KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

Sparse and Cosparse Signal Recovery for Compressive Sensing of Multi-Channel ECG Signals

Yurrit Avonds

Thesis voorgedragen tot het behalen
van de graad van Master of Science
in de ingenieurswetenschappen:
biomedische technologie

Promotor:
Prof. dr. ir. Sabine Van Huffel

Assessoren:
Prof. dr. ir. Robert Puers
Ir. Vladimir Matic

Begeleider:
Ir. Yipeng Liu

Academiejaar 2012 – 2013

Preface

The thesis in front of you is the result of a year of hard work. But it could never have been realised without the help of the following people. First of all, my supervisor Yipeng Liu, who advised me during our weekly meetings and even during the rest of the week through his quick replies to my e-mails. Next, I would like to thank my promotor, prof. dr. ir. Sabine Van Huffel for giving me the opportunity to work on this thesis in the SCD division of the ESAT department and for helping me throughout the year. I also want to thank Vladimir Matic for his additional advice during my meetings with Yipeng Liu. Finally, I would like to thank all of my friends and family for their help and support over the course of this year, most notably my mother Marleen Cabuy and my brother Niels Avonds.

Yurrit Avonds

Contents

Samenvatting

Mobiele ECG-metingen worden een steeds belangrijkere biomedische toepassing, aangezien het aantal mensen met hartafwijkingen stijgt door de vergrijzing van de bevolking en het verslechteren van de algemene levensstijl. Door de ECG bij de patiënt zelf, thuis of onderweg, te meten en ze naar een ziekenhuis te zenden, vormt het een alternatief voor continue meting in het ziekenhuis, wat tijdrovend en duur is voor patiënt en ziekenhuis. De toestellen die hierbij gebruikt worden beschikken vaak over een beperkte hoeveelheid energie en kunnen daardoor slechts beperkte hoeveelheden data opslaan, verwerken en verzenden. Gecomprimeerde metingen bieden hier een oplossing door data te comprimeren tijdens de meting.

Het is gebaseerd op twee concepten: willekeurig samplen en de spaarse voorstelling van een signaal. Bij willekeurig samplen worden verschillende willekeurige lineaire combinaties van de waarden van het originele signaal opgeslagen in een vector kleiner dan de vector van het kritiek gesamplede signaal. Bij een spaarse voorstelling, wordt een signaal voorgesteld door een vector met veel nullen, wat een vereiste is voor het correct reconstrueren van het originele signaal uit de meting. Bij de traditionele, spaarse aanpak, gebeurt dit door het signaal voor te stellen als een lineaire combinatie van een aantal basisfuncties die in een zogenaamd 'woordenboek' worden opgeslagen. Dit resulteert in een coefficiëntenvector met veel nullen. Recent onderzoek heeft geleid tot een cospaarse methode, waarbij een analysematrix wordt vermenigvuldigd met de waarden van het signaal om een spaarse voorstelling te bekomen.

Gebaseerd op beide soorten spaarse voorstellingen, zijn er in het verleden verschillende algoritmes voor signaalreconstructie ontwikkeld. Daarnaast bestaan er algoritmes die meerdere signalen met vergelijkbare karakteristieken tegelijkertijd reconstrueren. Dit is nuttig voor mobiele ECG-metingen, aangezien een ECG-meting uit meerdere kanalen kan bestaan.

In deze thesis worden bestaande algoritmes vergeleken en wordt een cospaars algoritme voor meerkanaals reconstructie met betere resultaten op vlak van snelheid en precisie, ontwikkeld. Bij het vergelijken van woordenboeken en analysematrices, blijken een woordenboek gebaseerd op wavelets en een analysematrix, gebaseerd op afgeleiden van de tweede orde, de beste resultaten op te leveren. Vervolgens worden een spaars en cospaars algoritme aangepast voor meer robuuste reconstructie wanneer de coefficiënten voor het willekeurig samplen en de meting zelf ruis bevatten. Ten slotte wordt een inleidende studie gepresenteerd over het nut van de gereconstrueerde signalen voor klinische diagnose, als basis voor toekomstig onderzoek.

Abstract

Mobile ECG monitoring is steadily becoming a more important biomedical application, as the number of people suffering from cardiac disorders increases due to the aging of the world population and worsening of our lifestyle. By recording data at the patient's location and transmitting it to the hospital, it provides an alternative to continuous monitoring in a hospital environment, which is both time consuming and expensive for the patient and the hospital. The devices used to perform this task are often limited in energy resources and therefore limited in the amount of data they can store, process and transmit. Compressive sensing can improve the energy efficiency of a mobile ECG monitor by compressing the data while it is being recorded.

Compressive sensing is based on two concepts: random sampling and sparse representation of the signal. In random sampling, several random linear combinations of the original signal values are stored in a vector that is smaller in size than the critically sampled signal would be. In sparse representation, a signal is represented as a vector with many zeros, which is a requirement for the correct reconstruction of the original signal from the compressive measurement. In the traditional, sparse approach, this is done by representing the signal as a linear combination of a limited number of basis functions stored in a dictionary, which leads to a coefficient vector with many zeros. Recent research has led to a cosparse approach where a so-called analysis matrix is multiplied with the signal values to obtain a sparse representation.

Algorithms for signal reconstruction, based on one of these types of sparse representation, have been developed. Additionally, methods to measure and reconstruct multiple signals with similar characteristics at once, have been proposed. This is useful for mobile ECG monitoring, since an ECG measurement can consist of multiple channels, i.e. recordings from different positions on the body.

In this thesis, several previously developed algorithms are compared and a single-channel cosparse algorithms is generalised to a multi-channel algorithm with better performance than any of the other algorithms in terms of reconstruction accuracy and processing time. It is found that a dictionary based on wavelets and an analysis matrix based on second order derivatives lead to the most accurate reconstructions. The cosparse method is also found to be independent of the type of random sampling that is used. Next, a sparse and cosparse algorithm are adjusted for more robust performance when the coefficients for random sampling and the measurement itself are corrupted by noise. Finally, an introductory study on the usefulness of the reconstructed signals for clinical diagnosis is presented as a basis for future research.

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

AMI	Acute myocardial infarction
BSBL	Block Sparse Bayesian Learning
BP	Basis Pursuit
CT	Computed Tomography
CoSaMP	Compressive Sampling Matching Pursuit
CS	Compressive Sensing
CHF	Congestive heart failure
CR	Compression Ratio
ECG	Electrocardiogram
GAP	Greedy Analysis Pursuit
GAPn	Noise aware GAP
IHT	Iterative Hard Thresholding
NMAE	Normalised Mean Absolute Error
NMSE	Normalised Mean Square Error
MRI	Magnetic Resonance Imaging
OMP	Orthogonal Matching Pursuit
OMPn	Noise aware OMP
OMMP	Orthogonal Multi-Matching Pursuit
OMMPn	Noise aware OMMP
PRD	Percentage Root Difference
SCoSaMP	Simultaneous CoSaMP
SGAP	Simultaneous GAP
SSIM	Structured Similarity
SOMP	Simultaneous OMP
SOMMP	Simultaneous OMMP

Symbols

$\boldsymbol{\alpha}$	Co-sparse projection of a signal
Λ	Co-Support
Ω	Support
$\boldsymbol{\Omega}$	Analysis Matrix
$\boldsymbol{\phi}$	Sensing Matrix
$\boldsymbol{\psi}$	Dictionary Matrix
ϕ	Wavelet scaling function
ψ	Wavelet wavelet function
$\mathbf{A}$	Multi-channel co-sparse projection
M	Length of the coefficient vector
n	Length of the measurement
N	Length of the original signal
P	Length of the sparse sequence in the cosparse approach
$\mathbf{r}$	Residual vector
$\mathbf{R}$	Residual matrix
$\mathbf{s}$	Coefficient vector
$\mathbf{S}$	Multi-channel coefficient matrix
$\mathbf{x}$	Original signal
$\mathbf{X}$	Multi-channel original signals matrix
$\mathbf{y}$	Compressive measurement
$\mathbf{Y}$	Multi-channel compressive measurements matrix

Chapter 1

Introduction

1.1 Biomedical Context

1.1.1 Data Storage in Biomedical Applications

Vast amounts of data are often acquired, transmitted and stored in applications based on biomedical signal and image processing, in order to capture and retain as much detail as possible to ensure a correct diagnosis based on these data types. In several imaging modalities, examining only a single patient may already require a lot of storage space. Magnetic Resonance Imaging (MRI) and Computed Tomography (CT), for example, store a scan of the whole body of a patient as a series of slices that show the contents of the body, perpendicular to a certain axis and at different positions along that axis. In case of a CT scan this may require more than 100 megabytes and in case of an MRI scan, more than a gigabyte of storage may even be required. In those cases, compression is useful to reduce the amount of data that has to be stored.[3]

This thesis focuses on applying compression to Electrocardiogram (ECG) signals. Although these signals do not necessarily require a lot of storage space, compression may be useful when only small amounts of data can be transmitted, for example due to limited energy resources in mobile ECG monitoring devices.

1.1.2 Mobile ECG Monitoring

Two major factors contribute to the current increase in the relative quantity of people that suffer from cardiac disorders: the aging of the world population and the fact that our eating and lifestyle habits are worsening. This places a huge pressure on the existing healthcare facilities, since those people that are prone to critical heart conditions such as congestive heart failure (CHF) or an acute myocardial infarction (AMI) should be monitored continously to spot early symptoms of such events. Since hospitals often lack the space to let every one of those patients stay over and the costs for doing so are too high, this is simply impossible.

Mobile ECG monitoring devices provide a solution to this problem. They can be placed at a patients home or even worn on the body, so as to not interfere with their daily activities. The data recorded by such a monitoring system is subsequently sent to a healthcare facility for analysis. The available energy in mobile ECG monitoring devices often limits the amount of data that can be recorded and transmitted. [19, 17]

Signal compression is an essential improvement for such a system. By compressing the data in the ECG monitor before sending it to the healthcare facility, the amount of data to be sent over the network is reduced. At the healthcare facility, the data can then be decompressed and analysed. This approach does, however, still require the ECG monitor to record the complete signal at a high sampling rate, store it temporarily and finally compress it before sending it over the network.

Compressive sensing (CS) eliminates the need to temporarily store the uncompressed signal by essentially performing the recording (sensing) and compression at the same time. By applying CS, the critical sampling rate defined in the classic Nyquist-Shannon sampling theorem is no longer a limitation on the minimum number of samples in a measurement. The complete ECG signal is thus never stored in the device, but instead an already compressed signal is written to memory before transmission. [5]

1.2 ECG Signals

1.2.1 Characteristics

An ECG is a recording of the activity of the heart. By measuring potential differences between two points on the human body, the propagation of the electrical waves that originate at the heart as a result of the different cardiac contractions can be measured. A projection of the electrical cardiac activity in the direction of the electrode is measured and visualised on a monitor.

In multi-channel ECG, signals are measured between several electrode pairs placed at different positions on the human body. By combining the information from each of the electrodes, the origin of the electrical activity can be located more precisely. Because each electrode records a projection of the same wave, the characteristics of the signals registered at all of these electrodes will be similar, a feature that will be used when reconstructing all channels from a multi-channel ECG at once. [30]

As described in [28], an ECG signal consists of 7 major characteristic parts. The P,Q,R,S and T peaks and 2 iso-electric parts that separate the P and T peak from the large QRS complex. These 7 parts are indicated in figure 1.1.

1.2.2 ECG Signal Simulator

An ECG signal simulator is used for some of the experiments that are part of this thesis. For the creation of a simple ECG signal model, knowledge of the possible dimensions of the characteristic peaks and iso-electric parts is required. In Table 1.1, the possible dimensions of the simulated ECG signal, defined as amplitude and duration intervals of each of the 7 characteristic parts, are given. The values in this table are based on those given in [28].

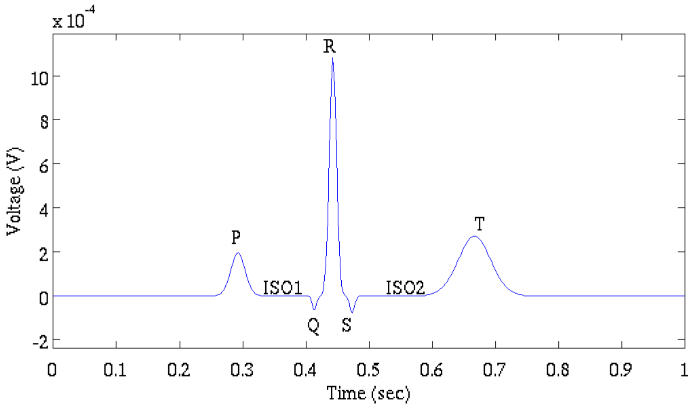


Figure 1.1: The 7 characteristic parts of an ECG in a simulated signal.

	Amplitude (mV)		Duration (ms)	
	Min	Max	Min	Max
P	0.1	0.2	60	80
ISO1	0	0	60	80
Q	0.06	0.08	20	30
R	0.9	1.1	30	40
S	0.06	0.08	20	30
ISO2	0	0	100	120
T	0.1	0.3	120	160

Table 1.1: Intervals of the dimensions of the characteristic parts of an ECG signal

The values in Table 1.1 are used to randomly generate ECG signals with peaks and iso-electric parts that have dimensions that are randomly picked from the intervals given in the table. All peaks are simulated as Gaussian peaks with a maximum that is randomly selected between the minimum and maximum amplitude and a support between the minimum and maximum duration. Of course this will not lead to a simulation of the full range of all possible ECG signals, but at least a basic dataset of clean signals can be generated.

Figure 1.2 shows an example of a 10 second ECG signal with a heart rate of 60 bpm at a sample rate $f_s = 360Hz$, simulated using this method. The random variation in the dimensions can clearly be seen from this example.

1.3 Compressive Sensing

The measurement $\mathbf{y}$ of a signal $\mathbf{x}$ is formulated in a discrete setting as the product of a sensing (or measurement) matrix $\boldsymbol{\phi}$ and the signal $\mathbf{x}$ in equation (1.1). In reality the measurement will not be an actual matrix multiplication, but rather the hardware equivalent thereof. However, for notation purposes, this matrix form is

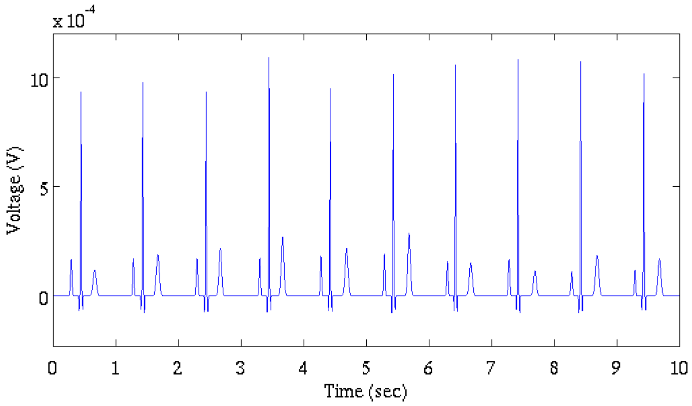


Figure 1.2: A 10 second ECG signal at $f_s = 360Hz$, generated with the simulator built for this thesis.

more convenient

$$\mathbf{y} = \boldsymbol{\phi}\mathbf{x} \tag{1.1}$$

with $\mathbf{y} \in \mathbb{R}^{n\times 1}$, $\boldsymbol{\phi} \in \mathbb{R}^{n\times N}$ and $\mathbf{x} \in \mathbb{R}^{N\times 1}$. If $n < N$, $\mathbf{y}$ is a compressed version of $\mathbf{x}$.

A requirement for a successful reconstruction of the original signal from the compressive measurement, is sparsity. This means that it must be possible to represent the signal as a sparse sequence (i.e. a vector that contains many zeros). This can be done in a sparse fashion (discussed in section 1.3.1) and a cosparse fashion (section 1.3.2). The importance of the sparsity lies in the fact that it reduces the number of possible signals that can be reconstructed from the measurement by assuming that the signal with the most sparse sequence corresponding to the measurement is the actual signal that was measured. For a multi-channel ECG, the sparse sequence of each channel will contain zeros at similar locations.

1.3.1 Sparse Reconstruction

In the sparse approach for signal reconstruction, it is assumed that a signal $\mathbf{x}$ can be represented as a linear combination of a number of (simple) functions. A collection of such functions is stored as the columns of a matrix referred to as a dictionary $\boldsymbol{\psi}$. If this dictionary can be used to represent every function in a certain set (e.g. the set $\mathbb{R}^{N\times 1}$ that contains all real functions of length N) then that dictionary is referred to as a basis for that set. The representation of $\mathbf{x}$ in a dictionary $\boldsymbol{\psi} \in \mathbb{R}^{N\times M}$ can be written as

$$\mathbf{x} = \boldsymbol{\psi}\mathbf{s} \tag{1.2}$$

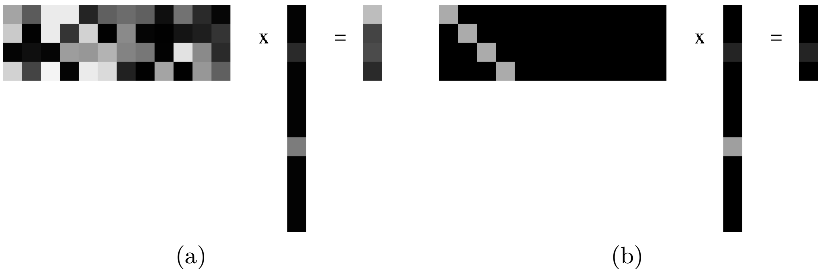
where $\mathbf{s} \in \mathbb{R}^{M\times 1}$ is a vector of coefficients used to create a linear combination of the elements in $\boldsymbol{\psi}$. If $\mathbf{s}$ contains many zeros, it is said to be sparse and the previously stated condition is fulfilled. Equations (1.1) and (1.2) can be combined to represent $\mathbf{y}$ as a function of $\mathbf{s}$

$$\mathbf{y} = \boldsymbol{\phi}\boldsymbol{\psi}\mathbf{s} \tag{1.3}$$

A second condition required for accurate reconstruction of a signal from its measurement is incoherence between $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$. This incoherence can be characterised by calculating the mutual coherence between $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ as

$$\mu(\boldsymbol{\phi}, \boldsymbol{\psi}) = \sqrt{N} \underset{1 \leq k,j \leq N}{max} | < \phi_k, \psi_j > | \tag{1.4}$$

If this mutual coherence is low, then $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are incoherent.[20] The importance of this incoherence is demonstrated in figure 1.3, which is a visual representation of equation (1.3). The matrix in figures 1.3a and 1.3b is the product $\boldsymbol{\phi\psi}$. This matrix is then multiplied by the long and sparse vector $\mathbf{s}$ (black squares are zeros) to obtain the shorter measurement vector $\mathbf{y}$.



(a)

(b)

Figure 1.3: Measuring a sparse coefficient vector using (a) an incoherent and (b) a coherent sensing matrix and dictionary combination.

In figure 1.3a, $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are incoherent, which is why the values (represented as gray values) in their product are seemingly randomly distributed. The vector $\mathbf{y}$ contains values that are random combinations of the values in $\mathbf{s}$ due to this randomness (and therefore due to the incoherence). Each element in $\mathbf{y}$ thus contains some information about each element in $\mathbf{s}$.

If the product of both matrices is coherent, one might obtain a situation like the one in figure 1.3b; although possibly less extreme. In this example, each element in $\mathbf{y}$ contains information about a single element from $\mathbf{s}$. It can be seen that some of the values of $\mathbf{s}$ are not captured in the measurement and this information is lost. [4]

When both the sparsity and the incoherence condition are satisfied, the following ℓ_0-minimisation can be used to retrieve the original signal $\mathbf{x}$.

$$\underset{\mathbf{s}}{min} ||\mathbf{s}||_0 \; s.t. \; \mathbf{y} = \boldsymbol{\phi\psi}\mathbf{s} \tag{1.5}$$

The desire for sparsity is expressed by the minimisation of $||\mathbf{s}||_0$ that counts the number of nonzero elements in $\mathbf{s}$. In practice, an ℓ_1- or ℓ_2-minimisation will be used in the optimisation for computational feasibility.

1.3.2 Cosparse Reconstruction

Instead of using a sparse coefficient vector to represent the signal in a dictionary, the cosparse approach obtains sparsity by transforming the signal into a sparse sequence

$\boldsymbol{\alpha}$. This sparse sequence is the product of an analysis matrix $\boldsymbol{\Omega} \in \mathbb{R}^{P \times N}$ and the signal $\mathbf{x}$. [22]

$$\boldsymbol{\alpha} = \boldsymbol{\Omega}\mathbf{x} \tag{1.6}$$

If $\boldsymbol{\alpha}$ is in fact sparse, then an ℓ_0-minimisation can be used to find $\mathbf{x}$ from $\mathbf{y}$, similar to the sparse approach.

$$\underset{\mathbf{x}}{min}||\boldsymbol{\Omega}\mathbf{x}||_0 \; s.t. \; \mathbf{y} = \boldsymbol{\phi}\mathbf{x} \tag{1.7}$$

Again, in practical implementations an ℓ_1- or ℓ_2-minimisation will be used in the optimisation for computational feasibility.

Note that incoherence is not an issue in the cosparse approach, as it is the original signal that is measured and not the sparse sequence. The only requirement is that $\boldsymbol{\phi}$ measures sufficient information of $\mathbf{x}$.

1.3.3 Multi-Channel Reconstruction

All of the previous paragraphs applied to the CS model for a single signal. It can easily be extended to a model for simultaneous measurement and reconstruction of several signals. This is convenient for, among others, multi-channel ECG applications where all channels could be compressed and reconstructed at once, instead of processing each channel one by one. Equation (1.1) can easily be adjusted to represent the measurement of multiple signals by replacing all the vectors by matrices with C columns, where each column contains a single signal.

$$\mathbf{Y} = \boldsymbol{\phi}\mathbf{X} \tag{1.8}$$

Similarly, this can be done for equations (1.2) and (1.6) and the ℓ_0-minimisation problems (1.5) and (1.7) to obtain the original signal matrix $\mathbf{X}$, measurements matrix $\mathbf{Y}$, the coefficients matrix $\mathbf{S}$ and the sparse sequence matrix $\mathbf{A}$ (the multi-channel equivalent of $\boldsymbol{\alpha}$).

In order to be able to solve a multi-channel ℓ_0-minimisation problem, either sparse or cosparse, it is required that the coefficients of all signal channels have a similar support, since this common support will be sought when solving these problems. As discussed in section 1.2 this is the case for the signals from a multi-channel ECG, since they are projections of the activity of a single source (i.e. the heart).

1.4 Signal Reconstruction from Noisy Compressive Sensing Measurements

In the previous section, it was assumed that the measurement is free of any noise (formula (1.1)), although this will seldom be the case in practical applications. Two types of noise may be present in the measurement. First, there is additive noise which is superimposed on the measurement. Several articles have already been written about additive noise and the algorithms that can be used to reliably reconstruct the original signal from the noisy measurement. [9, 16, 15]

The second type, multiplicative noise, disturbs the sensing matrix and thus the original signal is multiplied with this noise when it is measured. This may be problematic if the receiver assumes that the measurement was made with a certain known sensing matrix. At the moment, little research on this topic has happened. [15]

In chapter 4, two algorithms for more robust signal reconstruction from a noisy measurement are developed: one for the sparse approach and one for the cosparse approach.

1.5 Influence of Signal Reconstruction Accuracy on ECG Feature Detection

Since CS measurements store the information about a signal in less samples than the original signal, it can be expected that if too much compression is applied, valuable information will be lost and the original signal can no longer be completely reconstructed from the measurements. This may be problematic for a clinical diagnosis, where a loss of information may mean that a wrong conclusion is drawn from the data.

In chapter 5, the correlation between the amount of compression and the ability to perform certain analysis tasks on the reconstructed signal is sought, in order to determine when too much compression is applied to reliably perform a diagnosis based on the reconstructed signal. It is also investigated if there is a correlation between some accuracy metrics (to quantify the similarity between the original and reconstructed signal) and the detection of these features.

1.6 Contributions

The first major contribution of this thesis is the investigation of the possibilities of using the cosparse approach for signal reconstruction, which is still a relatively new topic in CS. Secondly, the speed and accuracy of multi-channel reconstruction algorithms is compared to those of their single-channel counterparts, to verify whether these are useful for application to multi-channel ECG reconstruction. The effect of using a binary matrix instead of a Gaussian one for the measurement on the computational requirements and reconstruction accuracy is also investigated. A fourth contribution is the adaptation of a sparse and a cosparse algorithm for the recovery of signals from measurements that have been altered by additive and multiplicative noise. Finally, a brief introductory study is made about the influence of the amount of compression that is applied to a signal on the ability of an algorithm to detect certain features in the reconstructed signal.

The rest of the thesis is structured as follows. In chapter 2, several existing methods to recover signals from CS measurements (both single- and multi-channel) are discussed. Next, in chapter 3, a cosparse algorithm is generalised for the reconstruction of all channels of a multi-channel ECG at once. In the second part of the chapter, reconstruction results of the methods discussed in chapter 2 and

SGAP are compared. In chapter 4 the influence of noise on the CS measurements is described and possible adaptations of a sparse and a cosparse algorithm to achieve better reconstruction quality from these noisy measurements are suggested. In chapter 5, the influence of the compression ratio on the ability to retrieve features from the reconstructed signal, is demonstrated. Finally, a conclusion about the thesis is drawn in chapter 6.

Chapter 2

Prior Work in Compressive Sensing

In this chapter, existing methods studied during the thesis are introduced in order to gain a better understanding of the contributions of the thesis. What will be referred to as a 'method' in the rest of the text consists of a sensing matrix, a dictionary and an algorithm for signal reconstruction in case of the sparse approach and a sensing matrix, an analysis matrix and an algorithm in case of the cosparse approach.

Sensing matrices are discussed in the first part of the chapter. The same sensing matrices were used for the sparse and cosparse methods, since the measurement phase is independent of the signal reconstruction. Next, the dictionaries and algorithms for sparse reconstruction are introduced and finally, the analysis matrices and algorithms used in cosparse methods are discussed.

Three major types of algorithms can be distinguished. The first group are the convex optimisation algorithms that convert the optimisation problem to a convex linear program. A popular algorithm of this type, Basis Pursuit (BP) was tested in this thesis. Secondly, there are greedy algorithms that iteratively estimate the support (sparse approach) or cosupport (cosparse approach) of $\mathbf{s}$ and $\boldsymbol{\alpha}$, respectively, and calculate the optimal coefficients corresponding to this estimate by solving a linear least-squares problem. The third type of algorithms are thresholding algorithms that iteratively adjust the coefficients by calculating an approximation of $\mathbf{s}$ and keeping only the larger values. The last algorithms that are discussed are the Block Sparse Bayesian Learning (BSBL) algorithms. They can not be categorised as any of the main types. BSBL algorithms assume that groups of coefficients are distributed according to a Gaussian distribution and learns the parameters of these distributions.

2.1 Sensing Matrices

It has been found that Gaussian and other random matrices are very suitable to perform compressive measurements, since there is a very high probability of such a matrix being incoherent with a dictionary or basis. [8] Two such random matrices were tested in this thesis.

Gaussian This is the most frequently used sensing matrix in compressive sensing. Its elements are selected randomly from a standard Gaussian distribution ($\mu = 0$, $\sigma = 1$). In order to reduce the computational complexity, the matrix can be made sparse. This means that a random selection of elements is kept equal to 0, while the others are still drawn randomly from a standard Gaussian distribution. A small example of a Gaussian sensing matrix with 50% sparsity that measures a 6-sample signal as a 4-sample vector is given in equation (2.1).

$$\boldsymbol{\phi}_{Gauss} = \begin{bmatrix} 0.925 & 0 & 0 & 0.225 & -0.846 & 0 \\ 0 & 0.081 & -1.474 & 0 & 0 & -0.717 \\ -0.805 & 0 & 1.779 & 0 & -0.585 & 0 \\ 0 & -0.236 & 1.074 & 0 & 0.510 & 0 \end{bmatrix} \tag{2.1}$$

Binary The binary sensing matrix is created by setting a random selection of elements to zero and the other elements to 1. This type of random matrix can be implemented more easily in a hardware device than the Gaussian matrix due to its binary nature. In equation (2.2) the binary equivalent of the matrix in equation (2.1) is shown.

$$\boldsymbol{\phi}_{binary} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{2.2}$$

2.2 Sparse Reconstruction

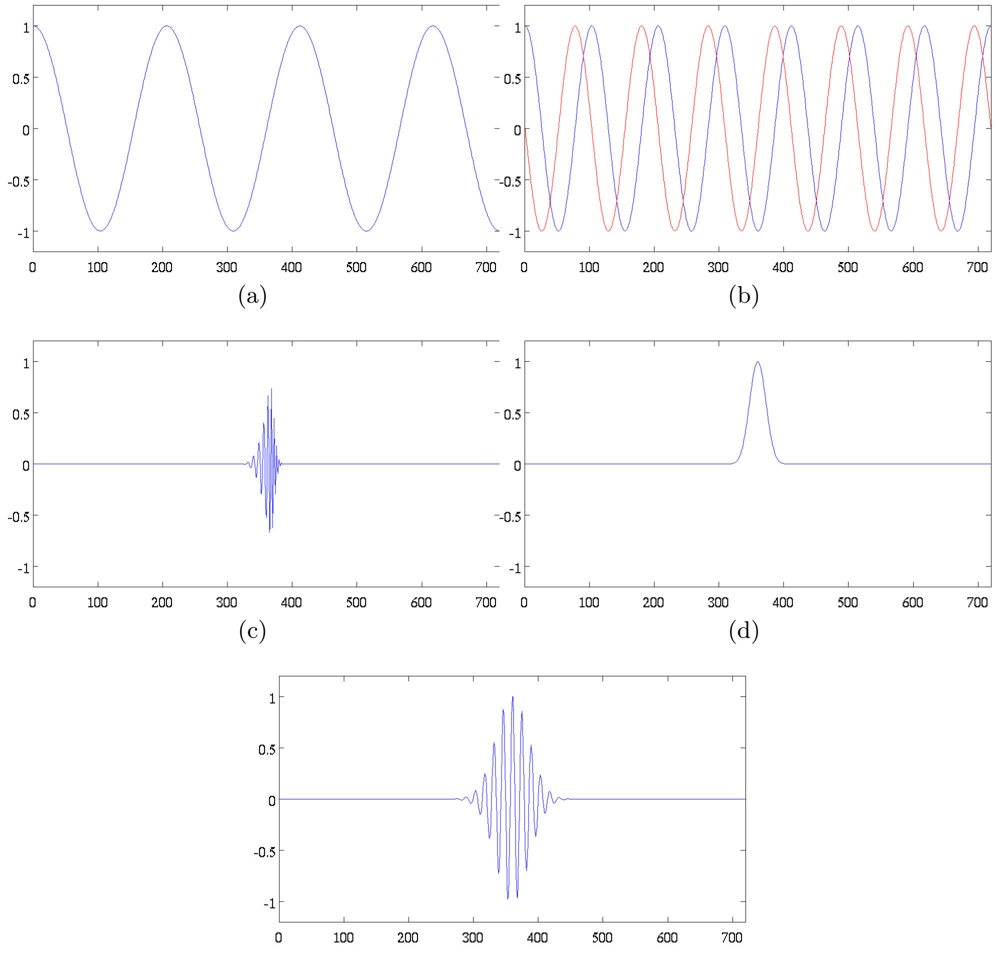
Sparse reconstruction methods require a dictionary in which the signal is sparse (i.e. coefficient vector $\mathbf{s}$ contains many zeros) and an algorithm to reconstruct $\mathbf{s}$. Equation (1.2) is subsequently used to reconstruct the signal from the recovered coefficients.

2.2.1 Dictionaries

There are several possible dictionaries that can represent a signal as a sparse coefficient vector. Some of the more popular dictionaries were tested in this thesis. Elements from the Discrete Cosine Transform dictionary and Discrete Fourier Transform dictionary both have infinite support in the time domain, while the Discrete Wavelet Transform, B-spline and Gabor dictionaries are localised both in space and in time. An example of a typical element from each of the dictionaries is shown in figure 2.1

Discrete Cosine Transform (DCT) The DCT dictionary is based on the concept of the Discrete Cosine Transform that transforms a signal into a set of cosine functions of different frequencies, based on the fact that any real signal can be represented as a linear combination of cosine functions. Each column $\boldsymbol{\psi}_k$ in the dictionary is a cosine function of a different frequency, defined as

$$\boldsymbol{\psi}_k(t) = cos(\pi k t) \tag{2.3}$$



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

Figure 2.1: An example of a single element from each dictionary: (a) DCT (b) DFT (real part in blue, imaginary part in red) (c) DWT (d) B-spline and (e) Gabor

where k is the number of the column $\boldsymbol{\psi}_k$ in the dictionary and t is a variable ranging from 0 to 1 in N steps. If the largest value of k equals N, then $\boldsymbol{\psi}$ is said to be complete, if it is larger than N, then $\boldsymbol{\psi}$ is over-complete. [20, 2]

Discrete Fourier Transform (DFT) The DFT dictionary is based on the Discrete Fourier Transform, a transformation that is quite similar to the DCT in that it transforms a signal into a series of functions with full support in the time domain. The main difference is that in case of the DFT, complex exponentials are used where the DCT uses cosine functions. The dictionary contains these complex exponentials of different frequencies in its columns and is defined as [20]

$$\boldsymbol{\psi}_k(t) = e^{i2\pi kt} \tag{2.4}$$

By using Euler's Identity

$$e^{it} = cos(t) + isin(t) \tag{2.5}$$

it can be seen that the dictionary is even more similar to the DCT dictionary than equation (2.4) would suggest, since each element is actually a combination of a cosine and sine multiplied by the imaginary unit i.

