. Training and testing are executed using an MSE_{limit} and FOMP. The normalized distortion is plotted as a function of the average number of selected vectors in the approximations. **Dashed**: DCT with different MSE_{limit}'s, **solid**: Frames using FOMP and MSE_{limit}'s at 40, 70, 100 and 150.

Chapter 6

Compression using one frame

This chapter starts by investigating the properties of the frame coefficients. Reference compression schemes are explained, and some compression experiments using frames are presented and compared to compression using reference compression schemes.

6.1 Investigation of frame coefficient properties

To get some insight into the properties of the frame coefficients, some investigation was done. This kind of insight is needed to find a good coding strategy for the coefficients, or just as explanations of why we get the results that we get.

Figure 6.1 and Figure 6.2 show examples of histograms of the number of times different frame vectors are chosen. The experiments are done using frames of size 32×64 trained and tested with OMP and MSE_{limit}. MIT100_{train} is the training signal and MIT100_{test} is the test signal for both the experiments. In the experiment in Figure 6.1 $\text{MSE}_{limit} = 70$ in both training and testing. This gives an $\text{ND}_{limit} = 0.24$ at training and $\text{ND}_{limit} = 0.23$ at testing. The experiment in Figure 6.2 used $\text{MSE}_{limit} = 40$ in both training and testing. This gives an $\text{ND}_{limit} = 0.18$ at training and $\text{ND}_{limit} = 0.17$ at testing. After the training the frame was sorted so that the most frequently used vector in the last training iteration becomes $\mathbf{f}_1$, i.e. the first frame vector, the second most frequently used becomes $\mathbf{f}_2$ an so forth.

Figure 6.1 a) and Figure 6.2 a) show how many times the different frame vectors are chosen *as the first* chosen frame vector when representing the test

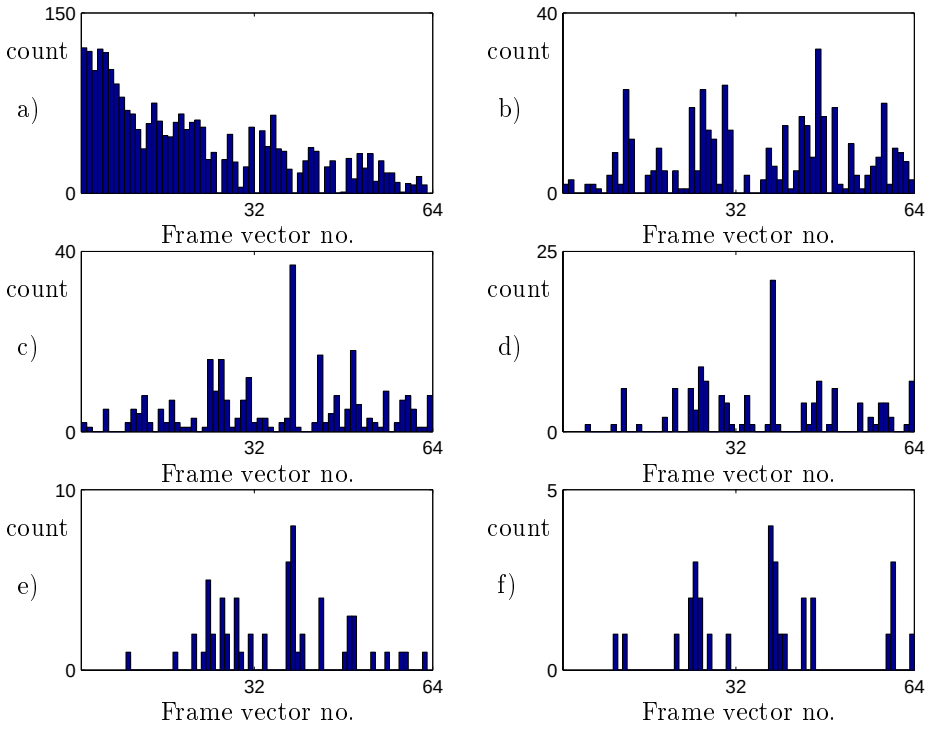


Figure 6.1: Histogram of the number of occurrence for the frame vectors. The frame is trained on MIT100_{train} with $\text{MSE}_{limit} = 70$ corresponding to $\text{ND}_{limit} = 0.24$, and tested on MIT100_{test} with $\text{MSE}_{limit} = 70$ corresponding to $\text{ND}_{limit} = 0.23$. a) shows how many times the different frame vectors are chosen as the *first* chosen frame vector, b) as the second, and for c), d), e), and f) chosen as no. 3, 4, 5, and 6 respectively.

signal. Figure 6.1 b) and Figure 6.2 b) show how many times the vectors are chosen as the *second* chosen frame vector and so forth. In a) we can see indications that the frame was sorted before the test in that the histogram shows a slightly decreasing tendency. Still we can see in all the plots in the figures that the frame vectors are selected fairly regularly from all the vectors in the set, without any obvious preferences.

This is not surprising since we optimize solely with respect to MSE over the set of frame vectors. We impose no other constraints, like entropy, in our frame design algorithm. This means that all the vectors in the frame are actively used to minimize the MSE, which for a fixed code word length would be very desirable. In traditional JPEG-like transform coding the situation is

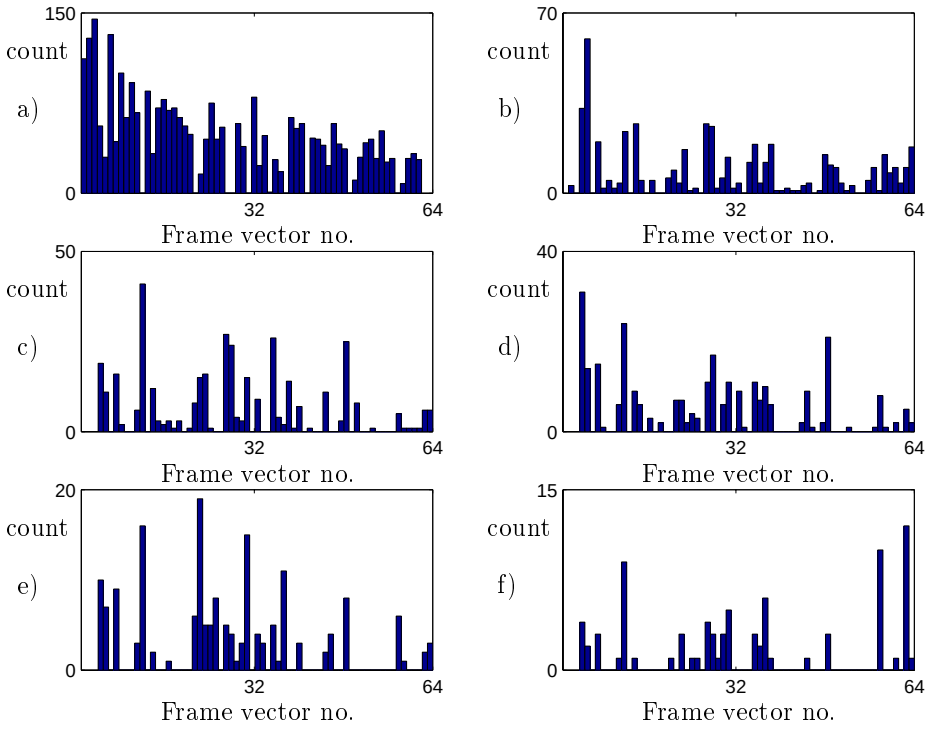


Figure 6.2: Histogram of the number of occurrence for the frame vectors. The frame is trained on MIT100_{train} with $\text{MSE}_{limit} = 40$ corresponding to $\text{ND}_{limit} = 0.18$, and tested on MIT100_{test} with $\text{MSE}_{limit} = 40$ corresponding to $\text{ND}_{limit} = 0.17$. a) shows how many times the different frame vectors are chosen as the *first* chosen frame vector, b) as the second, and for c), d), e), and f) chosen as no. 3, 4, 5, and 6 respectively.

very different. Figure 6.3 a) and b) shows histograms of the total number of occurrence of the frame vectors from the same experiments as described above. In c) and d) histograms of the number of occurrence of the different basis vectors are plotted after representing MIT100_{test} with DCT and two different thresholding values. From a) and b) we can see, as stated earlier, that there is a decreasing tendency in the number of occurrence, but that the frame vectors are selected fairly regularly from all the vectors in the set. In c) and d) on the other hand the number of occurrence decreases rapidly, and the distribution is far from even. 44 and 60 % of the basis vectors are never used in the two cases. The information of which of the coefficients in a sparse vector that are different from zero can be regarded as *position information*.

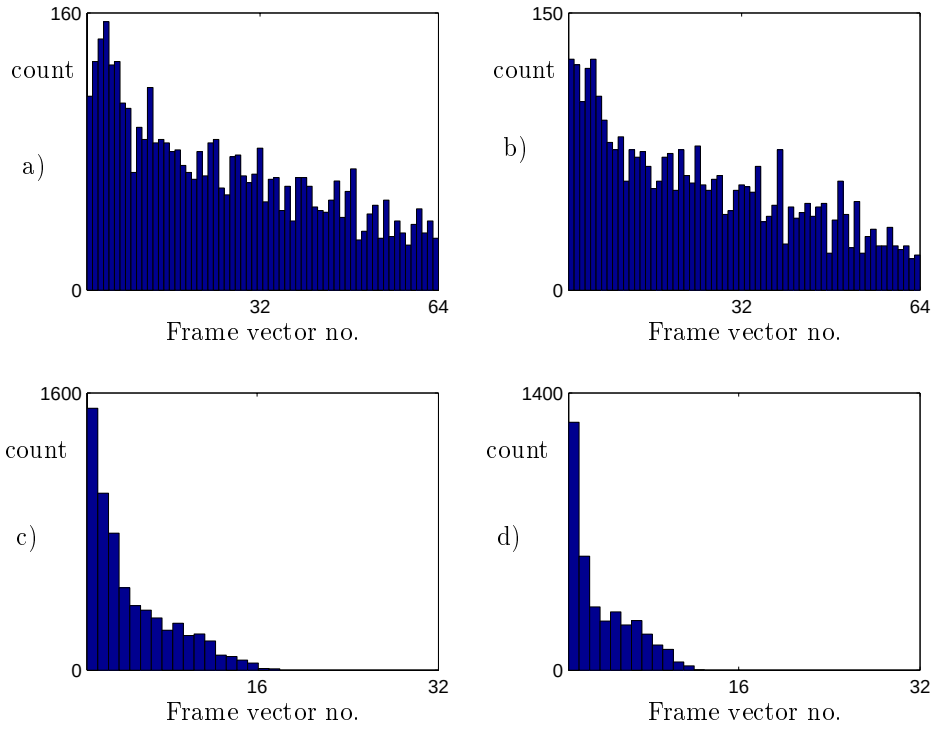


Figure 6.3: Histogram of the total number of times different frame or transform(basis) vectors are chosen, tested on MIT100_{test}. a) $\text{ND}_{limit} = 0.17$, b) $\text{ND}_{limit} = 0.23$, c) DCT, T=10, d) DCT, T=30.

The position information is far from uniform in the DCT case, and can clearly be entropy coded. This makes entropy coding a very useful coding strategy for ordinary transform coding. Fixed code word length on the other hand would not perform, by far, as good as an entropy based scheme. Figure 6.3 a) and b) shows that the distribution of the position information has some similarity to a uniform distribution in the frame test, and not much can be gained by entropy coding of the frame position information.

The Figures 6.4 and 6.5 show histograms of the coefficient *values* for the 6 frame vectors *chosen* first, second and so on. The same frames as before are tested on MIT100_{test} with the same MSE_{limits} as before. Figure 6.6 shows histograms of the values of the six first DCT coefficients. The variances of the eight first coefficient values for the DCT test (solid) are plotted in Figure 6.7 together with the variances of the coefficients chosen first, second and so forth of the two test examples with $\text{ND}_{limit} = 0.17$ (dash-dot) and $\text{ND}_{limit} = 0.23$

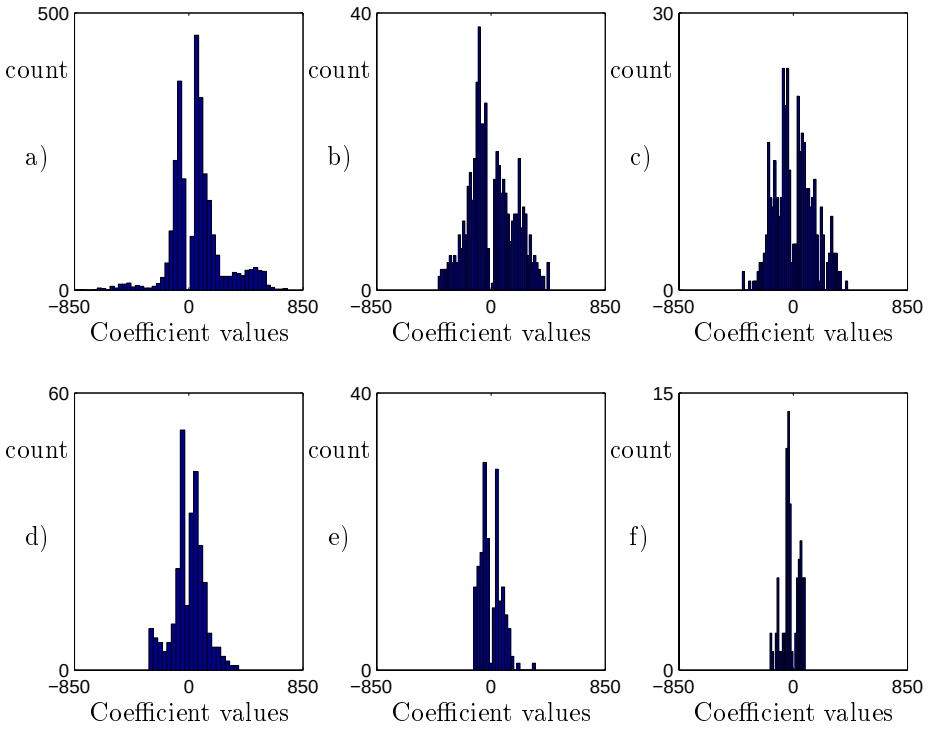


Figure 6.4: Histograms of coefficient *values*. The frame is trained on MIT100_{train} with OMP and $\text{MSE}_{limit} = 40$ corresponding to $\text{ND}_{limit} = 0.18$, and it is tested on MIT100_{test} with $\text{MSE}_{limit} = 40$ corresponding to $\text{ND}_{limit} = 0.17$. a) values of the coefficients chosen *first* b), c), d), e), and f) values of the coefficients chosen as no. 2, 3, 4, 5, and 6 respectively.

(dashed).

By studying Figure 6.6 and 6.7 we see that the signal variance of the DCT coefficients decreases approximately monotonically. This is very advantageous both when doing entropy coding with uniform quantizers, but also with no entropy coding, where a bit allocating scheme would take advantage of the decreasing variance. The situation for the frame coefficients is somewhat different. There is a distinct decreasing tendency, which is an advantage, but the decrease is not as monotonic or as rapid to approximately zero as for the DCT coefficients. Another observation is that the first frame vectors have a significantly larger variance than the first DCT coefficient. The fact that in the frame case there are different vectors chosen as no. 1 at different signal blocks is probably partly responsible for this. But it does also explain why

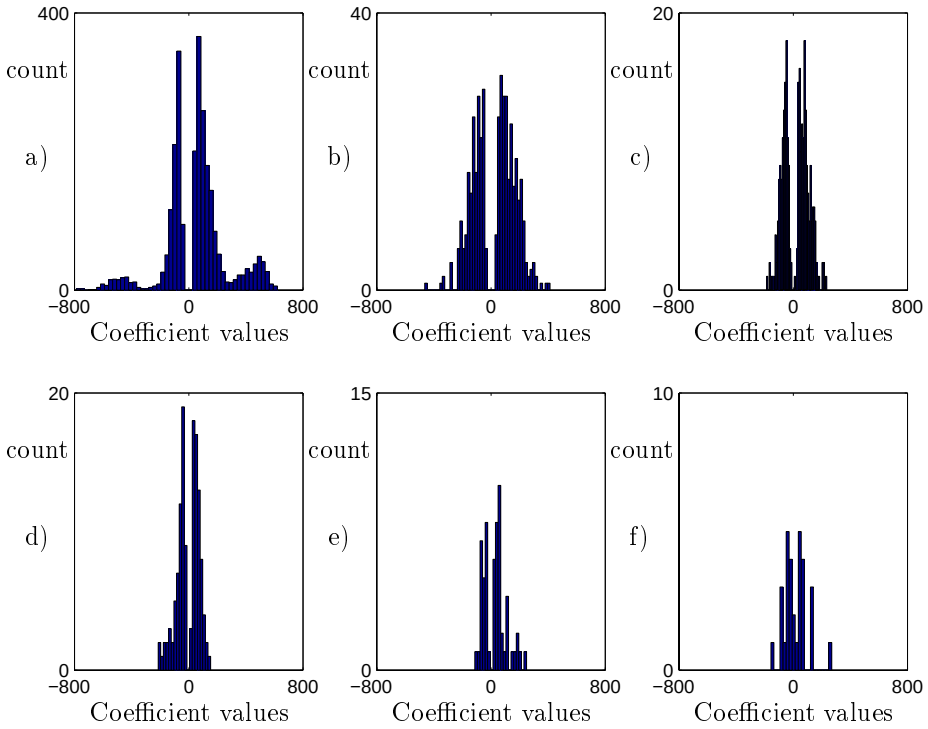


Figure 6.5: Histograms of coefficient *values*. The frame is trained on MIT100_{test} with OMP and $\text{MSE}_{limit} = 70$ corresponding to $\text{ND}_{limit} = 0.24$, and it is tested on MIT100_{test} with $\text{MSE}_{limit} = 70$ corresponding to $\text{ND}_{limit} = 0.23$. a) values of the coefficients chosen *first* b), c), d), e), and f) values of the coefficients chosen as no. 2, 3, 4, 5, and 6 respectively.

we get a better approximation using *one* frame vector than using *one* (usually the first) DCT vector. Another factor is that the distributions for each of the coefficient numbers are very peaky, and far from uniform in the DCT case. This makes entropy coding of the coefficient values advantageous. This is also the case for the frame coefficients but not as distinct as for the DCT coefficients.

If entropy coding is used for the position information and values, or symbols combining position information *and* value, the optimal quantization would be uniform quantization and thresholding. In the case of no entropy coding, the histograms show that a bit allocating scheme would be beneficial.