Discrete Wavelet Transform (DWT) Different from the basis functions used in the DCT and DFT, wavelets are localised not only in space but also in time. This means that when a signal is transformed to the wavelet domain, each of the coefficients will correspond to information that is both located at a certain position in the signal and describes a limited range of frequencies in the signal.

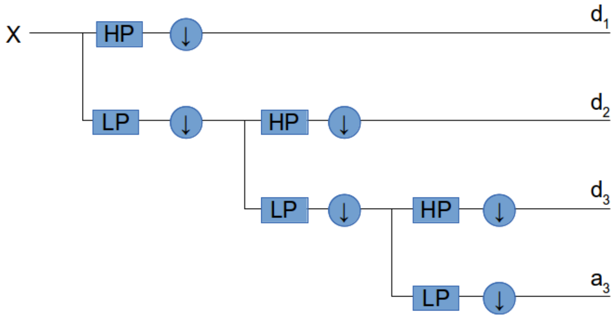


Figure 2.2: Example of a 3-level multi-resolution filter.

The concept of wavelet transforms can be represented by a multi-resolution filter. A signal x is first filtered with a high-pass filter and a low-pass filter that divide the frequency content in two parts of equal bandwidth. Both parts are subsequently downsampled by a factor 2 (as indicated by the arrows in figure 2.2). A schematic representation of a 3-level multi-resolution filter is shown in figure 2.2. The coefficients on the right side (d_1,d_2,d_3,a_1) are the so-called detail and approximation coefficients that are the result of the (consecutive) filter(s).

In the wavelet transform, the detail and approximation coefficients are used to represent the signal as a linear combination of translated and dilated versions of a scaling function and wavelet functions. The scaling function is constructed using

$$\phi_{j,n}(t) = \frac{1}{\sqrt{2^j}}\phi(\frac{t-n}{2^j}) \tag{2.6}$$

where j is the level, which determines the scale (and thus the frequency content) of the wavelet and n is the shift, which determines the position of the wavelet on the time axis. The wavelet functions, on the other hand, are constructed as

$$\psi_{j,n}(t) = \frac{1}{\sqrt{2^j}}\phi(\frac{t-2^j n}{2^j}) \tag{2.7}$$

The wavelet transform can be formulated in matrix notation, where the rows of the transformation matrix are shifted and scaled wavelet and scaling functions. A small example of how a signal can be represented in a wavelet dictionary is given in equation (2.8). Note that coefficients that no longer fit in a row are reflected to the first elements of that row.

$$\mathbf{x} = \boldsymbol{\psi}_{DWT}.\mathbf{s}$$

$$\Longrightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} & 0 & 0 & 0 \\ h_{2,4} & 0 & 0 & 0 & h_{2,1} & h_{2,2} & h_{2,3} \\ -h_{2,1} & -h_{2,2} & h_{2,3} & h_{2,4} & 0 & 0 & 0 \\ h_{2,4} & 0 & 0 & 0 & -h_{2,1} & -h_{2,2} & h_{2,3} \\ -h_{1,1} & h_{1,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -h_{1,1} & h_{1,2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -h_{1,1} & h_{1,2} & 0 \\ h_{1,2} & 0 & 0 & 0 & 0 & 0 & -h_{1,1} \end{bmatrix}^T . \begin{bmatrix} a_{2,1} \\ a_{2,2} \\ d_{2,1} \\ d_{2,2} \\ d_{1,1} \\ d_{1,2} \\ d_{1,3} \\ d_{1,4} \end{bmatrix} \tag{2.8}$$

B-spline A spline is a parameterised polynomial, defined as the combination of a number of simple polynomials between a series of points (called knots). In a dictionary, these B-spline elements will look like small smooth waves with a limited support on the time axis. As a consequence, they are located both in time and frequency. A B-spline dictionary is constructed as

$$\boldsymbol{\psi}_k(\Delta : \cup_{j=1}^{l}\Delta_j) := \cup_{j=1}^{l}\{B_{m,k}^{j} : k = 1, ..., m + \#\Delta_j\} \tag{2.9}$$

where the $\Delta_j, j = 1, ..., l$ are partitions of $\Delta := t_i{}_{i=0}^{N+1}, N \in \mathbb{N}, s.t.c = t_0 < t_1 < ... < t_N < t_{N+1} = d$. Each B-spline $B_{m,j}$ is recursively built as

$$\begin{cases} B_{1,k}(t) = \begin{cases} 1, b_k \leq t < b_{k+1} \\ 0, otherwise \end{cases} \\ B_{m,k}(t) = \frac{t - p_j}{p_{j+m-1} - p_j} B_{m-1,j}(t) + \frac{pj+m-t}{pj+m-pj+1} B_{m-1,j+1}(t) \end{cases} \tag{2.10}$$

where m is the step number in the recursive formula, b_k and b_{k+1} are border points associated with the B-spline for the first iteration and the p values are the knots defined as $p_{m+i} = t_i, i = 1, ..., N, t_1 < ... < t_N$. [29]

Gabor A Gabor dictionary consists of Gabor atoms that are cosine functions defined within a Gaussian window to make them localised in time. The atoms in the dictionary are defined as

$$\boldsymbol{\psi}_k(t) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-(t-\mu_k)^2/\sigma_k^2} cos(\omega_k t + \theta_k) \tag{2.11}$$

where μ and $\sigma > 0$ are the mean and standard deviation of the Gaussian window, respectively; and $\omega = 2\pi f \geq 0$ and $\theta \in [0, 2\pi]$ are the frequency and phase of the cosine within the Gaussian window, respectively. [1]

2.2.2 Algorithms

The algorithms for the sparse approach that were tested for this thesis include Basis Pursuit (BP), greedy algorithms for single-channel reconstruction (Orthogonal (Multi) Matching Pursuit (O(M)MP) and Compressive Sampling Matching Pursuit (CoSaMP)), Iterative Hard Thresholding (IHT) and Block Sparse Bayesian Learning (BSBL).

Basis Pursuit (BP) The basis pursuit algorithm converts equation (1.5) into a linear program in order to solve it. The standard form of the primal of a linear program is

$$min\, c^T s\; s.t.\, \boldsymbol{\phi\psi} s = y \tag{2.12}$$

If $c = [1\,1\;\ldots\;1]^T$ then this corresponds to an ℓ_1-minimisation equivalent of equation (1.5).

According to duality theory, this problem can be written in an equivalent dual form [10]

$$max\, y^T s\; s.t.\, \boldsymbol{\phi\psi}^T s - 2v = -c, 0 \leq v \leq c \tag{2.13}$$

This dual problem is then solved using the interior-point method implemented in MATLAB® through the `linprog`-command. [11]

Greedy Algorithms, Single-Channel Three different types of greedy algorithms were tested for this thesis: Orthogonal Matching Pursuit (OMP) [26, 31], Orthogonal Multi-Matching Pursuit (OMMP) [33] and Compressive Sampling Matching Pursuit (CoSaMP) [12]. OMP is the basic algorithm, while OMMP and CoSaMP both are variations thereof.

In OMP, the index of the element from $\boldsymbol{\psi}$ that corresponds most to $\mathbf{y}$, according to the estimate $(\boldsymbol{\psi\phi})^T\mathbf{y}$, is added to the current estimate of the support Ω_k. Next, a least-squares optimisation is used to calculate the optimal value of the coefficient at that position in the support. Finally a residual $r_k = y - \boldsymbol{\phi\psi}\mathbf{s_k}$ is calculated. In the following iterations, $\mathbf{r}_{k-1}$ is used instead of $\mathbf{y}$ to find the next element to add to the support Ω and all coefficients are recalculated whenever an index is added to the support.

The OMMP and CoSaMP algorithms both make small adjustments to OMP. In OMMP, several indices are added to Ω during each iteration, instead of just one, in order to have the greedy algorithm converge to a solution faster. CoSaMP does this as well but only keeps a predetermined amount of the largest coefficients in the estimate $\mathbf{s}_k$ and sets the rest to 0 at the end of each iteration. Ideally, the amount of coefficients that is kept corresponds to the actual sparsity of the coefficient vector.

The stopping criterions are used to check whether the residual is small enough, compared to the original measurement (e.g. 5% of its size) and whether this relative size is smaller than that of the previous residual. This last criterion is based on the fact that the decrease in error $||\mathbf{r}_k||_2$ becomes gradually smaller as the number of iterations increases. So if the residuals size (i.e. 2-norm), relative to the measurement, starts to increase compared to the previous iteration, then this is an indication that the optimal solution was found in the previous iteration.

```
Algorithm 1: OMP, OMMP & CoSaMP

In: $\mathbf{y}$, $\boldsymbol{\phi}$, $\boldsymbol{\psi}$
Out: $\hat{\mathbf{x}}$

Initialisation ($k = 0$)
    • Initialise Support: $\hat{\Omega}_k = \emptyset$
    • Initialise Residual: $\mathbf{r}_k = \mathbf{y}$

Iterations $k \to k + 1$
    • Update Support: $\hat{\Omega}_k = \hat{\Omega}_{k-1} \cup \{\underset{i \notin \hat{\Omega}_{k-1}}{argmax} \, |(\boldsymbol{\phi\psi})^T \mathbf{r_k}_i|\}$
        – (OMP) add maximising $i$ to $\hat{\Omega}_{k-1}$
        – (OMMP/CoSaMP): add number of most maximising $i$'s' to $\hat{\Omega}_{k-1}$
    • Update Solution: $\hat{\mathbf{s_k}} = \underset{\mathbf{s}}{argmax} ||\mathbf{y} - (\boldsymbol{\phi\psi})_{\Omega_k} x||_2$
    • (CoSaMP) Set smallest values in $\mathbf{s}$ to 0.
    • Calculate Residual: $\mathbf{r}_k = \mathbf{y} - \boldsymbol{\phi\psi}\mathbf{s_k}$

Stopping Criteria
    • $\frac{||\mathbf{r}_k||_2}{||\mathbf{y}||_2} < 0.05 \implies \mathbf{s_k}$ is optimal
    • $\frac{||\mathbf{r}_k||_2}{||\mathbf{y}||_2} \leq \frac{||\mathbf{r}_{k-1}||_2}{||\mathbf{y}||_2} \implies \mathbf{s_{k-1}}$ is optimal
```

Greedy Algorithms, Multi-Channel All of the aforementioned greedy algorithms can be generalised to enable simultaneous reconstruction of multiple channels. This requires two adjustments to be made to the existing algorithms.

First of all, the algorithm should locate the elements in $\boldsymbol{\psi}$ that correlate strongly to all or most of the measurements. This can be achieved by finding the largest elements in the row wise summation of $(\boldsymbol{\phi\psi})^T \mathbf{R}_{k-1}$, where $\mathbf{R}_{k-1}$ is now a matrix with residuals in its columns, corresponding to each of the channels.

$$\hat{\Omega}_k = \hat{\Omega}_{k-1} \cup \{\underset{i \notin \hat{\Omega}_{k-1}}{argmax} \, |(\boldsymbol{\phi\psi})^T \mathbf{R_k}_i [1\, 1\, \ldots\, 1]^T \tag{2.14}$$

Secondly, the stopping criteria should be altered, since there is now a residual matrix $\mathbf{R}_k$ instead of a residual vector $\mathbf{r}_k$. A vector containing the relative residual norm $||\mathbf{r}_k||_2/||\mathbf{y}||_2$ for each of the c channels can be calculated as

$$\mathbf{r}_{MC_k} = \begin{bmatrix} \frac{||\hat{\mathbf{R}}_k||_{2,col}(1)}{||\hat{\mathbf{Y}}||_{2,col}(1)}, & \frac{||\hat{\mathbf{R}}_k||_{2,col}(2)}{||\hat{\mathbf{Y}}||_{2,col}(2)}, & \ldots, & \frac{||\hat{\mathbf{R}}_k||_{2,col}(c)}{||\hat{\mathbf{Y}}||_{2,col}(c)} \end{bmatrix}^T \tag{2.15}$$

where $||\mathbf{C}||_{2,col}(i)$ is the i-th element in a vector that contains the column-wise norms of a matrix $\mathbf{C}$, calculated as

$$||\mathbf{C}||_{2,col} = \sqrt{\begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{C}}_{1,1}^2 & \hat{\mathbf{C}}_{1,2}^2 & \ldots & \hat{\mathbf{C}}_{1,c}^2 \\ \hat{\mathbf{C}}_{2,1}^2 & \hat{\mathbf{C}}_{2,2}^2 & \ldots & \hat{\mathbf{C}}_{2,c}^2 \\ \vdots & & \ddots & \vdots \\ \hat{\mathbf{C}}_{N,1}^2 & \hat{\mathbf{C}}_{N,2}^2 & \ldots & \hat{\mathbf{C}}_{N,c}^2 \end{bmatrix}} \tag{2.16}$$

It can then be verified that a) if each element in $\mathbf{r}_{MC_k}$ is smaller than 0.05, the current solution is optimal or b) if any one element in $\mathbf{r}_{MC_k}$ is larger than the corresponding element in $\mathbf{r}_{MC_{k-1}}$, then the previous solution is the optimal solution.

Iterative Hard Thresholding (IHT) The concept behind Iterative Hard Thresholding (IHT) is that a coefficient vector can be recovered from a measurement by calculating an estimate of the original signal and then thresholding the result of this estimate by setting all but s values to zero, where s is (an estimate of) the sparsity of $\mathbf{s}$. An iteration of the IHT algorithm can be expressed mathematically as

$$\mathbf{s}_{k+1} = H_s(\mathbf{s}_k + (\boldsymbol{\phi\psi})^T(y - \boldsymbol{\phi\psi}\mathbf{s}_k)) \tag{2.17}$$

where k is the iteration number of the previous iteration and H_s is the thresholding operator that sets all but s values in the estimate to zero. The main drawback of this algorithm is that the value s should be known beforehand, which is not always the case. [6]

Block Sparse Bayesian Learning (BSBL), Single-Channel The Block Sparse Bayesian Learning algorithm assumes that a coefficient vector is divided into g blocks, with only some of these blocks containing nonzero elements. BSBL assumes that each block satisfies a parameterised multivariate Gaussian distribution

$$p(\mathbf{s}; \gamma_i, \mathbf{B}_i) \sim \mathcal{N}(\mathbf{0}, \gamma_i \mathbf{B}_i), i = 1, ..., g \tag{2.18}$$

where $\mathbf{B}_i$ is a positive definite matrix that represents the correlation within the i-th block and $\gamma_i \geq 0$ controls the sparsity within the i-th block. If $\gamma_i = 0$ then all coefficients in the i-th block are equal to 0. The parameters will then be learned, based on the observed measurement. [35]

Three algorithms, each with its own set of learning rules for the parameters $\mathbf{B}_i$ and γ_i, are presented in [35]. Two of those algorithms, the Expectation-Maximisation based BSBL-EM [34] and the Bounded-Optimisation based BSBL-BO were tested for the thesis.

The learning rules in BSBL-EM are based on the concepts of Expectation-Maximisation, where the coefficient vector is calculated, based on a certain set of parameters (Expectation) and based on this vector, the parameters are updated to maximise the probability of getting this result in the presence of a certain measurement (Maximisation).

In BSBL-BO, the learning rules are derived from Bounded Optimisation methods, where a cost function is optimised in the presence of a set of constraints. The learning rule for $\mathbf{B}_i$ is the same as the one used in BSBL-EM, while the learning rule for γ_i is different, resulting in faster performance.

Since this is quite a complex algorithm and the source code[1] was made available by the author of [34], this code was used to test the algorithm.

2.3 Cosparse Reconstruction

The cosparse approach for signal reconstruction is rather new and up until today, only a few algorithms exist in this domain. In this thesis, three analysis matrices and a greedy algorithm are studied. This greedy algorithm is also generalised for simultaneous reconstruction of multiple signals in chapter 3.

2.3.1 Analysis Matrices

First Order Derivative This type of matrix can be used to calculate the first order derivative of a signal, based on the finite difference formula. It is defined as

$$\boldsymbol{\Omega}_1 = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & 0 \\ \vdots & & & & 1 & -1 \\ 0 & \dots & \dots & 0 & 0 & 1 \end{bmatrix} \tag{2.19}$$

so that each row in the matrix calculates the difference between a sample and the subsequent sample in the signal it is multiplied with. The resulting sequence is sparse in the sense that the differences between two subsequent samples are often very small or equal to zero. The last row only contains a 1 in the last position, which allows for an easy calculation of the original signal $\mathbf{x}$ from the product $\boldsymbol{\Omega}_1\mathbf{x}$. The inverse of $\boldsymbol{\Omega}_1$ is simply an upper triangular matrix with all nonzero elements equal to 1 and is therefore very easy to calculate.

[1] https://sites.google.com/site/researchbyzhang/software

Second Order Derivative Since the second order derivative of a function is the first order derivative of its first order derivative, the matrix for such an operation can be calculated as the product of two first order derivative matrices.

$$\boldsymbol{\Omega}_2 = \boldsymbol{\Omega}_1^2 \tag{2.20}$$

Like $\boldsymbol{\Omega}_1$, the inverse of $\boldsymbol{\Omega}_2$ has a very simple form.

$$\boldsymbol{\Omega}_2^{-1} = \begin{bmatrix} 1 & 2 & 3 & \dots & N \\ 0 & 1 & 2 & \dots & N-1 \\ \vdots & & \ddots & & \vdots \\ \vdots & & & 1 & 2 \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix} \tag{2.21}$$

Wavelet The wavelet analysis matrix is based on the DWT dictionary that was defined in section 2.2.1. In equation (1.2), the dictionary is used as a synthesis matrix, which is why the analysis matrix can be calculated as the (pseudo)inverse of the dictionary matrix.

$$\boldsymbol{\Omega}_{DWT} = \boldsymbol{\psi}_{DWT}^{\dagger} \tag{2.22}$$

2.3.2 Algorithms

Due to good experiences with greedy algorithms for the sparse approach and a lack of existing algorithms for the cosparse, only one greedy algorithm, called Greedy Analysis Pursuit, was tested.

Greedy Analysis Pursuit (GAP), Single-Channel Greedy Analysis Pursuit is a greedy algorithm that is somewhat similar to the OMP and OMMP algorithms discussed in section 2.2.2.

The signal reconstruction in GAP is based on knowledge of two things: the measurement made by compressive sensing with a known sensing matrix and the fact that the co-support of the cosparse sequence $\boldsymbol{\Omega}\mathbf{x}$ should contain only zeros. The former is defined in equation (1.1), while the latter can be defined as

$$\boldsymbol{\Omega}_{\Lambda}\mathbf{x} = \mathbf{0} \tag{2.23}$$

where $\boldsymbol{\Omega}_{\Lambda}$ is the analysis matrix with the rows whose indices are not in the co-support Λ removed. Both facts can be combined into the following system in matrix form.

$$\begin{bmatrix} \boldsymbol{\phi} \\ \boldsymbol{\Omega}_{\Lambda} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \tag{2.24}$$

Which can be solved using a Moore-Penrose pseudoinverse to find $\mathbf{x}$.

The co-support is initially unknown, but is iteratively estimated from the projection of the signal estimate $\boldsymbol{\alpha}_k = \boldsymbol{\Omega}\hat{\mathbf{x}}_{k-1}$ by determining the locations of the largest elements in $\boldsymbol{\alpha}_k$ and removing these from the co-support. Ideally this gives a better estimate of the co-support in each subsequent iteration, which aids in refining $\hat{\mathbf{x}}_k$ in each iteration and allows for detection of smaller elements outside of the co-support in later iterations. The complete algorithm is shown in Algorithm 2. [22, 24]

```
Algorithm 2: GAP

In: $\mathbf{y}$, $\boldsymbol{\phi}$, $\boldsymbol{\Omega}$
Out: $\hat{\mathbf{x}}$

Initialisation ($k = 0$)
    • Initialise Co-Support: $\hat{\Lambda}_k = \{1, 2, 3, ..., p\}$
    • Initialise Solution: $\hat{\mathbf{x}}_k = \begin{bmatrix} \boldsymbol{\phi} \\ \sqrt{\lambda}\boldsymbol{\Omega} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}$

Iterations $k \rightarrow k + 1$
    • Project: $\boldsymbol{\alpha}_k = \boldsymbol{\Omega}\hat{\mathbf{x}}_{k-1}$
    • Update Co-Support: $\hat{\Lambda}_k = \hat{\Lambda}_{k-1} \backslash \{\underset{i\in\hat{\Lambda}_{k-1}}{argmax}\, |\boldsymbol{\alpha}_i|\}$
    • Update Solution: $\widehat{\mathbf{x}_k} = \begin{bmatrix} \boldsymbol{\phi} \\ \sqrt{\lambda}\boldsymbol{\Omega}_{\hat{\Lambda}_k} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}$

Stopping Criteria
    • $k < K_{max}$
    • $r_k > r_{k-1}$
      $r_k = |1 - \frac{||\hat{\mathbf{x}}_k||}{||\hat{\mathbf{x}}_{k-1}||}|$
```

2.4 Discussion

A selection of the reconstruction algorithms that have been developed over the years was discussed in this chapter. All three major groups (basis pursuit, thresholding and greedy algorithms) are represented and an additional more complex algorithm (BSBL) is included. Several dictionaries and analysis matrices for the sparse and cosparse approach, respectively, have been discussed as well. In the next chapter, the GAP algorithm is generalised to a cosparse greedy algorithm that can reconstruct multiple signals simultaneously and its performance is compared to that of the methods presented in this chapter.

Chapter 3

Simultaneous Greedy Analysis Pursuit: Algorithm and Comparison to Other Methods

A new cosparse greedy algorithm for simultaneous reconstruction of multiple signals is developed in this chapter. It is based on the knowledge about the generalisation of sparse greedy algorithms for simultaneous signal reconstraction and about greedy algorithms for cosparse reconstruction,

Afterwards, the performance of the newly developed algorithm is compared to the methods discussed in chapter 2. Based on the results of a number of numerical experiments, a selection of the best algorithms, dictionaries, analysis matrices and sensing matrices for fast and accurate reconstruction of ECG signals is made.

Most importantly, this chapter provides evidence on why the use of the cosparse approach may be beneficial in terms of reconstruction accuracy and why using multi-channel reconstruction will decrease the total processing time. It is also shown why cosparse algorithms are more suitable for hardware implementation using a binary sensing matrix.

3.1 Simultaneous Greedy Analyis Pursuit

Based on a series of experiments (see also section 3.2), it seems that the cosparse approach is very successful at accurately reconstructing signals from measurements obtained by CS. However, it is a relatively new concept and so very little research has been done on the topic. While the concept of simultaneous reconstruction has already been explored in the sparse approach, it has barely been touched upon in the cosparse approach.

Similar to the shared support in sparse methods, it is required by this algorithm that the sparse sequences of all signals share a similar (co-)support. The research in this thesis focuses on ECG signals, for which this is definitely the case when using the presented analysis matrices. Due to its general nature, the algorithm could be applied to other groups of signals with similar (co-)support as well.

Algorithm 3: SGAP

In: $\mathbf{Y} \in \mathbb{R}^{n \times c}$, $\boldsymbol{\phi} \in \mathbb{R}^{n \times N}$, $\boldsymbol{\Omega} \in \mathbb{R}^{p \times N}$
Out: $\hat{\mathbf{X}} \in \mathbb{R}^{N \times c}$

Initialisation $(k = 0)$

- Initialise Co-Support: $\hat{\Lambda}_k = \{1, 2, 3, ..., p\}$
- Precalculations
  - $\boldsymbol{\phi}_\phi = \boldsymbol{\phi}^T \boldsymbol{\phi}$
  - $\boldsymbol{\phi}_Y = \boldsymbol{\phi}^T \mathbf{Y}$
- Initialise Solution: $\hat{\mathbf{X}}_k = (\boldsymbol{\phi}_\phi + \lambda \boldsymbol{\Omega}^T \boldsymbol{\Omega})^{-1} \boldsymbol{\phi}_Y$

Iterations $k \rightarrow k + 1$

- Project: $\mathbf{A} = \Omega \hat{\mathbf{x}}_{k-1}$
- Row-Wise Summation: $\boldsymbol{\alpha} = \mathbf{A}.\begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T$
- Update Co-Support: $\hat{\Lambda}_k = \hat{\Lambda}_{k-1} \backslash \{\underset{i \in \hat{\Lambda}_{k-1}}{argmax}\, |\boldsymbol{\alpha}_i|\}$
- Update Solution: $\hat{\mathbf{x}}_k = (\boldsymbol{\phi}_\phi + \lambda \boldsymbol{\Omega}^T_{\hat{\Lambda}_k} \boldsymbol{\Omega}_{\hat{\Lambda}_k})^{-1} \boldsymbol{\phi}_Y$, where $\boldsymbol{\Omega}_{\hat{\Lambda}_k}$ is $\boldsymbol{\Omega}$ with the rows with row numbers not in $\hat{\Lambda}_k$ set to 0

Stopping Criteria

- $k < K_{max}$
- $\mathbf{r}_k(i) > \mathbf{r}_{k-1}(i)$ for $i \in \{1, ..., c\}$

$$\mathbf{r}_k = \begin{bmatrix} 1 - \frac{||\hat{\mathbf{x}}_k||_{2,col}(1)}{||\hat{\mathbf{x}}_{k-1}||_{2,col}(1)} \\ 1 - \frac{||\hat{\mathbf{x}}_k||_{2,col}(2)}{||\hat{\mathbf{x}}_{k-1}||_{2,col}(2)} \\ \dots \\ 1 - \frac{||\hat{\mathbf{x}}_k||_{2,col}(c)}{||\hat{\mathbf{x}}_{k-1}||_{2,col}(c)} \end{bmatrix}$$

$$||\hat{\mathbf{x}}_k||_{2,col} = \sqrt{\begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}^T \begin{bmatrix} \hat{\mathbf{x}}^2_{k,11} & \hat{\mathbf{x}}^2_{k,12} & \cdots & \hat{\mathbf{x}}^2_{k,1c} \\ \hat{\mathbf{x}}^2_{k,21} & \hat{\mathbf{x}}^2_{k,22} & \cdots & \hat{\mathbf{x}}^2_{k,2c} \\ \vdots & & \ddots & \vdots \\ \hat{\mathbf{x}}^2_{k,N1} & \hat{\mathbf{x}}^2_{k,N2} & \cdots & \hat{\mathbf{x}}^2_{k,Nc} \end{bmatrix}}$$

3.1.1 Methods

In order to improve the computational efficiency of the GAP algorithm before generalising it, precalculations of some matrices that were previously calculated in each iteration but that did not change their values from iteration to iteration were included. According to [24], the solution update in GAP can be calculated as

$$\hat{\mathbf{x}_k} = \begin{bmatrix} \boldsymbol{\phi} \\ \sqrt{\lambda}\boldsymbol{\Omega}_{\hat{\Lambda}_k} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} = (\boldsymbol{\phi}^T\boldsymbol{\phi} + \lambda\boldsymbol{\Omega}^T_{\hat{\Lambda}_k}\boldsymbol{\Omega}_{\hat{\Lambda}_k})^{-1}\boldsymbol{\phi}^T\mathbf{y} \tag{3.1}$$

where λ is a small positive constant that ensures that $\boldsymbol{\phi}^T\boldsymbol{\phi} + \lambda\boldsymbol{\Omega}^T_{\hat{\Lambda}_k}\boldsymbol{\Omega}_{\hat{\Lambda}_k}$ is a well conditioned matrix and can therefore be inverted without any problems.

Matrix $\boldsymbol{\phi}$ and vector $\mathbf{y}$ are fixed over the course of the algorithm and so their products $\boldsymbol{\phi}^T\boldsymbol{\phi}$ and $\boldsymbol{\phi}^T\mathbf{y}$ will be, as well. Calculation of these matrix products can be done in the initialisation phase of the algorithm, and its stored result can be used in each iteration instead. These precalculations are included in the SGAP algorithm as well.

It is known from section 1.3.3 that the coefficients of the different channels of a multi-channel ECG in certain dictionaries share a similar support. Similarly, it can be shown that the sparse sequences (1.6) of such signals share a similar support and consequently a similar co-support. This is because of the fact that the signals themselves share (inversely) correlated levels of activity in corresponding sections, although their exact values may differ.

The generalisation of GAP to an algorithm that can process multiple signals at once requires two major changes. First of all, the projection $\mathbf{A}_k = \Omega\mathbf{X}\hat{}_{k-1}$ is now no longer a vector, but a matrix with the projections of the different channels stored in its columns. Instead of searching the locations that correspond to positions outside of the co-support estimate by finding large values in $\boldsymbol{\alpha}_k$, positions of large values in all or most columns of $\mathbf{A}_k$ need to be found. This can be achieved by performing a row-wise summation on the matrix $\mathbf{A}_k$

$$\boldsymbol{\alpha}_k = \mathbf{A}_k . \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T \tag{3.2}$$

which results in a vector $\boldsymbol{\alpha}$ not unlike the one obtained in GAP. This $\boldsymbol{\alpha}$ has large values at locations that are not part of the co-support of most channels and can therefore be removed from the co-support estimate.

The second stopping criterion is changed as well. A column-wise norm calculation is applied to the current signal estimate matrix in order to obtain a vector that contains the norms of each signal. Each element of this vector is compared to the corresponding value in the norm vector calculated from the previous signal estimate matrix to quantify the amount of change in signal estimate norm. If the change in norm for any signal is larger than that of the previous iteration, this indicates that the reconstruction accuracy is deteriorating and the previous estimate is returned as the final reconstruction. [3] The complete SGAP algorithm is presented in Algorithm 3.

3.2 Experiments

In the following paragraphs, the approach for the comparison of the methods discussed in chapter 2 and the newly developed SGAP algorithm, are discussed. The data that was used for the experiments are presented first. Next the different metrics for the quantification of the reconstruction accuracy and computational requirements are discussed. Finally, the settings that were used for the various parameters of the methods are described.

3.2.1 Data

All of the results are based on signals from the MIT-BIH Arrhythmia Database [21, 14], an ECG database that is commonly used in research. It contains 2-channel ECG recordings from 48 patients, with a 360 Hz sample rate. The first 29520 samples (82 seconds) of both channels of the first 23 patients in the database were split into segments of 720 samples (2 seconds). In this way, 943 2-channel, 2-second segments were obtained. These segments were compressed according to equation (1.1) and subsequently reconstructed using different combinations of dictionaries and algorithms. Segments of 2 seconds were used to keep the processing times low enough for the slower algorithms.

The compression Compression Ratio (CR) is expressed as the ratio between the number of samples in the compressed signal n and the number of samples in the original signal N.

$$CR = \frac{n}{N} \tag{3.3}$$

Eight different compression ratios, ranging from 0.9 to 0.2 in steps of 0.1, are used in the experiments to study the influence of the amount of compression on the reconstruction quality.

3.2.2 Reconstruction Accuracy Metrics

Four metrics are calculated to compare the quality of the reconstructions at different compression ratios. Three of these (Peak Root-mean-square Difference, Normalised Mean Absolute Error and Normalised Mean Square Error) are existing measures that are commonly used. The fourth measure (Structural Similarity) was originally developed to compare 2D-images based on three characteristics, but can just as well be used to compare signals.

Percentage Root-mean-square Difference (PRD) The first and most important measure is the Percentage Root-mean-square Difference (PRD). It calculates the 2-norm of the difference between the original signal $\mathbf{x}$ and reconstruction $\hat{\mathbf{x}}$. [19]

$$PRD = \frac{||\mathbf{x} - \hat{\mathbf{x}}||_2}{||\mathbf{x}||_2} \tag{3.4}$$

The PRD is an important measure because researches have established threshold values of the metric that indicate the clinical usefulness of a reconstructed signal.

Values of the PRD below 2% have been reported as 'very good', while values between 2% and 9% are still 'good'. All values above this 9% were found to be unacceptable for clinical diagnosis. [36, 19, 27]

Normalised Mean Absolute Error (NMAE) The Normalised Mean Absolute Error (NMAE) is used to give a more straightforward idea of what the actual error on the signal is. [13]

$$NMAE = \frac{\frac{1}{N}\sum_{i}|x_i - \hat{x}_i|}{x_{max} - x_{min}} \tag{3.5}$$

Normalised Mean Square Error (NMSE) The Normalised Mean Square Error (NMSE) is similar to the NMAE, but is a better indicator for the presence of large errors due to its quadratic nature. [18]

$$NMSE = \frac{\frac{1}{N}||x - \hat{x}||_2^2}{\frac{1}{N}||x||_2^2} \tag{3.6}$$

Structural Similarity (SSIM) For this thesis, the Structural Similarity (SSIM) that was originally developed in [32] to compare two 2D-images, is used to compare two signals. The SSIM is a linear combination of the results of three separate comparisons. The constants C_1, C_2 and C_3 in the following formulas are very small values to avoid division by zero. For the exact values that were used for these constants, please refer to [32].

First, the luminance of both images is compared. The luminance simply corresponds to the mean value μ of the image, so the mean of the signals can be used instead.

$$l = \frac{2\mu_x\mu_{\hat{\mathbf{x}}} + C_1}{\mu_x^2 + \mu_{\hat{\mathbf{x}}}^2 + C1} \tag{3.7}$$

The second equation compares the contrast in two images. For signals this can be seen as a comparison between the internal variation (characterised by the standard deviation σ) in both signals.

$$c = \frac{2\sigma_x\sigma_{\hat{\mathbf{x}}} + C_2}{\sigma_x^2 + \sigma_{\hat{\mathbf{x}}}^2 + C_2} \tag{3.8}$$

The third measure compares the structure similarity between both images, i.e. the correlation after removal of bias (the mean values) and after normalisation for variance.

$$s = \frac{\sigma_{x\hat{\mathbf{x}}} + C_3}{\sigma_x\sigma_{\hat{\mathbf{x}}} + C_3} \tag{3.9}$$

where $\sigma_{x\hat{x}}$ is the correlation coefficient between $\mathbf{x}$ and $\hat{\mathbf{x}}$, calculated as

$$\sigma_{x\hat{x}} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(\hat{x}_i - \mu_{\hat{x}}) \tag{3.10}$$

The three comparisons are finally combined into a single metric to obtain the SSIM, which will have a value between 0 and 1 where 0 indicates a complete lack of similarity and 1 indicates perfect similarity. For simplicity and to assure equal contributions of l, c and s to the final SSIM, the weighting factors were set to $\alpha = \beta = \gamma = 1$.

$$SSIM = l^{\alpha} c^{\beta} s^{\gamma} \tag{3.11}$$

3.2.3 Computational Requirements

To quantify the computational requirements of the algorithms, the required computation time and number of iterations until convergence were measured. In order to compare single-channel algorithms to their multi-channel counterparts, the single-channel algorithms were used to reconstruct each channel separately. The reconstruction times for both channels were then added to obtain an estimate of the total required reconstruction time. The same thing was done for the number of required iterations.

All of the experiments were performed in MATLAB® 2013a, on a quad core 3.10GHz Intel® Core$^{\text{TM}}$i5-3450 system with 8GB of memory, running the CentOS 6.4 operating system with Linux kernel version 2.6.32.

3.2.4 Parameter Settings

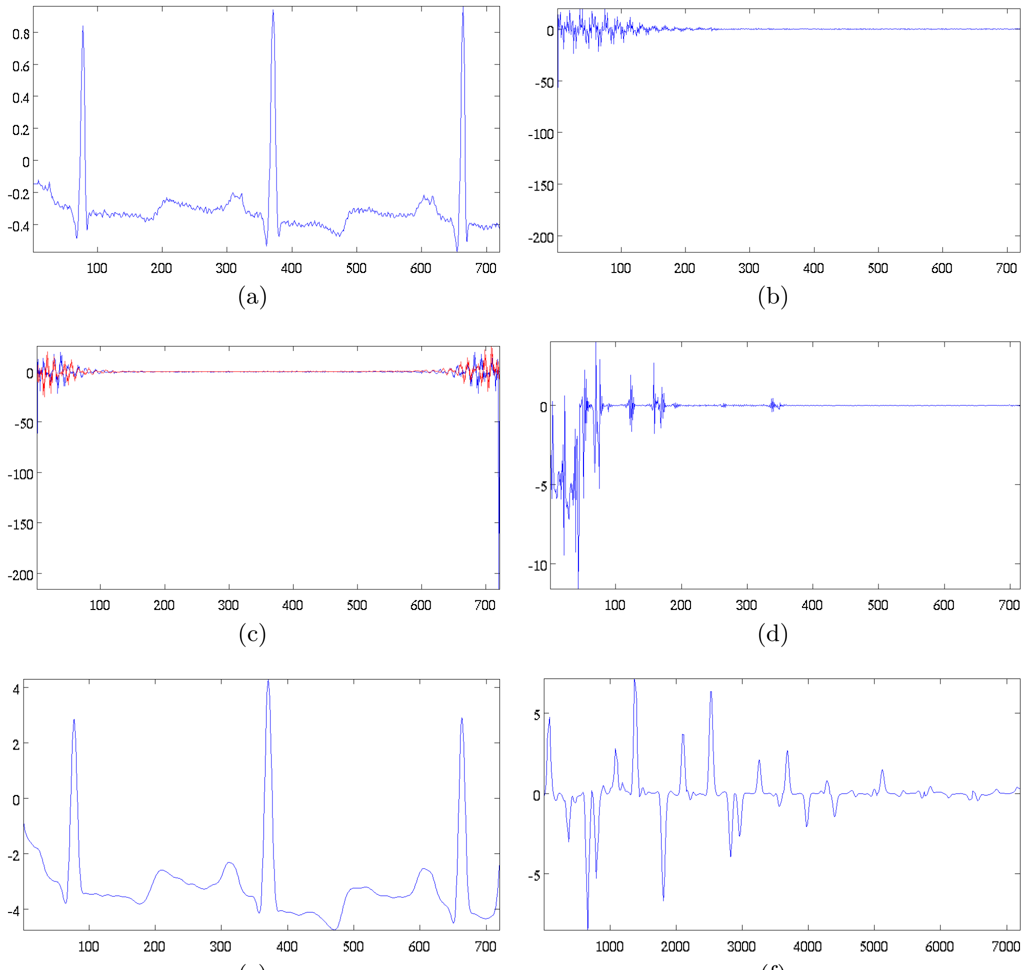
A 50% sparse Gaussian sensing matrix is used in all of the following experiments. There is hardly any difference in reconstruction accuracy when using a complete matrix (i.e. no 0 values) and this would only increase the computational complexity. In section 3.3.5, the effect of using a binary sensing matrix on the reconstruction accuracy and processing time is discussed.

Both the DWT matrix for sparse reconstruction and the wavelet analysis matrix are based on a level 4 wavelet decomposition with Daubechies-4 wavelets. That this is the optimal wavelet type and decomposition level for the dataset in these experiments, has been determined empirically through smaller scale experiments (i.e. using a selection of the signal segments discussed in section 3.2.1).

The number of elements t to be added to the support estimate or removed from the co-support estimate in OMMP and GAP, respectively, was also determined empirically. In OMMP, optimal results are obtained when adding $t = 4$ elements to the support in each iteration. GAP requires the removal of $t = 10$ elements from the co-support per iteration for optimal reconstruction. The maximum number of iterations in SGAP and GAP is set to $K_{max} = (p - t)/t$ in order to avoid removing t elements when less than t are left in the co-support.

A disadvantage of IHT and CoSaMP is the difficulty in deciding upon the required sparsity of the sought coefficient vector, since the sparsity will differ for each signals and dictionary. It is therefore necessary to determine an average sparsity for which the support estimates of some signals will contain too much elements, while others will contain too few. For the dataset in this thesis it was found empirically that setting the required number of nonzero elements in the coefficient vector to 75 returned the best average PRD values possible, while numbers higher and lower than 75 lead to deterioration of the mean PRD, especially at lower CR. Results for CoSaMP were excluded, since alternatives (OMP & OMMP) are available that do not require the sparsity to be known beforehand.

3.3 Results



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

Figure 3.1: Sparsity of an example signal (a) in each dictionary: (b) DCT (c) DFT (d) DWT (e) B-spline and (f) Gabor

The following sections discuss the results of the experiments described above. To avoid displaying an abundance of information, the results in each section are presented with the previous results in mind. This means that in the first section, the dictionaries and analysis matrices are discussed and a single dictionary and analysis matrix is decided upon as the best one for sparse and cosparse methods, respectively.

The remaining dictionaries and analysis matrices are no longer considered in the subsequent sections. In the following section, some of the algorithms that are either too slow or too inaccurate are discussed and excluded from further results. The next two sections contain a comparison of single- and multi-channel algorithms, including the (dis)advantages of multi-channel algorithms, compared to their corresponding single-channel versions. In the final section, the (dis)advantages of using a binary sensing matrix instead of a Gaussian one are discussed.

Most of the results are presented as boxplots containing 5 values. These are, in increasing order of magnitude: lower whisker (LOW), 25th percentile ($P25$), median (MED), 75th percentile ($P75$) and higher whisker ($HIGH$), where $LOW = P25 - w.(P75 - P25)$ and $HIGH = P75 + w.(P75 - P25)$, with $w = 1.5$. The boxplots show outliers (values outside the $[LOW, HIGH]$ range) as crosses. Exact values for all algorithms are included in appendix A.

3.3.1 Dictionaries & Analysis Matrices

The dictionaries that are not useful for ECG CS are first excluded. Next, the remaining useful dictionaries and analysis matrices are compared in terms of reconstruction quality to settle upon one dictionary and analysis matrix for the rest of the experiments.

Useless Dictionaries for ECG

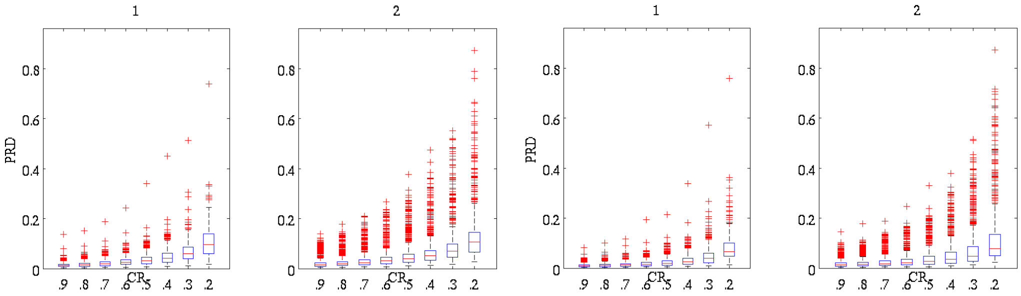
In order to exclude some of the proposed dictionaries from further experiments, the sparsity of ECG signals in each of the dictionaries is first investigated. Figure 3.1 shows an example of a 2-second ECG signal and its corresponding coefficient vector in each of the dictionaries. These coefficient vector representations were investigated for several signals, with similar results.

The DCT, DFT and DWT dictionaries prove to be very useful in creating a sparse representation of an ECG signal. All three have only a few nonzero values. Note how the DFT has complex coefficients with a real part (blue in figure 3.1) and an imaginary part (red in figure 3.1).

The B-spline dictionary seems to be hardly useful in creating a sparse representation of and ECG, since the coefficient vector is only a smoothed version of the original signal that contains barely any zeros (unless the original signal would contain many zeros to begin with). The resulting representation depends strongly on the parameters used to create the B-spline dictionary. The smoothness of the resulting representation varies with different parameter settings but it never becomes sufficiently sparse. This also leads to the algorithms converging to solutions that are often very poor in accuracy or not converging at all.

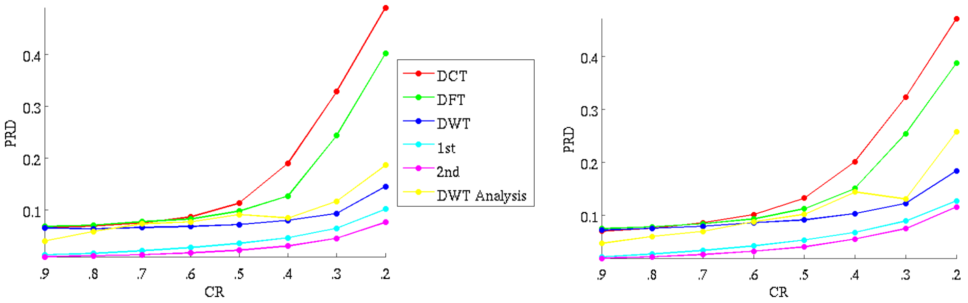
The Gabor dictionary presents another problem. It is a highly redundant dictionary (i.e. much more elements in the dictionary than there are samples in the signal), since it contains elements that account for variations in the different parameters in (2.11) (different phases and frequencies of the cosine and different widths of the Gaussian window). On top of this redundant nature, no settings for the parameters would generate really sparse coefficient vectors. This can be seen in the example in figure 3.1f that contains many smooth peaks (corresponding to series of nonzero coefficients) instead a number of spikes. The combination of the redundancy and the presence of many nonzero coefficients makes methods using the Gabor dictionary converge slowly, if they will even converge to a decent solution at all.

Best Dictionary and Analysis Matrix



(a) (b)

Figure 3.2: PRD values for the (a) 1st and (b) 2nd order derivative using the GAP algorithm. Numbers above the plots indicate the corresponding channel number.

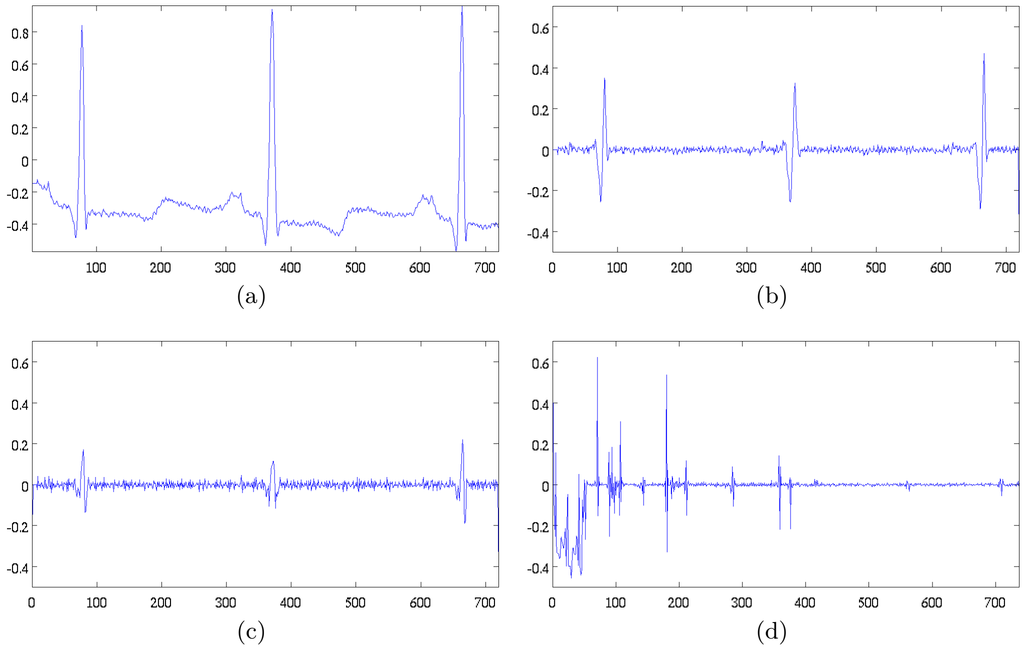


(a) (b)

Figure 3.3: Mean PRD values of the DCT, DFT and DWT dictionaries and 1st and 2nd order derivative and wavelet matrices for (a) channel 1 and (b) channel 2.

There are now three dictionaries left that can be used for ECG CS. In figure 3.3 the mean PRD values resulting from the reconstruction of CS measurements using these dictionaries and the most basic greedy algorithm (OMP) are presented.

Comparing the mean values of the accuracy metrics, the DFT dictionary proves to be slightly better than the DCT dictionary, although both are still worse than the DWT dictionary. The DWT dictionary also results in much less variation in the results, though it could be argued that it has more outliers with higher PRD and NMAE values at higher CR values (0.9 through 0.6). Since the interest of this thesis is in obtaining as much compression as possible (i.e. 0.5 and below) while still retaining good reconstruction accuracy, the use of the DWT dictionary is preferable to the others.



(a) (b) (c) (d)

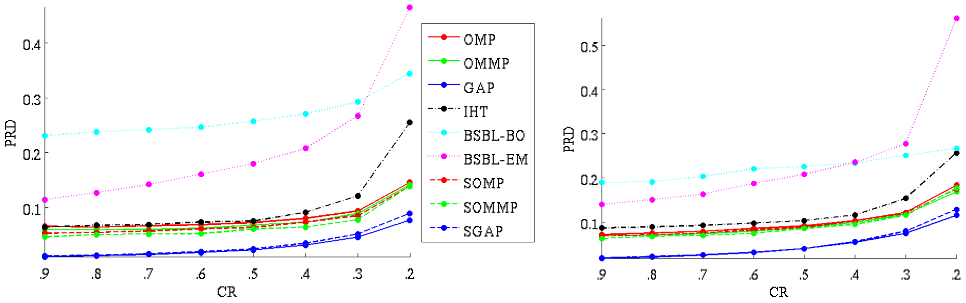
Figure 3.4: Examples of sparse sequences for a signal using different analysis matrices: (a) original Signal (b) 1st order derivative (c) 2nd order derivative and (d) wavelet

For the cosparse case, the most basic greedy algorithm (GAP) was used to compare the analysis matrices. Figure 3.3 presents the mean PRD values resulting from these analysis matrices. The wavelet analysis matrix can immediately be excluded, since its PRD is significantly higher than that of the other 2 analysis matrices. This can be explained by looking at the sparse sequences of an example signal in figure 3.4. Here it can be seen that the first and second order derivative analysis matrix actually result in a more sparse sequence than the wavelet analysis matrix. The reconstruction accuracy obtained using the wavelet analysis matrix in the cosparse approach and using the DWT dictionary in the sparse approach is remarkably similar.

Deciding between the first and second order derivative matrix is much more difficult, but since the mean PRD values corresponding to the second order derivative matrix are slightly lower (figure 3.3) and its outliers generally have lower values (figure 3.2) , this analysis matrix was used in the rest of the experiments.

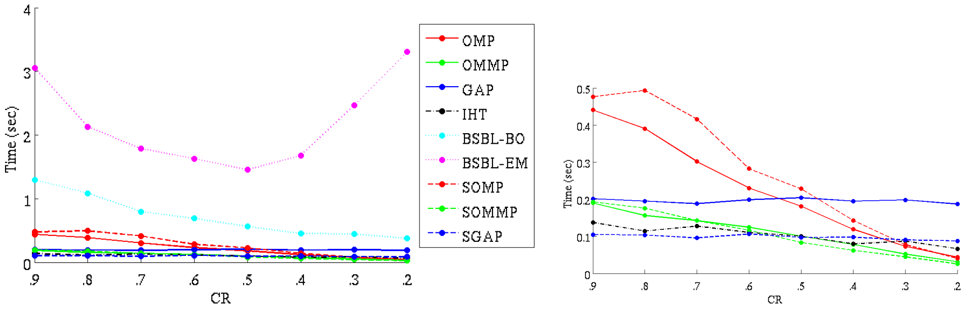
In conclusion, it can be seen that the DWT matrix is the best dictionary for sparse methods, while the second order derivative matrix is the best analysis matrix for cosparse methods. These two matrices will be used in further experiments for the comparison of the algorithms themselves.

3.3.2 Slow and Inaccurate Algorithms



(a) (b)

Figure 3.5: Mean PRD values of the different algorithms using the best dictionary and analysis matrix, at different CR values for (a) channel 1 and (b) channel 2.



(a) (b)

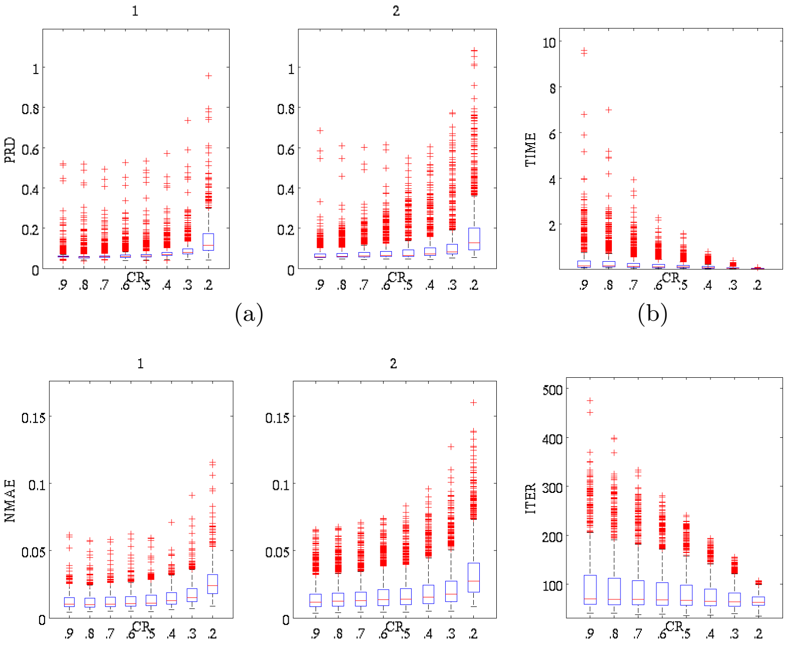
Figure 3.6: Mean processing times of the OMP, OMMP, SOMP, SOMMP, GAP and SGAP algorithms using the best dictionary/analysis matrix, at different CR values. Single-channel algorithm times are the sum of the processing times for each channel. (b) is a detail of (a).

The BP algorithm is the most problematic of all algorithms. Even for a signal of half the length (360 samples) and a very low maximum number of iterations (e.g. 20), it takes more than ten seconds before it finishes. Needless to say that when convergence is desired, much more iterations are needed and the restoration of a single signal would require possibly hundreds of iterations. Since the goal of this thesis is not only to have an accurate reconstruction, but a reasonably fast one as well. the BP algorithm was abandoned.

Since it was found in earlier experiments that the BSBL algorithms perform much slower and less accurate than the other algorithms, the experiments using these algorithms were performed on a smaller dataset in order to save time. Only the mean PRD values and processing times of these algorithms are included in figures 3.5 and 3.6. It can be seen that their processing times (especially those of BSBL-EM) are well above those of the other algorithms. On top of that, the PRD values of both algorithms are much higher than those of the other algorithms as well. This could be explained by the fact that the algorithm divides the coefficient vector in fixed blocks (it would be impossible to manually indicate the borders of each block in each signal, even though the algorithm allows the user to do this). These blocks may not correspond to the actual borders of the blocks. BSBL was further abandoned in order to focus the research on the other faster and more accurate algorithms.

The mean PRD and processing time of the IHT algorithm at each CR value are also included in figures 3.5 and 3.6 for reference. Its performance is not discussed in further detail, as it can be seen that it is inferior in reconstruction accuracy to the greedy sparse and cosparse algorithms. The reason behind this is that the sparsity could not be determined for each of the signals seperately. It could be argued that its speed is one of its main advantages, but in figure 3.6 it can be seen that the total IHT processing time for both channels is similar to that of the SGAP algorithm and is even higher than that of OMMP/SOMMP at lower CR.

3.3.3 Algorithm Comparison: Single-Channel



(a)

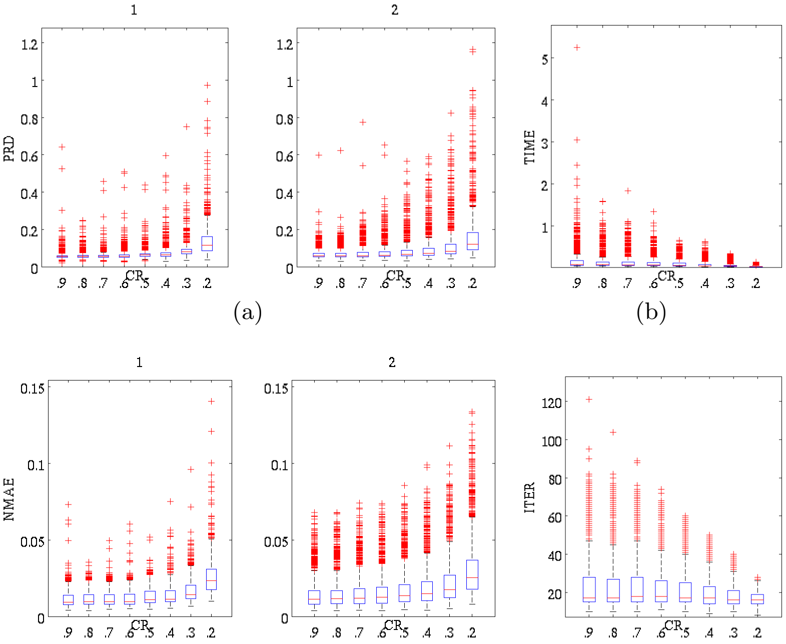
(b)

(c)

(d)

Figure 3.7: Accuracy and computational requirements of the OMP algorithm using a Wavelet Dictionary: (a) PRD (c) NMAE (b) processing time and (d) Iterations. Numbers above the plots indicate the corresponding channel number.

The only algorithms left that are both fast and accurate enough are the different greedy algorithms. For single-channel reconstruction, these are the sparse algorithms OMP and OMMP and the cosparse algorithm GAP. Two accuracy measures (PRD & NMAE), the processing times and the number of iterations for each algorithm



(a) (b)

(c) (d)

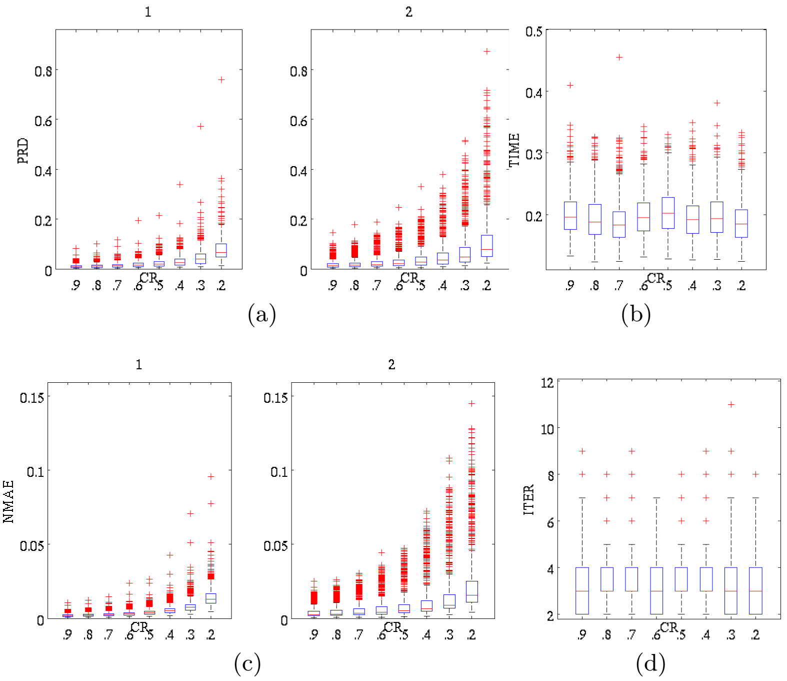
Figure 3.8: Accuracy and computational requirements of the OMMP algorithm using a Wavelet Dictionary: (a) PRD (c) NMAE (b) Time and (d) Iterations

are presented in figure 3.7 (OMP), figure 3.8 (OMMP) and figure 3.9 (GAP). The mean processing times and mean PRD values are included in figures 3.6 and 3.5, respectively.

The main advantage of using OMMP instead of OMP is immediately clear. The processing time is significantly shorter, because of the reduced amount of iterations required for convergence. Most importantly, there is hardly any loss in accuracy as a consequence of speeding up the algorithm. Note that only 4 elements were selected in each OMMP iteration and the algorithm may therefore be sped up even more by selecting a larger amount of elements per iteration, though this will be at the cost of having a worse reconstruction accuracy.

The quality measures of the GAP algorithm are all significantly better (lower PRD, NMAE and NMSE; higher SSIM) than those of the OMP and OMMP algorithm. This is the case for all CR values, with a reduction in accuracy at lower CR values similar to that of OMP and OMMP. This reduction is natural for all algorithms, since the same amount of information is stored in less samples at lower CR. It should also be noted that OMP and OMMP result in more outliers with bad accuracy compared to GAP.

In terms of mean processing time, the GAP algorithm is more consistent than the OMP and OMMP algorithms over the whole range of CR values. This makes that it is faster at high CR, but starting at CR=0.5, the sparse greedy algorithms will outperform GAP in terms of mean processing time. There is, however, much



(a) (b) (c) (d)

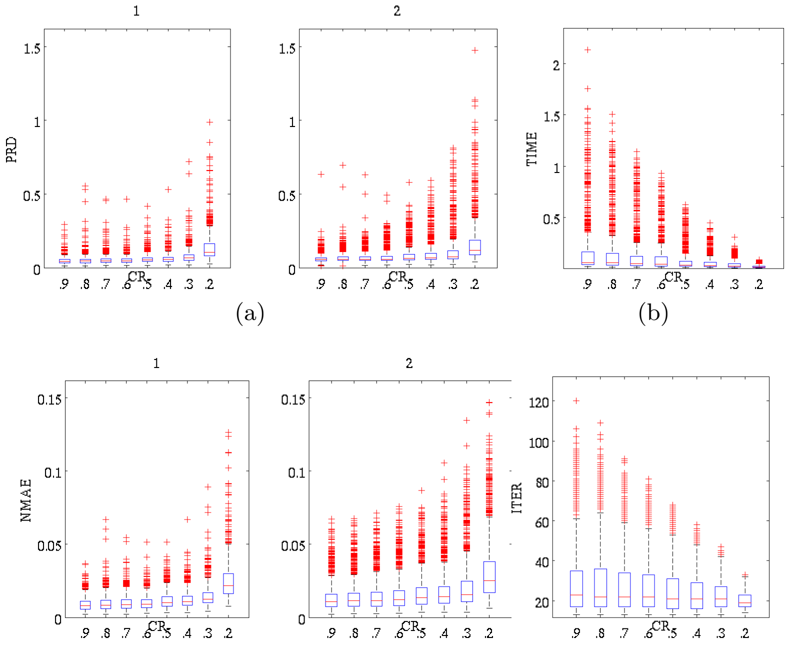
Figure 3.9: Accuracy and computational requirements of the GAP algorithm using a 2nd order derivative analysis matrix results: (a) PRD (c) NMAE (b) Time and (d) Iterations

variation in the processing times of the sparse algorithms that is not at all present in the GAP algorithm. Many outliers with values up to 3 seconds were noted in OMP, with extreme outliers up to 10 seconds; while in OMMP there were many outliers with values up to 1.5 seconds and some extreme cases with processing times up to about 5 seconds. In GAP, the highest processing time that was found was smaller than 0.5 seconds.

This consistency can be explained by looking at the number of iterations of each algorithm. The number of iterations of SOMMP decreases with increasing CR because its stopping criterion depends on the residual norm, which reduces with each iteration. The residual is based on the original measurement, and will initially be smaller and reduce to a sufficiently small value faster at lower CR. In SGAP, the number of iterations remains similar for each CR value, because its stopping criterion is based on the amount of change between the previous and current reconstruction, which will not be different for lower CR.

Each iteration of GAP takes up more time than those of OMMP. This is due to the fact that OMMP solves a smaller inversion problem in each iteration than GAP, since it slowly builds a support starting from an empty one, while GAP removes elements from an initially full co-support and therefore has to solve large systems in the first iterations of the algorithm. On the other hand, GAP requires much less iterations to converge, because it removes more elements from the co-support estimate than OMMP adds to the support estimate. GAP can do this, because the co-support estimate is only a constraint on the inversion problem of which the least-squares solution is sought. If some co-support elements are thought to be a part of the support, the constraint is less rigid, but nonetheless a decent solution can be found.

3.3.4 Algorithm Comparison: Multi-Channel



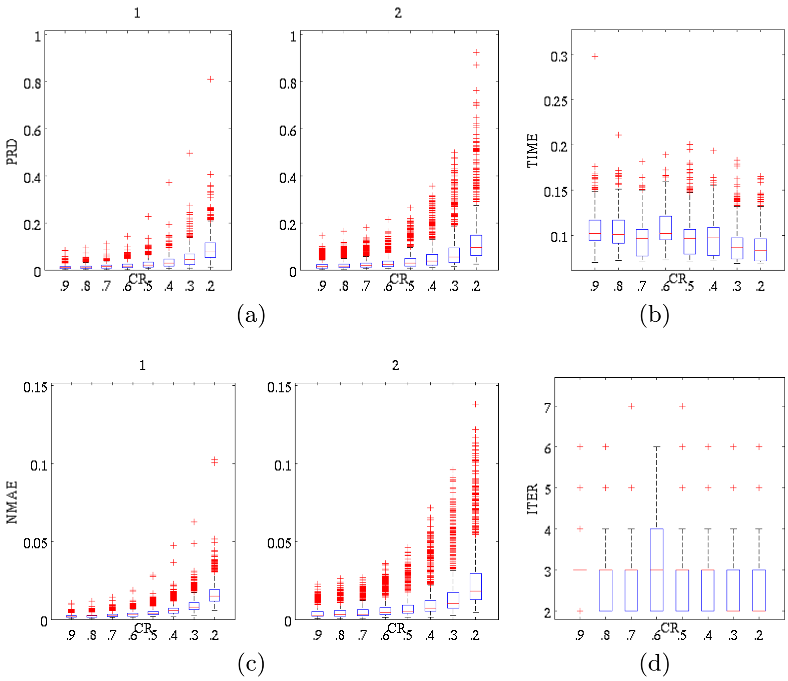
(a) (b)

(c) (d)

Figure 3.10: Accuracy and computational requirements of the SOMMP algorithm using a Wavelet Dictionary: (a) PRD (c) NMAE (b) Time and (d) Iterations

To compare the multi-channel greedy algorithms to their single-channel equivalents, the results of SOMMP (figure 3.10) and SGAP (figure 3.11) are presented here. An individual figure for SOMP is omitted due to its similarity to SOMMP in terms of accuracy. The mean processing times and mean PRD values of all multi-channel algorithms (including SOMP) are also included in figures 3.6 and 3.5, respectively.