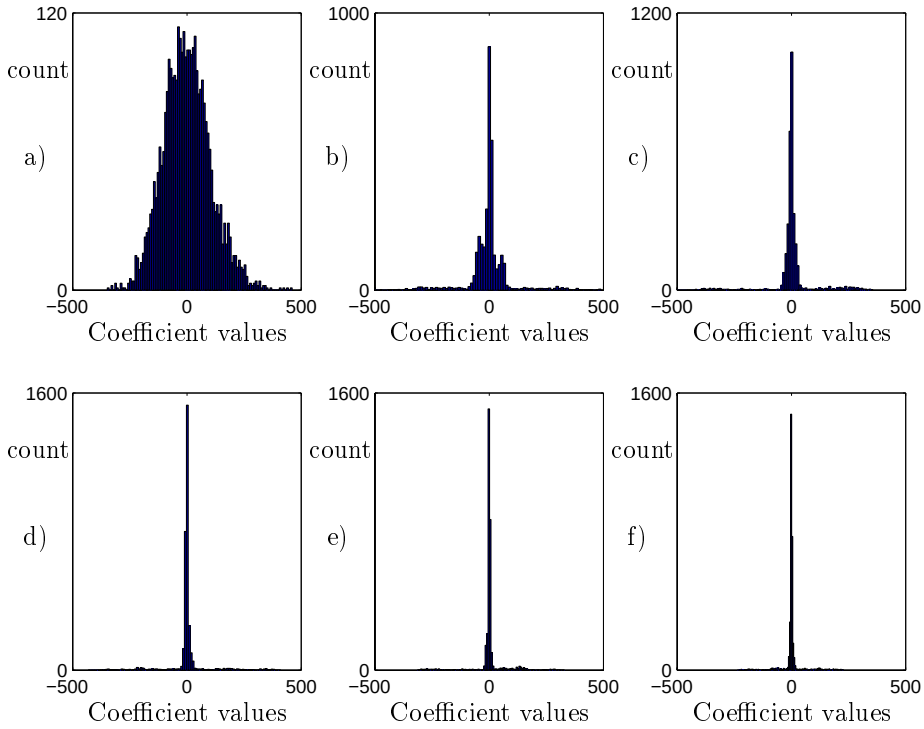


Figure 6.6: Histograms of DCT coefficient *values* for the signal MIT100$_{test}$. a) shows the values of the *first* DCT coefficient, b), c), d),), and f) coefficient no. 2, 3, 4, 5, and 6 respectively.

6.2 Reference compression schemes

A reference compression scheme is needed for comparison of rate-distortion performance. We want to make a fair comparison and have chosen reference compression schemes according to that. The scheme described in Section 6.2.1 is used in our ECG experiments. The coefficients are entropy coded, and thus have variable code word length. In our image experiments the widely used compression standard, JPEG, is used as the reference compression scheme.

6.2.1 Variable length coding scheme for 1D signals

Since the frame based compression scheme is block based, the reference scheme should be block based and the same block lengths should be used to ensure a fair comparison. We choose to use a transform based scheme since that is

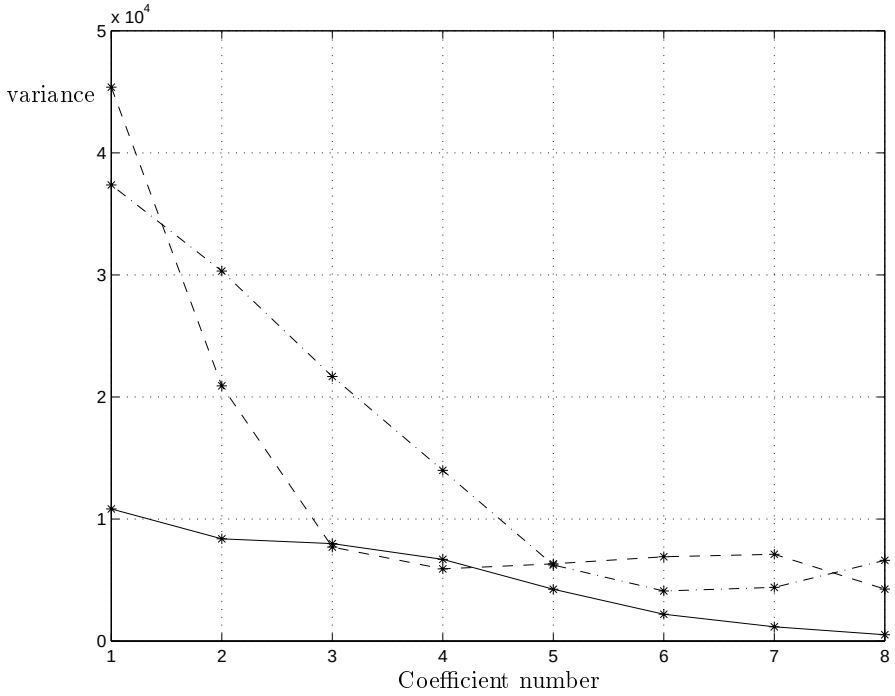


Figure 6.7: The variance of coefficient values, tested on MIT100$_{test}$. **Solid**: Variance of the eight first DCT coefficient, **dash-dot**: Variance of the coefficients chosen first, second and so forth for frame with OMP and ND$_{limit}$ = 0.17, **Dashed**: Variance of the coefficients chosen first, second and so forth for frame with OMP and ND$_{limit}$ = 0.23

widely used, and choose the most frequently used transform, the DCT. The quantization and coding of the coefficients are done as similarly as possible to the quantization and coding of the coefficients in the frame based compression scheme.

We have a transform based compression scheme as follows: A transform, e.g. the DCT, is used to find the transform coefficients for a signal vector. The coefficients are quantized by a uniform mid-tread quantizer with quantization step Δ. The coefficients are thresholded, that is all the coefficients with values $w \in [-T, T]$ are set to zero. This gives a dead zone in the quantizer as illustrated in Figure 6.8.

A run-length coder is used to indicate the position of the coefficients. A quantized coefficient and the associated run are combined into one symbol. These symbols, along with an End Of Block (EOB) symbol, are entropy coded.

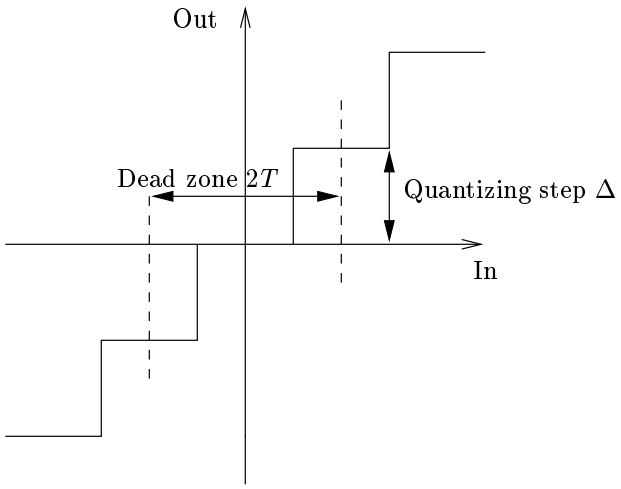


Figure 6.8: Uniform mid tread quantizer with dead zone due to thresholding.

This is somewhat JPEG-inspired. This scheme will work well for a DCT scheme whereas, as revealed in the previous section, entropy coding will not be that advantageous for frame based compression.

6.3 Compression of ECG signals

Some compression experiments using one frame is presented here. The experiments are done using the frames optimized with an MSE_{limit} and with OMP as the vector selection algorithm.

Four different experiments are done to compare frame based compression with the reference compression scheme with entropy coding. Thus the frame coefficients are uniformly quantized and thresholded, run-length and entropy coded as in the DCT scheme.

The four experiments are done with different quantizer steps, Δ, and thresholding factors, $T = \Delta$. In two of the experiments the frames are optimized with MIT100_{train} as the training signal and with different MSE_{limit}'s. The other two frames are optimized with MIT_{mix} as the training set, and also with two different MSE_{limit}'s. In all four experiments the frames are tested on MIT100_{test}. The results of the experiments are shown in Figure 6.9.

The frames trained on MIT100_{train} performs better than the DCT scheme, but the frames trained on MIT_{mix} performs poorer. This shows potential in

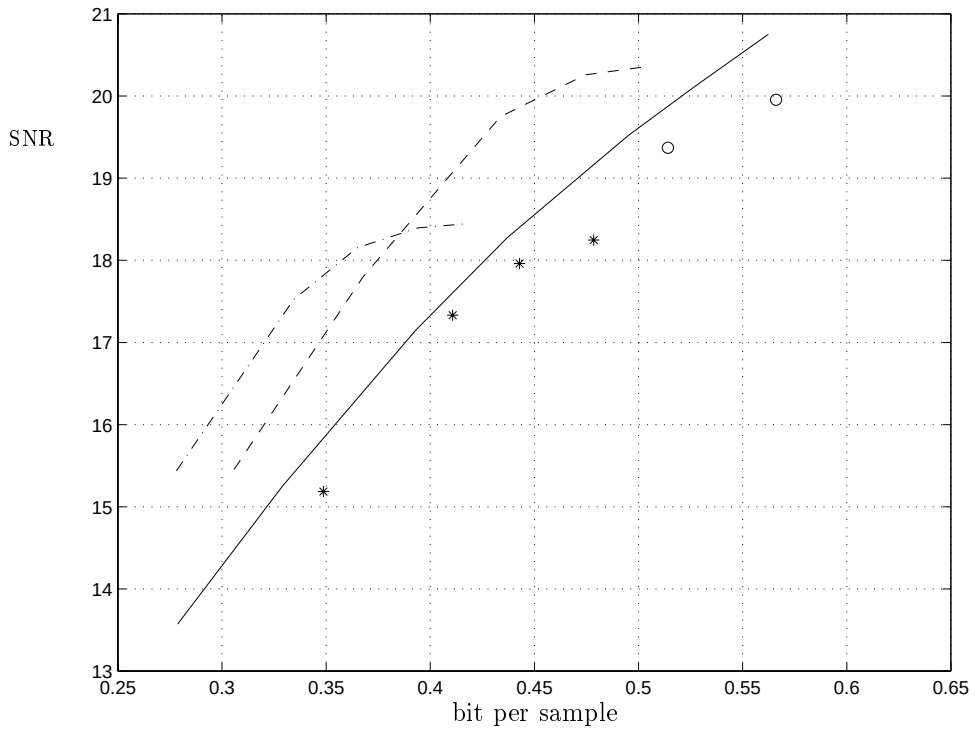


Figure 6.9: Compression experiments on MIT100$_{test}$. SNR in dB is plotted as a function of the number of bit per sample. **Solid**: DCT, **dashed**: ND$_{limit}$=0.17, and **dash-dotted**: ND$_{limit}$=0.23, both trained on MIT100$_{train}$. **o**: ND$_{limit}$=0.17 and ***:** ND$_{limit}$=0.23, both trained on MIT$_{mix}$.

using frame based compression, but also indicates that the frames need to be tailored for the type of signal to be compressed.

6.4 Compression of images

Some compression experiments using a frame based compression scheme was done on the test images Lena and Jet. The frames used in the experiments were all trained on the six training images using FOMP as the vector selection algorithm, and a limit on the MSE to provide sparsity. Six frames are used, trained with MSE$_{limit}$'s at 40, 70, 100, 150, 200, and 250 corresponding to ND$_{limit}$'s at 0.11, 0.14, 0.17, 0.20, 0.24, 0.26. JPEG experiments on the same images and with different quality factors were performed for comparison.

6.4.1 Coding of image representation

After using a frame to represent the image, in a non-separable way as described in Section 5.2, the representation consists of coefficient values and position information. Since a limit on the MSE is used to provide sparsity, the number of coefficients will vary from image block to image block. The values are quantized with a uniform quantizer with quantizer step Δ and thresholded with $T = \Delta$. Let $\mathbf{X}^M$ be a 64×4096 matrix ($M = 4096$) consisting of the image data of size 512×512, where a column in $\mathbf{X}^M$ is the lexicographic ordering of a 8×8 image block.

$$\mathbf{X}^M \simeq \hat{\mathbf{X}}^M = \mathbf{F}\mathbf{W}^M, \tag{6.1}$$

where $\mathbf{F}$ is the 64×128 frame and $\mathbf{W}^M$ is a matrix of size 128×4096 containing all the coefficient information, or the image representation.

Possible coefficient values are $\ldots -3\Delta, -2\Delta, -\Delta, \Delta, 2\Delta, 3\Delta \ldots$ and these are mapped into $\ldots 6, 4, 2, 1, 3, 5 \ldots$. The position information is handled as run, that is the number of zeros between two nonzero coefficients. Since we know the size of the frame EOB (End Of Block) information is not needed. We just continue to count zeros in the next block until we get to the first nonzero coefficient. To limit the number of different possible runs we make an exception when all the coefficients in a column of $\mathbf{W}$ are zero. In this case the run is set to $N = 64$, and the respective value to zero. After this reorganization of the coefficients we have a string of runs and a string of values. Both the runs and the values are now entropy coded using Huffman tables. The final bit per pixel rate is calculated including the Huffman side information for the image.

6.4.2 Image compression results

In Figure 6.10 the results of the experiments are plotted with Peak Signal to Noise Ratio (PSNR) as a function of bit per pixel. The PSNR is a common measure of image quality and is calculated as:

$$\text{PSNR} = 10 \log_{10}(\frac{255^2}{\text{MSE}}) \tag{6.2}$$

since 255 is the largest possible pixel value in an 8 bit per pixel image. The MSE is calculated as in Equation 2.9. When calculating the residuals in all the image experiments, the pixel values of the reconstructed image are quantized back to the 0-255 possible values of the 8 bit per pixel format of the original image.

The dashed curves show the JPEG performance on the test images with different quality factors. The solid curves show frame based compression with different MSE_{limit}'s. Figure 6.10 a) shows results from compression of the test image Lena. $\text{MSE}_{limit} = 70, 100, 150, 200$ and 250 are used, corresponding to ND_{limit}'s at 0.16, 0.19, 0.23, 0.27, and 0.30. Figure 6.10 b) shows results from compression of the test image Jet. $\text{MSE}_{limit} = 40, 70, 100, 150, 200$ and 250 are used, corresponding to ND_{limit}'s at 0.28, 0.38, 0.45, 0.55, 0.64, and 0.71.

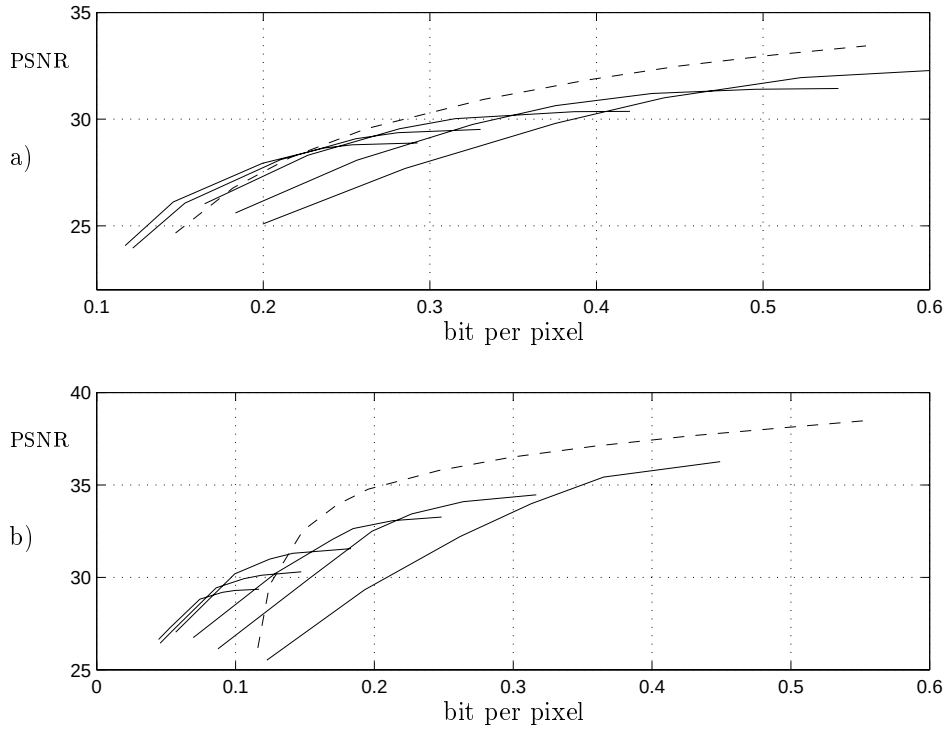


Figure 6.10: Compression experiment on a) Lena and b) Jet. Training and testing is done using MSE_{limit} and FOMP. PSNR in dB is plotted as a function of bit per pixel. **Dashed**: JPEG, **solid**: Different MSE_{limit}'s as a function of different quantizing step, Δ.

Figure 6.10 shows that JPEG outperforms the frame based compression scheme except for extremely low bit-rates where the frame based compression performs best.

Figure 6.11 shows the reconstructed test image Lena after being compressed with the frame based scheme to 0.15 bit per pixel. Figure 6.12 shows the

reconstructed test image Lena after being compressed with JPEG at the same bit rate, 0.15 bit per pixel. By inspecting the reconstructed images it is obvious that the frame based scheme performed better at this bit rate. The PSNR for the frame based scheme is 26.1 dB and PSNR= 24.7 dB for JPEG.

Figure 6.11: Reconstructed test image Lena after being compressed to 0.15 bit per pixel using a frame with $\mathrm{ND}_{limit} = 0.30$ and $\Delta = 100$. PSNR=26.1 dB

Figure 6.12: Reconstructed test image Lena after being compressed to 0.15 bit per pixel using JPEG. PSNR=24.7 dB

The reconstruction of the test image Jet after being compressed to 0.1 bit per pixel using the frame based scheme is depicted in Figure 6.13. The reconstructed image after JPEG compression at 0.11 bit per pixel (it is impossible to get a JPEG compression at 0.1 bit per pixel) can be seen in Figure 6.14. The frame based scheme performs much better as can easily be seen by comparing the images. The frame based scheme gives a PSNR at 30.3 dB whereas the JPEG gives PSNR= 25.9 dB.