In case of sparse greedy algorithms, there is no real reduction in mean processing time (simultaneous reconstruction by SOMP is even slightly slower than separate reconstruction by OMP, while OMMP and SOMMP processing times are nearly equal), but there are less outliers with large processing times present. On the other hand, it is indicated by all four similarity metrics that the reconstruction accuracy of multi-channel algorithms is similar to that of the corresponding single-channel algorithms, though it should be noted that some of the outliers in SOM(M)P have worse accuracy values than the worst outliers in OM(M)P.



(a)

(b)

(c)

(d)

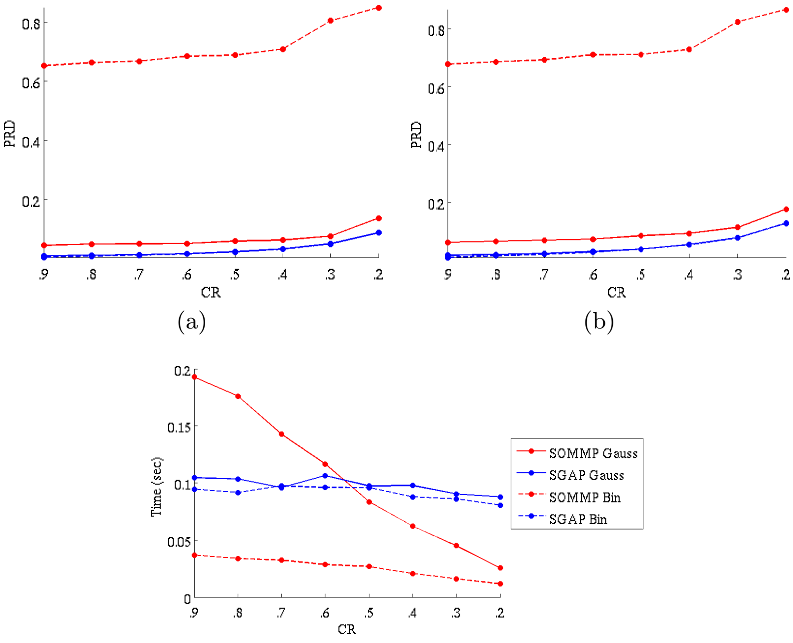
Figure 3.11: Accuracy and computational requirements of the SGAP algorithm using a 2nd order derivative analysis matrix results: (a) PRD (c) NMAE (b) Time and (d) Iterations

In the cosparse greedy algorithms, there is a vast improvement in processing time when using SGAP instead of GAP. The total time is nearly halved, compared to separate reconstruction by GAP. Like the GAP algorithm, the processing time of SGAP hardly changes with the CR, for similar reasons. In terms of reconstruction accuracy, there is, like in the sparse approach, hardly any difference between GAP and SGAP. This is very interesting in that one could easily use a faster multi-channel algorithm, without barely any loss in reconstruction accuracy.

3.3.5 Binary Sensing Matrix

The results of using a binary sensing matrix instead of a Gaussian one in SOMMP and SGAP are discussed here, but the conclusions similarly apply to their single-channel versions OM(M)P and GAP.

It is remarkable how different the effect of using a different sensing matrix on sparse and cosparse methods is. In SOMMP, the reconstruction accuracy degrades dramatically while the processing time also decreases. In SGAP, on the other hand, there is hardly any change at all (the lines for GAP and SGAP in the graphs overlap), neither in terms of reconstruction accuracy nor in terms of processing speed. This indicates that SGAP could easily be implemented in hardware using this much simpler sensing matrix.



(a)

(b)

(c)

Figure 3.12: Mean PRD values of the reconstruction of (a) channel 1 and (b) channel 2, when the measurement was obtained using a binary or a Gaussian sensing matrix; (c) corresponding processing times.

The reason for this large difference in reconstruction accuracy is the difference in how both algorithms handle the estimation of the (co-)support elements. (S)OM(M)P searches for the largest value(s) in $(\boldsymbol{\phi\psi})^T\mathbf{r}_{k-1}$, which depends strongly on the size of the values in $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$. When a Gaussian sensing matrix is used, its 2-norm is similar in size to that of a dictionary (provided that the columns of the dictionary have been normalised). On the other hand, when a binary sensing matrix is used to obtain the measurement, the 2-norm of the sensing matrix will be much larger than that of the dictionary. This results in a severe deformation of the vector $(\boldsymbol{\phi\psi})^T\mathbf{r}_{k-1}$, making it difficult to select the correct indices to add to the support estimate. In (S)GAP, the values of the projection $\boldsymbol{\alpha}_k = \Omega\hat{\mathbf{x}}_{k-1}$ are completely independent of the sensing matrix, which is why the binary matrix can be used as a substitute for the Gaussian matrix, despite its larger values.

3.4 Discussion

A generalisation of the GAP algorithm for simultaneous signal reconstruction (SGAP) was proposed. Like in the sparse methods for simultaneous reconstruction, two adjustments are made to the cosparse algorithm to allow it to reconstruct mulitple signals at once. The first one being the way it looks for elements that are part of the support and the second one being the adjusted stopping criteria.

The wavelet dictionary was found to be superior to the other four tested dictionaries for sparse reconstruction methods for ECG signals. In the cosparse methods, the second order derivative matrix was found to be the best candidate for an analysis matrix, although the reconstruction accuracy of the first order derivative matrix was nearly equal to that of the second order derivative matrix.

Some algorithms were immediately excluded based on overlong processing times (BP), poor quality in unsupervised circumstances (BSBL) and poor reconstruction quality (IHT). This leaves only the greedy algorithms that satisfy both desired requirements: fast processing times and good reconstruction quality.

Concerning cosparse algorithms, it can be concluded that it is definitely beneficial for the reconstruction accuracy to consider using such algorithms over traditional sparse algorithms. Despite their better accuracy, they do not require much more processing time than sparse greedy algorithms (even less at higher CR values). The processing times of (S)GAP were found to be nearly the same at all CR values.

It was also found that the use of multi-channel algorithms for greedy sparse algorithms lead to slower (SOMP) or equal (SOMMP) processing times while providing similar reconstruction accuracy. In greedy cosparse algorithms, simultaneous processing leads to faster processing times and similar reconstruction accuracy, which makes SGAP and ideal alternative for separate processing of all signals using GAP.

(S)GAP also seems more convenient for implementation in hardware than the sparse greedy algorithms, as there is no difference in performance, neither in terms of reconstruction accuracy, nor in terms of processing time, when a binary sensing matrix is used instead of a Gaussian one. This is because the sensing matrix and analysis matrix are used independently in (S)GAP, whereas (S)O(M)MP uses the multiplication of the two to estimate the coefficient vector support, returning a completely different result for both types of sensing matrices.

Chapter 4

Signal Reconstruction from Measurements with Multiplicative Noise

In the previous chapters, it was assumed that measurements are free of any noise. Although this may be true for simulated measurements, this is not always the case for real measurements in hardware. Additive noise may be placed on top of the measurement, thereby altering the values stored inside. Multiplical noise is more complex, it alters the values of the implemented sensing matrix, so that it will not contain the exact values that are expected at the receiver side. Both types of noise will thus alter the measurement values and make it difficult for the receiver to reconstruct the original signal, because to him it will seem as though these values are the result of measuring a different signal with the known sensing matrix.

In this chapter, possible adjustments for the most basic greedy algorithms of the sparse (OMP) and cosparse (GAP) approaches are suggested, in order to make them more robust to the influence of both additive and multiplicative noise. Naturally, these approaches can be implemented in more complex variations on these greedy algorithms as well.

4.1 Noise types

A distinction is made between additive and multiplicative noise, depending on whether the noise is added to the measurement or whether the sensing matrix $\boldsymbol{\phi}$ itself is noisy. Many types of noise exist, depending on the setting where the measurements happen. For this thesis, it is assumed that the additive and multiplicative noise are both random, with a certain maximum value.

4.1.1 Additive and Multiplicative Noise

Additive Noise Additive noise causes - as its name suggests - perturbations in the measurements by being added to the measurement samples. The model for a measurement with additive noise is defined as

$$\mathbf{y}_A = \boldsymbol{\phi}\mathbf{x} + \boldsymbol{\eta}_A \tag{4.1}$$

where $\boldsymbol{\eta}_A$ is a vector containing the values of the additive noise.

Several noise aware sparse reconstruction algorithms aimed at additive noise have already been developed for CS. In [7], it was suggested that the 2-norm of the noise can be used as a limit for the residual in each iteration, i.e. if $||\mathbf{r}_k|| \leq ||\boldsymbol{\eta}_A||$ then the best possible solution has been found and the algorithm should stop. This follows from the fact that at this point, OMP would start reconstructing the noise if it is not stopped. It has been found that IHT is also robust to additive noise [6] and in [16] a method based on Bayesian learning that accounts for additive noise was developed. Due to relatively recent development of cosparse methods, there has been little research on cosparse reconstruction from measurements with additive noise, though in [23] a noise-aware version of the GAP algorithm was suggested.

Multiplicative Noise It is essential for a correct reconstruction of the signal from the measurement that the receiver knows exactly what sensing matrix was used at the site where the compressive sensing took place. When the random measurement process is implemented in hardware, the measurement process will not use a matrix of exactly quantised values, but rather the electrical equivalent thereof, possibly resulting in small deviations from the expected sensing matrix values. These perturbated values will be multiplied with the signal to obtain the measurement and the receiver will thus receive a measurement that does correspond to the supposedly known sensing matrix. The model for a measurement with multiplicative noise is

$$\mathbf{y}_M = \boldsymbol{\phi}_M\mathbf{x} \tag{4.2}$$

where $\boldsymbol{\phi}_M$ is a noisy sensing matrix defined as

$$\boldsymbol{\phi}_M = \boldsymbol{\phi} + \boldsymbol{\eta}_M \tag{4.3}$$

with $\boldsymbol{\eta}_M$ a matrix of noise values that are added to the sensing matrix.

In a real situation, both types of noise might be present, which results in even further perturbations of the measurement values. Combining equations (4.1) and (4.2) leads to a model of the full noise situation.

$$\mathbf{y}_N = \boldsymbol{\phi}_M\mathbf{x} + \boldsymbol{\eta}_A \tag{4.4}$$

In [15] the concept of multiplicative noise was discussed, but overall there is little research on the topic of multiplicative noise in the literature and no algorithms aimed specifically at this type of noise were found.

4.1.2 Noise Characteristics

Two characteristics are used to define the noise. They are used as prior knowledge for the noise-aware sparse and cosparse algorithms discussed below. The first characteristic is the 2-norm of the total noise that is present in the noisy measurement, characterising the size of the noise.

$$\epsilon = ||\mathbf{y} - \mathbf{y}_N||_2 \tag{4.5}$$

Secondly, the covariance matrix $\mathbf{P}$ is used to characterise the uncertainty caused in the measurement by the multiplicative noise.

$$\mathbf{P} = E[(\mathbf{V} - E[\mathbf{V}])(\mathbf{V} - E[\mathbf{V}])^T] \tag{4.6}$$

In the sparse approach $V = (\boldsymbol{\phi} + \boldsymbol{\eta_M})\boldsymbol{\psi}$. The multiplication with $\boldsymbol{\psi}$ is used to characterise the noisy measurement of $\mathbf{s}$ instead of that of $\mathbf{x}$. In the cosparse approach $V = (\boldsymbol{\phi} + \boldsymbol{\eta_M})$. The multiplication by $\boldsymbol{\psi}$ is not necessary because the algorithm directly reconstructs $\mathbf{x}$ from $\mathbf{y}$.

4.2 Methods

4.2.1 Noise Aware OMP

As it was stated in [7] that the stopping criterion can be used to stop OMP before it starts to reconstruct based on a residual that consists mostly of noise, the stopping criterion that was suggested in that research can be adapted to be applicable to multiplicative noise as well. The new stopping criterion is defined as

$$\mathbf{s}_k^T P \mathbf{s}_k + ||\mathbf{r}_k||_2 \leq \epsilon \tag{4.7}$$

where the second term is the 2-norm of the residual that was already used in [7]. The additional term $\mathbf{s}_k^T P \mathbf{s}_k$ is used to characterise the uncertainty in the measurement, caused by the perturbated sensing matrix.

4.2.2 Noise Aware GAP

Contrary to OMP, the GAP algorithm searches the values of the signal $\mathbf{x}$ that satisfy the constraint that $||\boldsymbol{\Omega}_\Lambda \mathbf{x}||_2^2$ should be zero (or at least as small as possible) instead of iteratively finding more and more elements that belong to the support of the sparse sequence. Because an approximation of the complete signal, satisfying this constraint, is created at each iteration, the 2-norm of the residual $||\mathbf{r}_k||_2^2 = ||\mathbf{y} - \boldsymbol{\phi}\mathbf{x}||_2$ will become very small, even in the earliest iterations. This means that the residual can not be used as part of a stopping criterion.

Two adjustments can be made to make the GAP algorithm more robust to additive and multiplicative measurement noise. First of all, the inverse problem that was used in the original GAP algorithm is altered. This is done by starting from a new optimisation problem that includes constraints based on the noise characteristics that were already used in equation (4.7).

$$min||\boldsymbol{\Omega}_\Lambda \mathbf{x}||_2^2 \; s.t. \; \mathbf{x}^T \mathbf{P} \mathbf{x} + ||\mathbf{y} - \boldsymbol{\phi}\mathbf{x}||_2^2 \leq \epsilon \tag{4.8}$$

```
Algorithm 4: Noise Aware GAP

In: $\mathbf{y}$, $\boldsymbol{\phi}$, $\boldsymbol{\Omega}$
Out: $\hat{\mathbf{x}}$

Initialisation ($k = 0$)
    • Initialise Co-Support: $\hat{\Lambda}_k = \{1, 2, 3, ..., p\}$
    • Precalculations
        – $\boldsymbol{\phi}_\phi = \boldsymbol{\phi}^T\boldsymbol{\phi}$
        – $\boldsymbol{\phi}_Y = \boldsymbol{\phi}^T\mathbf{Y}$
    • Initialise Solution: $\widehat{\mathbf{x}_k} = (2\boldsymbol{\Omega}^T\boldsymbol{\Omega} + \mathbf{P} + \mathbf{P}^T + 2\boldsymbol{\phi}_\phi)^\dagger(\boldsymbol{\phi}^T\mathbf{y} + \mathbf{y}^T\boldsymbol{\phi})$

Iterations $k \to k + 1$
    • Project: $\boldsymbol{\alpha} = \Omega\hat{\mathbf{x}}_{k-1}$
    • Update Co-Support: $\hat{\Lambda}_k = \hat{\Lambda}_{k-1} \backslash \{\underset{i \in \hat{\Lambda}_{k-1}}{argmax} \, |\boldsymbol{\alpha}_i|\}$
    • Update Solution: $\widehat{\mathbf{x}_k} = (2\boldsymbol{\Omega}_{\hat{\Lambda}_k}^T\boldsymbol{\Omega}_{\hat{\Lambda}_k} + \mathbf{P} + \mathbf{P}^T + 2\boldsymbol{\phi}_\phi)^\dagger(\boldsymbol{\phi}^T\mathbf{y} + \mathbf{y}^T\boldsymbol{\phi})$
    • Calculate Sparse Signal: $\hat{\mathbf{x}}_{s,k} = (\boldsymbol{\Omega})^\dagger S_{\Lambda_k}(\boldsymbol{\Omega}\hat{\mathbf{x}}_k)$
    • Calculate Residual: $\mathbf{r}_k = \mathbf{y} - \boldsymbol{\phi}\hat{\mathbf{x}}_{s,k}$

Stopping Criteria
    • $\mathbf{x}_{s,k}^T\mathbf{P}\mathbf{x}_{s,k} + ||\mathbf{r}_k|| \leq \epsilon \implies \mathbf{x}_{s,k}$ is optimal
```

This optimisation problem can be written as the Lagrangian

$$\begin{aligned} \mathcal{L} &= \mathbf{x}^T\boldsymbol{\Omega}_\Lambda^T\boldsymbol{\Omega}_\Lambda\mathbf{x} + \mathbf{x}^T\mathbf{P}\mathbf{x} + (\mathbf{y}^T - \mathbf{x}^T\boldsymbol{\phi}^T)(\mathbf{y} - \boldsymbol{\phi}\mathbf{x}) \\ &= \mathbf{x}^T\boldsymbol{\Omega}_\Lambda^T\boldsymbol{\Omega}_\Lambda\mathbf{x} + \mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\boldsymbol{\phi}\mathbf{x} - \mathbf{x}^T\boldsymbol{\phi}^T\mathbf{y} + \mathbf{x}^T\boldsymbol{\phi}^T\boldsymbol{\phi}\mathbf{x} \end{aligned} \tag{4.9}$$

The goal of the optimisation is to find a signal $\mathbf{x}$ that satisfies the constraints in the optimisation problem defined above. In order to find an equation that can be used to optimise $\mathbf{x}$ the derivative of the Lagrangian with respect to $\mathbf{x}$ is taken. By setting this derivative equal to zero, a linear equation is found.

$$\frac{\partial \mathcal{L}}{\partial x} = 2\mathbf{x}^T\boldsymbol{\Omega}_\Lambda^T\boldsymbol{\Omega}_\Lambda + \mathbf{x}^T(\mathbf{P} + \mathbf{P}^T) - (\boldsymbol{\phi}^T\mathbf{y} + \mathbf{y}^T\boldsymbol{\phi}) + 2\mathbf{x}^T\boldsymbol{\phi}^T\boldsymbol{\phi} = 0 \tag{4.10}$$

The linear problem resulting from the derivative of the Lagrangian can be written in standard form as

$$\mathbf{A}\mathbf{x} = \mathbf{K}, where \begin{cases} \mathbf{A} = 2\boldsymbol{\Omega}_\Lambda^T\boldsymbol{\Omega}_\Lambda + \mathbf{P} + \mathbf{P}^T + 2\boldsymbol{\phi}^T\boldsymbol{\phi} \\ \mathbf{K} = \boldsymbol{\phi}^T\mathbf{y} + \mathbf{y}^T\boldsymbol{\phi} \end{cases} \tag{4.11}$$

which is a linear problem that can be solved just like the original set of equations in (2.24).

The second adjustment that has to be made has to do with the use of $||\mathbf{r}_k||_2^2$ in the stopping criterion. In OMP, the residual is based on a signal approximation using a sparse series of coefficients for a certain dictionary, while in GAP the signal approximation is not based on a series of coefficients at all. To make the residual calculation more similar to that of OMP, a sparse approximation based on the current signal estimate is calculated in each iteration. This is accomplished by calculating the projection $\boldsymbol{\alpha}$, setting the elements in this projection that are in the current co-support estimate to zero and finally recreating a sparse signal estimate based on this altered projection using the pseudo-inverse of $\mathbf{\Omega}$. The whole process can be summarised as

$$\hat{\mathbf{x}}_{s,k} = (\mathbf{\Omega})^{\dagger} S_{\Lambda_k}(\mathbf{\Omega}\hat{\mathbf{x}}_k) \tag{4.12}$$

where S_{Λ_k} is an operator that sets the elements of the vector in its argument that are part of the current co-support estimate Λ_k to zero.

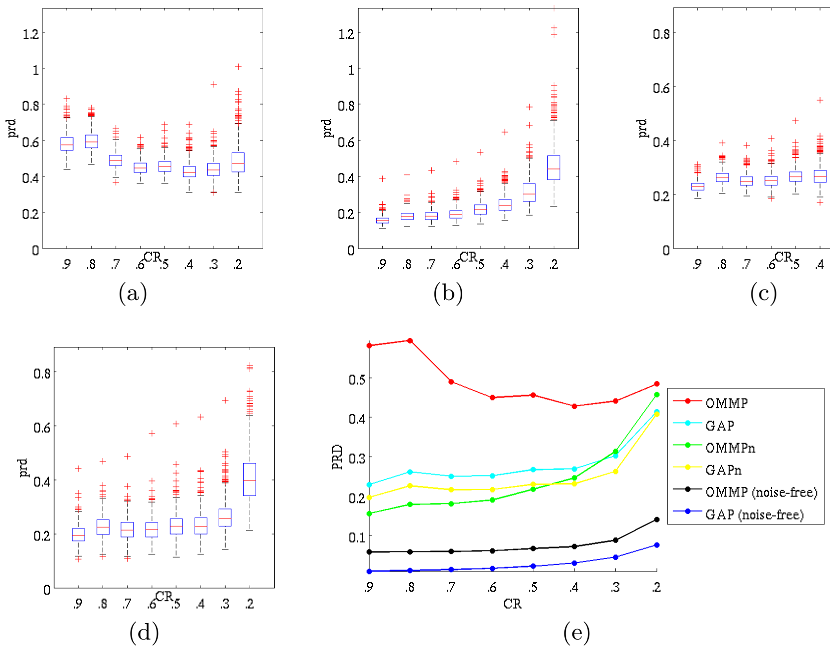
The whole algorithm still uses the complete estimate $\hat{\mathbf{x}}_k$ in the rest of each iteration, like in the original algorithm, but this sparse estimate $\mathbf{x}_{s,k}$ is now used to estimate a residual that is used as a part of the stopping criterion. The complete algorithm is summarised in Algorithm 4.

Out of the three analysis matrices studied in this thesis, the approach is only valid for the wavelet analysis matrix, as it is the only matrix that can be used to create a sparse version of the signal based on a limited number of coefficients by inverting it. This is not possible when using the 1st and 2nd order derivative matrix, since each value in the sparse sequences created using these matrices depend on the value of the following element, essentially making all of the elements depend on all of the following elements. This is a shame, as it was shown in chapter 3 that the wavelet analysis matrix was actually the worst of the analysis matrices.

4.3 Experiments

Two datasets are used in these experiments. The first one consists of the first channels of the 943 2-second 2-channel ECG segments from the MIT-BIH Arrhythmia Database used in chapter 3. In these signals, it might be difficult to distinguish between noise that is left in the signal due to poor performance of the noise-aware reconstruction algorithm or noise that was already present in the signal because of the recording equipment. Therefore, a second dataset was used, consisting of 100 simulated 2-second segments created using the simulator described in section 1.2.2 with the same sample rate $f_s = 360Hz$ as the signals from the MIT-BIH Database and beats per minute (BPM) varying between 50 and 75. Only 100 simulated signals were used since there is less variation between these signals than between the real signals. The same similarity measures as the ones described in section 3.2.2 were used here to compare signal reconstructions.

Two reconstructions are made for the sparse and cosparse methods. First, a measurement containing noise is reconstructed using the original algorithm to see



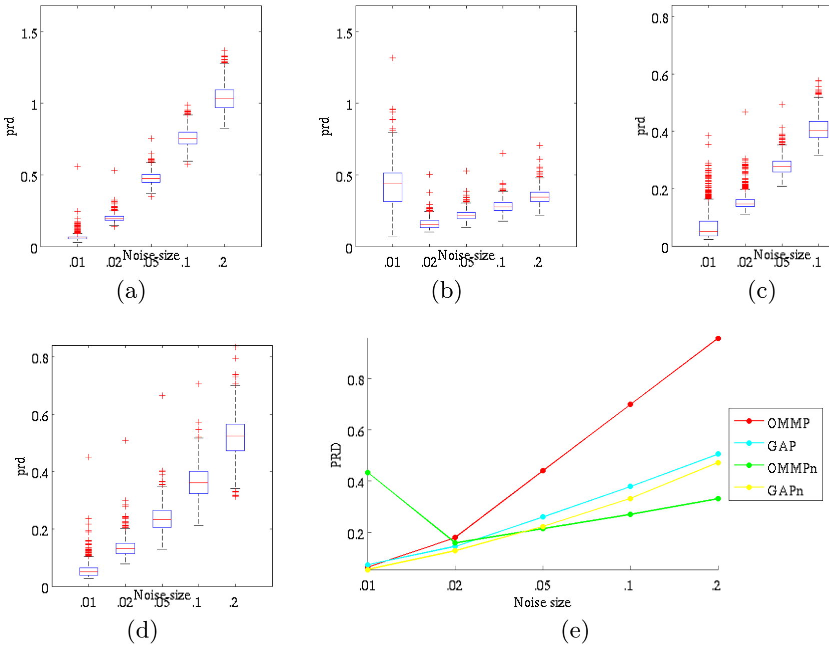
(a) (b) (c) (d) (e)

Figure 4.1: PRD values of noisy signal reconstruction of real signals at CR values with a fixed noise size (maximum size equal to 5% of the signal maximum) using (a) OMMP (b) Noise Aware OMMP (c) GAP and (d) Noise Aware GAP. (e) is an overview of mean PRD values. The horizontal axis shows the random noise maximum as a percentage of the signal maximum.

how large the damage done by the presence of noise is. Next, a reconstruction using the new adapted algorithm is performed to see the improvement in reconstruction accuracy, compared to the reconstruction from a noisy measurement with the original algorithm. To speed the processing up, an OMMP and Noise Aware OMMP algorithm were used instead of an OMP algorithm and its Noise Aware variation, as it was proven in chapter 2 that both are equally accurate.

To find out what the influence of the noise level and CR is on the reconstruction quality, the experiments were performed for a fixed value of CR = 0.5 and a varying noise level (random noise with a maximum size equal to 1%, 2%, 5%, 10% and 20% of the signal maximum, for both $\boldsymbol{\eta}_M$ and $\boldsymbol{\eta}_A$) and also for a fixed random noise with a maximum size equal to 5% of the signal maximum (again for both $\boldsymbol{\eta}_M$ and $\boldsymbol{\eta}_A$) and a varying CR (0.9 to 0.2 in steps of 0.1).

Since the goal of these experiments is solely to confirm the usefulness of the adapted algorithms, the 2-norm of the noise and the covariance was derived directly from the known noise. In real applications, these values would of course have to be derived in some other way.



(a) (b) (c)

(d) (e)

Figure 4.2: PRD values of noisy signal reconstruction of real signals at fixed CR=0.5 using (a) OMMP (b) Noise Aware OMMP (c) GAP and (d) Noise Aware GAP at different noise sizes. (e) is an overview of mean PRD values. The horizontal axis shows the random noise maximum as a percentage of the signal maximum.

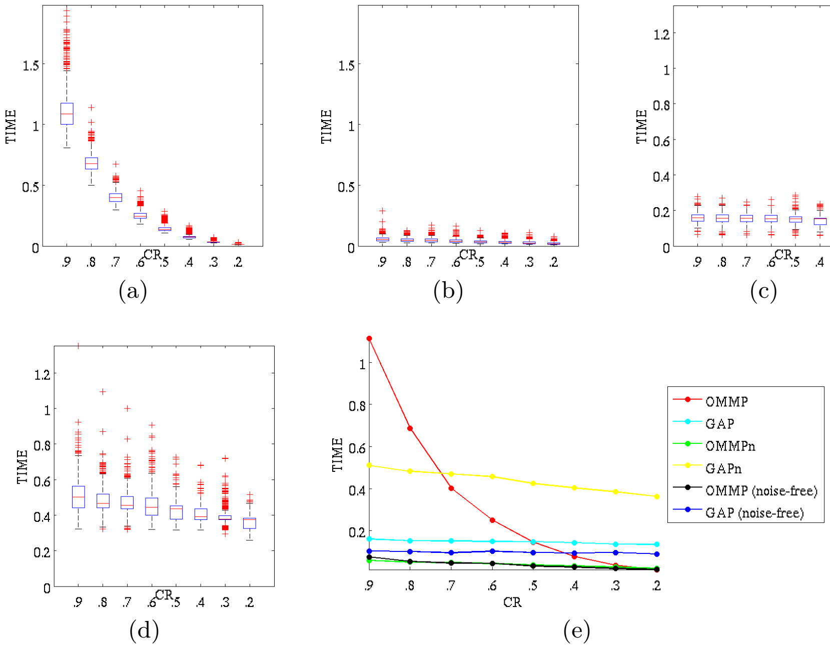
4.4 Results

To save space, the only accuracy metric presented here is the PRD value, since it leads to the same conclusions as the other metrics. For an overview of all metric values from this chapter, please refer to appendix B.

In figure 4.1, the PRD values for the reconstruction of the noisy signal with a fixed noise size at different CR values is shown. It can be seen that the reconstruction accuracy is greatly improved when using Noise Aware OMMP (OMMPn) instead of the regular OMMP algorithm, at least for higher CR values. At lower CR, the advantages of using a noise aware algorithm seem to be cancelled out as the two PRD values of both algorithms start to converge.

For Noise Aware GAP (GAPn), it seems that there is little advantage in using the noise aware version of the algorithm. It should be noted, however, that the GAP algorithm is already much more resistant to noisy measurement than OMMP and this slight improvement makes it nearly as accurate as OMMPn at higher CR and even more accurate at lower CR. Since the results are based on the use of the wavelet analysis matrix, it is not unthinkable that if a better analysis matrix - that can be used to create a sparse signal approximation - is used instead, the results might improve. However, no such matrices were developed in this thesis.

Figure 4.2e also shows the mean PRD values of a noise-free reconstruction of the



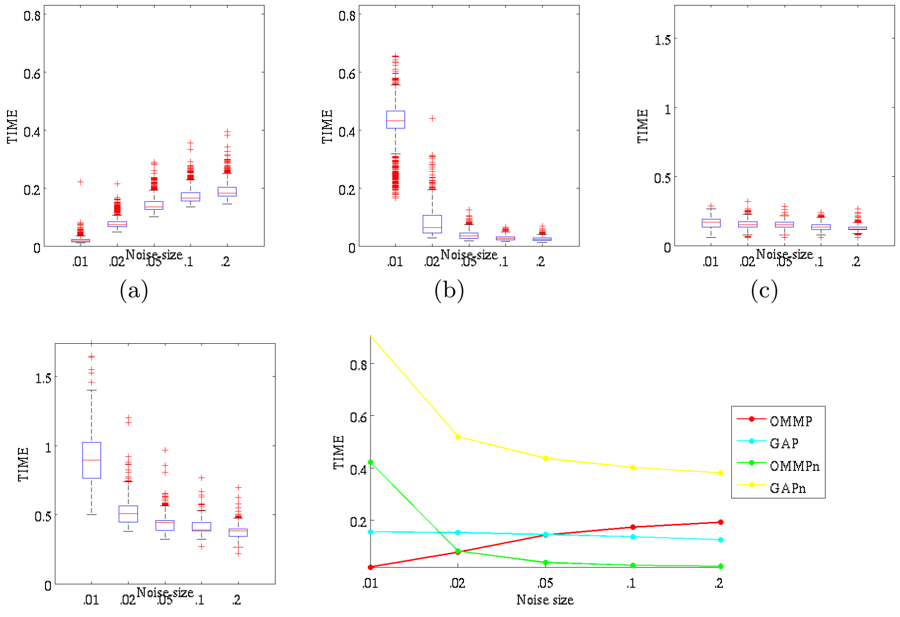
(a) (b) (c) (d) (e)

Figure 4.3: Processing times of noisy signal reconstruction of simulated signals at different noise sizes using (a) OMMP (b) Noise Aware OMMP (c) GAP and (d) Noise Aware GAP. (e) is an overview of mean processing times. The horizontal axis shows the random noise maximum as a percentage of the signal maximum.

same signals (using GAP and a second order derivative analysis matrix). It can be seen that the PRD values obtained using the noise-aware algorithms are still much higher than those of their equivalent noise-free reconstructions and so there is ample room for improvement, both in the sparse and in the cosparse methods.

When the amount of noise that is present in the measurement is changed while the compression is kept constant (as in figure 4.1), it can be seen that the PRD quickly grows larger than the critical 9% value. At low noise levels, GAPn outperforms the other algorithms, though it is quickly surpassed by OMMPn. The latter does shows some oddly large PRD values at the lowest noise level, which may be attributed to the stopping criterion that forces the algorithm to continue iterating until the norm of its residual has reached an extremely low value (lower than the 5% stopping criterion used in the original OMP algorithm and its variations) and thus to force it to start picking elements from the dictionary that may not be part of the actual support. This could easily be fixed by adding an additional check on the size of the stopping criterion in the noise aware algorithm and replacing it with the previously used 5% if it is smaller than that 5%.

Figures 4.3 and 4.4 show the processing times of the regular and noise-aware algorithms. It is remarkable that there is hardly any difference in mean processing time between the reconstruction of a clean signal using OMMP and reconstruction from a noisy signal using OMMPn at any CR. GAPn on the other hand is much slower



(a) (b) (c)

(d) (e)

Figure 4.4: Processing times of noisy signal reconstruction of simulated signals using (a) OMMP (b) Noise Aware OMMP (c) GAP and (d) Noise Aware GAP at different noise sizes. (e) is an overview of mean processing times (Or = original (noise-free) reconstruction). The horizontal axis shows the random noise maximum as a percentage of the signal maximum.