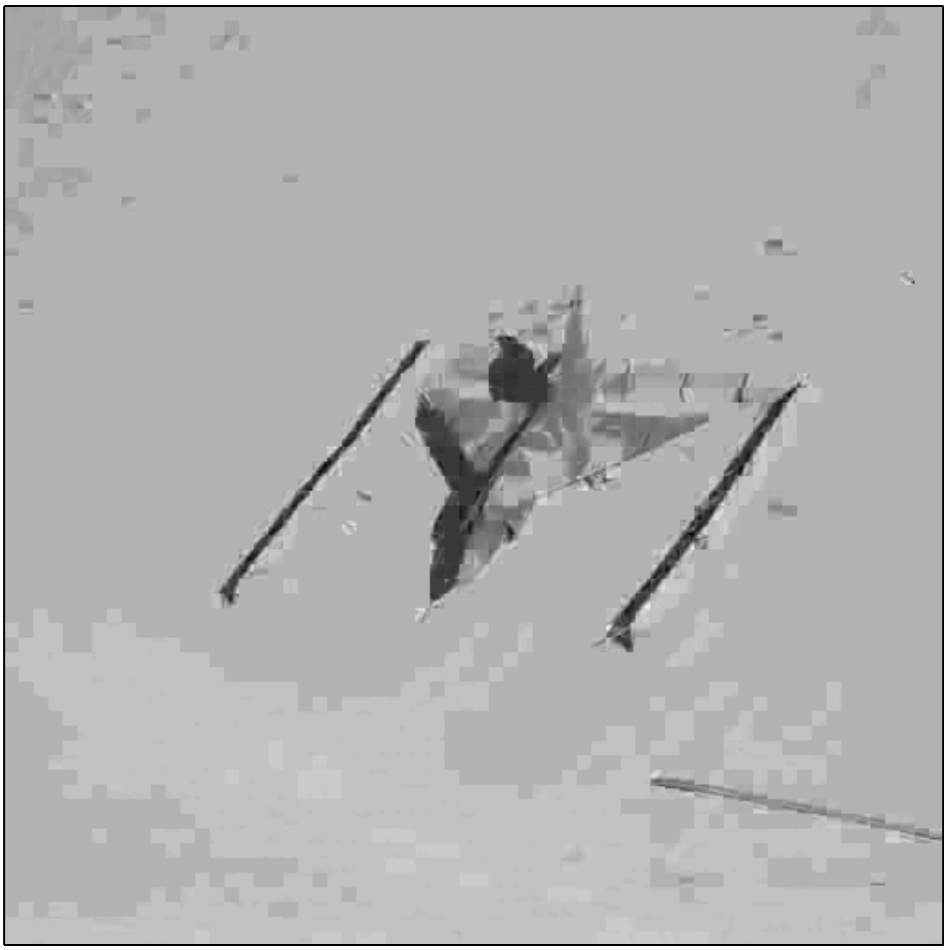


Figure 6.13: Reconstructed test image Jet after being compressed to 0.10 bit per pixel using a frame with $\mathrm{ND}_{limit} = 0.55$ and $\Delta = 60$. PSNR=30.3 dB

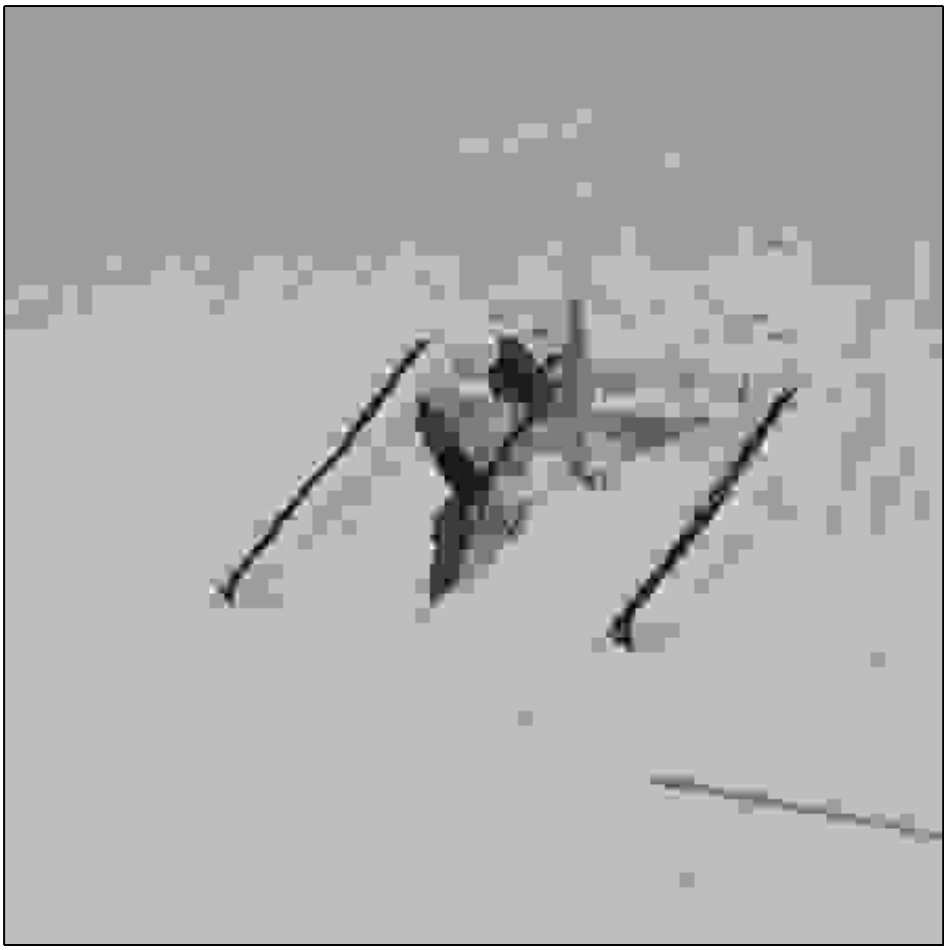


Figure 6.14: Reconstructed test image Jet after being compressed to 0.11 bit per pixel using JPEG. PSNR=25.9 dB

6.5 Discussion

The compression experiments show some good results, but are not as convincing as the approximation capability results presented in the previous chapter. The chosen coding strategy and the chosen reference compression scheme is however a matter of discussion. All the compression schemes use entropy coding, and as stated in Section 6.1 this favors DCT based schemes more than frame based schemes.

In compression of ECG signals the coding of the coefficients in the reference scheme and the frame based scheme are identical. In the image compression experiment JPEG is used as reference scheme and in JPEG the coding of the coefficients have been given a lot of consideration. Giving it more consideration, we may find a more efficient way of coding the image representation in the frame based scheme.

The approximation capabilities of frames trained for using a predetermined number of vectors in each approximation where shown in Chapter 5 to be significantly better than DCT. In Chapter 5 it was also shown that frame vectors in a frame trained for using i vector in each approximation where different from frame vectors in a frame trained trained for using j vector in each approximation, where $i \neq j$. This motivates for using several frames designed for different sparsity factors in a compression scheme. This scheme is explored in the next chapter, and it gives considerable better compression results than the experiments using *one* frame in this chapter.

Chapter 7

Multi Frame Compression, MFC

In this chapter the Multi Frame Compression (MFC) scheme [20, 19] is presented.

In Chapter 5 it was shown that frames designed for using *one* vector in each approximation was quite different from frames designed for using *two, three, four*, and so forth frame vectors in each approximation. This motivates the idea of using a set of frames designed for different numbers of frame vectors in each approximation instead of using a single frame as in Chapter 6. A frame set would give much more flexibility in forming the approximation than a single frame. Some extra side information will be needed to tell which frame is used approximating a signal block, but knowing which frame is used we also know exactly how many coefficients different from zero in that block, thus the extra side-information turns out to be very small.

7.1 The Multi Frame Compression (MFC) scheme

The main idea in MFC is to use a set of frames in the compression scheme, letting one frame be optimized when using just *one* vector in each approximation, one when using *two* and so forth. When compressing a signal vector the number of vectors required to fulfill an MSE condition will decide which frame to use. The frame notation is: $\mathbf{F}_i$, $i = 1, 2, \ldots, L$, where L is the maximum number of vectors allowed in an approximation, is a frame of size $N \times K, K \geq N$ optimized for using i vectors in each approximation.

Figure 7.1 illustrates the compression scheme. An MSE_{target} is decided. The vector selection scheme should be the same as the one used when designing the frames. There is an order relation between the frames. $\mathbf{F}_1$ is designed for, and used only if the signal block is approximated using only *one* vector and coefficient. The same goes for $\mathbf{F}_2$, but where *two* vectors and coefficients are used, and so forth. For compression of a signal it is desirable to use as few coefficients as possible to satisfy the MSE_{target}. Since entropy coding is used there is a possibility that using x vectors and coefficients could require fewer bits than $x-1$ vectors and coefficients for some signal vectors, but fewer coefficients means lower bit rate in general. The frame selection system in MFC works by letting a signal vector first be approximated using just one coefficient, if that is not good enough in the MSE sense, two is tried, and so forth. The coefficients corresponding to the selected frame vectors from the selected frame are quantized by a uniform mid tread quantizer and thresholded. The quantized coefficients are run-length and entropy coded. There is an entropy coder designed for each frame, and each signal block starts with a Beginning Of Block (BOB) symbol.

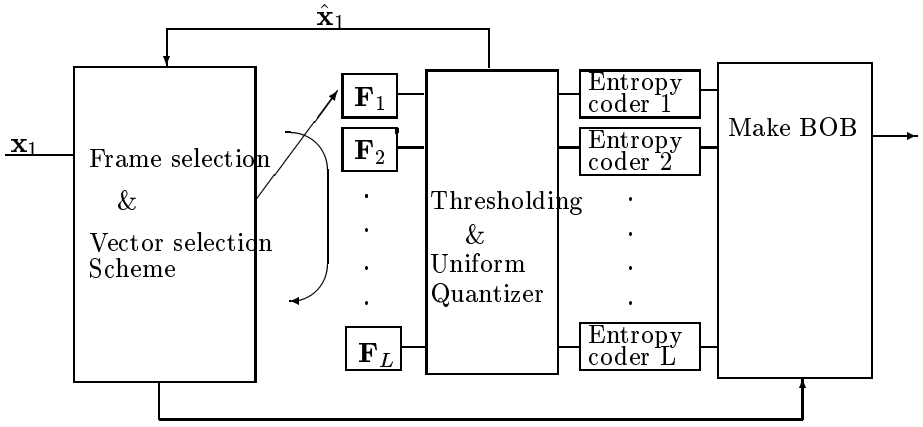


Figure 7.1: Illustration of the compression scheme.

7.1.1 Representation of multi frame coefficients

In the reference transform compression scheme the number of coefficients *not* quantized to zero will vary for different signal vectors. In the context of frame expansions this means that for a given quantizer step the number of vectors needed in the approximation will vary for different signal vectors if

the approximation quality in terms of MSE is to be approximately constant throughout the signal. The proposed compression scheme, MFC, uses several frames where each frame is designed for use with *a fixed number of frame vectors in each approximation.* In MFC the desired approximation quality will decide the number of vectors to be used in the approximation of a signal block.

If only one frame is used in conjunction with run-length entropy coding it makes sense to use an EOB symbol between each signal block in the same way as done in JPEG. When using MFC, a BOB symbol is needed to tell which frame is used when approximating the next signal vector. The BOB tells that frame $\mathbf{F}_i$ is used, and then it is clear that exactly i symbols, each consisting of an amplitude and a run, will be transmitted before the next BOB. This means that we can use run-length coding and entropy coding where each frame has its own entropy coder. This is a strong advantage because the optimal entropy coders can be very different for the different frames.

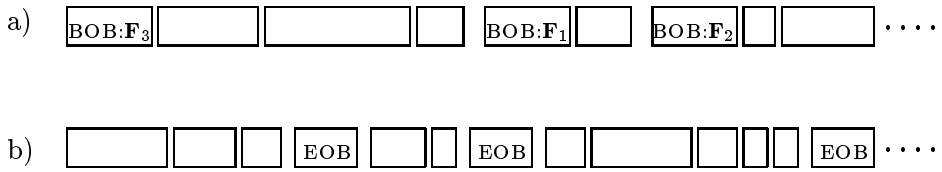


Figure 7.2: a) MFC scheme using BOB symbols to tell which frame is used in approximating the next signal block. b) Reference transform coding scheme, or compression with one frame. An EOB symbol is needed to separate the consecutive signal blocks.

Figure 7.1.1 illustrates the use of BOB and EOB. The boxes indicate an entropy coded symbol, i.e. a bit sequence of variable length. As opposed to the reference transform based compression scheme, in MFC we know exactly *when* the next BOB is coming. Thus there is no need to let BOB be a special word to be recognized, as an EOB has to be. This means the BOB symbol can also be entropy coded.

Experiments show that the BOB requires more bit than the EOB, but the difference is relatively small. Thus the use of several different frames, optimized for different numbers of vectors, and with their own entropy coder, requires very little extra side information compared to using one frame, one entropy coder and an EOB symbol.

For very low bit rates, like 0.2 - 1 bit per sample in ECG experiments, the probability of using $\mathbf{F}_1$ or $\mathbf{F}_2$ is much larger than using $\mathbf{F}_i$, $i = 3, 4, \ldots, L$.

Thus the entropy of the BOB symbols is low. If the BOB symbols are Huffman coded, the extra side information when using these BOB instead of EOB symbols is typically less than 0.03 bit per sample in our ECG experiments.

As in the reference transform based coder described in Section 6.2.1, the coefficients are quantized by a uniform mid tread quantizer with quantizing step, Δ, and thresholded by T. A run-length coder is used to indicate the position of the coefficients. A quantized coefficient and the associated run are combined into one symbol, and these symbols are entropy coded with separate entropy coders for each frame.

7.1.2 Multi frame compression: Main algorithm

The MFC scheme works as follows: When compressing a signal vector, the MFC scheme starts by using $\mathbf{F}_1$ to approximate the signal vector, the coefficients are quantized, and the residual, $\mathbf{r}_l$, is calculated. The error energy, $\|\mathbf{r}_l\|^2$, is compared to an MSE_{target}. If the approximation is good enough $\mathbf{F}_1$ is used, if not $\mathbf{F}_2$ is tried and so forth. For frame $\mathbf{F}_i$ a signal vector, $\mathbf{x}_l$, is approximated as shown in Equation 4.1 where only i of the $w_l(j)$'s are non-zero.

There is a strong connection between the quantization step, Δ, the threshold, T, and MSE_{target}. For a target bit-rate there is an optimal combination of MSE_{target}, Δ, and T. These factors can be incorporated into a *quality factor* as was done in JPEG image coders.

For a signal vector approximated with frame $\mathbf{F}_i$ it is possible that one or more of the i coefficients are quantized to zero even though the requirement on the MSE_{target} is *not* satisfied. In this situation it is not always beneficial to increment i, and try with the next frame. If that solution were to be chosen, and the Δ is large compared to the optimal Δ for the MSE_{target}, the scheme would often use the frames with the maximum numbers of vectors allowed in an approximation, which increases the bit-rate. If one coefficient is quantized to zero when using frame $\mathbf{F}_i$, only $i-1$ frame vectors are used in the approximation, and it would be better to use frame $\mathbf{F}_{i-1}$ which is designed for using $i-1$ vectors. An idea is therefore to decrement i and go back to the previous frame, even though we know that this approximation was not good enough compared to the MSE_{target}. This way the compression scheme always tries to use as few vectors as possible in each approximation, but we loose some control over the local MSE for some of the signal vectors. This resulted in good overall rate-distortion performance, but for some signal vectors the error energy became very large. We dealt with this problem in the following way: When the above described situation occurs for frame $\mathbf{F}_i$, calculate the

residual, and error energy, when using frame $\mathbf{F}_{i-1}$. If this error energy is less than $factor \times \text{MSE}_{target}$, the frame $\mathbf{F}_{i-1}$ is used in the approximation of the signal vector. If not, i is incremented and the approximation using frame $\mathbf{F}_{i+1}$ is calculated. This way we allow the error energy to be larger than MSE_{target} for some of the signal vectors. Typically this happens for signal vectors with large energy, that is in signal regions with large local variance. The signal to noise ratio (SNR) is

$$\text{SNR} = 10 \log_{10} \frac{\sigma_x^2}{\text{MSE}} \tag{7.1}$$

so the MSE can be larger in a signal region with large local variance but still the local SNR, and the visual quality, can be approximately the same. The algorithm implementing the MFC scheme can be summarized as follows:

1. A desired approximation quality, MSE_{target}, is chosen in terms of a target MSE for the overall signal. Assign counter variable $i = 1$.

2. A vector selection algorithm is used to find the approximation when using $\mathbf{F}_i$.

3. The coefficients are quantized with a uniform quantizer with quantizing step Δ, and they are thresholded with $T = \Delta$. The residual after quantization is calculated, and the MSE_i is compared to MSE_{target}.

4. If $i = L$ go to 8.

5. If $\text{MSE}_i < \text{MSE}_{target}$ go to 8.

6. If none of the coefficients are quantized to zero,
   $i = i + 1$ and go to 2.

7. If $\text{MSE}_{i-1} < factor \times \text{MSE}_{target}$, $i = i - 1$ and go to 8. Else $i = i + 1$ and go to 2.

8. $\mathbf{F}_i$ is used when approximating the signal vector. The approximation is entropy coded. Each frame has its own entropy coder.

9. A BOB symbol telling which frame was used is entropy coded and prepended to the bits resulting from step 8.

7.2 Variable sized frames

This section concerns the use of variable sized frames in the MFC scheme, and it was first tried in [21]. As in the previous section, let $\mathbf{F}_i$ denote an $N \times K_i$ matrix where $K_i \geq N$. The columns, $\{\mathbf{f}_j\}$, $j = 1, \ldots, K_i$, constitute a frame. The vector length, N, is constant whereas the *number of vectors in a frame*, K_i, can be different for different i's. The key idea here is to use large frames, i.e. large K, when using a small number of frame vectors in an approximation of a signal vector, and smaller frames when using more vectors in each approximation. When the approximation of a signal vector consists of many vectors an ordinary DCT will give good results, a large frame is not necessary, and using a smaller frame will lower the average entropy because there are fewer possible different output symbols. When using very few vectors in each approximation a large frame will provide a good chance of finding a small number of vectors whose linear combination match the signal vector well, and perform better than a small frame.

7.2.1 Rationale for using variable frame size

We here make the assumption that all the output symbols have equal probability. This means that for each frame in the MFC scheme every combination of quantized coefficient value and run has the same probability. This assumption makes it possible to do calculations on the bit rate for different sized frames. The assumption is unrealistic, but still this gives a rationale for using variable sized frames in the MFC scheme. In an experiment we use the actual histogram-based probabilities.

Let N be the block size, and Q the number of different quantizer steps. Let m be the number of different possible output symbols, i.e. combinations of a quantized value and a run. We assume that all the output symbols have the same probability of occurring, and this gives $\log_2 m$ bits per output symbol. For simplicity, we consider a scheme with two frames, $\mathbf{F}_1$ and $\mathbf{F}_2$, where the size of both frames is $N \times K$. With K different frame vectors, we have K different possible runs. This is obvious for frame $\mathbf{F}_1$, and it is also correct for frame $\mathbf{F}_2$ since there is a possibility for one of the coefficients to be quantized to zero. This gives KQ possible combinations of run and quantized coefficient, thus $m = KQ$.

For a given signal vector $\mathbf{x}_l$, let D_i be the distortion using frame $\mathbf{F}_i$, and R_i the bit rate. Let us assume that:

$$\mathbf{x}_l : D_1 > \mathrm{MSE}_{target} \Rightarrow \mathbf{F}_2 \text{ is used} \Rightarrow D_2, R_2. \tag{7.2}$$

We increase the size of $\mathbf{F}_1 \Rightarrow \tilde{\mathbf{F}}_1$, to $N \times (K+1)$, and try again. Assume now that:

$$\mathbf{x}_l : \tilde{D}_1 < \text{MSE}_{target} \Rightarrow \tilde{\mathbf{F}}_1 \text{ is used} \Rightarrow \tilde{D}_1, \tilde{R}_1. \tag{7.3}$$

We know that $\tilde{R}_1 > R_1$ and it is realistic to assume that $\tilde{D}_1 \approx D_2$. Assume that $\tilde{D}_1 = D_2$. First we show that $\tilde{R}_1 < R_2$, so that it is an improvement for $\mathbf{x}_l$ and the other signal vectors which now can use $\tilde{\mathbf{F}}_1$ instead of $\mathbf{F}_2$:

$$\begin{aligned}
\tilde{R}_1 &= \frac{1}{N}\log_2[(K+1)Q] = \frac{1}{N}[\log_2(K+1) + \log_2 Q] && (7.4)\\
R_2 &= \frac{2}{N}\log_2(KQ) = \frac{2}{N}[\log_2 K + \log_2 Q] && (7.5)\\
R_2 - \tilde{R}_1 &= \frac{1}{N}[\log_2 Q + 2\log_2 K - \log_2(K+1)] && (7.6)\\
&= \frac{1}{N}\log_2(\frac{QK^2}{K+1}) && (7.7)
\end{aligned}$$

If $\tilde{R}_1 < R_2$ then $R_2 - \tilde{R}_1 > 0$. This means that $\frac{QK^2}{K+1} > 1$. Q and K are positive integers, so this will be true for all $Q \geq 1$ and $K \geq 2$ which means for all practical purposes.

Secondly we will derive conditions indicating when the enlargement of $\mathbf{F}_1$ leads to a total improvement in the rate-distortion sense, considering all the signal vectors. All the signal vectors that first used $\mathbf{F}_1$ now have to use $\tilde{\mathbf{F}}_1$, and the bit rates for these signal vectors are enlarged from R_1 to $\tilde{R}_1$. It is possible that the distortion will get smaller in some of these blocks due to the fact that there is an extra vector to choose from. We look at the worst case where the distortion remains the same, and the bit rate increases.

Let p_1 be the probability that $\mathbf{F}_1$ is used, and p_2 that $\mathbf{F}_2$ is used when we compress with the same sized frames. After changing $\mathbf{F}_1$ to $\tilde{\mathbf{F}}_1$, $p_1 + q$ is the probability that $\tilde{\mathbf{F}}_1$ is used, and $p_2 - q$ that $\mathbf{F}_2$ is used. R_{tot} is the total bit rate using $\mathbf{F}_1$ and $\mathbf{F}_2$, and $\tilde{R}_{tot}$ is the total bit rate using $\tilde{\mathbf{F}}_1$ and $\mathbf{F}_2$.

$$\begin{aligned}
R_{tot} &= p_1 R_1 + p_2 R_2 && (7.8)\\
\tilde{R}_{tot} &= (p_1 + q)\tilde{R}_1 + (p_2 - q)R_2 && (7.9)
\end{aligned}$$

For $\tilde{R}_{tot}$ to be smaller than R_{tot}:

$$q(R_2 - \tilde{R}_1) > p_1(\tilde{R}_1 - R_1) \tag{7.10}$$

With $\tilde{R}_1$ and R_2 given by Equation 7.4 and 7.5, and

$$R_1 = \frac{1}{N}\log_2(KQ) = \frac{1}{N}[\log_2 K + \log_2 Q], \tag{7.11}$$

we have the requirement

$$\frac{q}{p_1} > \frac{\tilde{R}_1 - R_1}{R_2 - \tilde{R}_1} = \frac{\log_2 \frac{K+1}{K}}{\log_2 \frac{QK^2}{K+1}}. \tag{7.12}$$

If Equation 7.12 is true, it will be better in a rate-distortion sense to add an extra vector to $\mathbf{F}_1$. We can easily do the same calculations when the size is set to be for example $N \times jN,\ j = 1, 2 \ldots$. Let $\mathbf{F}_i,\ i = 1, 2$ have size $N \times jN$, and $\tilde{\mathbf{F}}_1$ have size $N \times (j+1)N$. The requirement for improvement in the bit rate will then be:

$$\frac{q}{p_1} > \frac{\tilde{R}_1 - R_1}{R_2 - \tilde{R}_1} = \frac{\log_2 \frac{j+1}{j}}{\log_2 \frac{NQj^2}{j+1}}. \tag{7.13}$$

For example, if $N = 16, Q = 100,$ and $j = 2$:

$$q > \frac{p_1}{18.9} \tag{7.14}$$

This means that the number of signal vectors that originally used $\mathbf{F}_2$ but now use $\tilde{\mathbf{F}}_1$, has to be greater than $\sim \frac{1}{19}$ times the number of signal vectors that used $\mathbf{F}_1$ in the first place (and now use $\tilde{\mathbf{F}}_1$).

An experiment using two frames was performed. $\mathbf{F}_1$ and $\mathbf{F}_2$ have size $N \times N$, and $\tilde{\mathbf{F}}_1$ has size $N \times 2N$. The frames were trained on MIT100_{train}, and test experiment was executed on MIT100_{test}. The test signal was compressed using $\mathbf{F}_1$ and $\mathbf{F}_2$, and then using $\tilde{\mathbf{F}}_1$ and $\mathbf{F}_2$. In both cases MSE_{target} was constant, and Δ was varied. The output symbols do not have equal probability, so instead of using Equation 7.14 directly, the total bit rates $\tilde{R}_{tot}$ and R_{tot} were computed using the histogram-based probabilities. They are compared in Table 7.1.

For $\Delta = 1$ and $\Delta = 2$, $\tilde{R}_{tot} < R_{tot}$ and the scheme using $\tilde{\mathbf{F}}_1$ and $\mathbf{F}_2$ performs better than the one using $\mathbf{F}_1$ and $\mathbf{F}_2$. From Table 7.1 it can be seen that Equation 7.14 here is changed to approximately

$$q > \frac{p_1}{12}, \tag{7.15}$$

due to the nonuniform probability distribution to the output symbols.

Figure 7.3 shows rate-distortion plots for the two experiments. The o's and the *'s in the figure represent the MSE and the bit rates for the different Δ's in Table 7.1. The figure shows that the use of $\tilde{\mathbf{F}}_1$ instead of $\mathbf{F}_1$ works better for $\Delta =$ 1, 2, 5, 7, 10, 12, 15, 18, 20, and 25. For $\Delta = 1, 2$ it is obvious because

Δ	q	p_1	$\tilde{R}_{tot}$	R_{tot}
1	0.0613	0.7233	0.4402	0.4508
2	0.0613	0.7233	0.4308	0.4358
5	0.0613	0.7227	0.4104	0.4069
7	0.0607	0.7230	0.3992	0.3924
10	0.0604	0.7215	0.3863	0.3770
12	0.0604	0.7218	0.3785	0.3687
15	0.0581	0.7247	0.3672	0.3558
18	0.0542	0.7298	0.3575	0.3437
20	0.0471	0.7416	0.3503	0.3345
25	0.0308	0.7733	0.3329	0.3130
30	0.0119	0.8092	0.3165	0.2919

Table 7.1: Experiment with two frames.

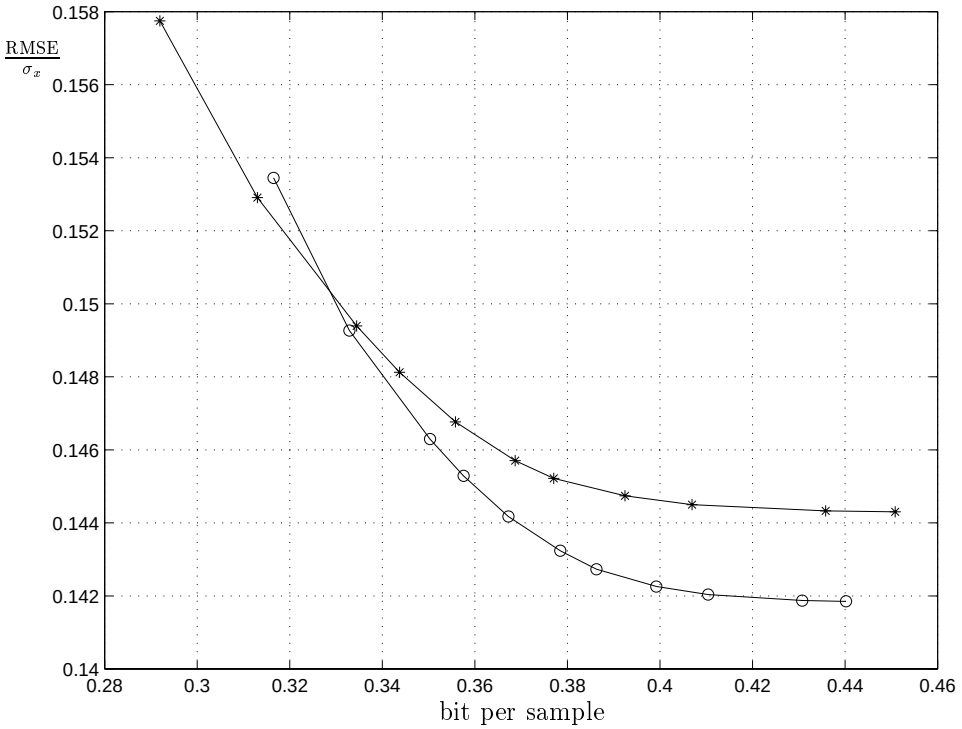


Figure 7.3: Rate-distortion plot for two experiments using two frames with variable size. *: $\mathbf{F}_1$ and $\mathbf{F}_2$, o: $\tilde{\mathbf{F}}_1$ and $\mathbf{F}_2$. Normalized distortion is plotted as function of bit per sample.

then Equation 7.15 is satisfied. For the rest of the Δ's the explanation lies in the assumption that the distortion is constant, and in fact the distortion is decreasing using $\tilde{\mathbf{F}}_1$ instead of $\mathbf{F}_1$ with the same Δ.

We have considered a scheme with only two frames for simplicity. The same rationale can be used for a scheme with more than two frames. Let the MFC scheme have L frames, $\mathbf{F}_i$, $i = 1, 2 \ldots L$, all of size $N \times K$. The same argument can be used by first enlarging $\mathbf{F}_1$, then enlarging $\mathbf{F}_2$ and so forth.

7.3 MFC experiments on ECG signals

We have done MFC experiments using ECG signals, and OMP is the vector selection algorithm in all the experiments in this section. All the frames are trained using the MOD algorithm of Chapter 4.

The trained frames for ECG signals from Experiment 2 in Chapter 5 are used to form frame sets with frames of equal size. The experiment has frames of size 32×64. Experiments are done with frames trained and tested on different segments on the same patient but also on frames trained on a mixed signals and tested on several signals as in Experiment 2 in Chapter 5.

Some of the same frames, of size 32×64, are used in the experiments with frame sets of variable size. In addition new frames with different sizes $N \times iN$ are trained to form sets of variable sized frames.

In all the experiments we use Huffman coded BOB symbols to tell which frame is used in the signal block representation. The frame coefficients are quantized with uniform quantizer and thresholded with $T = \Delta$, and run-length coded. Run and quantized value are combined into one symbol and entropy coded with different entropy coder for each frame. The *factor* in step 7 of the algorithm was experimently set to 5. The compression experiments are compared with experiments using the DCT based reference compression scheme described in Section 6.2.1.

7.3.1 ECG signal compression experiments using fixed size frames

In the experiment frames of size $N \times K$ were trained targeted at $1, 2, \ldots 12$ vectors in each approximation using MIT100_{train}, MIT207_{train}, and MIT_{mix}. The signal vector size, N, is 32. The number of frame vectors in each frame, K, is $2N = 64$. The initial frame vectors are normalized versions of signal

vectors in the training set as described in Section 5.1.3. Figure 5.5 shows training plots for some of the frames used in the compression experiments.

For different values of the desired approximation quality, MSE_{target}, the quantizing step, Δ was varied. Experiments on the same test signals were also done using the DCT based reference compression scheme described in Section 6.2.1.

The dashed lines in Figure 7.4 shows the rate-distortion results of compression experiments on MIT207_{test} and MIT100_{test} when the frame sets trained on MIT207_{train} and MIT100_{train} were used.

The dashed lines in Figure 7.5 shows the rate-distortion results of compression experiments on MIT100_{test} and MIT101_{test} when the frame set trained on MIT_{mix} was used. MIT101 is from a patient that has not contributed to the MIT_{mix} signal. MIT101_{test} is MIT101, 6:00 to 11:00 minutes.

The results show that the MFC scheme works well. In terms of rate-distortion it is better than traditional transform based techniques like the DCT for low bit rates. Figure 7.4 shows that when using the MFC scheme, the SNR reaches an almost constant level when the bit rate increases. The major reason for this is the specification of a desired approximation quality, MSE_{target}. For a given MSE_{target}, if Δ is reduced to be less than the optimal Δ, the improvement in SNR will be very small, especially for large MSE_{target}. The increase of the bit-rate will also be small. Different MSE_{target} have to be used for different target bit-rates. In our experiments we have been concentrating on low bit rates. The coding scheme is restricted never to use more than $L = 12$ vectors in an approximation. For higher target bit-rates a larger L would be used, but this compression scheme is expected to work best for *low* bit rates.

7.3.2 ECG signal compression experiments using variable sized frames

Several experiments with frames of different sizes were performed. For block size $N = 32$ frames of size $N \times jN \quad j = 1, 2 \ldots 7$, were used. Letting the size of frame $\mathbf{F}_i$ be $N \times N$ there are several possibilities. $\mathbf{F}_i$ can be a nonorthogonal frame designed using MOD, or $\mathbf{F}_i$ can be an ordinary orthogonal transform, e.g. the DCT. When using $\mathbf{F}_i$, only i vectors can be used in the approximation of a signal vector. Thus if $\mathbf{F}_i$ is an orthogonal transform we must set the $N - i$ smallest coefficients to zero, and quantize the i largest coefficients with the uniform quantizer.

There are at least three reasons for using the DCT for some of the frames, $\mathbf{F}_i$, with large i's. One is that the frames, $\mathbf{F}_i$, with large i's are computationally

size	$N \times N$	$N \times 2N$	...	$N \times jN$
	$\mathbf{F}_i^1$	$\mathbf{F}_i^2$	...	$\mathbf{F}_i^j$

Table 7.2: Notation for different sized frames $\mathbf{F}_i$ where i is the number of vectors used in each approximation the frame is designed for.

expensive to design, but since this is done off line it might not be relevant. Another reason is that when many vectors are used in the approximation, the approximation can be good with many different frames, as it can with an orthogonal basis like the DCT. Using DCT for some of the frames also solves the problem of what to do if one or several of the coefficients are quantized to zero. Let $\mathbf{F}_i$, $i = D, D+1, \ldots, L$ all be the ordinary DCT. If one of the D largest coefficients is quantized to zero when using $\mathbf{F}_D$ and the residual is larger than the target residual, we simply use $\mathbf{F}_D$ when approximating the signal vector. There are no reasons to find an approximation using $\mathbf{F}_{D+1}$ because the same coefficient will be quantized to zero. A third reason is that the suboptimality of the vector selection schemes like the OMP/FOMP becomes more significant when many vectors are chosen, and by using a transform like the DCT that problem is avoided.

Even if $\mathbf{F}_i$, $i = D, D+1, \ldots, L$ is the ordinary DCT there has to be $L - D + 1$ different frames in the sense that frame $\mathbf{F}_i$ is restricted to use no more than i vectors in an approximation, and the BOB symbol must distinguish between them.

Complete compression experiments were done using different test signals. The quantizing step, Δ, was varied for different values of the desired approximation quality, MSE_{target}. The results are compared to the DCT based reference compression scheme.

Experiments were done with block size $N = 32$. $\mathbf{F}_i$, $i = 1, 2, \ldots 12$ of size $N \times K_i$ were optimized using MIT100_{train}, MIT207_{train}, and MIT_{mix}. The number of frame vectors in a frame, K_i, is variable. The initial frame vectors are normalized versions of the first K_i signal vectors in the training set (for MIT_{mix} like described in Section 5.1.3), and the frames are trained using MOD. Table 7.2 shows the notation for the different sized frames.

We tried compression experiments with different frame sets, and for this application the following set worked well: $\mathbf{F}_1^7$, $\mathbf{F}_2^5$, $\mathbf{F}_3^3$, $\mathbf{F}_i^2$, $i = 4, 5 \ldots 7$, and DCT for $i = 8, 9 \ldots 12$. This is the set used in the compression experiments shown here. At this stage we lack a good way of deciding on the best frame size for an application, but it is always possible to try different alternatives and find a set that works well. This is a topic for further investigations.

The solid lines in Figure 7.4 shows the rate-distortion results of compression experiments on MIT207_{test} and MIT100_{test} when the frame sets trained on MIT207_{train} and MIT100_{train} were used. The experiments were done with a number of different values on the desired approximation quality, MSE_{target}. In the figure the results are compared to results using the DCT based reference compression scheme and the fixed size frame sets.