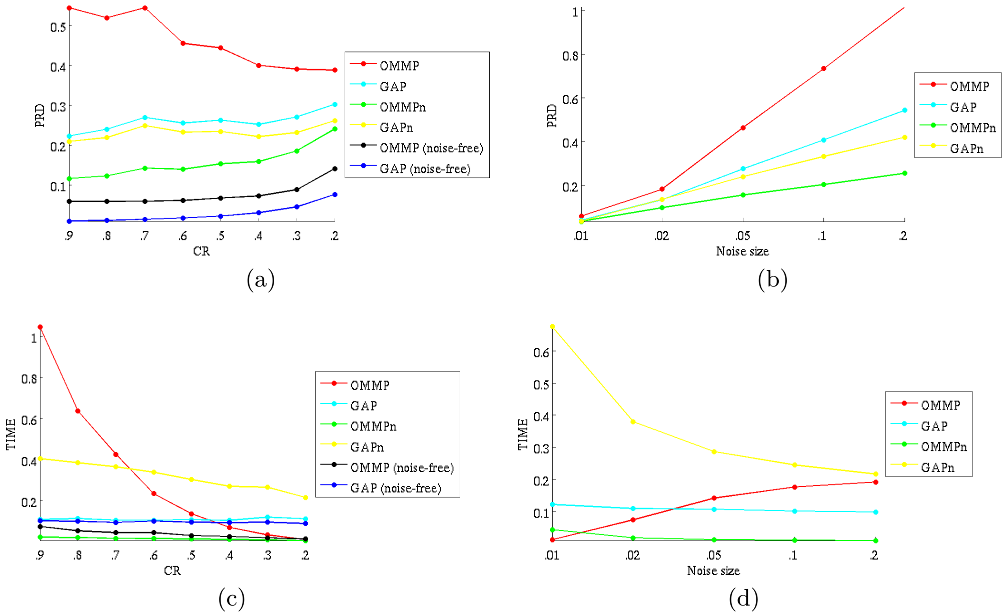
due to the additional computations that are required for the sparse approximation of the signal estimate.

For the simulated signal experiments, the results are shown in figure 4.5 as the mean PRD and processing time values only. Most of the results are quite similar to those obtained from the real signals, although it should be noted that the odd high PRD values at higher CR values in figure 4.2 are not present in figure 4.5b. This could be explained by the fact that the simulated signal is much simpler, making the stopping criterion based on the noise size and covariance not too strict at low noise sizes to reconstruct the whole signal.

OMMPn also works remarkably better, compared to its performance on the real signals. This can be explained by the simple nature of the simulated signal as well, since it can be represented by a very sparse coefficient vector.

4.5 Discussion

From the results of the experiments, it can be seen that the adjusted algorithms improve the reconstruction quality in the presence of additive and multiplicative noise in the measurements. In case of OMP (and OMMP), where the reconstruction



(a)

(b)

(c)

(d)

Figure 4.5: Mean PRD values for the simulated data when varying (a) the CR and (b) the maximum noise size; mean processing time when varying (c) the CR and (d) the maximum noise size.

accuracy is very low when using an algorithm that is not noise-aware, the noise-awareness provides a vast improvement. With GAP, the difference in accuracy is smaller, although the regular GAP algorithm is already quite resistant to the influence of noise.

Another problem here is the fact that the wavelet analysis matrix was used by lack of a better alternative. Developing a better analysis matrix that can also create a sparse approximation of the signal estimate in the GAPn algorithm (which the 1st and 2nd order derivative could not) could drastically improve the results.

Chapter 5

Influence of Signal Reconstruction Accuracy on ECG Feature Detection

The signals that are reconstructed from the compressive sensing measurement of an ECG will be used for clinical diagnosis. In order to draw a correct conclusion from the received signal, certain relevant information will have to be reconstructed accurately. What information is required will depend on the application.

In this chapter, QRS-complex detection is introduced as an example of a simple application of ECG signals. A slightly modified version of the classic Pan-Tompkins algorithm is used to perform this detection. Through numerical experiments, it is verified whether there is a correlation between the detected locations of the Q-,R- and S-peaks and the amount of compression.

This chapter is only a brief introduction to what could be a complete research on the usefulness of CS in several ECG applications. It should therefore be seen more as a starting ground for possible future research than as a complete study of the topic.

5.1 QRS-complex Detection

The detection of R-peaks in the ECG signal is most important for the calculation of the heart rate. The heart rate is a count of the number of ECG cycles in a given time period and can therefore be characterised as the number of R-peaks in that time period.

5.1.1 The Pan-Tompkins algorithm

The Pan-Tompkins algorithm is a classic R-peak detection algorithm. It consists of five steps that eventually lead to the detection of the R-peak locations. In the first and second step, the signal is filtered using a low-pass filter with transfer function

$$H(z) = \frac{(1-z^{-6})^2}{(1-z^{-1})^2} \tag{5.1}$$

and a high pass filter with transfer function

$$H(z) = \frac{-1 + 32z^{-16} + z^{-32}}{1 + z^{-1}} \tag{5.2}$$

These two filter form a bandpass filter with a pass region between 5 and 15 Hz. This removes the effects of the electrical net interference (50 or 60 Hz, depending on the geographical location where the ECG was recorded), muscle noise influence, baseline wander and interference from the T-wave in the ECG signal.

The third step of the algorithm is a differentiation that emphasises strong slopes in the bandpass filtered signal, such as the one present in the QRS-complex. This is accomplished by using the filter with the following transfer function for signals with sampling period T

$$H(z) = \frac{1}{8T}(-z^{-2} - 2z^{-1} + 2z + z^{2}) \tag{5.3}$$

In the fourth step, the value of each sample of the signal is squared to make all values positive and to make higher values in the output of the third step stand out more. Finally, a moving-window integration defined as

$$y(nT) = \frac{1}{W} \tag{5.4}$$

is applied to the signal to smooth out smaller peaks. Here W is the width of the integration window and was set to 150 ms. [25]

Different from the method proposed in [25], the signal resulting from the fifth step is thresholded using a threshold value that is equal to the product of the mean and maximum values in the signal resulting from the fifth step. The algorithm was also extended to determine Q- and S-peaks, based on the location of the R-peak. This was achieved by simply searching for local minima in a small interval to the left and right of the R-peak.

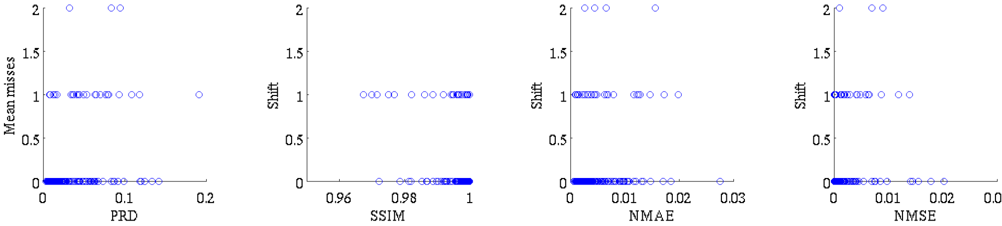
For this thesis, existing source code[1] was used with some alterations to make the algorithm more robust at the first and last samples of the signal.

5.2 Experiments

The goal of the experiments is to find a correlation a) between the compression ratio CR and the difference in peak locations from those in the original signal (denoted as 'shift') and b) between the quality measures and the shift.

This is achieved by first applying the Pan-Tompkins algorithm to the original signal and storing the locations of the Q-, R- and S-peaks that are found there. Next, the signal is compressed using different CR values (0.9 to 0.2 in steps of 0.1 like in chapter 3) and reconstructed. The reconstructed signals are subsequently processed using the Pan-Tompkins algorithm as well and the found locations are compared to those found earlier in the original signal. The absolute differences in

[1] http://matlabz.blogspot.be/2011/04/contents-cancellation-dc-drift-and.html



(a) (b) (c) (d)

Figure 5.1: Difference in number of detected peaks as function of (a) PRD (b) SSIM (c) NMAE and (d) NMSE

the locations between the original and reconstructed signals are stored for each CR value so they can be compared. Since it is possible that the algorithm may detect a different amount of peaks in the reconstructed signals, the difference in the amount of registered peaks was counted as well and in case of a difference, only the locations of the peaks closest to those in the original signal were compared. Since the detection of Q and S peaks depends on the detection the corresponding R peak, an equal amount of Q, R and S peaks will always be detected.

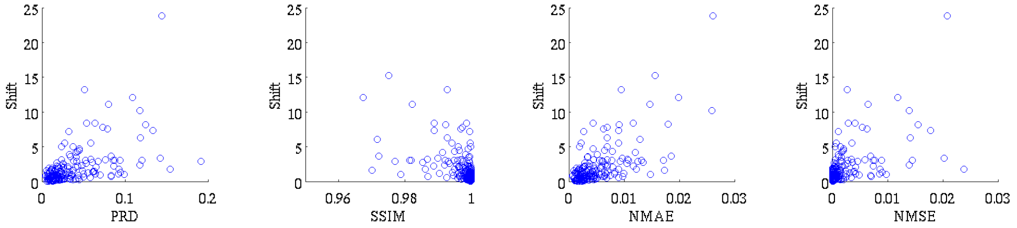
The signals used are the 82 first seconds of the second channel of the first 23 patients in the MIT-BIH Arrhythmia Database. For the reconstructions, the original signal is split into 2-second segments like before, but after the reconstruction, the 41 reconstructed segments are combined again to form a complete 82-second reconstructed signal.

For brevity, only the best method for single-channel reconstruction (GAP with a Second Order Derivative analysis matrix) is used for these experiments.

5.3 Results

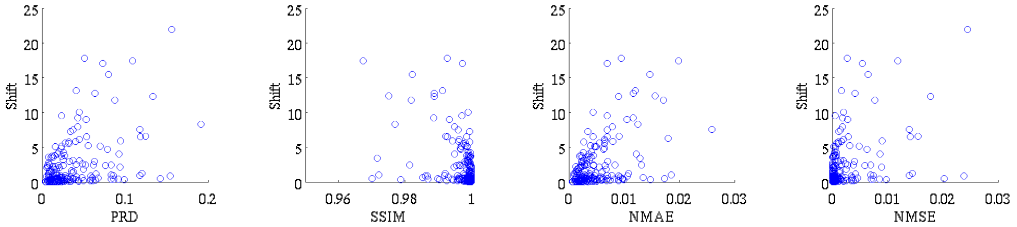
The difference in the amount of detected peaks is plotted as a function of the four accuracy metrics in figure 5.1. What stands out immediately, is that there are always maximally two peaks detected more or less than in the original signal, if there is any difference at all. It is interesting that even a rudimentary algorithm like Pan-Tompkins is able to detect a nearly equal number of peaks in the original signal and even the poorer reconstructions thereof. This means that the heart rate could even be detected from signals that were reconstructed from highly compressed measurements. Although there are some differences in the number of detected peaks even at the better values of the metrics (low PRD, NMAE & NMSE and high SSIM), it can be seen that at those better values, there are more signals where there was no difference in the number of detected peaks than at the worse values of the metrics.

In figure 5.5a the average number of peak differences is shown at each CR value. It can be seen that these mean values are generally quite low, as was already concluded from figure 5.1, and that there is hardly a correlation between the amount



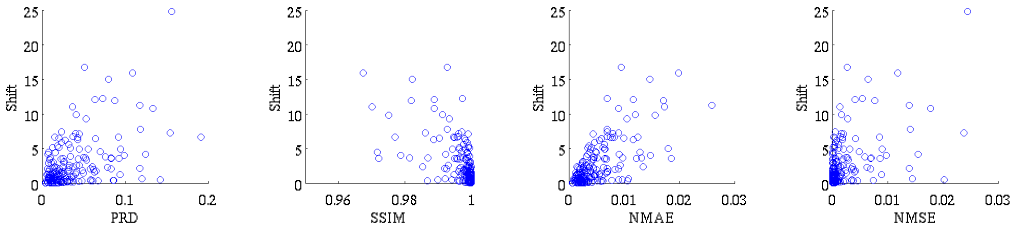
(a) (b) (c) (d)

Figure 5.2: Shift in the detected Q peak location (as a number of samples) as a function of (a) PRD (b) SSIM (c) NMAE and (d) NMSE



(a) (b) (c) (d)

Figure 5.3: Shift in the detected R peak location (as a number of samples) as a function of (a) PRD (b) SSIM (c) NMAE and (d) NMSE



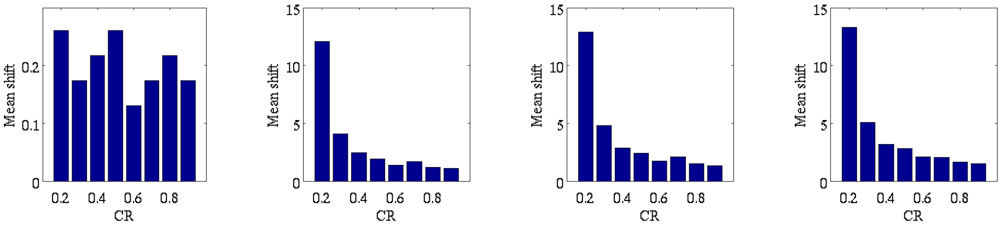
(a) (b) (c) (d)

Figure 5.4: Shift in the detected S peak location (as a number of samples) as a function of (a) PRD (b) SSIM (c) NMAE and (d) NMSE

of compression and the mean difference in the number of detected peaks.

Figures 5.2, 5.3 and 5.4 show scatter plots with the four accuracy metrics on the horizontal axis and the shift, expressed as a number of samples, on the vertical axis for the Q (5.2), R (5.3) and S (5.4) peaks. The range of both axes is limited to exclude some extreme outliers and make the plots clearer.

The maximum shift is reasonably low in all three cases, with most of them being smaller than 15 samples. Considering that the sample rate $f_s = 360Hz$,



(a) (b) (c) (d)

Figure 5.5: Difference in (a) number of detected peaks (b) Q peak location (c) R peak location and (d) S peak location as a function of the CR.

this means that most of the peak locations are detected at less than 0.05 seconds from the locations they are detected at in the original signal, even in really poor reconstructions.

Despite the fact that there is no really clear correlation between the metrics and the amount of shift of each peak, the lowest shifts are mostly concentrated around the better accuracy metric values, which may mean that these metrics may serve as an indicator to decide if a signal reconstruction is still useful for a certain application. However, this should of course be verified by comparing the metric values to ratings of the signal quality that are given by a professional cardiologist, something that was not possible for this thesis.

Finally, figures 5.5b, 5.5c & 5.5d show the mean shifts as a function of the CR. It is clear that with lower CR values (i.e. more compression), it becomes harder to detect the peak close to its original position, as poor reconstruction may lead to deformation of the peaks and therefore a shift of the position of the local extrema.

5.4 Discussion

The aim of this chapter was to find a correlation between (a) the accuracy metrics and the ability to detect the Q, R and S peaks at the same position as in an uncompressed signal and (b) the rate of compression and this ability.

No clear correlation was found between the accuracy metrics and the shift of the peaks in the reconstructed signals. Although it should be noted that at low PRD, NMAE & NMSE and high SSIM values, low shifts are much more common. On the other hand, there is an obvious correlation between the CR and the mean peak shifts. Logically, more compression leads to a more deformed reconstructed signal and therefore to a higher chance of detecting the peaks at a more different position.

This chapter was merely an introduction to this topic. With the help of professional signal quality ratings and a much larger dataset of reconstructed signals, it could be further investigated if there could after all be some correlation between the accuracy measures and the usefulness of the signals for certain applications and how much compression could be allowed for those applications.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Several conclusions can be drawn from the work in this thesis. Most importantly, the usefulness of cosparse algorithms for ECG was proven. The results from the conducted experiments indicate that cosparse algorithms do in fact lead to better reconstruction accuracy than their sparse counterparts. Three analysis matrices were tested for the cosparse approach, of which the second order derivative matrix was found to be the best one in terms of reconstruction accuracy. For the sparse approach, the DWT dictionary was found to lead to the best possible reconstructions.

Secondly, it was proven that simultaneous reconstruction of multiple ECG channels is beneficial in terms of processing time for the cosparse approach. The sparse approach did not show an improvement, but processing times stayed similar to those of the single-channel algorithms. For both approaches, the reconstruction accuracy remained the same, whether a single- or multi-channel algorithm was used. This means that, if the coefficient vectors or cosparse sequences of several signals have a similar (co-)support (as is the case in multi-channel ECG recordings), multi-channel algorithms can be used to reconstruct these signals in the same amount of time (sparse approach) or faster (cosparse approach).

Next, the sparse OM(M)P and cosparse GAP algorithm were adjusted to account for the presence of additive and multiplicative noise. Noise aware OM(M)P showed a vast improvement in accuracy, compared to a reconstruction from a noisy measurement using the original algorithm. In GAP, the improvement was smaller, but GAP itself proved to be quite robust to noise already. In real signals, the reconstruction accuracy of both algorithms was similar, while OMMPn was more accurate when applied to a basic set of simulated signals.

Finally, the shift, i.e. the difference in the position where the Q, R and S peaks were detected in the reconstructed signals compared to their positions in the original signal, was investigated. No clear correlation between the values of the accuracy metrics and the amount of shift was found, although most low shifts were found in signals with good accuracy metrics. On the other hand, it was found that when more compression is applied, on average the shift will be larger.

6.2 Future Work

The contents of chapters 4 and 5 may serve as the basis for further research on their respective topics. Although it was shown that it is possible to adjust the greedy sparse and cosparse algorithms to improve their performance in the presence of additive and multiplicative noise, there is still much room for improvement, since neither the sparse, nor the cosparse algorithms were able to achieve PRD values suitable for decent clinical analysis (i.e. below 9% according to the literature) except at the lowest noise levels. An important part of this work could be the search for a decent analysis matrix that is able to create a sparse approximation of the signal estimate in the GAPn-algorithm, but with better resulting reconstruction accuracy than the wavelet analysis matrix.

The second topic for possible future work is the testing of the applicability of compressive sensing for actual applications. A minor example was already shown in chapter 5, but this could be expanded to other applications than QRS-detection. By having a professional rate a series of reconstructed signals in terms of their usefulness for certain applications and correlating these ratings to the accuracy metrics, threshold values for each metric could be found so that the metrics could be used as indicators of signal usefulness for certain applications instead of only using them as measures of similarity to the original signal.

Appendices

Appendix A

Method Result Tables

The following tabels summarize the five characteristic values of a boxplot (see section 3.3) for all of the methods indicated in table Table A.1.

	Dictionary			Analysis Matrix		
	DCT	DFT	DWT	1st Order Derivative	2nd Order Derivative	Wavelet
BSBL-BO			G			
BSBL-EM			G			
IHT			G			
OMP	G	G	G			
OMMP	G	G	G			
SOMP	G	G	G			
SOMMP	G	G	G/B			
GAP				G	G	G
SGAP				G	G/B	G

Table A.1: Overview of the methods for which the results are included in appendix A. 'G' indicates results for a Gaussian sensing matrix, 'B' indicates results for a binary sensing matrix.

A.1 BSBLBO / Gaussian / DWT

	Channel 1					Channel 2				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**	**LOW**	**.25**	**MED**	**.75**	**HIGH**
PRD										
.9	.065	.192	.229	.276	.403	0	.130	.170	.254	.441
.8	.060	.194	.234	.282	.415	0	.131	.173	.253	.437
.7	.060	.196	.233	.287	.422	0	.138	.185	.264	.454
.6	.063	.201	.235	.292	.430	0	.143	.205	.285	.498
.5	.067	.213	.251	.310	.457	0	.144	.216	.292	.513
.4	.060	.218	.266	.323	.481	0	.146	.223	.298	.527
.3	.017	.227	.278	.368	.578	0	.152	.234	.327	.590
.2	.109	.280	.333	.395	.566	0	.165	.252	.342	.607
NMAE										
.9	.007	.021	.025	.031	.045	0	.021	.034	.058	.112
.8	.007	.022	.026	.032	.047	0	.022	.035	.057	.111
.7	.009	.023	.028	.032	.046	0	.025	.036	.058	.108
.6	.010	.024	.028	.033	.047	0	.030	.039	.059	.103
.5	.013	.026	.030	.035	.049	0	.032	.041	.060	.103
.4	.015	.028	.032	.037	.050	0	.035	.043	.061	.101
.3	.018	.032	.036	.041	.055	0	.038	.048	.063	.101
.2	.024	.039	.043	.049	.064	0	.039	.051	.069	.115
NMSE										
.9	0	.037	.052	.076	.136	0	.017	.029	.065	.136
.8	0	.037	.055	.080	.143	0	.017	.030	.064	.135
.7	0	.038	.054	.082	.148	0	.019	.034	.070	.146
.6	0	.040	.055	.085	.153	0	.020	.042	.081	.172
.5	0	.045	.063	.096	.173	0	.021	.047	.085	.182
.4	0	.048	.071	.104	.190	0	.021	.050	.089	.191
.3	0	.052	.077	.135	.261	0	.023	.055	.107	.233
.2	0	.079	.111	.156	.271	0	.027	.063	.117	.251
SSIM										
.9	.743	.859	.895	.936	1.051	.355	.713	.793	.951	1.309
.8	.721	.846	.892	.930	1.055	.317	.698	.797	.951	1.332
.7	.709	.837	.894	.922	1.051	.353	.690	.771	.914	1.251
.6	.666	.818	.884	.920	1.072	.329	.661	.742	.883	1.216
.5	.629	.800	.880	.914	1.086	.311	.650	.727	.876	1.214
.4	.603	.785	.867	.906	1.088	.362	.645	.711	.834	1.118
.3	.565	.759	.845	.888	1.081	.338	.596	.669	.767	1.025
.2	.286	.624	.752	.850	1.188	.310	.553	.620	.714	.957

Table A.2: BSBLBO accuracy (Gaussian sensing / DWT dictionary)

	Channel 1+2				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**
TIME					
.9	.257	.953	1.135	1.417	2.113
.8	.225	.810	.933	1.199	1.784
.7	.232	.620	.701	.880	1.268
.6	.190	.508	.584	.720	1.038
.5	.157	.415	.489	.587	.846
.4	.179	.355	.407	.473	.650
.3	.107	.302	.355	.432	.627
.2	.024	.282	.339	.454	.713
ITERS					
.9	8.2	36	42	54.5	82.2
.8	12	36	42	52	76
.7	11.5	35.5	41	51.5	75.5
.6	2.5	34	42	55	86.5
.5	9.5	35	43	52	77.5
.4	9.2	37	43.5	55.5	83.2
.3	0	36	46	60.5	97.2
.2	0	42	59	92	167

Table A.3: BSBLBO computations (Gaussian sensing / DWT dictionary)

A.2 BSBLEM / Gaussian / DWT

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
					PRD					
.9	0	.081	.109	.145	.242	.002	.104	.137	.171	.273
.8	0	.088	.123	.165	.280	.021	.114	.144	.175	.268
.7	0	.099	.141	.181	.305	.012	.122	.155	.195	.305
.6	0	.111	.158	.202	.339	.019	.143	.177	.226	.350
.5	0	.123	.184	.226	.379	.008	.151	.193	.246	.389
.4	.010	.155	.212	.251	.396	0	.161	.226	.286	.475
.3	.085	.217	.251	.306	.438	0	.179	.264	.339	.578
.2	.064	.361	.435	.558	.855	0	.354	.545	.743	1.327
					NMAE					
.9	5.2e-4	.011	.014	.017	.028	0	.015	.024	.052	.108
.8	2.9e-4	.012	.015	.020	.031	0	.017	.027	.055	.112
.7	7.4e-4	.013	.017	.021	.034	0	.018	.029	.057	.116
.6	0	.015	.020	.025	.040	0	.023	.035	.058	.109
.5	.004	.017	.022	.026	.039	0	.028	.036	.059	.104
.4	.009	.021	.026	.029	.042	0	.035	.042	.063	.106
.3	.007	.027	.033	.039	.059	0	.038	.048	.067	.109
.2	0	.048	.068	.089	.151	0	.052	.095	.231	.499
					NMSE					
.9	0	.007	.012	.021	.043	0	.011	.019	.029	.057
.8	0	.008	.015	.027	.056	0	.013	.021	.031	.058
.7	0	.010	.020	.033	.068	0	.015	.024	.038	.073
.6	0	.012	.025	.041	.084	0	.020	.031	.051	.097
.5	0	.015	.034	.051	.104	0	.023	.037	.061	.117
.4	0	.024	.045	.063	.122	0	.026	.051	.082	.166
.3	0	.047	.063	.093	.163	0	.032	.070	.115	.239
.2	0	.130	.189	.312	.584	0	.125	.297	.553	1.193
					SSIM					
.9	.922	.962	.976	.988	1.028	.546	.812	.930	.989	1.255
.8	.906	.953	.972	.984	1.030	.514	.796	.908	.984	1.266
.7	.892	.944	.965	.980	1.032	.437	.762	.869	.978	1.302
.6	.856	.926	.952	.973	1.043	.396	.731	.836	.955	1.291
.5	.824	.908	.942	.964	1.048	.419	.714	.808	.911	1.206
.4	.796	.888	.921	.949	1.040	.377	.666	.752	.859	1.147
.3	.719	.835	.873	.912	1.028	.332	.607	.694	.791	1.066
.2	.098	.501	.654	.769	1.172	0	.217	.473	.642	1.279

Table A.4: BSBLEM accuracy (Gaussian sensing / DWT dictionary)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
			TIME		
.9	.755	1.906	2.238	2.674	3.826
.8	.749	1.679	1.907	2.299	3.230
.7	.436	1.250	1.404	1.792	2.606
.6	.443	1.098	1.231	1.535	2.191
.5	.495	1.169	1.322	1.619	2.294
.4	.152	1.269	1.477	2.014	3.132
.3	0	1.667	2.324	3.024	5.059
.2	.366	2.494	3.055	3.912	6.040
			ITERS		
.9	0	35.5	45.5	63	104.2
.8	0.5	38	49	63	100.5
.7	0	39	49	65	104
.6	0	43	55.5	79	133
.5	0	56	71	98	161
.4	0	76	98	157.5	279.8
.3	0	137	208	326	609.5
.2	0	235.5	374	506.5	913

Table A.5: BSBLEM computations (Gaussian sensing / DWT dictionary)

A.3 IHT / Gaussian / DWT

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
PRD										
.9	0	.038	.056	.079	.139	0	.045	.066	.100	.183
.8	0	.041	.058	.083	.146	0	.046	.067	.104	.191
.7	0	.042	.060	.085	.149	0	.048	.070	.107	.196
.6	0	.045	.063	.090	.158	0	.051	.074	.114	.207
.5	0	.044	.065	.093	.166	0	.054	.080	.120	.220
.4	0	.053	.078	.112	.200	0	.062	.091	.133	.239
.3	0	.074	.106	.150	.265	0	.083	.125	.185	.338
.2	0	.180	.240	.316	.522	0	.163	.229	.324	.566
NMAE										
.9	5.2e-4	.008	.010	.013	.020	0	.009	.012	.018	.033
.8	0	.008	.011	.014	.022	0	.009	.012	.019	.034
.7	4.7e-5	.008	.011	.014	.022	0	.010	.013	.020	.034
.6	0	.009	.012	.015	.024	0	.010	.014	.021	.037
.5	6.0e-5	.009	.012	.015	.025	0	.011	.015	.022	.038
.4	0	.011	.015	.019	.031	0	.013	.017	.025	.043
.3	0	.016	.020	.027	.043	0	.018	.024	.035	.061
.2	0	.036	.047	.061	.097	0	.035	.044	.060	.097
NMSE										
.9	0	.001	.003	.006	.013	0	.002	.004	.010	.022
.8	0	.002	.003	.007	.015	0	.002	.004	.011	.024
.7	0	.002	.004	.007	.015	0	.002	.005	.011	.025
.6	0	.002	.004	.008	.017	0	.003	.006	.013	.028
.5	0	.002	.004	.009	.019	0	.003	.006	.014	.032
.4	0	.003	.006	.013	.027	0	.004	.008	.018	.038
.3	0	.005	.011	.023	.048	0	.007	.016	.034	.075
.2	0	.032	.057	.100	.202	0	.026	.053	.105	.223
SSIM										
.9	.983	.991	.994	.996	1.004	.952	.978	.990	.996	1.023
.8	.981	.990	.994	.996	1.005	.948	.977	.990	.996	1.025
.7	.980	.990	.993	.996	1.005	.945	.975	.989	.996	1.026
.6	.978	.988	.992	.995	1.005	.938	.972	.988	.995	1.029
.5	.977	.988	.992	.995	1.006	.932	.969	.987	.994	1.031
.4	.967	.982	.989	.993	1.008	.915	.961	.982	.992	1.039
.3	.939	.968	.980	.987	1.016	.852	.932	.971	.985	1.065
.2	.708	.847	.900	.940	1.079	.597	.806	.903	.945	1.154

Table A.6: IHT accuracy (Gaussian sensing / DWT dictionary)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
TIME					
.9	.079	.119	.130	.146	.186
.8	.069	.100	.109	.121	.152
.7	.082	.113	.121	.133	.165
.6	.076	.101	.107	.117	.141
.5	.066	.090	.097	.106	.129
.4	.048	.071	.078	.086	.109
.3	.048	.077	.086	.097	.126
.2	.035	.058	.066	.074	.098
ITERS					
.9	17	29	32	37	49
.8	20.5	31	33	38	48.5
.7	21.5	32	35	39	49.5
.6	22.5	33	37	40	50.5
.5	24	36	40	44	56
.4	26.5	40	44	49	62.5
.3	34	49	54	59	74
.2	28	49	57	63	84

Table A.7: IHT computations (Gaussian sensing / DWT dictionary)

A.4 OMP / Gaussian / DCT

	Channel 1					Channel 2				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**	**LOW**	**.25**	**MED**	**.75**	**HIGH**
					PRD					
.9	.042	.059	.064	.070	.087	.036	.058	.063	.073	.095
.8	.038	.059	.065	.074	.095	.031	.059	.066	.079	.107
.7	.036	.062	.068	.080	.106	.027	.064	.072	.089	.126
.6	.024	.065	.075	.093	.134	.010	.066	.076	.102	.158
.5	0	.073	.090	.135	.229	0	.073	.090	.142	.247
.4	0	.086	.147	.265	.533	0	.084	.130	.237	.466
.3	0	.151	.287	.457	.917	0	.137	.223	.410	.819
.2	0	.293	.437	.647	1.179	0	.238	.381	.585	1.104
					NMAE					
.9	0	.010	.013	.016	.027	0	.009	.012	.019	.034
.8	0	.010	.013	.017	.028	0	.010	.013	.020	.036
.7	0	.011	.014	.019	.031	0	.010	.015	.023	.042
.6	0	.012	.016	.022	.037	0	.011	.017	.026	.048
.5	0	.016	.021	.028	.046	0	.014	.023	.032	.061
.4	0	.024	.033	.045	.076	0	.020	.033	.050	.095
.3	0	.040	.061	.077	.133	0	.038	.059	.075	.131
.2	.024	.075	.091	.108	.158	.017	.067	.087	.101	.152
					NMSE					
.9	.001	.003	.004	.005	.007	4.5e-4	.003	.004	.005	.008
.8	6.5e-4	.004	.004	.005	.008	0	.004	.004	.006	.010
.7	1.5e-4	.004	.005	.006	.010	0	.004	.005	.008	.014
.6	0	.004	.006	.009	.015	0	.004	.006	.011	.020
.5	0	.005	.008	.018	.038	0	.005	.008	.020	.043
.4	0	.007	.022	.070	.164	0	.007	.017	.056	.130
.3	0	.023	.082	.209	.489	0	.019	.050	.168	.391
.2	0	.086	.191	.419	.918	0	.057	.145	.342	.769
					SSIM					
.9	.971	.986	.991	.996	1.011	.947	.977	.991	.997	1.027
.8	.968	.985	.991	.996	1.012	.943	.975	.989	.997	1.029
.7	.962	.982	.989	.995	1.015	.923	.967	.986	.996	1.040
.6	.949	.976	.986	.994	1.021	.907	.960	.983	.995	1.048
.5	.916	.961	.979	.990	1.034	.842	.932	.972	.993	1.083
.4	.771	.896	.950	.980	1.105	.619	.837	.944	.983	1.201
.3	.357	.710	.840	.945	1.298	.202	.649	.796	.948	1.395
.2	0	.455	.641	.780	1.268	0	.415	.574	.843	1.484

Table A.8: OMP accuracy (Gaussian sensing / DCT dictionary)

	Channel 1+2				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**
			TIME		
.9	0	.139	.207	.414	.827
.8	0	.167	.245	.438	.846
.7	0	.145	.218	.396	.773
.6	0	.130	.188	.343	.661
.5	0	.106	.157	.287	.558
.4	0	.086	.128	.219	.419
.3	0	.057	.077	.110	.188
.2	.011	.031	.038	.044	.064
			ITERS		
.9	0	146	181	252.7	412.9
.8	0	146	179	250	406
.7	0	142	176	243	394.5
.6	0	142	174	240.7	388.9
.5	0	137	169	230.7	371.4
.4	14.5	139	167	222	346.5
.3	41	125	147	181	265
.2	53.5	97	110	126	169.5

Table A.9: OMP computations (Gaussian sensing / DCT dictionary)