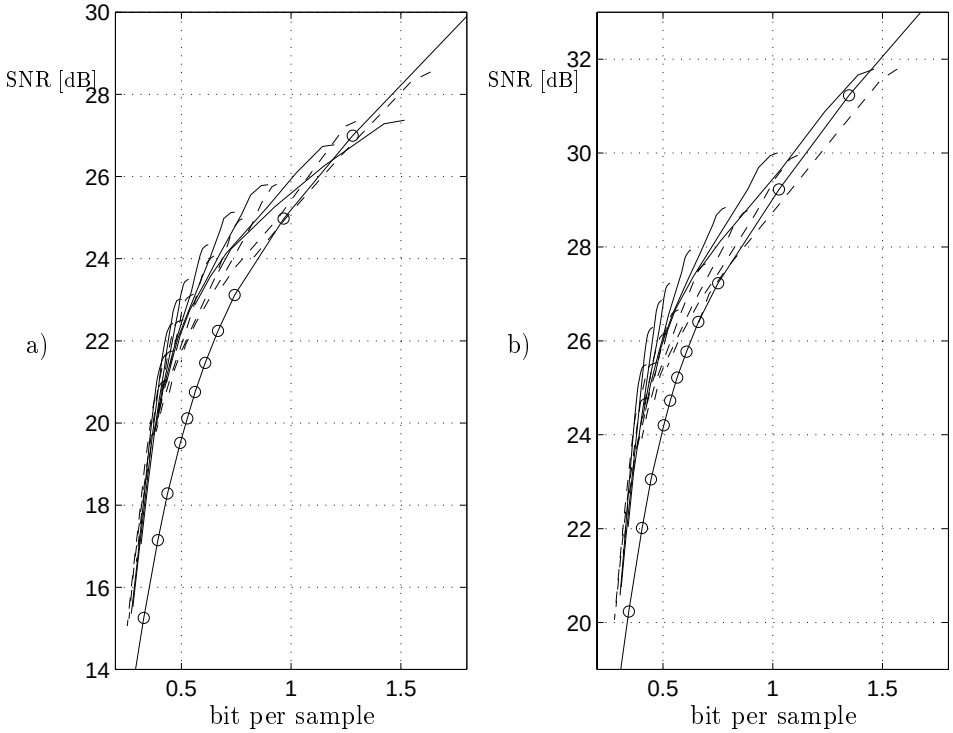


Figure 7.4: Rate-distortion plots. **solid with o's**: DCT, **dashed**: MFC scheme with different MSE_{target} using fixed sized frames, **solid** MFC scheme with different MSE_{target} using variable sized frames. a) Trained on MIT100_{train}, tested on MIT100_{test} b) Trained on MIT207_{train}, tested on MIT207_{test}.

The solid lines in Figure 7.5 shows the rate-distortion results of compression experiments on MIT100_{test} and MIT101_{test} when the frame set trained on MIT_{mix} was used.

A heartbeat from MIT100_{test} and MIT207_{test} are plotted in Figure 7.6 together with reconstructed versions of the same heartbeat compressed using MFC

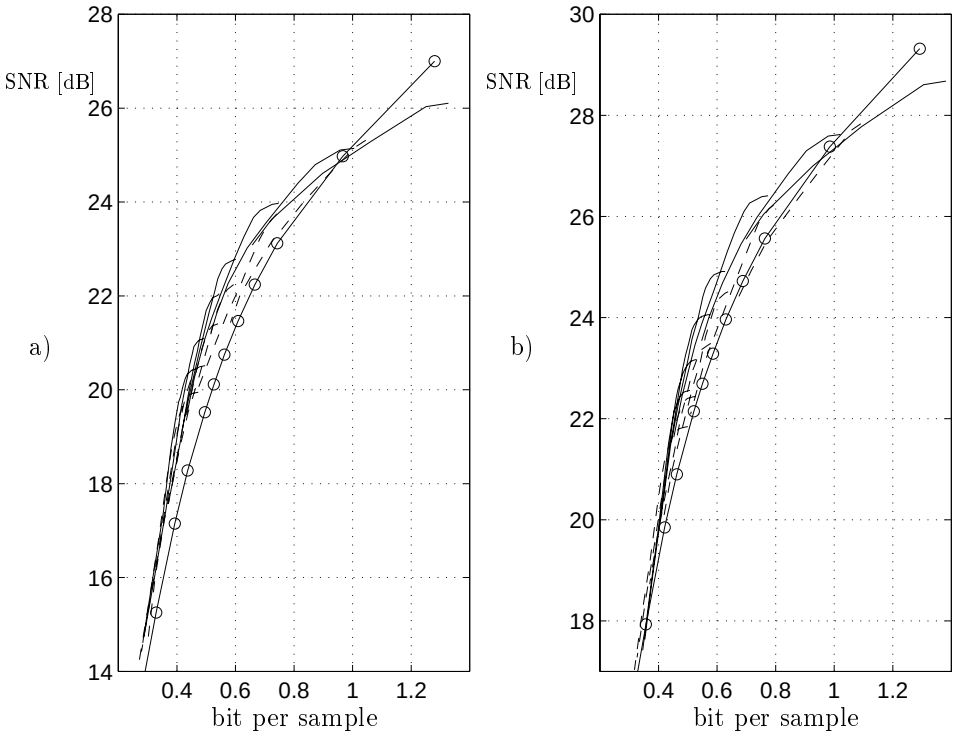


Figure 7.5: Rate-distortion plots. **solid with o's**: DCT, **dashed**: MFC scheme with different MSE_{target} using fixed sized frames, **solid** MFC scheme with different MSE_{target} using variable sized frames. All frames were trained on MIT_{mix}. a) Tested on MIT100_{test} b) Tested on MIT101_{test}.

scheme and the DCT based reference compression scheme. The original signal has 12 bit per sample, and the reconstructed signals were compressed to 0.4 bit per sample.

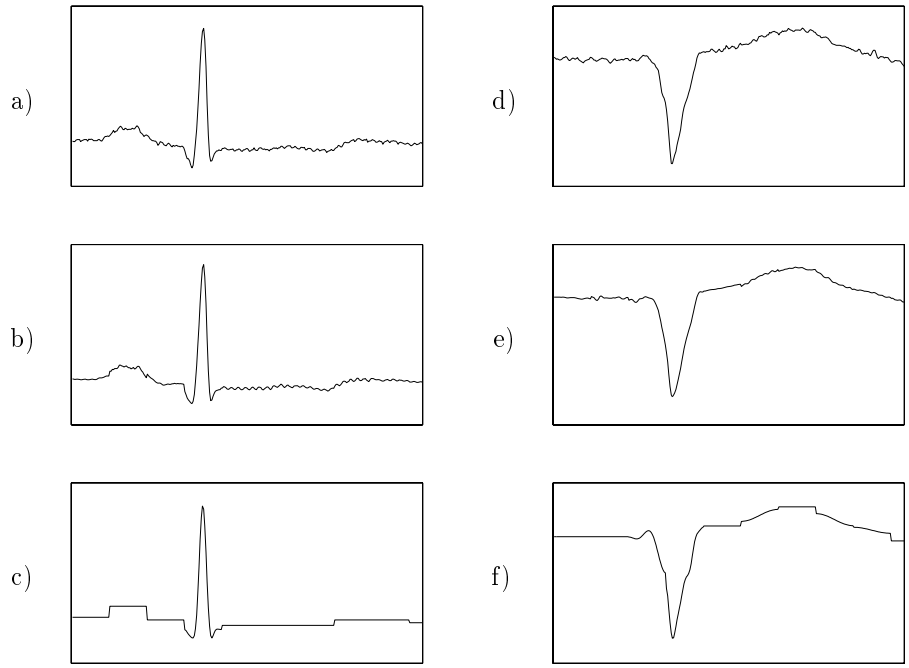


Figure 7.6: Part of test signals and reconstructed signals. a) Part of MIT100_{test}, original b) MFC, 0.4 bit per sample c) DCT, 0.4 bit per sample. d) Part of MIT207_{test}, original e) MFC, 0.4 bit per sample f) DCT, 0.4 bit per sample.

Compared to the results from the DCT based reference compression scheme, the variable sized frames perform very well at low bit rates.

7.3.3 Discussion

The optimized frames are used in an MFC scheme which performs very well at low bit rates when tested on ECG signals. At low bit rates the experiments demonstrate improved rate-distortion performance by 2-4 dB for the MFC scheme when compared to a reference DCT coding scheme, when trained and tested on different time segments from the same patients. We also show that

using variable sized frames instead of fixed sized frames in the MFC scheme, a further improvement of approximately 0.2-1 dB is achieved.

When compressing ECG signals there is sometimes the need of continuously recording the heart beat of a person during a long time period (weeks) for diagnostic reasons. In a situation like this it would be natural to train the scheme for that person before using it. In other applications a more general system that can be used on different persons is needed. On other types of signals similar issues may occur. Therefore we have done experiments covering both these situations.

Comparing Figure 7.4 and Figure 7.5 it can bee seen that the performance is dependent on the training set, as expected. The test on the same patient as the training in Figure 7.4 performs better than the test shown in Figure 7.5. The latter experiment is a frame set trained on a set of signals to design a frame set that can be used on a broader class of signals. The set is tested on both a patient that has produced a part of the training signal, but on a different time segment, and on a patient that has not contributed to the training set. The MFC scheme performs better than the DCT for low bit rates in all the cases, but there is less to gain when the match between the training set and the test signal is decreased.

Comparing Figure 7.4 and Figure 7.5 with Figure 6.9 in Chapter 6 it is easily seen that the MFC scheme performs significantly better than the compression scheme with *one* frame and MSE_{limit}.

7.4 MFC experiments on images

We have done some MFC experiments on images using a set of 12 fixed size frames, size 64×128, trained on the set of training images. The training and testing are done using FOMP as the vector selection algorithm. JPEG experiments on the same images and with different quality factors were performed for comparison.

The BOB symbols, i.e. the information of which frame is used for each image block, are Huffman coded as in the ECG experiments. The value and position information for each of the frames are coded the exact same way as described in section 6.4.1. The final bit rate is calculated including the Huffman side information for the image.

7.4.1 Image compression experiments using fixed size frames

The result of the MFC experiment on the test image Lena is shown in Figure 7.7. The dashed curve shows JPEG compression experiments of the test image with different quality factors. The solid curves show MFC experiments with different MSE_{target}'s. For each of the MSE_{target}, the Δ is varied. The experiments demonstrate improved rate-distortion performance by 0.1-1 dB for the MFC scheme when compared to JPEG at these low bit rates.

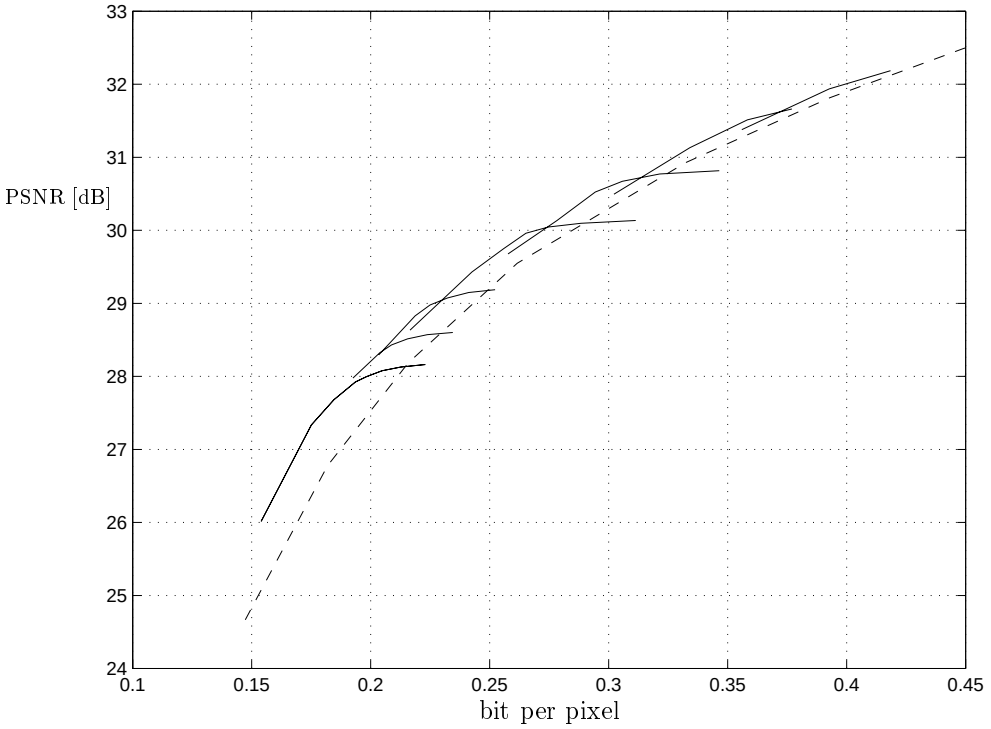


Figure 7.7: MFC experiments on test image Lena. PSNR in dB is plotted as a function of bit per pixel. **Dashed**: JPEG, **solid**: MFC with different MSE_{target}.

Figure 7.8 shows the reconstructed test image Lena after being compressed with the MFC scheme to 0.2 bit per pixel and PSNR = 28.2 dB. Corresponding, Figure 7.9 shows the reconstructed test image Lena after being compressed with JPEG to 0.2 bit per pixel and PSNR = 27.6 dB. Visual inspection proves that the MFC scheme performs better than JPEG for this bit rate.

Figure 7.7 should be compared with Figure 6.10 a) in Chapter 6 to confirm

Figure 7.8: Reconstruction of test image Lena. Compressed to 0.2 bit per pixel using MFC scheme with 12 fixed size frames. PSNR = 28.2 dB

Figure 7.9: Reconstruction of test image Lena. Compressed to 0.2 bit per pixel using JPEG. PSNR = 27.6 dB

that the MFC scheme performs better than compression using one frame in the image experiments, as well as in the ECG experiments.

Chapter 8

Other applications of frames

Again we consider the signal model:

$$\mathbf{x} = \mathbf{F}\mathbf{w} + \mathbf{n}, \tag{8.1}$$

where $\mathbf{x}$ is an $N \times 1$ data vector, $\mathbf{F}$ is an $N \times K$ matrix where $K \geq N$, $\mathbf{w}$ is an $K \times 1$ *sparse* coefficient vector and $\mathbf{n}$ is an $N \times 1$ noise vector. The columns of the matrix $\mathbf{F}$ form an overcomplete set, and spans the space $\mathbf{R}^N$.

Equation (8.1) shows up in several important applications. It can be used as a convenient signal representation model useful for compression, and it can also be a model for the true underlying system that produced the available dataset $\mathbf{x}$. The earlier chapters have mainly been occupied by lossy signal compression where $\mathbf{n}$ represents the reconstruction error, and $\mathbf{F}\mathbf{w} = \hat{\mathbf{x}}$ the approximation of the signal vector $\mathbf{x}$.

In the case of lossy signal compression the *quality of the approximation*, for a given sparsity of $\mathbf{w}$ and a specified $\mathbf{F}$, is of primary importance. On the other hand, if we want to find the true underlying structure that produced the data, finding *the true* $\mathbf{F}$ is essential. This can be the case in applications such as signal reconstruction, estimation, and denoising, or blind source separation for the case when we have fewer sensors than sources.

The next sections will address some of these applications. In an experimental section we show that the frame design algorithm, MOD, described in Chapter 4, used with a noise robust version of FOCUSS called regularized FOCUSS, described in Chapter 3, works well in reconstructing the true $\mathbf{F}$ from the dataset $\mathbf{x}$. The parameters in the regularized FOCUSS are selected according to the modified L-curve method described in Section 3.2.3. The MOD

algorithm has already been shown to work well in designing frames for compression purposes.

The regularized FOCUSS algorithm is a parallel vector selection algorithm, based on minimizing a diversity measure. It works well finding the *correct* sparse vector when the data is made from a *true underlying sparse structure*. The greedy methods, like OMP and FOMP works very well for compression when choosing a relatively small number of vectors from a frame. These algorithms are based on minimizing the MSE in each step, selecting a new vector. In this chapter, where we want to reconstruct the true $\mathbf{F}$, we therefore use the regularized FOCUSS.

8.1 Signal reconstruction and estimation

The problem of finding localized energy solutions from *limited* data arises in many applications. Linear extrapolation problems can be represented as Equation 8.1, with or without the noise vector $\mathbf{n}$. The application can be reconstruction or estimation of data.

In this situation the overcomplete matrix $\mathbf{F}$ represents an operator that maps the unknown data $\mathbf{w}$ to a limited data set $\mathbf{x}$, and the noise vector, $\mathbf{n}$, is discarded. Equation 8.1 is underdetermined and has an infinite number of solutions. A common solution is the minimum norm solution, which is computed from the pseudo inverse:

$$\mathbf{w}_{mn} = \mathbf{F}^T(\mathbf{F}\mathbf{F}^T)^{-1}\mathbf{x}. \tag{8.2}$$

This solution tends to spread the energy over all of or a large number of the entries in $\mathbf{w}$. If a localized energy solution is desired as a consequence of information about the problem, so that the energy is concentrated in only a few of the entries in $\mathbf{w}$, a sparse solution to Equation 8.1 is needed.

Gorodinitsky and Rao [29] have done work using this sparsity model for functional imaging of the brain using EEG or MEG signals. In their work, the content of the frame $\mathbf{F}$ was obvious, and there was no need for frame design. Still the example shows that the model in Equation 8.1 can be useful in many applications and the content of $\mathbf{F}$ may in general not be known so that the frame needs to be reconstructed or estimated as well as the data set, $\mathbf{w}$.

In the latter case Olshausen and Field have done work where they try to find a model of some of the response properties of neurons in primary visual cortex [57]. If the theory is that the neurons actually work according to the model of Equation (8.1) with a sparse $\mathbf{w}$, it is desirable to find the true $\mathbf{F}$ and $\mathbf{w}$.

8.2 Blind source separation

Blind source (signal) separation, also called the *cocktail party problem* due to the way the human brain can distinguish between different speakers in a noisy environment, is a kind of filtering problem that occurs in many different situations [36]. The blind separation problem was introduced by Herault and Jutten in 1986 [37] and has been given considerable attention since that. Many of the earlier suggested solutions were somewhat ad hoc but in recent years more mathematical methods have been proposed, like Bell and Sejnowski's infomax approach [5].

To formulate the blind source separation problem, consider a set of unknown source signals $w_i(n), i = 1, 2 \dots K$. The source signals are mutually independent of each other. Unknown factors, represented by an unknown nonsingular matrix $\mathbf{F}$, mixes the source signals linearly, and a set of observation signals $x_i(n), i = 1, 2 \dots N$ results:

$$\mathbf{x}(n) = \mathbf{F}\mathbf{w}(n). \tag{8.3}$$

The need for blind source separation arises in many application including

- *Speech separation* where the independent sources are different speakers. The speech signals have been mixed together and needs to be separated [5].
- *Array antenna processing:* Separation of multiple co-channel digital signals received by an antenna array [66].
- *Multisensor biomedical records* where the observed signals can be recordings from a multitude of sensors used to monitor biological signals of interest [9].
- *Financial market data analysis* where the observed signals are different stock market data and one wants to find the set of independent dominant components in the marked [4].

In the traditional blind source problem there is as many observed signals as independent sources: $N = K$. In this case we know that $\mathbf{F}$ is an $N \times N$ matrix, thus it is invertible since the independence of the sources provides full rank. The blind source problem is usually formulated as finding an estimate of the $\mathbf{F}^{-1}$ matrix and use that to find estimates of the source signals $\hat{\mathbf{w}}(n)$:

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{F}}^{-1}\mathbf{x}(n). \tag{8.4}$$

The blind source separation problem is sometimes summarized as:

- *Given L independent realizations of the observation vector* $\mathbf{x}$ *find an estimate of the inverse of the mixing matrix* $\mathbf{F}$ [36].

An illustration of the problem is showed in Figure 8.1.

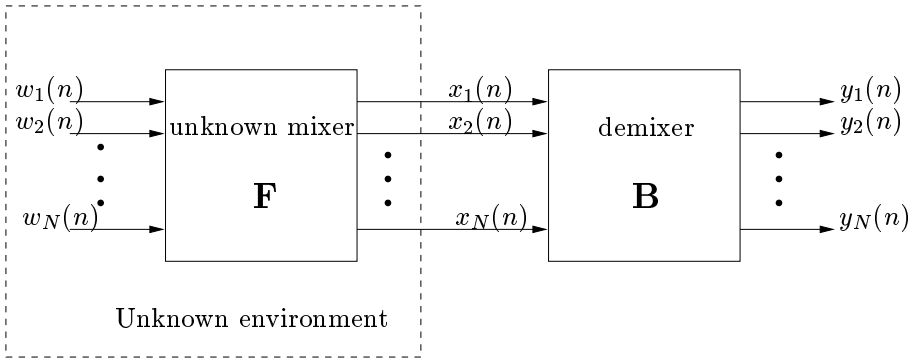


Figure 8.1: The blind source separation problem as it is solved traditionally. $y_i(n)$ is an estimator of $w_i(n)$ and $\mathbf{B}$ is an estimator of $\mathbf{F}^{-1}$.

Independent Component Analysis (ICA) has received attention in blind source separation [5, 46], and it is a generalization of the Principal Component Analysis (PCA). PCA is the same as KLT, thus it removes the correlation between the input signals. Using the PCA to estimate the $\mathbf{F}^{-1}$, the vectors in the resulting matrix, $\mathbf{B}$, are constrained to be orthogonal. The ICA not only decorrelate the signals but also reduces higher-order statistical dependencies, attempting to make the signals dependency as weak as possible. The ICA imposes no orthogonality constraints on the vectors in the matrix. The basis of the ICA is that the sources, $\mathbf{w}(n)$, at each point in time are instant mutually independent. A common assumption is that the number of sensors is greater or equal to the number of sources, i.e. $N \geq K$, to assure that the matrix $\mathbf{F}$ is full rank. This is necessary because the ICA finds an estimation of $\mathbf{F}^{-1}$. Another common assumption is no noise or only low additive noise.

Let the problem of blind source separation be reformulated as:

- *Given L independent realizations of the observation vector* $\mathbf{x}$*, reconstruct or estimate the original signals and the mixing matrix.*

The solution possibilities are no longer limited to estimate the inverse of $\mathbf{F}$. This way it is also possible to allow for more sources than sensors, i.e. $K \geq N$,

if at each point in time, n, there are only $s \leq N$ sources active. This means that at each point in time the number of nonzero sources is less than or equal to the number of sensors, but that the *total* number of sources are greater than the number of sensors. In other words the source vector $\mathbf{w}(n)$ needs to be *sparse*, with no more than N nonzero entries. The mixing matrix $\mathbf{F}$ is now of dimension $N \times K$ where $K \geq N$, thus, it is overcomplete. Assuming that all the sources are active, i.e. none of the sources are zero at all times, the matrix $\mathbf{F}$ spans the N dimensional space, and it is a frame. With an additive noise vector this gives us the Equation 8.1 for each point in time. A time series will give a set of data, as in the experiments in the following section.

Some work in this area is done by Lewicki and Sejnowski [49] and Lee et al. [47].

8.3 Experiments on reconstructing the true frame

In this section we show some experiments on reconstructing the true frame from a data set. This can be useful when we know that a physical system has a true underlying sparse structure, and we only have access to the data vectors produced by such a model. As indicated in the previous sections, this can be used for signal reconstruction and estimation as well as in blind source separation problems. Some of these experiments were shown in [22]

The experiments are done using a 20×30 original matrix, $\mathbf{F}_{orig}$ with random entries, chosen from a normal distribution with mean zero and variance one. The columns in $\mathbf{F}_{orig}$ are normalized. The noise free data vector is obtained as a linear combination of m randomly picked vectors from $\mathbf{F}_{orig}$ where the coefficients are Gaussian random variables with zero mean and unit variance. The constructed coefficient vector is denoted $\check{\mathbf{w}}$. The noise free data vector is normalized $\Rightarrow \check{\mathbf{x}}$. The noisy data vector, $\mathbf{x}$, is $\check{\mathbf{x}} + \mathbf{n}$ where $\mathbf{n}$ is a noise vector with Gaussian random entries with zero mean and variance depending on the SNR in the experiment. Experiments were done without noise and with SNR at 20 dB.

Mathematically, the synthetic data set can be described as:

$$\begin{aligned} \mathbf{F}_{org} \frac{\check{\mathbf{w}}_l}{\|\mathbf{F}_{org}\check{\mathbf{w}}_l\|} &= \check{\mathbf{x}}_l \\ \mathbf{x}_l &= \check{\mathbf{x}}_l + \mathbf{n}_l \end{aligned} \tag{8.5}$$

where $l = 1, 2 \ldots 1000$.

Experiments are done with m fixed at 4 and then at 7. Experiments with m varying within the training vector set is also done. In this case m is uniformly distributed between 1 and 10, this gives a mean $\bar{m} = 5.5$.

In the experiment the only available data is the training set $\mathbf{x}_l, l = 1, 2 \ldots 1000$. An initial frame is constructed by using a normalized version of the first 30 vectors from the training set, and this matrix is called $\mathbf{F}_0$. The frame that the training converges to is called $\mathbf{F}_{conv}$. If $\mathbf{F}_{conv} \simeq \mathbf{F}_{orig}$ the procedure has worked well in reconstructing the generative model of the underdetermined system with sparsity constraint.

The training of the frames is done by using MOD on the training set, and by using the regularized FOCUSS described in Chapter 3 as the vector selection algorithm required in the MOD. The parameters in the regularized FOCUSS are selected according to the modified L-curve method described in Section 3.2.3.

Two issues are of special interest: The number of vectors used in an approximation, i.e. the sparsity, and the error. Therefore the average number of vectors, called $\bar{r}$, and the normalized distortion are plotted as a function of training iterations in the experiments. All the experiments converges completely.

Figures 8.2 and 8.3 show plots of the average number of vectors used in the approximations and the normalized distortion as a function of training iterations for the experiments without noise for $m = 4$ and $m = 7$ respectively. In the experiment with $m = 4$ all the 30 frame vectors where reconstructed from the data, so that $\mathbf{F}_{conv} = \mathbf{F}_{orig}$[1].

In the experiment with $m = 7$, 29 of the 30 frame vectors were reconstructed to within 1% error. In both these experiments, it can be seen from the figure that the average number of vectors used in the approximations at convergence is lower than the number of vectors used to *produce* the data set. When $m = 4$ the average number of vectors used in the approximations converges at $\bar{r} = 3.222$ instead of 4, and for $m = 7$ it converges at $\bar{r} = 5.103$ instead of 7.

This means that the reconstruction of the set $\frac{\check{\mathbf{w}}_l}{\|\mathbf{F}_{org}\check{\mathbf{w}}_l\|}, l = 1, 2 \ldots 1000$ is not quite accurate. This is not surprising since we use a version of FOCUSS that allows for noise. Even if no noise were added in this training set, we start the training with a wrong $\mathbf{F}$ since it is unknown, thus we have to allow for noise if we want sparse solutions when we do the vector selection. In terms of compression the results are encouraging since the reconstructed coefficient vector is even *sparser* than the true coefficient vector, and this is true for

[1] a small difference, (1%) measured by the norm of the error for each vector, is allowed in all the experiments

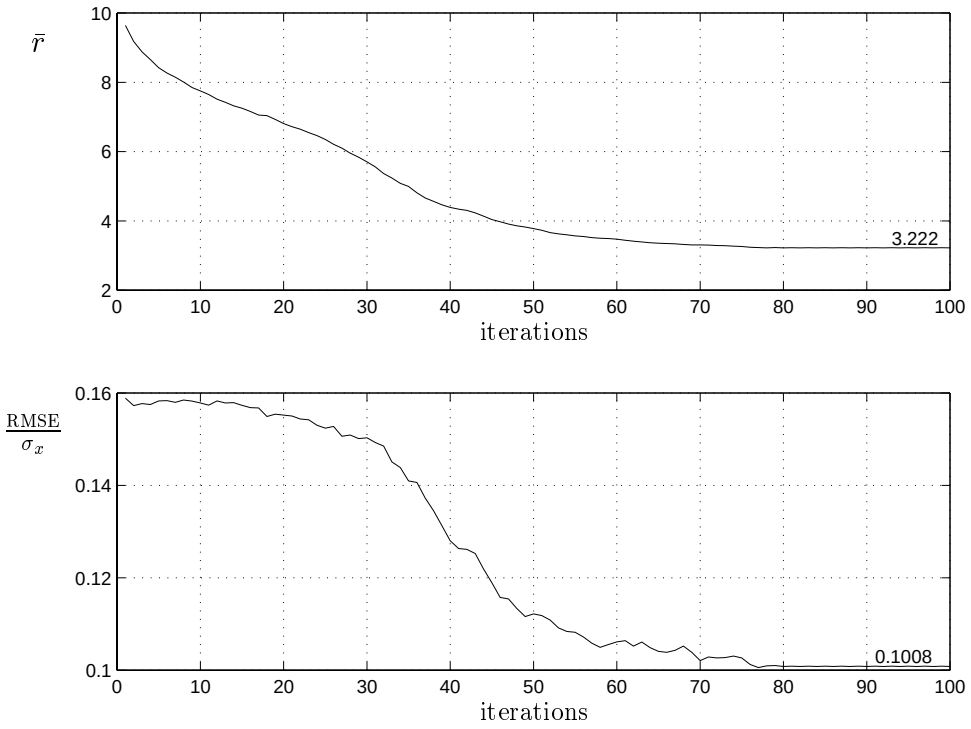


Figure 8.2: Training sequence of reconstructing experiment, $m = 4$, no noise. The average number of vectors used in the approximations, $\bar{r}$, and the normalized distortion are plotted as functions of iterations.

all the experiments presented here. For model reconstruction this may not be desired, but since the true $\mathbf{F}_{orig}$ is reconstructed, a FOCUSS version that allows less noise can be used together with the true $\mathbf{F}_{orig}$, or the $\mathbf{F}_{conv}$ after the training. This will result in a more accurate reconstruction of the set $\frac{\check{\mathbf{w}}_l}{\|\mathbf{F}_{org}\check{\mathbf{w}}_l\|}, l = 1, 2 \ldots 1000$.

The normalized distortion in all the experiments is also plotted, and converges to a lower value than the start value. Since we use the regularized FOCUSS as the vector selection algorithm, we allow for noise in our attempt to model the data vector. The regularized FOCUSS require a target SNR as input, and the approximated vector will have an SNR somewhere around this value. Therefore it is not expected for the distortion values to drop dramatically, as it does in the training experiments in Chapter 5. In most of the training experiments in Chapter 5 the OMP and FOMP are used as vector selection algorithms, and a sparsity criterion is used. This way the number of vectors used in the

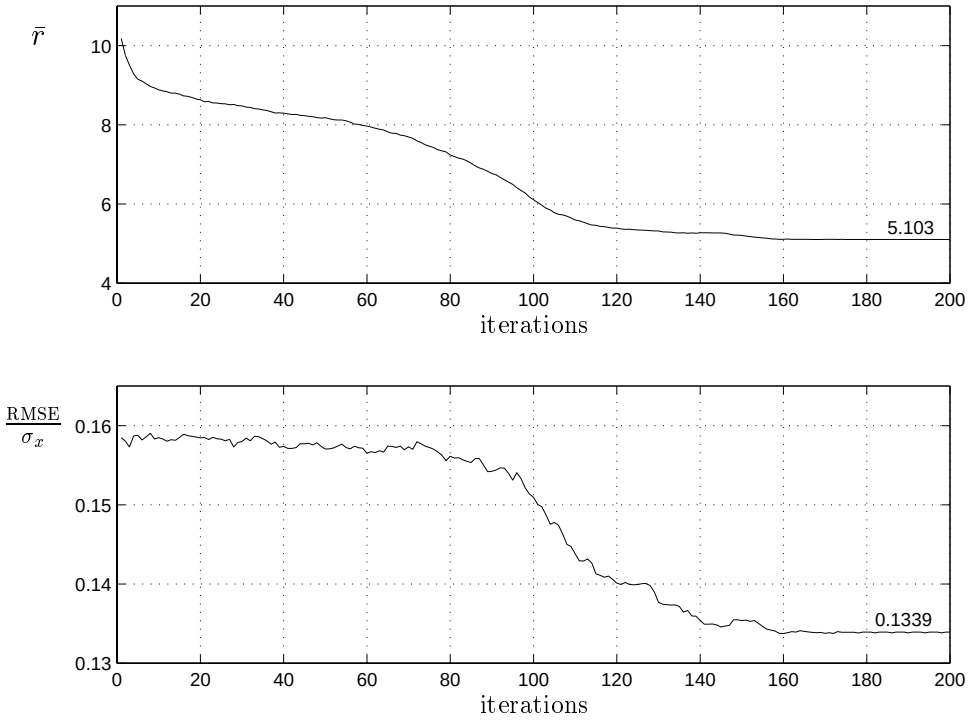


Figure 8.3: Training sequence of reconstructing experiment, $m = 7$, no noise. The average number of vectors used in the approximations, $\bar{r}$, and the normalized distortion are plotted as functions of iterations.

approximations is constant, and the normalized distortion decreases. In the experiments in this section both the average number of vectors used in the approximations and the normalized distortion decreases during training. The most dramatic development is in the average number of vectors used in the approximations due to the target SNR when using the regularized FOCUSS, but also the normalized distortion decreases when the iterations approach convergence.

Figure 8.4 shows the experiments with $m = 4$ and noise level at 20 dB. In this experiment all the 30 frame vectors where found, so that $\mathbf{F}_{conv} = \mathbf{F}_{orig}$. The experiment shown in Figure 8.5 was done with $m = 7$ and noise level at 20 dB, and here 29 of the 30 frame vectors from $\mathbf{F}_{orig}$ was reconstructed in $\mathbf{F}_{conv}$. The average number of vectors used in the approximations, $\bar{r}$, in these two experiments converges at slightly higher values than in the two earlier experiments and this is a consequence of the noise that is added to the

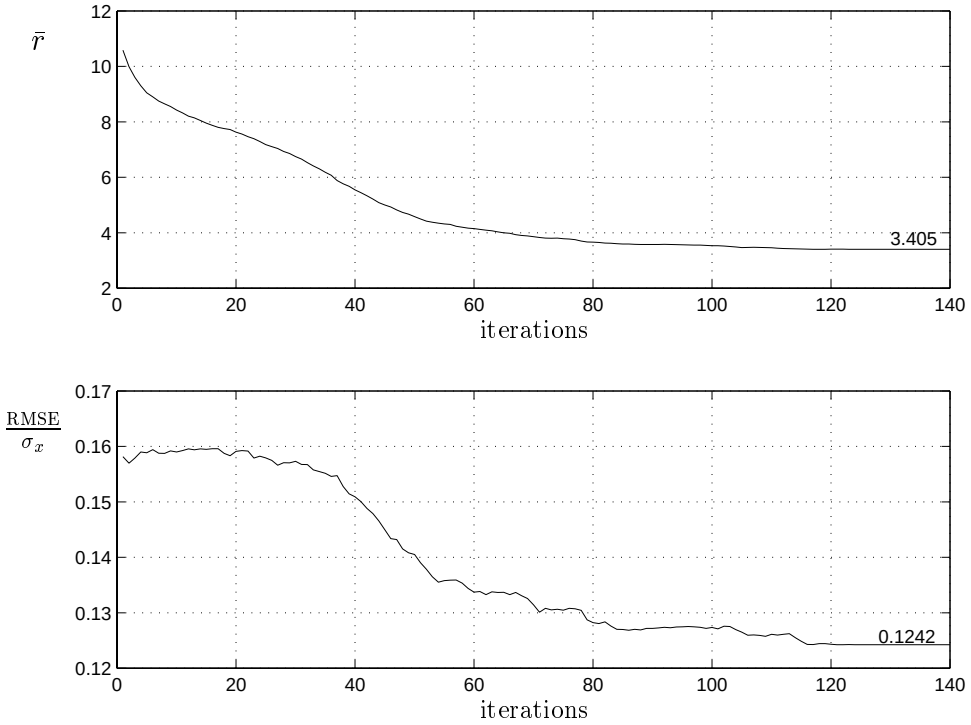


Figure 8.4: Training sequence of reconstructing experiment, $m = 4$, noise level 20 dB. The average number of vectors used in the approximations, $\bar{r}$, and the normalized distortion are plotted as functions of iterations.

training set.

For the two experiments with uniformly distributed m, shown in Figure 8.6 and 8.7, all the 30 frame vectors were reconstructed, so that $\mathbf{F}_{conv} = \mathbf{F}_{orig}$. Also here the average number of vectors used in the approximations at convergence, $\bar{r} = 3.752$ and $\bar{r} = 3.969$, is less than the $\bar{m} = 5.5$, and it is less in the no noise case than in the 20 dB case.

The experiment results indicate that MOD works very well with a good vector selection algorithm. The MOD algorithm has already produced good results on designing frames for compression of ECG signals and images as shown in [17, 19] and in Chapter 6, and 7, and the results in this section provides complimentary evidence of its good properties.

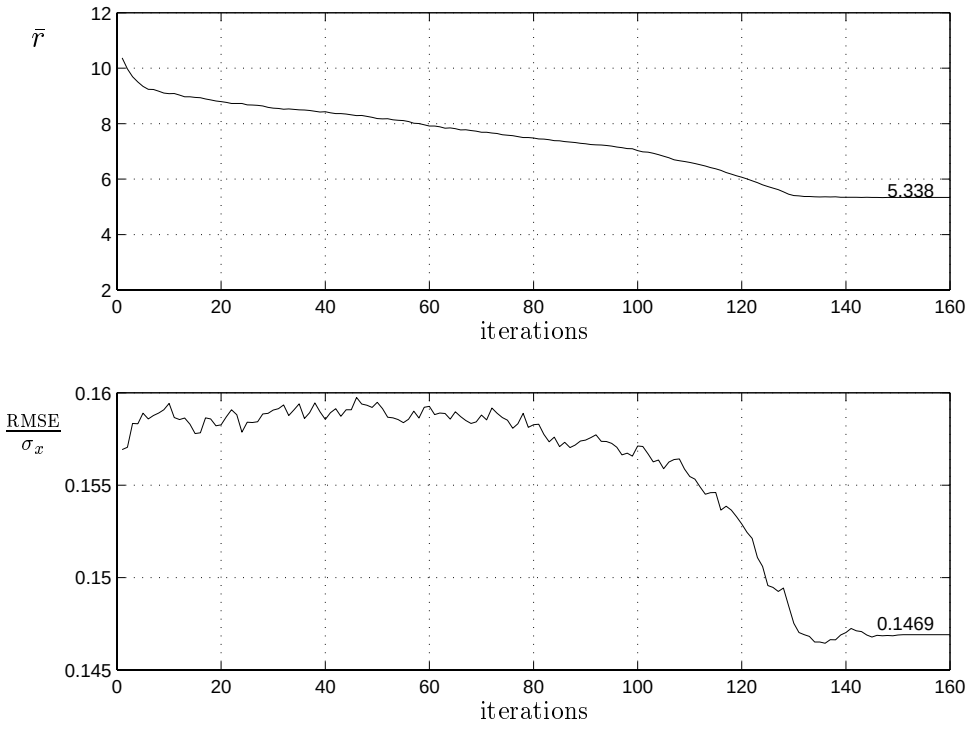


Figure 8.5: Training sequence of reconstructing experiment, $m = 7$, noise level 20 dB. The average number of vectors used in the approximations, $\bar{r}$, and the normalized distortion are plotted as functions of iterations.