A.5 OMP / Gaussian / DFT

	Channel 1					Channel 2				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**	**LOW**	**.25**	**MED**	**.75**	**HIGH**
PRD										
.9	.040	.057	.062	.069	.086	.032	.058	.064	.076	.102
.8	.037	.059	.064	.073	.095	.026	.059	.066	.081	.113
.7	.040	.064	.070	.080	.104	.019	.061	.068	.088	.130
.6	.028	.063	.072	.087	.123	.012	.064	.074	.098	.150
.5	.015	.069	.080	.105	.159	0	.069	.083	.123	.202
.4	0	.073	.094	.158	.285	0	.077	.103	.171	.312
.3	0	.105	.202	.336	.683	0	.110	.180	.308	.606
.2	0	.211	.365	.535	1.022	0	.186	.295	.489	.944
NMAE										
.9	0	.009	.012	.016	.027	0	.009	.012	.019	.035
.8	0	.010	.012	.017	.028	0	.010	.013	.020	.035
.7	0	.010	.014	.019	.032	0	.010	.014	.021	.039
.6	0	.011	.015	.020	.034	0	.011	.016	.023	.042
.5	0	.013	.017	.024	.040	0	.013	.019	.027	.050
.4	0	.016	.022	.030	.052	0	.016	.024	.037	.067
.3	0	.029	.043	.058	.100	0	.028	.044	.060	.110
.2	.003	.055	.076	.090	.141	0	.050	.070	.084	.136
NMSE										
.9	.001	.003	.004	.005	.007	0	.003	.004	.006	.009
.8	5.4e-4	.003	.004	.005	.008	0	.003	.004	.006	.011
.7	6.4e-4	.004	.005	.006	.010	0	.004	.005	.008	.014
.6	0	.004	.005	.008	.013	0	.004	.005	.010	.018
.5	0	.005	.006	.011	.021	0	.005	.007	.015	.030
.4	0	.005	.009	.025	.054	0	.006	.011	.029	.064
.3	0	.011	.041	.113	.266	0	.012	.032	.095	.219
.2	0	.044	.133	.286	.649	0	.035	.087	.239	.546
SSIM										
.9	.970	.986	.992	.996	1.012	.943	.975	.991	.997	1.028
.8	.968	.985	.991	.996	1.012	.944	.975	.990	.996	1.028
.7	.961	.981	.989	.995	1.015	.936	.972	.988	.996	1.032
.6	.956	.979	.988	.994	1.017	.926	.968	.985	.995	1.036
.5	.937	.971	.984	.993	1.026	.899	.956	.980	.993	1.050
.4	.904	.955	.977	.989	1.040	.803	.914	.970	.988	1.100
.3	.611	.826	.915	.969	1.185	.421	.750	.908	.970	1.299
.2	.095	.563	.741	.876	1.344	0	.522	.692	.913	1.501

Table A.10: OMP accuracy (Gaussian sensing / DFT dictionary)

	Channel 1+2				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**
TIME					
.9	0	.559	.905	1.762	3.566
.8	0	.547	.896	1.638	3.275
.7	0	.454	.740	1.365	2.731
.6	0	.394	.616	1.159	2.307
.5	0	.318	.504	.887	1.740
.4	0	.266	.412	.683	1.309
.3	0	.188	.284	.423	.777
.2	.020	.091	.112	.138	.209
ITERS					
.9	0	182	233	322	532
.8	0	181	233	316.7	520.4
.7	0	175	226	307	505
.6	0	175	222	300	487.5
.5	0	168.2	213	285	460.1
.4	11.5	169	208	274	431.5
.3	42.5	155	190	230	342.5
.2	78	129	145	163	214

Table A.11: OMP computations (Gaussian sensing / DFT dictionary)

A.6 OMP / Gaussian / DWT

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
					PRD					
.9	.047	.056	.058	.062	.070	.025	.054	.058	.073	.102
.8	.040	.052	.056	.060	.073	.027	.056	.060	.076	.105
.7	.044	.055	.059	.063	.074	.021	.056	.062	.080	.116
.6	.038	.055	.060	.066	.083	.018	.059	.063	.086	.126
.5	.040	.058	.063	.070	.088	.011	.059	.065	.092	.140
.4	.042	.065	.070	.080	.102	.009	.064	.073	.102	.157
.3	.033	.071	.080	.097	.135	0	.071	.082	.120	.192
.2	0	.089	.114	.173	.300	0	.093	.127	.200	.361
					NMAE					
.9	0	.008	.010	.015	.026	0	.008	.012	.018	.032
.8	0	.008	.010	.015	.025	0	.009	.012	.018	.033
.7	0	.008	.010	.016	.026	0	.009	.013	.019	.034
.6	0	.009	.011	.016	.026	0	.009	.014	.021	.038
.5	0	.009	.011	.017	.029	0	.010	.014	.022	.040
.4	0	.010	.013	.019	.032	0	.011	.015	.024	.045
.3	0	.012	.015	.022	.036	0	.012	.018	.028	.051
.2	0	.018	.024	.032	.053	0	.019	.027	.041	.073
					NMSE					
.9	.002	.003	.003	.004	.005	0	.003	.003	.005	.009
.8	.001	.003	.003	.004	.005	0	.003	.004	.006	.010
.7	.002	.003	.003	.004	.005	0	.003	.004	.006	.011
.6	.001	.003	.004	.004	.006	0	.003	.004	.007	.013
.5	.001	.003	.004	.005	.007	0	.004	.004	.008	.016
.4	9.0e-4	.004	.005	.006	.010	0	.004	.005	.010	.020
.3	0	.005	.006	.009	.016	0	.005	.007	.014	.028
.2	0	.008	.013	.030	.063	0	.009	.016	.040	.087
					SSIM					
.9	.971	.986	.993	.997	1.012	.951	.979	.991	.998	1.026
.8	.975	.988	.993	.997	1.010	.945	.977	.990	.997	1.029
.7	.972	.987	.993	.997	1.011	.941	.975	.990	.997	1.031
.6	.973	.987	.992	.996	1.010	.930	.970	.988	.997	1.037
.5	.970	.985	.991	.996	1.012	.924	.968	.988	.997	1.041
.4	.961	.981	.989	.995	1.015	.906	.960	.985	.996	1.050
.3	.954	.977	.986	.993	1.016	.884	.950	.980	.994	1.061
.2	.908	.952	.972	.982	1.026	.779	.903	.958	.986	1.110

Table A.12: OMP accuracy (Gaussian sensing / DWT dictionary)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
			TIME		
.9	0	.107	.160	.374	.773
.8	0	.110	.163	.354	.719
.7	0	.099	.138	.276	.541
.6	0	.086	.127	.229	.444
.5	0	.085	.114	.183	.329
.4	0	.061	.085	.131	.236
.3	0	.047	.059	.081	.133
.2	.018	.036	.042	.049	.067
			ITERS		
.9	0	120	147	201.7	324.4
.8	0	120	147	200	320
.7	12.4	120	146	191.7	299.4
.6	18.9	119	144	185.7	285.9
.5	30.5	119	143	178	266.5
.4	42.5	116	137	165	238.5
.3	53	113	131	153	213
.2	77	116	128	142	181

Table A.13: OMP computations (Gaussian sensing / DWT dictionary)

A.7 OMMP / Gaussian / DCT

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
PRD										
.9	.042	.059	.064	.070	.087	.036	.058	.063	.073	.095
.8	.036	.059	.064	.073	.096	.030	.059	.065	.078	.107
.7	.031	.061	.067	.080	.110	.025	.062	.070	.087	.124
.6	.019	.063	.073	.093	.138	.006	.065	.076	.104	.163
.5	0	.071	.088	.127	.211	0	.071	.091	.145	.256
.4	0	.089	.140	.259	.514	0	.085	.129	.232	.453
.3	0	.149	.276	.427	.843	0	.139	.211	.391	.770
.2	0	.281	.433	.605	1.091	0	.236	.374	.570	1.070
NMAE										
.9	0	.009	.013	.016	.027	0	.009	.012	.020	.035
.8	0	.010	.013	.017	.028	0	.009	.013	.020	.036
.7	0	.011	.014	.019	.031	0	.010	.015	.022	.041
.6	0	.012	.016	.022	.037	0	.011	.017	.025	.047
.5	0	.014	.020	.027	.045	0	.013	.022	.032	.060
.4	0	.023	.033	.043	.074	0	.021	.033	.048	.089
.3	0	.040	.058	.073	.122	0	.038	.057	.072	.122
.2	.026	.071	.087	.101	.147	.020	.066	.082	.096	.142
NMSE										
.9	.001	.004	.004	.005	.007	5.1e-4	.003	.004	.005	.008
.8	4.8e-4	.003	.004	.005	.008	0	.004	.004	.006	.010
.7	0	.004	.005	.006	.011	0	.004	.005	.008	.013
.6	0	.004	.005	.009	.016	0	.004	.006	.011	.021
.5	0	.005	.008	.016	.033	0	.005	.008	.021	.045
.4	0	.008	.020	.067	.156	0	.007	.017	.054	.124
.3	0	.022	.076	.182	.422	0	.019	.044	.153	.354
.2	0	.079	.187	.366	.796	0	.056	.140	.324	.728
SSIM										
.9	.970	.986	.991	.996	1.012	.945	.976	.991	.997	1.029
.8	.969	.985	.991	.996	1.012	.944	.976	.989	.997	1.029
.7	.962	.982	.989	.995	1.015	.927	.969	.987	.996	1.038
.6	.952	.977	.987	.994	1.019	.910	.961	.983	.995	1.047
.5	.928	.966	.981	.991	1.029	.843	.933	.973	.993	1.083
.4	.792	.904	.952	.979	1.091	.635	.843	.944	.982	1.190
.3	.375	.715	.844	.941	1.281	.246	.668	.809	.949	1.371
.2	5.5e-4	.479	.653	.798	1.276	0	.423	.579	.842	1.470

Table A.14: OMMP accuracy (Gaussian sensing / DCT dictionary)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
TIME					
.9	0	.063	.089	.148	.275
.8	0	.066	.090	.145	.263
.7	0	.062	.086	.134	.243
.6	0	.057	.076	.124	.224
.5	0	.049	.065	.101	.178
.4	0	.043	.057	.083	.142
.3	.006	.035	.043	.054	.082
.2	.012	.022	.025	.028	.037
ITERS					
.9	0	36.2	45	63	103.1
.8	0	37	45	63	102
.7	0	37	45	62	99.5
.6	0	36	45	62	101
.5	0	36	43	60	96
.4	3	36	44	58	91
.3	13	34	40	48	69
.2	15	27	31	35	47

Table A.15: OMMP computations (Gaussian sensing / DCT dictionary)

A.8 OMMP / Gaussian / DFT

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
					PRD					
.9	.038	.058	.063	.071	.090	.033	.060	.066	.079	.106
.8	.039	.060	.065	.074	.096	.023	.058	.064	.081	.116
.7	.037	.062	.068	.079	.104	.019	.062	.070	.090	.133
.6	.020	.060	.070	.087	.127	.003	.062	.074	.102	.162
.5	.016	.071	.082	.107	.161	0	.067	.084	.130	.224
.4	0	.076	.103	.174	.323	0	.082	.116	.192	.356
.3	0	.108	.218	.367	.755	0	.113	.193	.348	.701
.2	0	.233	.398	.593	1.132	0	.204	.322	.546	1.059
					NMAE					
.9	0	.009	.012	.017	.028	0	.009	.013	.020	.036
.8	0	.010	.013	.018	.030	0	.009	.013	.020	.035
.7	0	.010	.014	.019	.032	0	.010	.015	.022	.039
.6	0	.011	.015	.021	.035	0	.011	.016	.024	.043
.5	0	.013	.018	.025	.043	0	.013	.020	.028	.052
.4	0	.017	.024	.032	.055	0	.018	.027	.041	.076
.3	0	.030	.047	.062	.112	0	.030	.048	.067	.122
.2	.005	.062	.083	.101	.158	0	.055	.080	.097	.160
					NMSE					
.9	8.3e-4	.003	.004	.005	.007	0	.004	.004	.006	.010
.8	7.1e-4	.004	.004	.006	.008	0	.003	.004	.007	.011
.7	3.6e-4	.004	.005	.006	.010	0	.004	.005	.008	.015
.6	0	.004	.005	.008	.013	0	.004	.006	.010	.020
.5	0	.005	.007	.011	.021	0	.004	.007	.017	.035
.4	0	.006	.011	.030	.067	0	.007	.014	.037	.082
.3	0	.012	.048	.135	.319	0	.013	.037	.121	.284
.2	0	.054	.159	.351	.796	0	.042	.103	.298	.682
					SSIM					
.9	.969	.985	.991	.996	1.012	.940	.974	.990	.996	1.030
.8	.966	.984	.990	.995	1.013	.946	.976	.990	.996	1.026
.7	.961	.981	.989	.995	1.016	.936	.972	.987	.996	1.032
.6	.957	.979	.988	.994	1.017	.925	.967	.984	.995	1.037
.5	.933	.969	.983	.992	1.028	.888	.951	.980	.993	1.056
.4	.889	.948	.973	.987	1.047	.748	.890	.962	.985	1.127
.3	.567	.807	.907	.966	1.206	.349	.717	.888	.963	1.332
.2	0	.505	.699	.863	1.399	0	.456	.641	.902	1.572

Table A.16: OMMP accuracy (Gaussian sensing / DFT dictionary)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
			TIME		
.9	0	.195	.295	.541	1.059
.8	0	.178	.264	.457	.876
.7	0	.151	.224	.390	.748
.6	0	.137	.198	.341	.648
.5	0	.112	.165	.271	.510
.4	0	.094	.136	.216	.401
.3	0	.080	.108	.148	.249
.2	.022	.050	.059	.069	.097
			ITERS		
.9	0	45	58	81	135
.8	0	46	58	79	128.5
.7	0	45	57	78	127.5
.6	0	46	57	77	123.5
.5	0.5	44	55	73	116.5
.4	2.5	43	53	70	110.5
.3	7.5	39	49	60	91.5
.2	21.5	35	39	44	57.5

Table A.17: OMMP computations (Gaussian sensing / DFT dictionary)

A.9 OMMP / Gaussian / DWT

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
PRD										
.9	.037	.050	.054	.059	.072	.026	.054	.058	.072	.099
.8	.037	.051	.056	.061	.076	.027	.054	.059	.072	.099
.7	.035	.051	.057	.062	.078	.018	.053	.059	.077	.112
.6	.030	.050	.055	.063	.082	.017	.055	.061	.080	.117
.5	.038	.057	.062	.069	.088	.012	.058	.065	.088	.133
.4	.029	.056	.064	.074	.101	.004	.061	.071	.100	.157
.3	.032	.068	.079	.092	.128	0	.070	.083	.120	.195
.2	0	.086	.115	.162	.276	0	.090	.120	.183	.321
NMAE										
.9	0	.008	.009	.014	.023	0	.008	.011	.017	.030
.8	0	.008	.010	.014	.024	0	.008	.012	.017	.030
.7	0	.008	.010	.014	.023	0	.008	.012	.018	.033
.6	0	.008	.010	.014	.024	0	.009	.012	.019	.035
.5	0	.009	.011	.016	.027	0	.009	.014	.021	.038
.4	0	.010	.011	.017	.028	0	.010	.015	.023	.042
.3	0	.012	.014	.020	.034	0	.012	.018	.027	.049
.2	0	.018	.023	.031	.051	0	.018	.025	.037	.065
NMSE										
.9	.001	.002	.003	.003	.005	0	.003	.003	.005	.008
.8	9.8e-4	.003	.003	.004	.005	0	.003	.003	.005	.009
.7	7.5e-4	.003	.003	.004	.006	0	.003	.003	.006	.010
.6	3.0e-4	.002	.003	.004	.006	0	.003	.004	.006	.011
.5	8.6e-4	.003	.004	.005	.007	0	.003	.004	.008	.014
.4	0	.003	.004	.006	.009	0	.004	.005	.010	.019
.3	0	.005	.006	.009	.014	0	.005	.007	.014	.029
.2	0	.007	.013	.026	.055	0	.008	.014	.033	.071
SSIM										
.9	.979	.990	.994	.997	1.008	.953	.980	.992	.998	1.025
.8	.978	.989	.994	.997	1.008	.952	.979	.992	.998	1.025
.7	.978	.989	.994	.997	1.008	.949	.978	.991	.998	1.027
.6	.979	.990	.994	.997	1.007	.942	.975	.990	.997	1.031
.5	.972	.987	.992	.996	1.010	.934	.972	.989	.997	1.035
.4	.974	.987	.992	.995	1.009	.920	.966	.987	.996	1.042
.3	.958	.979	.987	.993	1.015	.890	.953	.981	.994	1.057
.2	.911	.954	.973	.984	1.027	.821	.921	.963	.987	1.087

Table A.18: OMMP accuracy (Gaussian sensing / DWT dictionary)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
TIME					
.9	0	.053	.076	.167	.338
.8	0	.054	.072	.135	.255
.7	0	.051	.068	.137	.266
.6	0	.048	.064	.128	.247
.5	0	.041	.053	.107	.208
.4	0	.035	.045	.079	.145
.3	0	.028	.035	.051	.085
.2	.002	.021	.026	.034	.054
ITERS					
.9	1.0	30.2	37	49.7	79.0
.8	5.5	31	37	48	73.5
.7	5.5	31	37	48	73.5
.6	7	31	37	47	71
.5	9	30	36	44	65
.4	12	30	35	42	60
.3	14	29	33	39	54
.2	19.5	30	33	37	47.5

Table A.19: OMMP computations (Gaussian sensing / DWT dictionary)

A.10 SOMP / Gaussian / DCT

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
					PRD					
.9	0	.035	.050	.062	.103	.032	.055	.062	.071	.095
.8	0	.038	.055	.066	.110	.030	.057	.064	.075	.103
.7	0	.040	.056	.068	.110	.024	.057	.064	.078	.111
.6	0	.045	.061	.076	.123	.016	.060	.069	.089	.132
.5	0	.054	.074	.101	.171	0	.064	.078	.113	.187
.4	0	.072	.107	.225	.455	0	.076	.107	.208	.407
.3	0	.137	.239	.416	.834	0	.119	.201	.413	.853
.2	0	.279	.431	.625	1.144	0	.240	.374	.631	1.217
					NMAE					
.9	0	.007	.010	.013	.021	0	.008	.011	.018	.032
.8	0	.008	.011	.014	.022	0	.009	.012	.019	.034
.7	0	.008	.011	.014	.023	0	.009	.013	.019	.034
.6	0	.009	.012	.016	.026	0	.010	.015	.021	.038
.5	0	.011	.015	.020	.033	0	.011	.018	.026	.048
.4	0	.017	.025	.038	.070	0	.017	.027	.042	.080
.3	0	.034	.054	.073	.131	0	.032	.055	.074	.137
.2	.020	.073	.092	.108	.161	.007	.067	.091	.106	.166
					NMSE					
.9	0	.001	.003	.004	.008	7.6e-5	.003	.004	.005	.008
.8	0	.001	.003	.004	.009	0	.003	.004	.006	.009
.7	0	.002	.003	.005	.009	0	.003	.004	.006	.010
.6	0	.002	.004	.006	.012	0	.004	.005	.008	.014
.5	0	.003	.005	.010	.021	0	.004	.006	.013	.026
.4	0	.005	.011	.051	.119	0	.006	.011	.043	.100
.3	0	.019	.057	.173	.404	0	.014	.040	.170	.405
.2	0	.078	.186	.391	.860	0	.058	.140	.398	.908
					SSIM					
.9	.983	.992	.995	.998	1.007	.954	.980	.992	.998	1.024
.8	.981	.991	.995	.998	1.008	.949	.978	.991	.997	1.026
.7	.979	.990	.995	.997	1.008	.951	.979	.990	.997	1.025
.6	.973	.987	.993	.997	1.010	.937	.973	.987	.996	1.032
.5	.955	.979	.989	.995	1.019	.904	.959	.983	.995	1.050
.4	.829	.925	.972	.989	1.085	.720	.881	.961	.988	1.148
.3	.410	.741	.871	.961	1.292	.250	.676	.833	.960	1.387
.2	0	.475	.653	.798	1.283	0	.404	.561	.848	1.514

Table A.20: SOMP accuracy (Gaussian sensing / DCT dictionary)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
			TIME		
.9	0	.156	.240	.474	.952
.8	0	.136	.210	.422	.852
.7	0	.130	.201	.382	.761
.6	0	.108	.169	.330	.663
.5	0	.093	.152	.299	.607
.4	0	.079	.132	.242	.488
.3	0	.057	.087	.133	.246
.2	.006	.031	.038	.048	.073
			ITERS		
.9	0	93	115	163	268
.8	0	91.2	113	162	268.1
.7	0	94	114	162.7	265.9
.6	0	92	113	159	259.5
.5	0	90	109	155	252.5
.4	0	88	110	152.7	249.9
.3	17.9	85	106	129.7	196.9
.2	41	71	81	91	121

Table A.21: SOMP computations (Gaussian sensing / DCT dictionary)

A.11 SOMP / Gaussian / DFT

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
PRD										
.9	0	.038	.056	.064	.103	.021	.052	.062	.072	.102
.8	0	.040	.057	.067	.108	.018	.053	.063	.077	.112
.7	0	.044	.063	.074	.118	.018	.057	.069	.084	.123
.6	0	.047	.065	.078	.126	.011	.059	.070	.092	.140
.5	0	.055	.074	.092	.148	.002	.065	.079	.107	.171
.4	0	.062	.086	.126	.221	0	.069	.089	.133	.230
.3	0	.089	.152	.259	.514	0	.097	.142	.226	.421
.2	0	.176	.312	.458	.882	0	.160	.254	.434	.844
NMAE										
.9	0	.007	.010	.012	.020	0	.008	.011	.016	.027
.8	0	.007	.010	.013	.021	0	.009	.012	.017	.028
.7	0	.008	.011	.015	.024	0	.010	.013	.019	.032
.6	0	.009	.012	.015	.025	0	.010	.014	.020	.034
.5	0	.010	.014	.019	.031	0	.012	.016	.024	.041
.4	0	.013	.017	.023	.038	0	.014	.019	.028	.050
.3	0	.022	.032	.044	.078	0	.022	.032	.046	.083
.2	0	.045	.064	.080	.133	0	.041	.062	.078	.133
NMSE										
.9	0	.001	.003	.004	.008	0	.003	.004	.005	.009
.8	0	.002	.003	.005	.009	0	.003	.004	.006	.010
.7	0	.002	.004	.005	.011	0	.003	.005	.007	.013
.6	0	.002	.004	.006	.012	0	.004	.005	.008	.016
.5	0	.003	.005	.009	.017	0	.004	.006	.012	.022
.4	0	.004	.007	.016	.034	0	.005	.008	.018	.037
.3	0	.008	.023	.067	.156	0	.009	.020	.051	.114
.2	0	.031	.097	.210	.479	0	.026	.065	.188	.432
SSIM										
.9	.980	.991	.995	.998	1.008	.961	.983	.993	.997	1.018
.8	.978	.990	.995	.998	1.009	.960	.982	.992	.997	1.018
.7	.973	.987	.994	.997	1.011	.949	.977	.991	.996	1.024
.6	.970	.986	.993	.997	1.012	.948	.977	.989	.995	1.024
.5	.957	.980	.990	.995	1.018	.924	.966	.985	.994	1.036
.4	.934	.970	.986	.993	1.029	.892	.952	.980	.992	1.051
.3	.752	.891	.952	.984	1.122	.623	.838	.941	.982	1.197
.2	.250	.648	.808	.914	1.312	.032	.574	.773	.936	1.477

Table A.22: SOMP accuracy (Gaussian sensing / DFT dictionary)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
TIME					
.9	0	.543	.867	1.777	3.628
.8	0	.443	.749	1.455	2.974
.7	0	.372	.596	1.093	2.174
.6	0	.365	.592	.900	1.703
.5	0	.269	.419	.634	1.181
.4	0	.229	.321	.475	.845
.3	0	.140	.205	.290	.516
.2	0	.061	.078	.102	.164
ITERS					
.9	0	105	139	199	340
.8	0	104	138	191	321.5
.7	0	99	131	181	304
.6	0	101	131	170	273.5
.5	6	96	123	156	246
.4	22.5	96	118	145	218.5
.3	28.5	81	97	116	168.5
.2	30.5	62	72	83	114.5

Table A.23: SOMP computations (Gaussian sensing / DFT dictionary)

A.12 SOMP / Gaussian / DWT

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
					PRD					
.9	0	.033	.046	.057	.092	.015	.050	.056	.072	.107
.8	2.4e-4	.035	.047	.058	.092	.013	.050	.056	.075	.112
.7	3.6e-4	.036	.049	.060	.096	.010	.051	.058	.079	.120
.6	.004	.040	.053	.064	.100	.010	.054	.061	.084	.129
.5	.004	.042	.054	.067	.106	1.7e-5	.054	.062	.090	.144
.4	.005	.048	.061	.077	.120	0	.060	.070	.103	.167
.3	0	.055	.069	.092	.147	0	.065	.077	.120	.202
.2	0	.078	.108	.168	.304	0	.085	.115	.184	.334
					NMAE					
.9	0	.006	.009	.012	.021	0	.008	.011	.017	.030
.8	0	.006	.009	.012	.021	0	.008	.012	.017	.031
.7	0	.007	.009	.013	.022	0	.008	.012	.018	.033
.6	0	.007	.010	.014	.023	0	.009	.013	.020	.036
.5	0	.008	.010	.014	.024	0	.009	.013	.020	.037
.4	0	.009	.011	.016	.027	0	.010	.016	.023	.042
.3	0	.011	.013	.018	.029	0	.012	.017	.026	.047
.2	0	.017	.022	.030	.050	0	.017	.025	.038	.070
					NMSE					
.9	0	.001	.002	.003	.006	0	.002	.003	.005	.009
.8	0	.001	.002	.003	.006	0	.002	.003	.006	.010
.7	0	.001	.002	.004	.007	0	.003	.003	.006	.011
.6	0	.002	.003	.004	.008	0	.003	.004	.007	.013
.5	0	.002	.003	.005	.009	0	.003	.004	.008	.016
.4	0	.002	.004	.006	.011	0	.004	.005	.011	.021
.3	0	.003	.005	.008	.017	0	.004	.006	.014	.029
.2	0	.006	.012	.028	.062	0	.007	.013	.034	.074
					SSIM					
.9	.983	.992	.996	.998	1.007	.959	.982	.992	.998	1.021
.8	.983	.992	.996	.998	1.007	.954	.980	.992	.998	1.024
.7	.982	.992	.995	.998	1.008	.950	.979	.991	.998	1.026
.6	.979	.990	.995	.998	1.009	.940	.974	.990	.997	1.032
.5	.978	.990	.995	.997	1.009	.936	.972	.989	.997	1.033
.4	.974	.987	.993	.996	1.010	.919	.965	.986	.996	1.042
.3	.967	.983	.991	.995	1.011	.902	.958	.983	.995	1.050
.2	.919	.959	.977	.986	1.026	.799	.912	.966	.988	1.101

Table A.24: SOMP accuracy (Gaussian sensing / DWT dictionary)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
			TIME		
.9	0	.103	.157	.466	1.011
.8	0	.101	.175	.500	1.099
.7	0	.081	.161	.461	1.032
.6	0	.070	.117	.307	.663
.5	0	.062	.105	.241	.510
.4	0	.050	.079	.162	.330
.3	0	.038	.054	.090	.169
.2	.002	.030	.037	.048	.076
			ITERS		
.9	0	67	92	159.7	298.9
.8	0	67	91	155	287
.7	0	67	89	147	267
.6	0	65	85	133.7	236.9
.5	0	66	87	129	223.5
.4	0	64	82	115	191.5
.3	1.9	63	79	103.7	164.9
.2	25.5	63	74	88	125.5

Table A.25: SOMP computations (Gaussian sensing / DWT dictionary)

A.13 SOMMP / Gaussian / DCT

	Channel 1					Channel 2				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**	**LOW**	**.25**	**MED**	**.75**	**HIGH**
					PRD					
.9	0	.034	.048	.060	.098	.027	.052	.059	.069	.094
.8	0	.036	.050	.063	.103	.024	.053	.060	.072	.100
.7	0	.040	.055	.067	.108	.021	.055	.064	.078	.112
.6	0	.046	.063	.077	.124	.015	.060	.070	.090	.135
.5	0	.054	.075	.104	.179	0	.065	.079	.113	.186
.4	0	.070	.108	.217	.437	0	.073	.107	.202	.397
.3	0	.130	.234	.394	.791	0	.115	.202	.399	.823
.2	0	.282	.431	.597	1.069	0	.242	.377	.598	1.133
					NMAE					
.9	0	.007	.009	.012	.020	0	.008	.011	.017	.030
.8	0	.007	.010	.013	.021	0	.008	.011	.017	.030
.7	0	.008	.011	.014	.023	0	.009	.013	.019	.033
.6	0	.009	.013	.016	.026	0	.010	.015	.022	.041
.5	0	.012	.015	.020	.033	0	.011	.018	.026	.048
.4	0	.017	.024	.036	.065	0	.017	.027	.041	.077
.3	0	.032	.051	.068	.122	0	.031	.053	.072	.133
.2	.021	.072	.089	.105	.155	.010	.066	.087	.104	.160
					NMSE					
.9	0	.001	.002	.004	.007	0	.003	.003	.005	.008
.8	0	.001	.003	.004	.008	0	.003	.004	.005	.009
.7	0	.002	.003	.005	.009	0	.003	.004	.006	.011
.6	0	.002	.004	.006	.012	0	.004	.005	.008	.015
.5	0	.003	.006	.011	.023	0	.004	.006	.013	.026
.4	0	.005	.012	.047	.110	0	.005	.011	.041	.095
.3	0	.017	.055	.156	.363	0	.013	.041	.159	.377
.2	0	.080	.186	.356	.771	0	.059	.142	.358	.807
					SSIM					
.9	.985	.993	.996	.998	1.006	.960	.983	.993	.998	1.020
.8	.983	.992	.996	.998	1.007	.959	.982	.992	.997	1.021
.7	.980	.990	.995	.997	1.008	.952	.979	.990	.997	1.024
.6	.971	.986	.993	.997	1.012	.931	.970	.987	.996	1.036
.5	.954	.978	.989	.995	1.019	.897	.956	.981	.995	1.053
.4	.848	.933	.973	.989	1.074	.733	.886	.962	.989	1.142
.3	.451	.758	.881	.964	1.271	.256	.680	.847	.963	1.387
.2	0	.475	.651	.802	1.294	0	.408	.571	.844	1.497

Table A.26: SOMMP accuracy (Gaussian sensing / DCT dictionary)

	Channel 1+2				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**
			TIME		
.9	0	.061	.088	.154	.293
.8	0	.056	.078	.135	.255
.7	0	.049	.075	.126	.242
.6	0	.044	.059	.105	.197
.5	0	.039	.055	.087	.160
.4	0	.034	.047	.077	.142
.3	0	.028	.037	.051	.085
.2	.007	.017	.020	.024	.034
			ITERS		
.9	0	24	30	42.7	70.9
.8	0	24	30	42	69
.7	0	24	29	41	66.5
.6	0	23	28	40	65.5
.5	0	23	28	40	65.5
.4	0	23	29	39	63
.3	6.5	23	28	34	50.5
.2	11.5	19	22	24	31.5

Table A.27: SOMMP computations (Gaussian sensing / DCT dictionary)

A.14 SOMMP / Gaussian / DFT

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
PRD										
.9	0	.037	.055	.064	.105	.019	.051	.061	.072	.104
.8	0	.039	.056	.066	.106	.014	.051	.062	.076	.113
.7	0	.042	.059	.070	.113	.012	.054	.064	.082	.124
.6	0	.046	.063	.077	.123	.006	.057	.068	.090	.141
.5	0	.054	.071	.093	.151	0	.063	.077	.108	.175
.4	0	.063	.088	.137	.248	0	.070	.092	.145	.258
.3	0	.087	.163	.316	.660	0	.093	.153	.273	.542
.2	0	.199	.364	.559	1.098	0	.180	.308	.531	1.057
NMAE										
.9	0	.007	.010	.013	.021	0	.008	.011	.016	.028
.8	0	.007	.010	.013	.022	0	.009	.012	.017	.029
.7	0	.008	.011	.014	.024	0	.009	.013	.018	.032
.6	0	.009	.012	.015	.026	0	.010	.014	.020	.035
.5	0	.011	.014	.019	.031	0	.012	.016	.024	.044
.4	0	.013	.018	.025	.042	0	.014	.020	.030	.055
.3	0	.021	.036	.053	.101	0	.023	.036	.055	.102
.2	0	.052	.076	.097	.165	0	.048	.075	.097	.170
NMSE										
.9	0	.001	.003	.004	.008	0	.003	.004	.005	.009
.8	0	.002	.003	.004	.009	0	.003	.004	.006	.011
.7	0	.002	.003	.005	.010	0	.003	.004	.007	.012
.6	0	.002	.004	.006	.012	0	.003	.005	.008	.016
.5	0	.003	.005	.009	.017	0	.004	.006	.012	.023
.4	0	.004	.008	.019	.041	0	.005	.008	.021	.045
.3	0	.008	.027	.100	.239	0	.009	.023	.074	.173
.2	0	.040	.133	.312	.721	0	.033	.095	.282	.656
SSIM										
.9	.980	.991	.995	.998	1.008	.961	.983	.993	.997	1.019
.8	.979	.990	.995	.998	1.009	.960	.982	.992	.997	1.018
.7	.975	.988	.994	.997	1.011	.957	.980	.991	.996	1.020
.6	.971	.986	.993	.997	1.012	.945	.976	.989	.996	1.026
.5	.958	.980	.991	.995	1.018	.922	.966	.985	.994	1.038
.4	.929	.967	.985	.993	1.031	.883	.948	.979	.991	1.056
.3	.670	.858	.947	.984	1.172	.571	.816	.940	.980	1.225
.2	.048	.562	.765	.905	1.419	0	.476	.720	.923	1.593