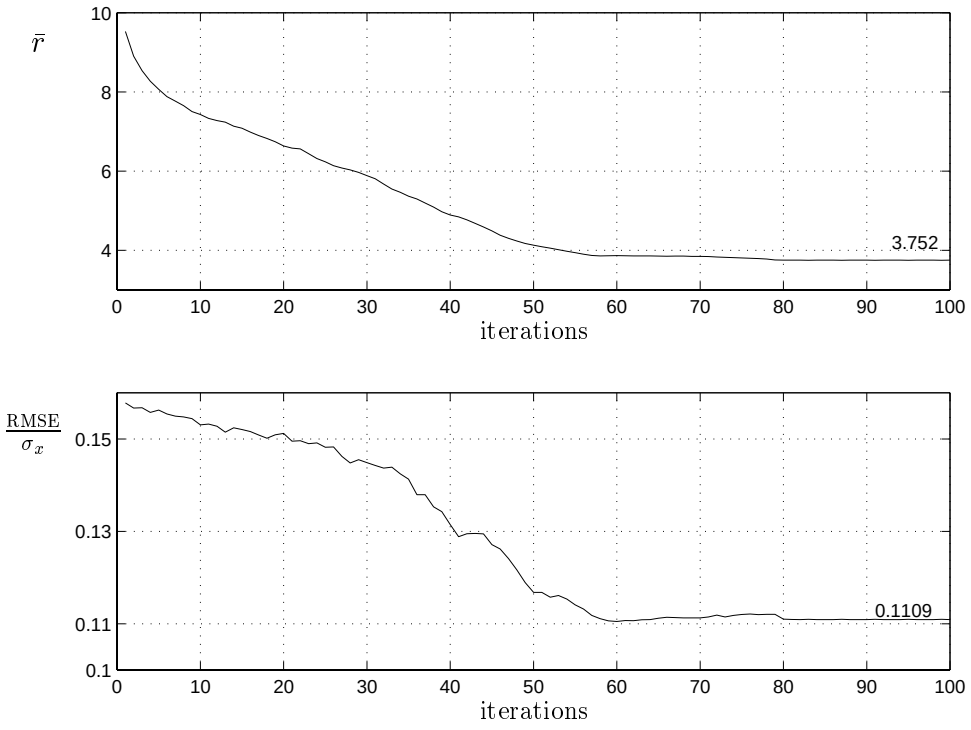


Figure 8.6: Training sequence of reconstructing experiment, uniformly distributed m, no noise. The average number of vectors used in the approximations, $\bar{r}$, and the normalized distortion are plotted as functions of iterations.

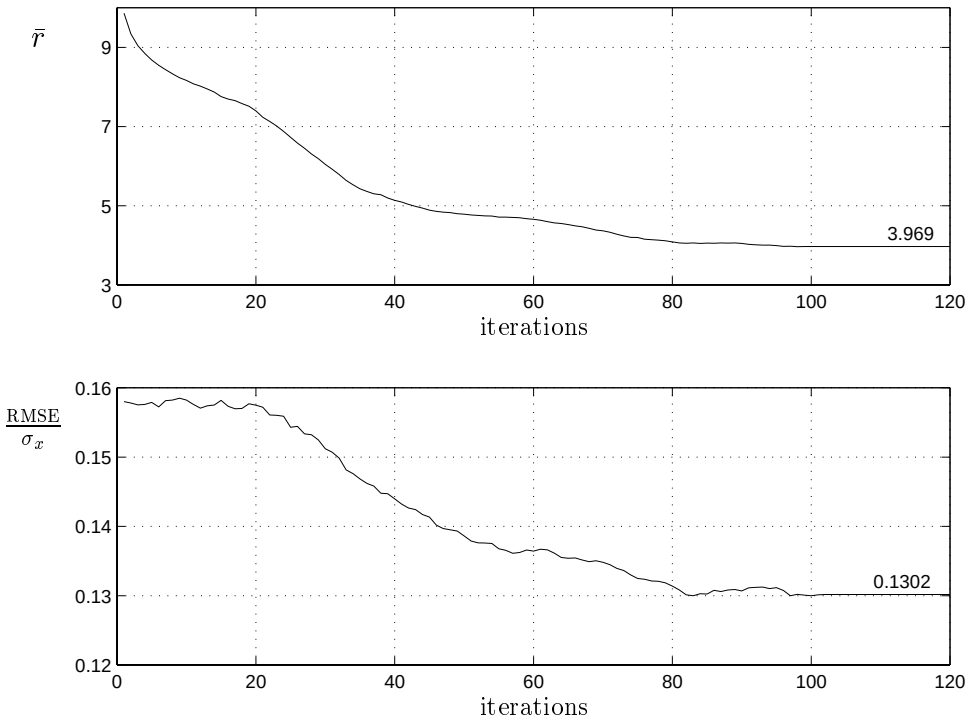


Figure 8.7: Training sequence of reconstructing experiment, uniformly distributed m, noise level 20 dB. The average number of vectors used in the approximations, $\bar{r}$, and the normalized distortion are plotted as functions of iterations.

Chapter 9

Conclusions

The aim of this thesis was to investigate the use of overcomplete vector sets, frames, for the purposes of signal representation and compression.

Emphasizing the synthesis part of a traditional analysis-synthesis setting, we are not restricted to the traditional transforms, filter bank, and wavelets and we can use overcomplete sets of vectors. The increased freedom does make a signal expansion *non-unique*, and the problem of finding a good expansion becomes more involved. We use existing matching pursuit techniques, and also we developed a robust version of regularized FOCUSS. It seems like the MP techniques are well suited for compression purposes where we need a good approximation using a few vectors. On the other hand, if we want to find the true underlying expansion that produced a noisy data set, or an approximation of it, the robust version of the regularized FOCUSS seems to be a good vector selection procedure. It is, however, very computationally expensive.

We have presented a frame design algorithm, the MOD. The design algorithm is iterative and inspired by the GLA, and it requires a training set. We show improved approximation capabilities compared to frames designed in an ad hoc manner. Experiments show typical reduction in normalized distortion by $25 - 40$ %. A simple way to solve the problem of finding an initial frame is to use a collection of the first training vectors. Experiments show good signal representation performance for ECG signals, speech and images, using this approach.

In general a frame based compression scheme is expected to work better than ordinary transform coding for *low bit rates*, and for signals that are not stationary, i.e. all real life signals. We use ECG signals and images in our compression experiments, both are non stationary, and our results shows improved performance at low bit rates.

A multi frame compression scheme, MFC, is developed, and improves both the ECG and the images compression results.

With other purposes than compression in mind, e.g. blind source separation with fewer signals than sources, we show that we are able to reconstruct a frame from a data set using the MOD and the robust version of the regularized FOCUSS.

9.1 Directions for future research

Based on experience gained during this work, we would suggest some possible directions for future research.

- *Convergence issues*
  We have not developed any formal proofs of convergence of the MOD algorithm. In our experiments using MOD with regularized FOCUSS on synthetic data, we observe convergence in all our experiments. In the experiments using MP techniques and real world data the training converges when we select *one* frame vector in each iteration. Note that the vector selection is optimal in that situation. In all other situations the vector selection is suboptimal. From the training experiments we observe that the change in the MSE becomes small after a number of iterations. At the point where the training is terminated, the change in the MSE is as minor fluctuations.

- *Deciding optimal frame sizes*
  The frame sizes used in this thesis are all chosen rather than optimized. Experimental results and ad hoc based reasoning have been the basis for choosing frame sizes. Our rationale for using variable sized frames, however, indicates that a more formal approach for finding suitable frame sizes could be developed. One possible problem is that it would be dependent upon the probability distribution of the signals, which are in general unknown. The pdf can probably be estimated using a training set.

- *Quality factor*
  The connection between the desired approximation quality and the quantizing step, i.e. MSE_{target} and Δ, can be embedded into a JPEG like quality factor.

- *Robust coding*
  For transmission over noisy channels, robust coding is important. Entropy coding is not very robust, and a different coding scheme is needed for the purpose of robust coding. Since the probability distribution over the position information of the frame coefficients are much more even than the probability distribution over the position information of traditional transform coefficients, the entropy schemes used in this thesis favors the traditional transform schemes. In a robust coding scheme, not using entropy coding, the advantages of the frame based system over a traditional transform based scheme may very well be more significant than shown in this work.

  When entropy coding is used, uniform quantization is optimal. Without entropy coding a pdf optimized quantizer is optimal. In a frame based system a possible approach could be to use a large training set to estimate the pdf of the coefficient selected first, second and so on, and make pdf optimized quantizers. Another approach is inspired by the shape gain VQ. As described in Chapter 2 there is a strong relation between shape gain VQ and frame based representation. We could, as in shape gain VQ, train the codebook of gains as well as the codebook of shapes. This can be done iteratively; for a given gain codebook and partition the optimal shape codebook is found and after that the new suboptimal partition. Next, given the shape codebook and partition, the optimal gain codebook is found and again the new suboptimal partition is found, and so forth. Note that in this context the suboptimal partition refers to a suboptimal vector selection algorithm together with an optimal method of finding the best gain values when the vector selection is decided. The shape codebook refers to the frame. The optimization is done solely with respect to MSE. This was tried on ECG signals. The position information, as run, and the coefficient value were combined into one symbol, as before. The probability distribution of the combined symbols turned out to be fairly even. As expected, the distribution of the position information had an even stronger similarity to a uniform distribution than in the tests shown in Chapter 6.

- *Other frame design methods*
  The MOD is quite computationally expensive having to use a vector selection algorithm for each of the vectors in the training set for each iterations along with the inversion of the matrix in Equation 4.20. A less computational expensive algorithm could be desirable in many applications. One possible approach for the second part of MOD, where a new frame is computed according to Equation 4.20 and a matrix inver-

sion is needed, could be a iterative updating of the frame vectors instead of recalculating them according to Equation 4.20. A gradient descent approach gives:

$$\hat{\mathbf{F}}^{(k+1)} = \hat{\mathbf{F}}^{(k)} - \alpha^{(k)} \sum_{l=1}^{M} \mathbf{r}_l^{(k)} \hat{\mathbf{w}}_l^T, \tag{9.1}$$

where $\alpha^{(k)}$ is a variable step-size parameter. When looking at the problem

$$\arg \min_{\mathbf{F}, \mathbf{W}^M} \langle \|\mathbf{x} - \mathbf{F}\mathbf{w}\|^2 + \lambda d(\mathbf{w}) \rangle_M \tag{9.2}$$

we discovered that it had the trivial solution that both the error $\|\mathbf{x} - \mathbf{F}\mathbf{w}\|^2$ and the sparsity measure $d(\mathbf{w})$ can go to zero in the case where $\mathbf{w} \to 0$ and $\mathbf{F} \to \infty$. To prevent this from happening $\mathbf{F}$ has to be bounded in some way. One possible approach is to keep the Frobenius norm of the matrix constant.

Letting the step size ensure the constant Frobenius norm, we get:

$$\alpha^{(k)} = \frac{2}{M} \frac{\text{trace}(\mathbf{F}^{(k)T} \tilde{\mathbf{R}}_{\mathbf{r}^{(k)}\hat{\mathbf{w}}})}{\text{trace}(\tilde{\mathbf{R}}_{\hat{\mathbf{w}}\mathbf{r}^{(k)}} \tilde{\mathbf{R}}_{\mathbf{r}^{(k)}\hat{\mathbf{w}}})}. \tag{9.3}$$

This is shown in Appendix A.2.

The first part of MOD requires recalculated coefficient vectors in every iteration. A possible less computational approach would be an algorithm iteratively updating the coefficient vectors instead of recalculating them for each iteration in the MOD algorithm. Kreutz-Delgado et. al. adress this topic in [40, 43]. The problem of Equation 9.2 is interpreted as a Lyapunov function, and this gives a possible approach for iterating both the frame vectors and the coefficient vectors.

- *Total Least Square*
  The MOD algorithm for frame design uses a Least Square (LS) approach. A possible improvement could be obtained by using a Total Least Square (TLS) approach.

- *Adaptive frame design*
  The frame design in this thesis is done off line using a training set. A scheme adapting the frame to the signal in real time could be an advantage if we have a signal with slowly changing characteristics, or in

other examples where we want to tailor a frame to a specific signal for better performance. One example of the latter could be in speech coding: A frame trained on a broad training set of different speakers could be the initial frame. In many situations just one speaker would use the system for a period, and adapting the frame would lead to improved performance.

- *Lapped frames*
  In this thesis we have concentrated on a block based scheme with no overlapping, like in traditional transform coding. This leads to well known blocking artifacts. A natural generalization would be to lapped frames. Comparing to the traditional analysis-synthesis setting this will correspond to the more general lapped transform or filter banks and wavelets. A general algorithm for lapped frames is developed by Aase et al. in [1].

- *MSVQ*
  In Chapter 2 we do a theoretical comparison of frame based representation and MSVQ. Experiments could be done both to verify the theoretical comparison as it is, and in compression schemes including coding of the coefficients/indices.

Appendix A

Mathematical details

A.1

We here show that Equation 4.20 is equivalent to Equation 4.21. That is the equation

$$\tilde{\mathbf{F}} = \mathbf{F} + \tilde{\mathbf{R}}_{rw}\tilde{\mathbf{R}}_{ww}^{-1} \tag{A.1}$$

is equivalent to

$$\tilde{\mathbf{F}} = \tilde{\mathbf{R}}_{xw}\tilde{\mathbf{R}}_{ww}^{-1}. \tag{A.2}$$

From the equations 4.1 and 4.2 we know that:

$$\mathbf{x}_l = \mathbf{F}\mathbf{w}_l + \mathbf{r}_l \tag{A.3}$$

The estimated cross correlation matrix $\tilde{\mathbf{R}}_{xw}$ can be written:

$$\tilde{\mathbf{R}}_{xw} = \frac{1}{M}\sum_{l=1}^{M}\mathbf{x}_l\mathbf{w}_l^T. \tag{A.4}$$

Inserting Equation A.3 in A.4 gives:

$$\begin{aligned}
\tilde{\mathbf{R}}_{xw} &= \frac{1}{M}\sum_{l=1}^{M}(\mathbf{F}\mathbf{w}_l + \mathbf{r}_l)\mathbf{w}_l^T \\
&= \frac{1}{M}\sum_{l=1}^{M}\mathbf{F}\mathbf{w}_l\mathbf{w}_l^T + \frac{1}{M}\sum_{l=1}^{M}\mathbf{r}_l\mathbf{w}_l^T \\
&= \mathbf{F}\frac{1}{M}\sum_{l=1}^{M}\mathbf{w}_l\mathbf{w}_l^T + \frac{1}{M}\sum_{l=1}^{M}\mathbf{r}_l\mathbf{w}_l^T \\
&= \mathbf{F}\tilde{\mathbf{R}}_{ww} + \tilde{\mathbf{R}}_{rw}.
\end{aligned}$$

Inserting this in Equation A.2 gives:

$$\begin{aligned}\tilde{\mathbf{F}} &= (\mathbf{F}\tilde{\mathbf{R}}_{ww} + \tilde{\mathbf{R}}_{rw})\tilde{\mathbf{R}}_{ww}^{-1} \\ &= \mathbf{F} + \tilde{\mathbf{R}}_{rw}\tilde{\mathbf{R}}_{ww}^{-1}\end{aligned}$$

which is identical with Equation A.1.