Table A.28: SOMMP accuracy (Gaussian sensing / DFT dictionary)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
TIME					
.9	0	.129	.208	.394	.791
.8	0	.153	.245	.496	1.010
.7	0	.134	.216	.388	.768
.6	0	.109	.171	.305	.598
.5	0	.092	.145	.240	.463
.4	0	.081	.121	.178	.322
.3	0	.069	.103	.158	.291
.2	1.7e-4	.037	.048	.062	.099
ITERS					
.9	0	27	36	52	89.5
.8	0	27	36	50	84.5
.7	0	27	35	48.7	81.4
.6	0	26	34	46	76
.5	0.5	26	33	43	68.5
.4	2.5	25	32	40	62.5
.3	8.5	25	30	36	52.5
.2	12	21	24	27	36

Table A.29: SOMMP computations (Gaussian sensing / DFT dictionary)

A.15 SOMMP / Gaussian / DWT

	Channel 1					Channel 2				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**	**LOW**	**.25**	**MED**	**.75**	**HIGH**
					PRD					
.9	0	.032	.043	.055	.090	.017	.048	.055	.068	.098
.8	0	.033	.045	.057	.093	.012	.048	.056	.072	.108
.7	0	.035	.046	.058	.094	.010	.049	.057	.074	.113
.6	.004	.037	.048	.060	.093	.009	.050	.058	.078	.119
.5	.003	.042	.054	.068	.108	.003	.055	.065	.089	.141
.4	.002	.044	.057	.072	.114	0	.056	.066	.098	.161
.3	0	.049	.065	.089	.148	0	.060	.074	.115	.197
.2	0	.079	.106	.162	.286	0	.088	.119	.189	.340
					NMAE					
.9	0	.006	.008	.011	.019	0	.007	.011	.016	.029
.8	0	.006	.008	.012	.021	0	.008	.011	.016	.029
.7	0	.006	.009	.012	.021	0	.008	.012	.017	.032
.6	0	.007	.009	.012	.020	0	.008	.012	.018	.033
.5	0	.008	.010	.014	.024	0	.009	.013	.020	.037
.4	0	.008	.011	.014	.024	0	.010	.014	.021	.038
.3	0	.010	.013	.017	.027	0	.011	.016	.025	.045
.2	0	.016	.022	.030	.050	0	.017	.025	.038	.069
					NMSE					
.9	0	.001	.002	.003	.006	0	.002	.003	.005	.008
.8	0	.001	.002	.003	.007	0	.002	.003	.005	.010
.7	0	.001	.002	.003	.007	0	.002	.003	.006	.010
.6	0	.001	.002	.004	.007	0	.003	.003	.006	.011
.5	0	.002	.003	.005	.009	0	.003	.004	.008	.015
.4	0	.002	.003	.005	.010	0	.003	.004	.010	.019
.3	0	.002	.004	.008	.016	0	.004	.005	.013	.028
.2	0	.006	.011	.026	.056	0	.008	.014	.036	.078
					SSIM					
.9	.984	.993	.996	.999	1.007	.961	.983	.993	.998	1.020
.8	.983	.992	.996	.998	1.008	.958	.982	.992	.998	1.022
.7	.983	.992	.996	.998	1.007	.952	.980	.992	.998	1.025
.6	.983	.992	.996	.998	1.007	.948	.978	.992	.998	1.027
.5	.977	.989	.994	.997	1.009	.933	.972	.989	.997	1.035
.4	.977	.989	.994	.997	1.008	.933	.971	.988	.997	1.035
.3	.972	.986	.992	.995	1.009	.916	.963	.986	.995	1.043
.2	.913	.957	.978	.987	1.031	.802	.913	.965	.988	1.099

Table A.30: SOMMP accuracy (Gaussian sensing / DWT dictionary)

	Channel 1+2				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**
			TIME		
.9	0	.042	.064	.169	.360
.8	0	.038	.060	.153	.326
.7	0	.035	.051	.125	.260
.6	0	.031	.046	.121	.256
.5	0	.028	.038	.078	.155
.4	0	.024	.034	.068	.133
.3	0	.021	.028	.055	.105
.2	0	.015	.019	.030	.052
			ITERS		
.9	0	17	23	35	62
.8	0	17	22	36	64.5
.7	0	17	22	34	59.5
.6	0	17	22	33	57
.5	0	16	21	31	53.5
.4	0	16	21	29	48.5
.3	2	17	21	27	42
.2	8	17	19	23	32

Table A.31: SOMMP computations (Gaussian sensing / DWT dictionary)

A.16 SOMMP / Binary / DWT

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
PRD										
.9	0	.285	.786	.981	2.026	0	.264	.828	1.040	2.205
.8	0	.285	.808	.986	2.037	0	.273	.839	1.036	2.182
.7	0	.293	.804	.991	2.038	0	.271	.860	1.049	2.217
.6	0	.298	.833	1.003	2.061	0	.290	.868	1.067	2.232
.5	0	.307	.829	1.012	2.069	0	.301	.868	1.068	2.217
.4	0	.341	.856	1.019	2.036	0	.326	.879	1.071	2.188
.3	0	.357	.941	1.216	2.505	0	.359	.967	1.268	2.633
.2	0	.433	.979	1.240	2.450	0	.427	1.022	1.281	2.562
NMAE										
.9	0	.035	.151	.299	.695	0	.029	.147	.381	.909
.8	0	.036	.156	.294	.680	0	.029	.147	.375	.894
.7	0	.039	.157	.299	.690	0	.029	.150	.380	.906
.6	0	.039	.154	.308	.712	0	.031	.155	.388	.924
.5	0	.040	.161	.303	.699	0	.031	.154	.379	.901
.4	0	.045	.163	.309	.706	0	.035	.155	.385	.909
.3	0	.049	.177	.361	.830	0	.039	.173	.438	1.037
.2	0	.061	.194	.366	.823	0	.048	.188	.451	1.055
NMSE										
.9	0	.081	.618	.963	2.285	0	.070	.686	1.082	2.601
.8	0	.081	.653	.972	2.307	0	.074	.703	1.074	2.573
.7	0	.086	.647	.982	2.325	0	.073	.739	1.101	2.643
.6	0	.089	.694	1.006	2.383	0	.084	.753	1.139	2.720
.5	0	.094	.688	1.023	2.417	0	.091	.753	1.140	2.714
.4	0	.117	.733	1.039	2.422	0	.107	.772	1.148	2.709
.3	0	.128	.885	1.480	3.508	0	.129	.935	1.609	3.829
.2	0	.188	.958	1.537	3.562	0	.182	1.045	1.641	3.830
SSIM										
.9	0	.164	.436	.934	2.088	0	.062	.396	.961	2.309
.8	0	.144	.420	.933	2.117	0	.068	.390	.958	2.294
.7	0	.150	.441	.926	2.091	0	.070	.363	.957	2.288
.6	0	.147	.409	.920	2.080	0	.066	.359	.951	2.280
.5	0	.144	.428	.923	2.093	0	.067	.351	.948	2.269
.4	0	.129	.389	.906	2.072	0	.054	.342	.939	2.267
.3	0	.086	.323	.889	2.094	0	.029	.278	.922	2.262
.2	0	.076	.263	.845	1.998	0	.034	.230	.897	2.193

Table A.32: SOMMP accuracy (Binary sensing / DWT dictionary)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
TIME					
.9	.007	.019	.021	.027	.039
.8	.007	.018	.020	.024	.035
.7	.005	.016	.019	.023	.034
.6	.006	.015	.017	.021	.030
.5	.007	.014	.015	.019	.026
.4	.008	.013	.014	.016	.021
.3	.004	.010	.012	.014	.021
.2	.004	.008	.010	.011	.016
ITERS					
.9	0	5	6	11	20
.8	0	5	5	10	17.5
.7	0	5	5	10	17.5
.6	0	5	5	11	20
.5	0	5	5	10	17.5
.4	0	5	5	10	17.5
.3	0	4	5	10	19
.2	0	4	5	9	16.5

Table A.33: SOMMP computations (Binary sensing / DWT dictionary)

A.17 GAP / Gaussian / 1st Order Derivative

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
PRD										
.9	0	.008	.012	.018	.032	0	.010	.015	.023	.043
.8	0	.009	.015	.021	.040	0	.013	.019	.028	.052
.7	0	.012	.019	.028	.052	0	.016	.024	.036	.066
.6	0	.017	.025	.035	.064	0	.020	.031	.045	.082
.5	0	.020	.032	.045	.082	0	.025	.040	.057	.106
.4	0	.026	.042	.062	.115	0	.033	.052	.073	.132
.3	0	.039	.060	.086	.157	0	.046	.071	.098	.175
.2	0	.061	.096	.138	.255	0	.067	.107	.145	.262
NMAE										
.9	0	.002	.002	.003	.006	0	.002	.003	.005	.009
.8	0	.002	.003	.004	.006	0	.003	.004	.006	.010
.7	0	.003	.004	.005	.009	0	.003	.004	.007	.013
.6	0	.004	.005	.007	.011	0	.004	.006	.009	.016
.5	0	.004	.006	.008	.015	0	.005	.007	.011	.020
.4	0	.006	.008	.011	.018	0	.007	.009	.014	.025
.3	0	.008	.011	.015	.024	0	.009	.013	.018	.031
.2	8.3e-4	.013	.017	.022	.034	0	.014	.018	.025	.042
NMSE										
.9	0	6.9e-5	1.5e-4	3.1e-4	6.8e-4	0	1.0e-4	2.3e-4	5.4e-4	.001
.8	0	7.8e-5	2.2e-4	4.5e-4	.001	0	1.6e-4	3.8e-4	8.0e-4	.002
.7	0	1.4e-4	3.7e-4	7.7e-4	.002	0	2.4e-4	5.8e-4	.001	.003
.6	0	2.8e-4	6.1e-4	.001	.003	0	4.0e-4	9.5e-4	.002	.004
.5	0	4.1e-4	.001	.002	.004	0	6.2e-4	.002	.003	.007
.4	0	6.8e-4	.002	.004	.008	0	.001	.003	.005	.012
.3	0	.002	.004	.007	.016	0	.002	.005	.010	.021
.2	0	.004	.009	.019	.042	0	.004	.011	.021	.046
SSIM										
.9	.999	1	1	1	1.000	.997	.999	1	1	1.001
.8	.999	.999	1	1	1.000	.996	.998	.999	1	1.002
.7	.998	.999	.999	1	1.001	.994	.997	.999	1	1.003
.6	.997	.998	.999	.999	1.001	.991	.996	.998	.999	1.004
.5	.995	.997	.998	.999	1.002	.985	.993	.997	.999	1.007
.4	.991	.995	.997	.998	1.002	.976	.989	.995	.998	1.011
.3	.984	.991	.994	.996	1.003	.956	.980	.992	.996	1.021
.2	.964	.979	.985	.990	1.005	.912	.959	.982	.991	1.038

Table A.34: GAP accuracy (Gaussian sensing / 1st Order Derivative)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
TIME					
.9	.093	.178	.201	.234	.319
.8	.085	.165	.190	.218	.297
.7	.079	.163	.185	.219	.303
.6	.090	.168	.190	.220	.298
.5	.104	.170	.191	.214	.280
.4	.095	.180	.206	.237	.322
.3	.086	.174	.202	.232	.319
.2	.087	.171	.195	.227	.311
ITERS					
.9	1.5	6	7	9	13.5
.8	1.5	6	7	9	13.5
.7	1.5	6	7	9	13.5
.6	3	6	7	8	11
.5	3	6	7	8	11
.4	1.5	6	7	9	13.5
.3	3	6	7	8	11
.2	1.5	6	7	9	13.5

Table A.35: GAP computations (Gaussian sensing / 1st Order Derivative)

A.18 GAP / Gaussian / 2nd Order Derivative

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
PRD										
.9	0	.005	.009	.013	.024	0	.008	.013	.021	.039
.8	0	.006	.010	.015	.029	0	.009	.015	.024	.047
.7	0	.007	.013	.018	.035	0	.011	.018	.029	.056
.6	0	.009	.015	.023	.044	0	.013	.022	.036	.072
.5	0	.011	.019	.029	.057	0	.016	.027	.047	.094
.4	0	.014	.026	.040	.080	0	.020	.035	.064	.131
.3	0	.022	.040	.059	.115	0	.028	.048	.086	.174
.2	0	.047	.067	.100	.179	0	.050	.079	.135	.261
NMAE										
.9	0	.001	.002	.002	.004	0	.002	.003	.005	.010
.8	0	.002	.002	.003	.005	0	.002	.003	.006	.011
.7	0	.002	.003	.003	.005	0	.003	.003	.007	.013
.6	0	.002	.003	.004	.006	0	.003	.004	.008	.015
.5	0	.003	.004	.005	.008	0	.004	.005	.009	.018
.4	0	.004	.005	.007	.011	0	.005	.007	.012	.023
.3	9.2e-5	.006	.007	.009	.015	0	.007	.009	.016	.030
.2	2.6e-4	.010	.013	.017	.027	0	.011	.016	.025	.046
NMSE										
.9	0	2.8e-5	7.9e-5	1.6e-4	3.7e-4	0	6.4e-5	1.7e-4	4.2e-4	9.6e-4
.8	0	3.7e-5	1.1e-4	2.3e-4	5.1e-4	0	8.2e-5	2.3e-4	5.8e-4	.001
.7	0	4.9e-5	1.6e-4	3.3e-4	7.5e-4	0	1.1e-4	3.1e-4	8.3e-4	.002
.6	0	7.5e-5	2.3e-4	5.2e-4	.001	0	1.6e-4	4.7e-4	.001	.003
.5	0	1.1e-4	3.8e-4	8.5e-4	.002	0	2.4e-4	7.1e-4	.002	.005
.4	0	2.0e-4	6.9e-4	.002	.004	0	4.1e-4	.001	.004	.010
.3	0	4.7e-4	.002	.003	.008	0	7.8e-4	.002	.007	.017
.2	0	.002	.004	.010	.022	0	.003	.006	.018	.041
SSIM										
.9	1	1	1	1	1.000	.998	.999	1	1	1.001
.8	.999	1	1	1	1.000	.997	.999	1	1	1.002
.7	.999	1	1	1	1.000	.996	.998	.999	1	1.002
.6	.999	.999	1	1	1.000	.993	.997	.999	1	1.003
.5	.998	.999	.999	1	1.001	.990	.996	.999	.999	1.005
.4	.996	.998	.999	.999	1.001	.984	.993	.998	.999	1.008
.3	.991	.996	.998	.999	1.003	.971	.987	.996	.998	1.014
.2	.973	.986	.992	.995	1.009	.919	.964	.985	.995	1.041

Table A.36: GAP accuracy (Gaussian sensing / 2nd Order Derivative)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
TIME					
.9	.109	.176	.196	.221	.288
.8	.095	.168	.188	.216	.289
.7	.102	.164	.183	.205	.266
.6	.105	.173	.195	.219	.288
.5	.103	.178	.202	.227	.302
.4	.102	.169	.192	.215	.282
.3	.096	.171	.193	.221	.296
.2	.097	.163	.185	.208	.275
ITERS					
.9	3	6	7	8	11
.8	3	6	7	8	11
.7	3	6	7	8	11
.6	3	6	7	8	11
.5	3	6	7	8	11
.4	3	6	7	8	11
.3	3	6	7	8	11
.2	2	5	6	7	10

Table A.37: GAP computations (Gaussian sensing / 2nd Order Derivative)

A.19 GAP / Gaussian / Wavelet

	Channel 1					Channel 2				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**	**LOW**	**.25**	**MED**	**.75**	**HIGH**
					PRD					
.9	0	.015	.030	.050	.102	0	.021	.038	.058	.113
.8	0	.028	.051	.080	.158	0	.030	.053	.078	.150
.7	0	.036	.068	.104	.207	0	.035	.061	.092	.177
.6	0	.033	.064	.112	.231	0	.042	.076	.119	.233
.5	0	.036	.069	.137	.288	0	.049	.083	.148	.297
.4	0	.032	.062	.115	.239	0	.062	.118	.216	.446
.3	0	.052	.092	.167	.339	0	.059	.102	.171	.339
.2	0	.102	.162	.254	.481	0	.133	.223	.370	.725
					NMAE					
.9	0	.003	.005	.011	.022	0	.004	.007	.014	.029
.8	0	.005	.009	.019	.042	0	.005	.010	.018	.039
.7	0	.006	.011	.026	.055	0	.005	.011	.023	.049
.6	0	.006	.010	.023	.049	0	.006	.013	.028	.061
.5	0	.006	.011	.027	.059	0	.007	.014	.033	.073
.4	0	.007	.011	.018	.036	0	.009	.019	.051	.115
.3	0	.010	.016	.028	.056	0	.011	.018	.036	.075
.2	0	.019	.028	.045	.084	0	.021	.038	.082	.175
					NMSE					
.9	0	2.3e-4	8.7e-4	.003	.006	0	4.6e-4	.001	.003	.008
.8	0	7.6e-4	.003	.006	.015	0	9.0e-4	.003	.006	.014
.7	0	.001	.005	.011	.025	0	.001	.004	.008	.019
.6	0	.001	.004	.013	.030	0	.002	.006	.014	.033
.5	0	.001	.005	.019	.045	0	.002	.007	.022	.051
.4	0	.001	.004	.013	.031	0	.004	.014	.046	.111
.3	0	.003	.008	.028	.066	0	.003	.010	.029	.068
.2	0	.010	.026	.064	.145	0	.018	.050	.137	.315
					SSIM					
.9	.988	.995	.999	1	1.007	.979	.991	.998	.999	1.012
.8	.958	.983	.996	.999	1.024	.957	.982	.996	.999	1.024
.7	.923	.969	.994	.999	1.044	.940	.975	.995	.999	1.034
.6	.935	.973	.995	.998	1.036	.902	.960	.993	.998	1.056
.5	.913	.964	.994	.998	1.049	.859	.942	.991	.998	1.081
.4	.959	.982	.995	.998	1.021	.690	.874	.981	.997	1.180
.3	.904	.959	.989	.996	1.051	.795	.915	.986	.995	1.115
.2	.783	.904	.961	.984	1.105	.284	.704	.921	.984	1.404

Table A.38: GAP accuracy (Gaussian sensing / Wavelet analysis)

	Channel 1+2				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**
			TIME		
.9	.139	.308	.363	.420	.588
.8	.070	.243	.311	.358	.531
.7	.069	.248	.319	.367	.545
.6	.102	.280	.338	.399	.577
.5	.119	.290	.348	.403	.574
.4	.139	.304	.356	.414	.579
.3	.167	.330	.377	.439	.602
.2	.161	.293	.336	.381	.512
			ITERS		
.9	1.5	9	12	14	21.5
.8	0.5	8	11	13	20.5
.7	0.5	8	11	13	20.5
.6	1.5	9	11	14	21.5
.5	1.5	9	12	14	21.5
.4	4	10	12	14	20
.3	5	11	12	15	21
.2	3	9	11	13	19

Table A.39: GAP computations (Gaussian sensing / Wavelet analysis)

A.20 SGAP / Gaussian / 1st Order Derivative

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
					PRD					
.9	0	.010	.015	.021	.038	0	.012	.018	.026	.048
.8	0	.012	.018	.025	.045	0	.014	.022	.032	.058
.7	0	.015	.022	.031	.056	0	.018	.027	.039	.071
.6	0	.019	.029	.041	.074	0	.023	.034	.050	.090
.5	0	.025	.037	.053	.094	0	.029	.044	.064	.116
.4	0	.032	.048	.069	.125	0	.039	.059	.082	.146
.3	0	.048	.071	.098	.174	0	.056	.083	.111	.193
.2	0	.072	.109	.150	.267	0	.085	.123	.167	.289
					NMAE					
.9	0	.002	.003	.004	.006	0	.002	.003	.005	.009
.8	0	.003	.003	.005	.007	0	.003	.004	.006	.011
.7	0	.003	.004	.006	.010	0	.004	.005	.008	.014
.6	0	.004	.005	.007	.012	0	.005	.006	.010	.017
.5	0	.005	.007	.010	.016	0	.006	.008	.012	.022
.4	0	.007	.009	.012	.020	0	.008	.010	.016	.028
.3	0	.010	.013	.017	.027	0	.011	.014	.021	.037
.2	3.8e-4	.015	.020	.024	.039	0	.016	.022	.032	.054
					NMSE					
.9	0	9.1e-5	2.1e-4	4.3e-4	9.5e-4	0	1.4e-4	3.2e-4	6.9e-4	.002
.8	0	1.3e-4	3.1e-4	6.3e-4	.001	0	2.0e-4	4.7e-4	.001	.002
.7	0	2.1e-4	4.8e-4	9.7e-4	.002	0	3.1e-4	7.4e-4	.002	.003
.6	0	3.8e-4	8.4e-4	.002	.004	0	5.2e-4	.001	.002	.005
.5	0	6.4e-4	.001	.003	.006	0	8.6e-4	.002	.004	.009
.4	0	.001	.002	.005	.011	0	.002	.003	.007	.014
.3	0	.002	.005	.010	.021	0	.003	.007	.012	.026
.2	0	.005	.012	.022	.048	0	.007	.015	.028	.059
					SSIM					
.9	.999	.999	1	1	1.000	.996	.998	.999	1	1.002
.8	.998	.999	.999	1	1.001	.995	.998	.999	1	1.003
.7	.997	.999	.999	.999	1.001	.992	.996	.999	.999	1.004
.6	.995	.998	.998	.999	1.002	.986	.994	.998	.999	1.007
.5	.992	.996	.998	.999	1.002	.977	.990	.996	.999	1.011
.4	.988	.994	.996	.998	1.003	.963	.984	.993	.997	1.018
.3	.976	.987	.992	.995	1.006	.929	.969	.987	.995	1.034
.2	.952	.973	.980	.986	1.007	.833	.926	.972	.988	1.081

Table A.40: SGAP accuracy (Gaussian sensing / 1st Order Derivative)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
			TIME		
.9	.039	.072	.085	.094	.128
.8	.030	.071	.089	.098	.139
.7	.030	.071	.087	.098	.139
.6	.034	.066	.075	.088	.120
.5	.031	.065	.081	.088	.123
.4	.012	.065	.083	.100	.153
.3	.026	.062	.074	.086	.122
.2	.022	.061	.070	.086	.125
			ITERS		
.9	0.5	2	2	3	4.5
.8	0.5	2	3	3	4.5
.7	0.5	2	3	3	4.5
.6	0.5	2	2	3	4.5
.5	0.5	2	2	3	4.5
.4	0.5	2	3	3	4.5
.3	0.5	2	2	3	4.5
.2	0.5	2	2	3	4.5

Table A.41: SGAP computations (Gaussian sensing / 1st Order Derivative)

A.21 SGAP / Gaussian / 2nd Order Derivative

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
					PRD					
.9	0	.005	.010	.015	.029	0	.008	.014	.022	.043
.8	0	.006	.011	.017	.032	0	.010	.016	.026	.049
.7	0	.007	.013	.020	.039	0	.011	.019	.030	.058
.6	0	.009	.016	.025	.049	0	.013	.023	.038	.075
.5	0	.011	.021	.033	.065	0	.016	.029	.049	.098
.4	0	.015	.030	.047	.094	0	.021	.038	.066	.134
.3	0	.024	.044	.069	.136	0	.031	.056	.095	.191
.2	0	.052	.076	.116	.211	0	.062	.097	.148	.278
					NMAE					
.9	0	.001	.002	.003	.004	0	.002	.003	.005	.010
.8	0	.002	.002	.003	.005	0	.002	.003	.006	.011
.7	0	.002	.003	.003	.006	0	.003	.004	.006	.012
.6	0	.002	.003	.004	.007	0	.003	.004	.008	.014
.5	0	.003	.004	.005	.008	0	.004	.005	.009	.017
.4	0	.004	.006	.007	.012	0	.005	.007	.012	.022
.3	0	.006	.008	.011	.017	0	.007	.010	.017	.032
.2	3.0e-4	.012	.015	.019	.030	0	.013	.018	.029	.055
					NMSE					
.9	0	3.0e-5	9.3e-5	2.2e-4	5.1e-4	0	7.1e-5	1.9e-4	4.9e-4	.001
.8	0	4.0e-5	1.2e-4	2.8e-4	6.4e-4	0	9.2e-5	2.6e-4	6.5e-4	.001
.7	0	5.5e-5	1.7e-4	4.0e-4	9.2e-4	0	1.2e-4	3.5e-4	9.0e-4	.002
.6	0	7.9e-5	2.6e-4	6.2e-4	.001	0	1.7e-4	5.2e-4	.001	.003
.5	0	1.3e-4	4.5e-4	.001	.002	0	2.6e-4	8.3e-4	.002	.006
.4	0	2.3e-4	9.0e-4	.002	.005	0	4.5e-4	.001	.004	.010
.3	0	5.5e-4	.002	.005	.011	0	9.5e-4	.003	.009	.021
.2	0	.003	.006	.013	.029	0	.004	.009	.022	.049
					SSIM					
.9	.999	1	1	1	1.000	.997	.999	1	1	1.001
.8	.999	1	1	1	1.000	.996	.998	1	1	1.002
.7	.999	.999	1	1	1.000	.995	.998	.999	1	1.002
.6	.998	.999	1	1	1.001	.993	.997	.999	1	1.003
.5	.997	.999	.999	1	1.001	.990	.995	.999	.999	1.005
.4	.995	.998	.999	.999	1.002	.982	.992	.997	.999	1.009
.3	.989	.995	.997	.998	1.004	.963	.984	.995	.997	1.018
.2	.965	.982	.989	.994	1.012	.887	.951	.981	.993	1.057

Table A.42: SGAP accuracy (Gaussian sensing / 2nd Order Derivative)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
			TIME		
.9	.061	.095	.102	.117	.150
.8	.053	.091	.101	.117	.155
.7	.033	.077	.097	.107	.151
.6	.057	.095	.102	.121	.160
.5	.038	.079	.097	.107	.147
.4	.031	.078	.097	.109	.155
.3	.039	.074	.086	.097	.132
.2	.035	.072	.083	.096	.133
			ITERS		
.9	3	3	3	3	3
.8	0.5	2	3	3	4.5
.7	0.5	2	3	3	4.5
.6	0	2	3	4	7
.5	0.5	2	3	3	4.5
.4	0.5	2	3	3	4.5
.3	0.5	2	2	3	4.5
.2	0.5	2	2	3	4.5

Table A.43: SGAP computations (Gaussian sensing / 2nd Order Derivative)

A.22 SGAP / Binary / 2nd Order Derivative

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
					PRD					
.9	0	.003	.006	.009	.016	0	.005	.008	.013	.024
.8	0	.005	.009	.013	.025	0	.008	.013	.019	.037
.7	0	.007	.012	.018	.033	0	.010	.017	.027	.051
.6	0	.009	.016	.024	.046	0	.013	.022	.035	.068
.5	0	.011	.021	.032	.064	0	.016	.029	.048	.096
.4	0	.015	.029	.046	.092	0	.021	.038	.067	.135
.3	0	.023	.044	.069	.138	0	.031	.057	.095	.191
.2	0	.055	.078	.115	.206	0	.064	.098	.151	.280
					NMAE					
.9	0	8.5e-4	.001	.002	.003	0	.001	.002	.003	.006
.8	0	.001	.002	.002	.004	0	.002	.003	.005	.009
.7	0	.002	.002	.003	.005	0	.002	.003	.006	.011
.6	0	.002	.003	.004	.007	0	.003	.004	.008	.014
.5	0	.003	.004	.005	.009	0	.004	.005	.010	.018
.4	0	.004	.005	.007	.012	0	.005	.007	.012	.022
.3	0	.006	.008	.011	.017	0	.007	.010	.017	.031
.2	9.6e-4	.012	.015	.019	.030	0	.013	.018	.030	.056
					NMSE					
.9	0	1.2e-5	3.2e-5	7.3e-5	1.6e-4	0	2.8e-5	7.2e-5	1.7e-4	3.7e-4
.8	0	2.7e-5	7.6e-5	1.7e-4	3.8e-4	0	6.4e-5	1.7e-4	3.8e-4	8.5e-4
.7	0	4.7e-5	1.4e-4	3.1e-4	7.0e-4	0	1.1e-4	2.9e-4	7.1e-4	.002
.6	0	7.5e-5	2.5e-4	5.6e-4	.001	0	1.7e-4	4.9e-4	.001	.003
.5	0	1.3e-4	4.6e-4	.001	.002	0	2.6e-4	8.3e-4	.002	.005
.4	0	2.3e-4	8.7e-4	.002	.005	0	4.5e-4	.001	.004	.010
.3	0	5.3e-4	.002	.005	.011	0	9.6e-4	.003	.009	.021
.2	0	.003	.006	.013	.029	0	.004	.010	.023	.050
					SSIM					
.9	1	1	1	1	1.000	.999	1	1	1	1.001
.8	.999	1	1	1	1.000	.998	.999	1	1	1.001
.7	.999	1	1	1	1.000	.996	.998	.999	1	1.002
.6	.998	.999	1	1	1.001	.993	.997	.999	1	1.003
.5	.997	.999	.999	1	1.001	.989	.995	.999	.999	1.005
.4	.995	.998	.999	.999	1.002	.982	.992	.998	.999	1.009
.3	.989	.995	.997	.998	1.004	.963	.984	.995	.997	1.018
.2	.965	.982	.989	.993	1.011	.874	.946	.979	.993	1.065

Table A.44: SGAP accuracy (Binary sensing / 2nd Order Derivative)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
			TIME		
.9	.041	.078	.094	.103	.140
.8	.040	.075	.095	.098	.133
.7	.036	.079	.097	.107	.150
.6	.036	.078	.097	.106	.148
.5	.032	.077	.095	.108	.154
.4	.034	.070	.090	.094	.131
.3	.030	.069	.089	.094	.133
.2	.034	.066	.075	.088	.120
			ITERS		
.9	0.5	2	3	3	4.5
.8	0.5	2	3	3	4.5
.7	0.5	2	3	3	4.5
.6	0.5	2	3	3	4.5
.5	0.5	2	3	3	4.5
.4	0.5	2	3	3	4.5
.3	0.5	2	3	3	4.5
.2	0.5	2	2	3	4.5

Table A.45: SGAP computations (Binary sensing / 2nd Order Derivative)

A.23 SGAP / Gaussian / Wavelet

	Channel 1					Channel 2				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
					PRD					
.9	0	.042	.056	.070	.113	0	.046	.062	.080	.131
.8	.011	.061	.079	.094	.144	.010	.067	.086	.105	.161
.7	.006	.071	.097	.115	.181	.011	.080	.105	.126	.195
.6	.009	.107	.146	.173	.271	.002	.112	.157	.185	.295
.5	0	.116	.164	.205	.339	0	.115	.176	.218	.372
.4	0	.121	.195	.255	.457	0	.127	.208	.282	.513
.3	0	.140	.223	.320	.591	0	.151	.230	.355	.662
.2	0	.216	.327	.428	.746	0	.209	.325	.460	.835
					NMAE					
.9	0	.007	.011	.018	.034	0	.008	.013	.021	.042
.8	0	.010	.016	.023	.043	0	.011	.017	.029	.057
.7	0	.011	.019	.027	.051	0	.012	.020	.035	.068
.6	0	.017	.028	.040	.074	0	.016	.028	.053	.108
.5	0	.018	.030	.044	.084	0	.018	.030	.056	.113
.4	0	.021	.035	.053	.100	0	.020	.034	.065	.133
.3	0	.024	.039	.060	.114	0	.024	.040	.070	.140
.2	0	.040	.057	.081	.143	0	.036	.058	.094	.182
					NMSE					
.9	0	.002	.003	.005	.010	0	.002	.004	.006	.013
.8	0	.004	.006	.009	.017	0	.004	.007	.011	.021
.7	0	.005	.009	.013	.026	0	.006	.011	.016	.030
.6	0	.012	.021	.030	.057	0	.013	.025	.034	.067
.5	0	.013	.027	.042	.085	0	.013	.031	.048	.099
.4	0	.015	.038	.065	.141	0	.016	.043	.079	.174
.3	0	.020	.050	.103	.227	0	.023	.053	.126	.281
.2	0	.047	.107	.183	.388	0	.044	.105	.211	.462
					SSIM					
.9	.970	.987	.994	.998	1.015	.949	.978	.992	.998	1.027
.8	.945	.976	.989	.996	1.026	.903	.959	.985	.996	1.052
.7	.923	.966	.984	.994	1.036	.859	.940	.979	.994	1.076
.6	.841	.929	.967	.988	1.077	.704	.876	.961	.991	1.163
.5	.806	.914	.961	.987	1.095	.631	.845	.955	.987	1.201
.4	.745	.887	.947	.982	1.124	.533	.804	.943	.985	1.256
.3	.643	.841	.928	.973	1.172	.422	.754	.912	.976	1.308
.2	.451	.736	.858	.927	1.212	.047	.583	.830	.940	1.476

Table A.46: SGAP accuracy (Gaussian sensing / Wavelet analysis)

	Channel 1+2				
CR	LOW	.25	MED	.75	HIGH
			TIME		
.9	.018	.086	.107	.131	.198
.8	.018	.085	.105	.129	.195
.7	.015	.084	.106	.130	.199
.6	.017	.082	.103	.125	.191
.5	.016	.086	.104	.132	.202
.4	.032	.101	.111	.147	.216
.3	.021	.101	.126	.155	.235
.2	.046	.108	.130	.150	.212
			ITERS		
.9	0	2	3	4	7
.8	0	2	3	4	7
.7	0	2	3	4	7
.6	0	2	3	4	7
.5	0	2	3	4	7
.4	0	3	3	5	8
.3	0	3	4	5	8
.2	0	3	4	5	8

Table A.47: SGAP computations (Gaussian sensing / Wavelet analysis)

Appendix B

Noise Aware Reconstruction Result Tables

This appendix contains the resulting metric values when reconstructing noisy measurements using the standard OMMP and GAP algorithms and their noise-aware versions OMMPn and GAPn. Each section is titled as follows: Algorithm Name / Type of Signals / Variable. The first term is the name of the algorithm that is used. The results of the normal and noise-aware algorithm are shown in one table. The second term indicates what type of signals were used for these results: real data from the MIT-BIH Arrhythmia database or simulated signals from the ECG simulator. The last term indicates whether the results are shown as a function of varying CR or noise level. The complete description of the experiments and the data (real and simulated) can be found in chapter 4.