A.2

In this section we show how a constant Frobenius norm of $\mathbf{F}$ led to Equation 9.3 in Chapter 9. Let $\|\mathbf{F}\|_F$ denote the Frobenius norm of the matrix $\mathbf{F}$;

$$\|\mathbf{F}\|_F^2 = \text{trace}(\mathbf{F}^T\mathbf{F}).$$

Using a gradient descent approach to update the frame vectors we have:

$$\hat{\mathbf{F}}^{(k+1)} = \hat{\mathbf{F}}^{(k)} - \alpha^{(k)} \sum_{l=1}^{M} \mathbf{r}_l^{(k)} \hat{\mathbf{w}}_l^T, \tag{A.5}$$

where $\mathbf{r}_l = \hat{\mathbf{F}}\hat{\mathbf{w}}_l - \mathbf{x}_l$, $l = 1, 2, \ldots M$, k is an iteration variable, and $\alpha^{(k)}$ is a step-size parameter. Using the step size to keep the Frobenius norm constant gives:

$$\text{trace}(\mathbf{F}^{(k+1)T}\mathbf{F}^{(k+1)}) = \text{trace}(\mathbf{F}^{(k)T}\mathbf{F}^{(k)}). \tag{A.6}$$

Substituting Equation A.5 in Equation A.6 we get:

$$\begin{aligned}&\text{trace}(\mathbf{F}^{(k+1)T}\mathbf{F}^{(k+1)}) = \\ &\quad \text{trace}[(\mathbf{F}^{(k)} - \alpha^{(k)} \sum_{l=1}^{M} \mathbf{r}_l^{(k)} \hat{\mathbf{w}}_l^T)^T (\mathbf{F}^{(k)} - \alpha^{(k)} \sum_{l=1}^{M} \mathbf{r}_l^{(k)} \hat{\mathbf{w}}_l^T)] = \\ &\quad \text{trace}[\mathbf{F}^{(k)T}\mathbf{F}^{(k)} - \mathbf{F}^{(k)T}\alpha^{(k)} \sum_{l=1}^{M} \mathbf{r}_l^{(k)} \hat{\mathbf{w}}_l^T - \alpha^{(k)} \sum_{l=1}^{M} \hat{\mathbf{w}}_l \mathbf{r}_l^{(k)T} \mathbf{F}^{(k)} \\ &\quad + \alpha^{(k)2} \sum_{l=1}^{M} \hat{\mathbf{w}}_l \mathbf{r}_l^{(k)T} \sum_{i=1}^{M} \mathbf{r}_i^{(k)} \hat{\mathbf{w}}_i^T] = \\ &\quad \text{trace}(\mathbf{F}^{(k)T}\mathbf{F}^{(k)}) - 2\alpha^{(k)} \text{trace}(\mathbf{F}^{(k)T} \sum_{l=1}^{M} \mathbf{r}_l^{(k)} \hat{\mathbf{w}}_l^T) \\ &\quad + \alpha^{(k)2} \text{trace}(\sum_{l=1}^{M} \hat{\mathbf{w}}_l \mathbf{r}_l^{(k)T} \sum_{i=1}^{M} \mathbf{r}_i^{(k)} \hat{\mathbf{w}}_i^T) = \\ &\quad \text{trace}(\mathbf{F}^{(k)T}\mathbf{F}^{(k)})\end{aligned}$$

This gives:

$$\alpha^{(k)} = \frac{2\text{trace}(\mathbf{F}^{(k)T} \sum_l \mathbf{r}_l^{(k)} \hat{\mathbf{w}}_l^T)}{\text{trace}(\sum_l \hat{\mathbf{w}}_l \mathbf{r}_l^{(k)T} \sum_i \mathbf{r}_i^{(k)} \hat{\mathbf{w}}_i^T)} \tag{A.7}$$

Defining:

$$\begin{aligned} \tilde{\mathbf{R}}_{\mathbf{r}^{(k)}\hat{\mathbf{w}}} &= \frac{1}{M} \sum_{l=1}^{M} \mathbf{r}_l^{(k)} \hat{\mathbf{w}}_l^T \\ \tilde{\mathbf{R}}_{\hat{\mathbf{w}}\mathbf{r}^{(k)}} &= \frac{1}{M} \sum_{l=1}^{M} \hat{\mathbf{w}}_l \mathbf{r}_l^{(k)T}, \end{aligned}$$

We have:

$$\alpha^{(k)} = \frac{2}{M} \frac{\text{trace}(\mathbf{F}^{(k)T} \tilde{\mathbf{R}}_{\mathbf{r}^{(k)}\hat{\mathbf{w}}})}{\text{trace}(\tilde{\mathbf{R}}_{\hat{\mathbf{w}}\mathbf{r}^{(k)}} \tilde{\mathbf{R}}_{\mathbf{r}^{(k)}\hat{\mathbf{w}}})}. \tag{A.8}$$

Appendix B

Tables

Size 16×32			
No. of vectors in the approximation	Speech_{test}		
	DCT	Initial frame: DCT+Haar	MOD opt. frames
1	0.6518	0.6273	0.4599
2	0.4510	0.4266	0.2874
3	0.3194	0.3040	0.1950
4	0.2323	0.2256	0.1395
5	0.1700	0.1688	0.1050

Table B.1: Normalized distortion after test on Speech_{test}. Frames trained on Speech_{train}. Ad hoc based initial frame (DCT+Haar), size 16×32.

Size 16×41			
No. of vectors in the approximation	MIT100_{test}		
	DCT	Ad hoc based initial frame	MOD opt. frames
1	0.6275	0.1901	0.1428
2	0.3772	0.1067	0.0687
3	0.2290	0.0658	0.0449
4	0.0724	0.0455	0.0325
5	0.0471	0.0348	0.0260

Table B.2: Normalized distortion after test on MIT100_{test}. Frames trained on MIT100_{train}. Ad hoc based initial frame (7 DCT vectors + 34 ad hoc designed vectors), size 16×41.

Size 16×41			
No. of vectors in the approximation	MIT113_{test}		
	DCT	Ad hoc based initial frame	MOD opt. frames
1	0.4637	0.1683	0.1180
2	0.2570	0.0773	0.0556
3	0.1497	0.0514	0.0372
4	0.0754	0.0384	0.0275
5	0.0477	0.0310	0.0227

Table B.3: Normalized distortion after test on MIT113_{test}. Frames trained on MIT100_{train}. Ad hoc based initial frame (7 DCT vectors + 34 ad hoc designed vectors), size 16×41.

Size 32×64			
No. of vectors in the approximation	Speech_{test}		
	DCT	Initial frame	MOD opt. frames
1	0.7372	0.7843	0.5883
2	0.5788	0.6920	0.4243
3	0.4627	0.6261	0.3217
4	0.3786	0.5688	0.2606
5	0.3147	0.5190	0.2181
6	0.2636	0.4739	0.1832

Table B.4: Normalized distortion after test on Speech_{test}. Frames trained on Speech_{train}. Initial frame from training vectors, size 32×64.

Size 32×64					
No. of vectors in the approximation	MIT100_{test}				
	DCT	MIT100_{train}		MIT_{mix}	
		Initial frame	MOD opt. frames	Initial frame	MOD opt. frames
1	0.7203	0.4474	0.2391	0.4917	0.2802
2	0.5571	0.3118	0.1142	0.3429	0.1563
3	0.4257	0.2567	0.0825	0.2683	0.1183
4	0.3183	0.2169	0.0640	0.2241	0.0998
5	0.2369	0.1910	0.0536	0.1948	0.0787
6	0.1713	0.1723	0.0497	0.1737	0.0679

Table B.5: Normalized distortion after test on MIT100_{test}. Frames trained on MIT100_{train} and MIT_{mix}. Initial frame from training vectors, size 32×64.

Size 32×64					
No. of vectors in the approximation	MIT207$_{test}$				
	DCT	MIT207$_{train}$		MIT$_{mix}$	
		Initial frame	MOD opt. frames	Initial frame	MOD opt. frames
1	0.4707	0.1739	0.1049	0.2486	0.1434
2	0.2226	0.1220	0.0643	0.1577	0.0828
3	0.1317	0.0988	0.0480	0.1115	0.0615
4	0.0859	0.0831	0.0391	0.0878	0.0516
5	0.0611	0.0708	0.0331	0.0736	0.0398
6	0.0465	0.0606	0.0291	0.0613	0.0352

Table B.6: Normalized distortion after test on MIT207$_{test}$. Frames trained on MIT207$_{train}$ and MIT$_{mix}$. Initial frame from training vectors, size 32×64.

Bibliography

[1] S. O. Aase, J. H. Husøy, K. Skretting, and K. Engan, "Optimized signal expansions for sparse representation," *IEEE Transactions on Signal Processing*, 1999. Submitted Sept–1999.

[2] N. Abramson, *Information theory and coding.* USA: McGraw-Hill, 1963.

[3] J. Anderson and S. Mohan, *Source and Channel Coding: An Algorithmic Approach.* Boston: Kluwer Academic Publishers, 1991.

[4] A. Back and A. Weigend, "What drives stock returns? an independent component analysis," in *Proceedings of the IEEE/IAFE/INFORMS Conference on Computational Intelligence for Financial Engineering*, (New York, USA), pp. 141–156, 1998.

[5] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[6] A. Berg and W. Mikhael, "Signal representation using adaptive parallel mixed transform techniques," in *Proc. 38th IEEE Midwest Symp. Circuits Syst.*, (Rio de Janeiro), Aug. 1995.

[7] A. Berg and W. Mikhael, "An efficient structure and algorithm for the mixed transform representation of signals," in *Proc. of the 29th Asilomar Conference on Signals, Systems and Computers*, (Monterey, California), pp. 1056–1060, 1996.

[8] A. Berg and W. Mikhael, "An efficient structure and algorithm for image representation using nonorthogonal basis images," *IEEE Trans. Circuits, Syst. II: Analog and Digital Signal Processing*, vol. 44, no.10, pp. 818–828, Oct. 1997.

[9] J. Cardoso, "Multidimensional independent component analysis," in *Int. Conf. on Acoust. Speech and Signal Proc.*, vol. 4, (Seattle, USA), pp. 1941–1944, May 1998.

[10] S. Chen and D. Donoho, "Application of Basis Pursuit in Spectrum Estimation," in *Proc. ICASSP*, vol. III, (Seattle, WA), pp. 1865–1868, May 1998.

[11] S. S. Chen, *Basis Pursuit*. PhD thesis, Stanford University, Nov. 1995.

[12] G. Davis, *Adaptive Nonlinear Approximations*. PhD thesis, New York University, Sept. 1994.

[13] V. DeBrunner, L. Chen, and H. Li, "Lapped multiple bases realizations for the transform coding of still images," in *Proc. Asilomar*, vol. 2, (CA, USA), pp. 943–947, 1994.

[14] V. DeBrunner, L. Chen, and H. Li, "On the use of (lapped) multiple transforms in still image compression," in *Proc. International Conference on Image Processing*, vol. 1, (CA, USA), pp. 294–297, 1995.

[15] K. Engan, S. O. Aase, and J. H. Husøy, "Transform based ECG signal compression using nonorthogonal vectors," in *Proc. NORSIG '97*, (Tromsø, Norway), pp. 140–145, May 1997.

[16] K. Engan, S. O. Aase, and J. H. Husøy, "Designing frames for matching pursuit algorithms," in *Proc. ICASSP '98*, (Seattle, USA), pp. 1817–1820, May 1998.

[17] K. Engan, S. O. Aase, and J. H. Husøy, "Frame based signal compression using method of optimal directions (MOD)," in *Proc. ISCAS'99*, (Orlando, USA), pp. IV–1–IV–4, June 1999.

[18] K. Engan, S. O. Aase, and J. H. Husøy, "Method of optimal directions for frame design," in *Proc. ICASSP '99*, (Phoenix, USA), pp. 2443–2446, Mar. 1999.

[19] K. Engan, S. O. Aase, and J. H. Husøy, "Multi-frame compression: Theory and design," *Signal Processing*, vol. 80, (to be published in issue 10) 2000.

[20] K. Engan, J. H. Husøy, and S. O. Aase., "Optimized frame design for a matching pursuit based compression scheme," in *Proc. EUSIPCO '98*, (Rhodos, Greece), pp. 153–156, Sept. 1998.

[21] K. Engan, J. H. Husøy, and S. O. Aase, "Optimized frames of variable size for use in a matching pursuit based compression scheme," in *Proc. NORSIG '98*, (Vigsø, Denmark), pp. 277–280, June 1998.

[22] K. Engan, B. Rao, and K. Kreutz-Delgado, "Frame design using FOCUSS with method of optimized directions (MOD)," in *Proc. NORSIG '99*, (Oslo, Norway), pp. 65–69, Sept. 1999.

[23] K. Engan, B. Rao, and K. Kreutz-Delgado, "Regularized FOCUSS for subset selection in noise," in *Proc. of NORSIG 2000*, (Sweden), pp. 247–250, June 2000.

[24] H. W. Engl and W. Grever, "Using the L-curve for determining optimal regularization parameters," *Nummer.Math.*, vol. 69, pp. 25–31, 1994.

[25] J. H. Friedman and W. Stuetzle, "Projection pursuit regression," *Journal of the American Statistical Association*, vol. 76, pp. 817–823, Dec. 1981.

[26] A. Gersho, *Vector Quantization and Signal Compression*. Boston: Kluwer Academic Publishers, 1992.

[27] M. Gharavi-Alkhansari and T. S. Huang, "A fast orthogonal matching pursuit algorithm," in *Int. Conf. on Acoust. Speech and Signal Proc.*, (Seattle, U.S.A), pp. 1389–1392, May 1998.

[28] M. Goodwin, "Matching pursuit with damped sinusoids," in *Proc. ICASSP '97*, (Munich), pp. 2037–2040, Apr. 1997.

[29] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Trans. Signal Processing*, vol. 45, pp. 600–616, Mar. 1997.

[30] V. K. Goyal, "Quantized overcomplete expansions: Analysis, synthesis and algorithms," tech. rep., Electronics research laboratory, memorandum No. UCB/ERL M95/97, July 1995.

[31] V. K. Goyal, M. Vetterli, and N. T. Thao, "Quantization of overcomplete expansions," in *Proc. IEEE Data Compression Conf.*, (Utah), pp. 13–22, Mar. 1995.

[32] V. K. Goyal, M. Vetterli, and N. T. Thao, "Quantized overcomplete expansions in RN: Analysis, synthesis, and algorithms," *IEEE Trans. Inform. Theory*, vol. 44, pp. 16–31, Jan. 1998.

[33] M. Hanke, "Limitations of the L-curve method in ill-posed problems," *BIT*, vol. 36, pp. 287–301, June 1996.

[34] P. C. Hansen, "Analysis of Discrete Ill-posed Problems by Means of the L-curve," *SIAM Review*, vol. 34, pp. 561–580, Dec. 1992.

[35] P. C. Hansen and D. P. O'Leary, "The use of the L-Curve in the regularization of discrete ill-posed problems," *SIAM J. Sct. Comput.*, vol. 14, pp. 1487–1503, Nov. 1993.

[36] S. Haykin, *Neural networks: a comprehensive foundation*. New Jersey, USA: Prentice Hall, 1999.

[37] J. Herault and C. Jutten, "Space or time adaptive signal processing by neural network models," in *Neural Networks for Computing: AIP Conference Proceedings 151*, (American Institute for Physics, New York), pp. 206–211, 1986.

[38] J. H. Husøy, S. O. Aase, K. Skretting, and K. Engan, "Design of general block oriented expansions for efficient signal representation," in *Proc. ISCAS'99*, (Orlando, USA), pp. III–9–III–12, June 1999.

[39] K. Kreutz-Delgado and B. D. Rao, "Measures and Algorithms for Best Basis Selection," in *Proc. ICASSP 98*, vol. III, (Seattle, Washington), pp. 1881–1884, May 1998.

[40] K. Kreutz-Delgado, B. D. Rao, and K. Engan, "Novel algorithms for learning overcomplete dictionaries," in *Proc. of the 33 Asilomar Conference on Signals, Systems and Computers*, (Monterey, California), 1999.

[41] K. Kreutz-Delgado, B. D. Rao, K. Engan, T.-W. Lee, and T. Sejnowski, "Convex/schur-convex (CSC) log-priors and sparse coding," in *Proc. 6th joint symp. on Neural Comp.*, vol. 9, (Pasadena, USA), pp. 65–71, May 1999.

[42] K. Kreutz-Delgado, B. D. Rao, K. Engan, T.-W. Lee, and T. Sejnowski, "Learning overcomplete dictionaries: Convex/schur-convex (CSC) log-priors, factorial codes, and independent/dependent component analysis (I/DCA)," in *Proc. 6th joint symp. on Neural Comp.*, vol. 9, (Pasadena, USA), pp. 72–78, May 1999.

[43] K. Kreutz-Delgado, B. Rao, K. Engan, T.-W. Lee, and T. Sejnowski, "Learning overcomplete dictionaries and sparse representations," *In preparation for submission to Neural Computation.*, 2000.

[44] E. Kreyszig, *Advanced Engineering Mathematics*. New York: Wiley, sixth edition ed., 1988.

[45] W. LeBlanc, B. Bhattacharya, S. A. Mahmoud, and V. Cuperman, "Efficient search and design procedures for robust multi-stage vq of lpc parameters for 4 kb/s speech coding," *IEEE Trans. Speech and Audio Processing*, vol. 1, pp. 373–385, Oct. 1993.

[46] T. W. Lee, M. Girolami, A. J. Bell, and T. J. Sejnowski, "A unifying information-theoretic framework for independent component analysis," *International journal of computers and mathematics with applications*, In press, 1999.

[47] T. W. Lee, M. S. Lewicki, M. Girolami, and T. J. Sejnowski, "Blind source separation of more sources than mixtures using overcomplete representaions," *IEEE Signal Processing Letters*, vol. 6, pp. 87–90, Apr. 1999.

[48] C. M. Leung and W. S. Lu, "An l-curve approach to optimal determination of regularization parameter in image restoration," in *Proc. CCECE'93*, (Vancouver, Canada), pp. 1021–1024, Sept. 1993.

[49] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural Computation*, vol. 12, pp. 337–365, Feb. 2000.

[50] S. Mallat, *A wavelet tour of signal processing*. San Diego, USA: Academic Press, 1998.

[51] S. Mallat and F. Falzon, "Analysis of low bit rate image transform coding," *IEEE Trans. Signal Processing*, vol. 46, pp. 1027–1042, Apr. 1998.

[52] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, pp. 3397–3415, Dec. 1993.

[53] Massachusetts Institute of Technology, *The MIT-BIH Arrhythmia Database CD-ROM*, 2nd ed., 1992.

[54] W. Mikhael and A. Berg, "Image representation using nonorthogonal basis images with adaptive weight optimization," *IEEE Signal Processing Letters*, vol. 3, no.6, pp. 165–167, June 1996.

[55] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, pp. 227–234, Apr. 1995.

[56] R. Neff and A. Zakhor, "Very low bit-rate video coding based on matching pursuit," *IEEE Trans. Circuits, Syst. for Video Tech.*, vol. 7, pp. 158–171, Feb. 1997.

[57] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed in V1," *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[58] W. H. Pun and B. D. Jeffs, "Adaptive image restoration using a generalized gaussian model for unknown noise," *IEEE Trans. Image Processing*, vol. 4, pp. 1451–1456, Oct. 1995.

[59] B. D. Rao, "Analysis and Extensions of the FOCUSS Algorithm," in *Proc. of the 30th Asilomar Conference on Signals, Systems and Computers*, vol. 2, (Monterey, California), pp. 1218–1223, Nov. 1996.

[60] B. D. Rao, "Signal processing with the sparseness constraint," in *Proc. ICASSP '98*, (Seattle, USA), pp. 1861–1864, May 1998.

[61] B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Diversity measure minimization based basis selection methods from noisy observations," *In preparation for submission to the IEEE Trans. on Signal Processing*, 2000.

[62] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. Signal Processing*, vol. 47, pp. 187–200, Jan. 1999.

[63] B. Rao, K. Kreutz-Delgado, and S. Dharanipragada, "Improving spectral resolution using basis selection," in *Proc. of the 9th IEEE Signal Processing Workshop on Statistical Signal and Array Processing*, (Portland, Oregon), Sept. 1998.

[64] T. Reginska, "A regularization parameter in discrete ill-posed problems," *SIAM J. Sci. Comput.*, vol. 17, pp. 740–749, May 1996.

[65] B. Sigurd and E. Sandøe, *Klinisk Elektrokardiologi.* Bingen: Publishing Partners Verlags GmbH, 1991.

[66] A. Swindlehurst, M. Goris, and B. Ottersten, "Some experiments with array data collected in actual uban and suburban environments," in *IEEE Workshop on Signal Processing advances in Wireless Communications*, (Paris, France), pp. 301–304, 1997.

[67] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding.* Englewood Cliffs: Prentice-Hall, 1995.

[68] W. Weldon, F. Galton, and K. Pearson, *Biometrika.* Cambridge: University Press, 1953.

[69] W.Y.Chan, S. Gupta, and A. Gersho, "Enhanced multistage vector quantization by joint codebook design," *IEEE Trans. Commun.*, vol. 40, pp. 1693–1697, Nov. 1992.