B.1 OMMP / Real / Compression

	OMMP					OMMPn				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
					PRD					
.9	.097	.140	.155	.169	.213	.097	.545	.575	.617	.213
.8	.103	.159	.177	.196	.252	.103	.557	.591	.628	.252
.7	.101	.160	.178	.199	.257	.101	.460	.487	.518	.257
.6	.106	.167	.187	.208	.270	.106	.422	.446	.475	.270
.5	.115	.190	.213	.241	.316	.115	.427	.454	.481	.316
.4	.119	.211	.238	.273	.365	.119	.396	.423	.455	.365
.3	.116	.261	.301	.358	.503	.116	.406	.437	.471	.503
.2	.182	.382	.440	.515	.715	.182	.425	.471	.532	.715
					NMAE					
.9	0	.020	.027	.043	.077	0	.084	.109	.156	.077
.8	0	.023	.030	.049	.089	0	.086	.110	.158	.089
.7	0	.023	.030	.051	.092	0	.070	.091	.132	.092
.6	0	.024	.032	.053	.097	0	.064	.082	.118	.097
.5	0	.028	.037	.061	.112	0	.064	.084	.119	.112
.4	0	.031	.041	.073	.136	0	.062	.078	.110	.136
.3	0	.039	.052	.093	.175	0	.063	.079	.114	.175
.2	0	.059	.080	.132	.242	0	.070	.090	.124	.242
					NMSE					
.9	.006	.020	.024	.029	.042	.006	.297	.331	.380	.042
.8	.005	.025	.031	.038	.058	.005	.310	.350	.395	.058
.7	.005	.025	.032	.039	.060	.005	.212	.237	.269	.060
.6	.005	.028	.035	.043	.067	.005	.178	.199	.226	.067
.5	.004	.036	.045	.058	.090	.004	.182	.206	.232	.090
.4	1.2e-4	.045	.057	.074	.119	1.2e-4	.157	.179	.207	.119
.3	0	.068	.091	.128	.218	0	.165	.191	.222	.218
.2	0	.146	.193	.266	.445	0	.180	.222	.283	.445
					SSIM					
.9	.798	.909	.961	.983	1.093	.798	.445	.637	.786	1.093
.8	.737	.881	.951	.977	1.121	.737	.442	.627	.776	1.121
.7	.722	.875	.950	.977	1.131	.722	.547	.710	.835	1.131
.6	.708	.868	.946	.974	1.134	.708	.578	.742	.855	1.134
.5	.632	.833	.931	.967	1.168	.632	.574	.737	.855	1.168
.4	.531	.788	.917	.959	1.216	.531	.622	.757	.860	1.216
.3	.324	.692	.874	.938	1.307	.324	.596	.753	.858	1.307
.2	.011	.521	.742	.861	1.372	.011	.544	.703	.820	1.372
					TIME					
.9	.010	.044	.055	.067	.101	.010	1.001	1.085	1.178	.101
.8	.008	.038	.046	.058	.088	.008	.635	.678	.727	.088
.7	.003	.035	.045	.057	.089	.003	.367	.397	.432	.089
.6	.003	.032	.040	.051	.080	.003	.227	.245	.268	.080
.5	.002	.027	.034	.044	.069	.002	.127	.137	.152	.069
.4	6.1e-5	.023	.030	.039	.062	6.1e-5	.068	.073	.080	.062
.3	0	.018	.024	.034	.058	0	.032	.034	.036	.058
.2	0	.014	.018	.026	.043	0	.013	.014	.016	.043
					ITER					
.9	11	20	22	26	35	11	115	118	120	35
.8	10	19	21	25	34	10	99	101	103	34
.7	8.5	19	22	26	36.5	8.5	80	82	84	36.5
.6	8.5	19	22	26	36.5	8.5	66	68	69	36.5
.5	7.5	18	21	25	35.5	7.5	55	56	58	35.5
.4	3.5	17	21	26	39.5	3.5	43	45	46	39.5
.3	3.5	17	20	26	39.5	3.5	33	34	35	39.5
.2	1	16	19	26	41	1	23	24	25	41

Table B.1: OMMP and OMMPn metrics for varying CR for real signals

B.2 OMMP / Real / Noise

	OMMP					OMMPn				
noise	**LOW**	**.25**	**MED**	**.75**	**HIGH**	**LOW**	**.25**	**MED**	**.75**	**HIGH**
PRD										
.01	.119	.372	.463	.541	.794	.119	.052	.059	.068	.794
.02	.052	.130	.151	.181	.259	.052	.163	.177	.191	.259
.05	.110	.186	.210	.236	.312	.110	.414	.438	.464	.312
.1	.158	.240	.265	.294	.376	.158	.653	.698	.738	.376
.2	.210	.299	.326	.358	.446	.210	.900	.951	1.012	.446
NMAE										
.01	0	.037	.062	.095	.181	0	.009	.010	.015	.181
.02	0	.019	.026	.049	.094	0	.026	.033	.044	.094
.05	0	.027	.036	.061	.113	0	.063	.080	.114	.113
.1	0	.035	.045	.075	.135	0	.101	.127	.182	.135
.2	0	.044	.057	.090	.160	0	.136	.175	.251	.160
NMSE										
.01	0	.138	.214	.293	.524	0	.003	.003	.005	.524
.02	0	.017	.023	.033	.057	0	.027	.031	.036	.057
.05	.003	.034	.044	.056	.088	.003	.171	.192	.215	.088
.1	.014	.058	.070	.087	.130	.014	.426	.487	.545	.130
.2	.031	.089	.106	.128	.186	.031	.809	.905	1.024	.186
SSIM										
.01	0	.487	.693	.844	1.378	0	.989	.993	.996	1.378
.02	.759	.894	.966	.984	1.119	.759	.907	.947	.972	1.119
.05	.638	.836	.935	.968	1.166	.638	.598	.753	.859	1.166
.1	.510	.772	.895	.947	1.210	.510	.370	.542	.707	1.210
.2	.371	.699	.847	.918	1.246	.371	.234	.390	.559	1.246
TIME										
.01	.315	.406	.432	.466	.557	.315	.015	.018	.023	.557
.02	0	.045	.064	.106	.196	0	.067	.074	.083	.196
.05	0	.027	.034	.046	.074	0	.126	.136	.153	.074
.1	.005	.021	.026	.032	.048	.005	.155	.165	.185	.048
.2	.007	.020	.023	.028	.041	.007	.172	.183	.204	.041
ITER										
.01	88	91	92	93	96	88	15	18	21	96
.02	0	26	32	44	71	0	40	42	43	71
.05	6	18	21	26	38	6	55	56	57	38
.1	7.5	15	17	20	27.5	7.5	61	62	63	27.5
.2	5.5	13	15	18	25.5	5.5	64	65	66	25.5

Table B.2: OMMP and OMMPn metrics for varying noise for real signals

B.3 OMMP / Simulated / Compression

	OMMP					OMMPn				
CR	LOW	.25	MED	.75	HIGH	LOW	.25	MED	.75	HIGH
					PRD					
.9	.086	.107	.115	.121	.143	.086	.516	.546	.569	.143
.8	.085	.113	.121	.132	.160	.085	.485	.520	.548	.160
.7	.098	.128	.138	.148	.178	.098	.511	.548	.569	.178
.6	.092	.126	.137	.148	.182	.092	.428	.449	.484	.182
.5	.103	.138	.151	.162	.197	.103	.415	.443	.471	.197
.4	.112	.146	.156	.168	.202	.112	.369	.394	.431	.202
.3	.122	.166	.182	.196	.239	.122	.361	.391	.417	.239
.2	.138	.209	.233	.257	.328	.138	.356	.383	.416	.328
					NMAE					
.9	.007	.010	.010	.012	.014	.007	.058	.061	.065	.014
.8	.007	.010	.011	.013	.016	.007	.054	.058	.062	.016
.7	.008	.012	.013	.014	.018	.008	.056	.060	.064	.018
.6	.008	.011	.012	.014	.017	.008	.046	.049	.053	.017
.5	.008	.013	.014	.016	.020	.008	.045	.048	.052	.020
.4	.009	.013	.014	.016	.020	.009	.040	.044	.046	.020
.3	.010	.015	.016	.019	.024	.010	.038	.042	.046	.024
.2	.010	.018	.020	.023	.032	.010	.037	.041	.045	.032
					NMSE					
.9	.007	.011	.013	.015	.020	.007	.266	.298	.324	.020
.8	.006	.013	.015	.017	.024	.006	.236	.270	.301	.024
.7	.008	.016	.019	.022	.030	.008	.261	.300	.324	.030
.6	.007	.016	.019	.022	.031	.007	.183	.201	.235	.031
.5	.009	.019	.023	.026	.037	.009	.172	.196	.221	.037
.4	.011	.021	.024	.028	.039	.011	.136	.156	.185	.039
.3	.012	.028	.033	.038	.054	.012	.130	.153	.174	.054
.2	.011	.044	.054	.066	.099	.011	.127	.147	.173	.099
					SSIM					
.9	.999	1	1	1	1	.999	.992	.993	.994	1
.8	.999	1	1	1	1	.999	.993	.994	.995	1
.7	.999	.999	1	1	1	.999	.992	.993	.994	1
.6	.999	1	1	1	1	.999	.994	.995	.996	1
.5	.999	.999	.999	1	1	.999	.995	.995	.996	1
.4	.999	.999	.999	1	1	.999	.996	.996	.997	1
.3	.999	.999	.999	.999	1	.999	.996	.996	.997	1
.2	.998	.998	.999	.999	1	.998	.996	.996	.997	1
					TIME					
.9	.014	.020	.022	.025	.031	.014	.967	1.030	1.100	.031
.8	.013	.018	.020	.022	.026	.013	.591	.626	.680	.026
.7	.012	.016	.017	.018	.022	.012	.398	.417	.455	.022
.6	.011	.014	.016	.017	.021	.011	.214	.232	.253	.021
.5	.008	.013	.014	.016	.021	.008	.123	.131	.143	.021
.4	.008	.011	.012	.013	.016	.008	.062	.068	.074	.016
.3	.005	.009	.010	.012	.016	.005	.030	.032	.038	.016
.2	.005	.007	.008	.008	.009	.005	.010	.011	.012	.009
					ITER					
.9	6	9	10	11	14	6	114	116	119	14
.8	6	9	9	11	14	6	96	98	100	14
.7	5	8	9	10	13	5	83	84	86	13
.6	6.5	8	9	9	10.5	6.5	66	68	69	10.5
.5	6.5	8	9	9	10.5	6.5	54.5	56	57	10.5
.4	4	7	8	9	12	4	42	43	44	12
.3	5.5	7	8	8	9.5	5.5	31	32	33	9.5
.2	4.5	6	7	7	8.5	4.5	21	22	22	8.5

Table B.3: OMMP and OMMPn metrics for varying CR for simulated signals

B.4 OMMP / Simulated / Noise

	OMMP					OMMPn				
noise	**LOW**	**.25**	**MED**	**.75**	**HIGH**	**LOW**	**.25**	**MED**	**.75**	**HIGH**
					PRD					
.01	.020	.029	.032	.036	.045	.020	.054	.057	.062	.045
.02	.064	.087	.095	.103	.126	.064	.170	.181	.192	.126
.05	.110	.143	.155	.165	.197	.110	.433	.461	.482	.197
.1	.132	.185	.204	.220	.272	.132	.685	.742	.773	.272
.2	.163	.230	.256	.275	.343	.163	.952	1.007	1.080	.343
					NMAE					
.01	.002	.003	.003	.004	.005	.002	.005	.005	.006	.005
.02	.006	.008	.009	.010	.013	.006	.018	.019	.021	.013
.05	.008	.013	.014	.016	.020	.008	.047	.050	.054	.020
.1	.010	.016	.018	.020	.026	.010	.075	.080	.084	.026
.2	.011	.019	.022	.025	.033	.011	.103	.111	.118	.033
					NMSE					
.01	2.3e-4	8.5e-4	.001	.001	.002	2.3e-4	.003	.003	.004	.002
.02	.003	.008	.009	.011	.015	.003	.029	.033	.037	.015
.05	.010	.020	.024	.027	.037	.010	.187	.212	.232	.037
.1	.013	.034	.041	.048	.069	.013	.470	.550	.597	.069
.2	.019	.053	.065	.076	.110	.019	.906	1.013	1.166	.110
					SSIM					
.01	1	1	1	1	1	1	1	1	1	1
.02	1	1	1	1	1	1	.999	.999	.999	1
.05	.999	.999	.999	1	1	.999	.994	.995	.996	1
.1	.998	.999	.999	.999	1	.998	.986	.987	.989	1
.2	.997	.998	.998	.999	1	.997	.973	.976	.979	1
					TIME					
.01	.021	.037	.039	.047	.063	.021	.011	.012	.014	.063
.02	.010	.016	.018	.020	.027	.010	.065	.072	.082	.027
.05	.010	.013	.013	.014	.017	.010	.126	.138	.157	.017
.1	.009	.011	.012	.012	.014	.009	.160	.169	.184	.014
.2	.008	.010	.011	.011	.013	.008	.175	.185	.201	.013
					ITER					
.01	16.5	22.5	24	26.5	32.5	16.5	12	13	14	32.5
.02	8	11	12	13	16	8	39.5	41	43	16
.05	4	7	8	9	12	4	55	56	58	12
.1	4.5	6	6	7	8.5	4.5	61	62	63	8.5
.2	3.5	5	6	6	7.5	3.5	64	65	66	7.5

Table B.4: OMMP and OMMPn metrics for varying noise for simulated signals

B.5 GAP / Real / Compression

	GAP					GAPn				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**	**LOW**	**.25**	**MED**	**.75**	**HIGH**
					PRD					
.9	.109	.175	.194	.219	.285	.109	.216	.228	.241	.285
.8	.117	.198	.224	.252	.333	.117	.246	.261	.277	.333
.7	.108	.189	.215	.242	.322	.108	.235	.249	.264	.322
.6	.109	.188	.215	.241	.320	.109	.234	.250	.266	.320
.5	.113	.198	.229	.256	.341	.113	.248	.264	.284	.341
.4	.111	.200	.227	.259	.348	.111	.245	.266	.288	.348
.3	.130	.228	.258	.293	.390	.130	.271	.299	.328	.390
.2	.161	.342	.397	.462	.643	.161	.349	.405	.468	.643
					NMAE					
.9	3.0e-4	.027	.033	.045	.071	3.0e-4	.032	.042	.062	.071
.8	.002	.031	.037	.050	.079	.002	.037	.047	.069	.079
.7	9.8e-4	.029	.036	.048	.077	9.8e-4	.035	.045	.067	.077
.6	8.0e-4	.030	.036	.050	.079	8.0e-4	.035	.045	.065	.079
.5	.001	.032	.039	.053	.084	.001	.037	.048	.070	.084
.4	.003	.033	.040	.054	.084	.003	.038	.048	.070	.084
.3	.003	.038	.046	.062	.097	.003	.043	.055	.081	.097
.2	0	.058	.074	.103	.170	0	.058	.077	.107	.170
					NMSE					
.9	.005	.030	.038	.048	.074	.005	.047	.052	.058	.074
.8	.003	.039	.050	.063	.100	.003	.061	.068	.077	.100
.7	.001	.036	.046	.059	.093	.001	.055	.062	.070	.093
.6	.001	.035	.046	.058	.092	.001	.055	.063	.071	.092
.5	5.0e-4	.039	.052	.065	.104	5.0e-4	.062	.070	.080	.104
.4	0	.040	.051	.067	.108	0	.060	.071	.083	.108
.3	.001	.052	.067	.086	.136	.001	.073	.090	.107	.136
.2	0	.117	.158	.214	.359	0	.122	.164	.219	.359
					SSIM					
.9	.792	.895	.937	.964	1.067	.792	.836	.918	.961	1.067
.8	.721	.860	.920	.953	1.093	.721	.799	.894	.948	1.093
.7	.757	.876	.925	.955	1.074	.757	.812	.904	.953	1.074
.6	.755	.875	.925	.955	1.075	.755	.812	.905	.951	1.075
.5	.722	.859	.914	.950	1.087	.722	.796	.890	.943	1.087
.4	.731	.861	.915	.947	1.077	.731	.794	.890	.942	1.077
.3	.641	.818	.891	.935	1.112	.641	.753	.862	.925	1.112
.2	.266	.628	.759	.869	1.230	.266	.609	.764	.863	1.230
					TIME					
.9	.260	.442	.502	.563	.744	.260	.140	.158	.176	.744
.8	.324	.440	.466	.517	.633	.324	.138	.157	.175	.633
.7	.334	.437	.456	.505	.609	.334	.136	.155	.174	.609
.6	.255	.400	.443	.497	.642	.255	.136	.154	.173	.642
.5	.266	.378	.434	.453	.565	.266	.135	.154	.165	.565
.4	.288	.376	.391	.435	.522	.288	.120	.153	.155	.522
.3	.342	.375	.378	.397	.429	.342	.107	.134	.153	.429
.2	.238	.325	.375	.383	.470	.238	.116	.134	.153	.470
					ITER					
.9	4	7	8	9	12	4	6	7	8	12
.8	5.5	7	7	8	9.5	5.5	6	7	8	9.5
.7	5.5	7	7	8	9.5	5.5	6	7	8	9.5
.6	3	6	7	8	11	3	6	7	8	11
.5	4.5	6	7	7	8.5	4.5	6	7	7	8.5
.4	4.5	6	6	7	8.5	4.5	5	7	7	8.5
.3	6	6	6	6	6	6	4	6	7	6
.2	3.5	5	6	6	7.5	3.5	5	6	7	7.5

Table B.5: GAP and GAPn metrics for varying CR for real signals

B.6 GAP / Real / Noise

	GAP					GAPn				
noise	**LOW**	**.25**	**MED**	**.75**	**HIGH**	**LOW**	**.25**	**MED**	**.75**	**HIGH**
					PRD					
.01	1.9e-4	.037	.049	.062	.098	1.9e-4	.035	.050	.086	.098
.02	.060	.109	.124	.142	.191	.060	.126	.137	.152	.191
.05	.107	.192	.220	.249	.334	.107	.240	.256	.277	.334
.1	.173	.292	.331	.371	.490	.173	.350	.375	.405	.490
.2	.282	.424	.471	.518	.660	.282	.467	.498	.540	.660
					NMAE					
.01	0	.008	.009	.013	.022	0	.007	.010	.015	.022
.02	.001	.018	.022	.029	.046	.001	.020	.026	.039	.046
.05	.002	.031	.038	.051	.081	.002	.037	.047	.067	.081
.1	.003	.047	.057	.076	.119	.003	.054	.069	.094	.119
.2	.002	.067	.081	.110	.175	.002	.072	.090	.131	.175
					NMSE					
.01	0	.001	.002	.004	.007	0	.001	.003	.007	.007
.02	0	.012	.015	.020	.033	0	.016	.019	.023	.033
.05	0	.037	.049	.062	.099	0	.058	.066	.077	.099
.1	.006	.085	.110	.138	.217	.006	.123	.141	.164	.217
.2	.046	.179	.222	.269	.402	.046	.218	.248	.291	.402
					SSIM					
.01	.985	.993	.995	.997	1.004	.985	.990	.995	.997	1.004
.02	.911	.955	.972	.984	1.028	.911	.931	.966	.984	1.028
.05	.749	.871	.920	.952	1.073	.749	.802	.898	.947	1.073
.1	.516	.746	.837	.899	1.129	.516	.669	.801	.888	1.129
.2	.213	.577	.719	.819	1.182	.213	.525	.694	.816	1.182
					TIME					
.01	.381	.765	.897	1.022	1.407	.381	.134	.170	.189	1.407
.02	.272	.448	.506	.565	.740	.272	.134	.153	.172	.740
.05	.276	.386	.442	.459	.568	.276	.134	.152	.170	.568
.1	.293	.382	.389	.442	.531	.293	.116	.134	.152	.531
.2	.265	.344	.383	.397	.477	.265	.115	.121	.135	.477
					ITER					
.01	6	12	14	16	22	6	6	8	9	22
.02	4	7	8	9	12	4	6	7	8	12
.05	4.5	6	7	7	8.5	4.5	6	7	7.7	8.5
.1	4.5	6	6	7	8.5	4.5	5	6	7	8.5
.2	3.5	5	6	6	7.5	3.5	5	5	6	7.5

Table B.6: GAP and GAPn metrics for varying noise for real signals

B.7 GAP / Simulated / Compression

	GAP					GAPn				
CR	**LOW**	**.25**	**MED**	**.75**	**HIGH**	**LOW**	**.25**	**MED**	**.75**	**HIGH**
					PRD					
.9	.151	.194	.209	.223	.266	.151	.212	.221	.232	.266
.8	.163	.204	.217	.230	.271	.163	.226	.239	.253	.271
.7	.170	.229	.247	.269	.328	.170	.253	.269	.281	.328
.6	.149	.210	.234	.250	.310	.149	.239	.253	.273	.310
.5	.157	.215	.230	.254	.312	.157	.243	.258	.281	.312
.4	.153	.203	.217	.236	.286	.153	.232	.249	.266	.286
.3	.144	.207	.223	.249	.312	.144	.252	.267	.291	.312
.2	.162	.234	.257	.281	.353	.162	.269	.293	.321	.353
					NMAE					
.9	.011	.017	.019	.021	.026	.011	.023	.024	.026	.026
.8	.012	.018	.019	.021	.027	.012	.025	.026	.027	.027
.7	.014	.020	.022	.024	.030	.014	.027	.029	.031	.030
.6	.009	.017	.020	.023	.032	.009	.026	.027	.029	.032
.5	.010	.018	.020	.024	.032	.010	.026	.028	.031	.032
.4	.011	.018	.019	.022	.029	.011	.025	.027	.028	.029
.3	.011	.018	.020	.023	.030	.011	.025	.028	.031	.030
.2	.011	.020	.023	.026	.034	.011	.027	.030	.033	.034
					NMSE					
.9	.020	.038	.044	.050	.068	.020	.045	.049	.054	.068
.8	.024	.041	.047	.053	.071	.024	.051	.057	.064	.071
.7	.023	.053	.061	.072	.102	.023	.064	.072	.079	.102
.6	.016	.044	.055	.062	.090	.016	.057	.064	.074	.090
.5	.019	.046	.053	.065	.092	.019	.059	.066	.079	.092
.4	.019	.041	.047	.056	.078	.019	.054	.062	.070	.078
.3	.014	.043	.050	.062	.091	.014	.063	.071	.085	.091
.2	.018	.055	.066	.079	.116	.018	.072	.086	.103	.116
					SSIM					
.9	.998	.999	.999	.999	1	.998	.999	.999	.999	1
.8	.998	.999	.999	.999	1	.998	.998	.999	.999	1
.7	.998	.998	.999	.999	.999	.998	.998	.998	.998	.999
.6	.998	.998	.999	.999	1	.998	.998	.998	.999	1
.5	.998	.998	.999	.999	1	.998	.998	.998	.999	1
.4	.998	.999	.999	.999	1	.998	.998	.998	.999	1
.3	.998	.999	.999	.999	1	.998	.998	.998	.999	1
.2	.997	.998	.998	.999	1	.997	.997	.998	.998	1
					TIME					
.9	.181	.336	.387	.440	.595	.181	.096	.099	.115	.595
.8	.223	.325	.382	.394	.496	.223	.096	.114	.132	.496
.7	.203	.321	.365	.399	.517	.203	.095	.104	.113	.517
.6	.132	.280	.325	.380	.528	.132	.094	.112	.113	.528
.5	.152	.257	.314	.326	.431	.152	.094	.112	.113	.431
.4	.216	.256	.256	.282	.322	.216	.093	.095	.111	.322
.3	.220	.255	.255	.278	.313	.220	.093	.111	.137	.313
.2	.110	.195	.196	.252	.338	.110	.092	.110	.128	.338
					ITER					
.9	2	5	6	7	10	2	4	4	5	10
.8	3.5	5	6	6	7.5	3.5	4	5	6	7.5
.7	3.5	5	5	6	7.5	3.5	4	4	5	7.5
.6	1	4	5	6	9	1	4	5	5	9
.5	2.5	4	5	5	6.5	2.5	4	5	5	6.5
.4	4	4	4	4	4	4	4	4	5	4
.3	1.5	3	4	4	5.5	1.5	4	5	6	5.5
.2	1.5	3	3	4	5.5	1.5	4	5	6	5.5

Table B.7: GAP and GAPn metrics for varying CR for simulated signals

B.8 GAP / Simulated / Noise

	GAP					GAPn				
noise	**LOW**	**.25**	**MED**	**.75**	**HIGH**	**LOW**	**.25**	**MED**	**.75**	**HIGH**
						PRD				
.01	.022	.032	.035	.039	.049	.022	.032	.036	.043	.049
.02	.102	.126	.134	.142	.167	.102	.123	.134	.146	.167
.05	.169	.220	.235	.253	.304	.169	.254	.274	.297	.304
.1	.221	.304	.328	.360	.444	.221	.375	.403	.437	.444
.2	.267	.378	.415	.453	.564	.267	.504	.537	.584	.564
						NMAE				
.01	.002	.003	.003	.004	.005	.002	.004	.004	.005	.005
.02	.008	.011	.013	.014	.017	.008	.013	.014	.015	.017
.05	.012	.019	.021	.023	.029	.012	.027	.029	.031	.029
.1	.013	.024	.028	.032	.043	.013	.040	.042	.046	.043
.2	.016	.029	.034	.039	.052	.016	.053	.057	.061	.052
						NMSE				
.01	3.3e-4	.001	.001	.002	.002	3.3e-4	.001	.001	.002	.002
.02	.009	.016	.018	.020	.027	.009	.015	.018	.021	.027
.05	.024	.048	.055	.064	.088	.024	.064	.075	.088	.088
.1	.037	.093	.108	.130	.186	.037	.141	.163	.191	.186
.2	.050	.143	.172	.205	.298	.050	.254	.288	.341	.298
						SSIM				
.01	1	1	1	1	1	1	1	1	1	1
.02	.999	.999	1	1	1	.999	1	1	1	1
.05	.998	.998	.999	.999	.999	.998	.998	.998	.998	.999
.1	.996	.997	.997	.998	.999	.996	.995	.996	.997	.999
.2	.992	.995	.996	.997	.999	.992	.992	.993	.994	.999
						TIME				
.01	.433	.606	.655	.721	.894	.433	.108	.112	.130	.894
.02	.170	.325	.372	.429	.584	.170	.094	.112	.130	.584
.05	.169	.257	.267	.315	.403	.169	.094	.112	.127	.403
.1	.117	.204	.257	.262	.349	.117	.094	.094	.112	.349
.2	.113	.199	.200	.257	.343	.113	.076	.094	.112	.343
						ITER				
.01	7	10	10.5	12	15	7	4.5	5	6	15
.02	2.8	5	6	6.5	8.8	2.8	4	5	6	8.8
.05	2.5	4	4	5	6.5	2.5	4	5	5	6.5
.1	1.5	3	4	4	5.5	1.5	4	4	5	5.5
.2	1.5	3	3	4	5.5	1.5	3	4	5	5.5

Table B.8: GAP and GAPn metrics for varying noise for simulated signals

Bibliography

[1] A. M. Abdulghani, A. J. Casson, and E. Rodriguez-Villegas. Quantifying the performance of compressive sensing on scalp eeg signals. In *Applied Sciences in Biomedical and Communication Technologies (ISABEL), 2010 3rd International Symposium on*, pages 1–5. IEEE, 2010.

[2] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *Computers, IEEE Transactions on*, 100(1):90–93, 1974.

[3] Y. Avonds, Y. Liu, and S. Van Huffel. Simultaneous greedy analysis pursuit for compressive sensing of multi-channel ecg signals. Paper written as part of thesis, 2013.

[4] S. R. Becker. *Practical Compressed Sensing: modern data acquisition and signal processing.* PhD thesis, California Institute of Technology, 2011.

[5] T. Blu, P.-L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot. Sparse sampling of signal innovations. *Signal Processing Magazine, IEEE*, 25(2):31–40, 2008.

[6] T. Blumensath and M. E. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5-6):629–654, 2008.

[7] T. T. Cai and L. Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *Information Theory, IEEE Transactions on*, 57(7):4680–4688, 2011.

[8] E. J. Candes, Y. C. Eldar, D. Needell, and P. Randall. Compressed sensing with coherent and redundant dictionaries. *Applied and Computational Harmonic Analysis*, 31(1):59–73, 2011.

[9] E. J. Candes, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

[10] K. M. Cheman. *Optimization Techniques for Solving Basis Pursuit Problems.* PhD thesis, North Carolina State University, 2006.

[11] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[12] R. Giryes, M. Elad, et al. Cosamp and sp for the cosparse analysis model. In *The 20th European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, August 27–31, 2012.

[13] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

[14] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13).

[15] M. A. Herman and T. Strohmer. General deviants: An analysis of perturbations in compressed sensing. *Selected topics in signal processing, IEEE Journal of*, 4(2):342–349, 2010.

[16] S. Ji, Y. Xue, and L. Carin. Bayesian compressive sensing. *Signal Processing, IEEE Transactions on*, 56(6):2346–2356, 2008.

[17] K. Kanoun, H. Mamaghanian, N. Khaled, and D. Atienza. A real-time compressed sensing-based personal electrocardiogram monitoring system. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011*, pages 1–6. IEEE, 2011.

[18] Y. G. Li, J. H. Winters, and N. R. Sollenberger. Mimo-ofdm for wireless communication: signal detection with enhanced channel estimation. *Communications, IEEE Transactions on*, 50(9):1471–1477, 2002.

[19] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst. Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes. *Biomedical Engineering, IEEE Transactions on*, 58(9):2456–2466, 2011.

[20] J. S. Medina. Compressive sampling and reconstruction of biopotential signals. Master's thesis, KU Leuven, 2011.

[21] G. B. Moody and R. G. Mark. The impact of the MIT-BIH arrhythmia database. *Engineering in Medicine and Biology Magazine, IEEE*, 20(3):45–50, 2001.

[22] S. Nam, M. E. Davies, M. Elad, and R. Gribonval. Cosparse analysis modeling-uniqueness and algorithms. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5804–5807. IEEE, 2011.

[23] S. Nam, M. E. Davies, M. Elad, and R. Gribonval. Recovery of cosparse signals with greedy analysis pursuit in the presence of noise. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2011 4th IEEE International Workshop on*, pages 361–364. IEEE, 2011.

[24] S. Nam, M. E. Davies, M. Elad, and R. Gribonval. The cosparse analysis model and algorithms. *Applied and Computational Harmonic Analysis*, 34(1):30–56, 2013.

[25] J. Pan and W. J. Tompkins. A real-time qrs detection algorithm. *Biomedical Engineering, IEEE Transactions on*, (3):230–236, 1985.

[26] Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[27] L. F. Polania, R. E. Carrillo, M. Blanco-Velasco, and K. E. Barner. Compressed sensing based method for ecg compression. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 761–764. IEEE, 2011.

[28] R. M. Rangayyan. *Biomedical signal analysis: a case-study approach.* IEEE Press, 2002.

[29] L. Rebollo-Neira and Z. Xu. Sparse signal representation by adaptive non-uniform b-spline dictionaries on a compact interval. *Signal processing*, 90(7):2308–2313, 2010.

[30] R. Sameni, M. Shamsollahi, and C. Jutten. Multi-channel electrocardiogram denoising using a bayesian filtering framework. In *Computers in Cardiology*, pages 185–188, Valencia, Spain, September 17–20, 2006. IEEE.

[31] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[33] Z. Xu. The performance of orthogonal multi-matching pursuit under rip. *arXiv preprint arXiv:1210.5323*, 2012.

[34] Z. Zhang and B. D. Rao. Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning. *Selected Topics in Signal Processing, IEEE Journal of*, 5(5):912–926, 2011.

[35] Z. Zhang and B. D. Rao. Extension of sbl algorithms for the recovery of block sparse signals with intra-block correlation. *Signal Processing, IEEE Transactions on*, 61(8):2009–2015, 2013.

[36] Y. Zigel, A. Cohen, and A. Katz. The weighted diagnostic distortion (wdd) measure for ecg signal compression. *Biomedical Engineering, IEEE Transactions on*, 47(11):1422–1430, 2000.