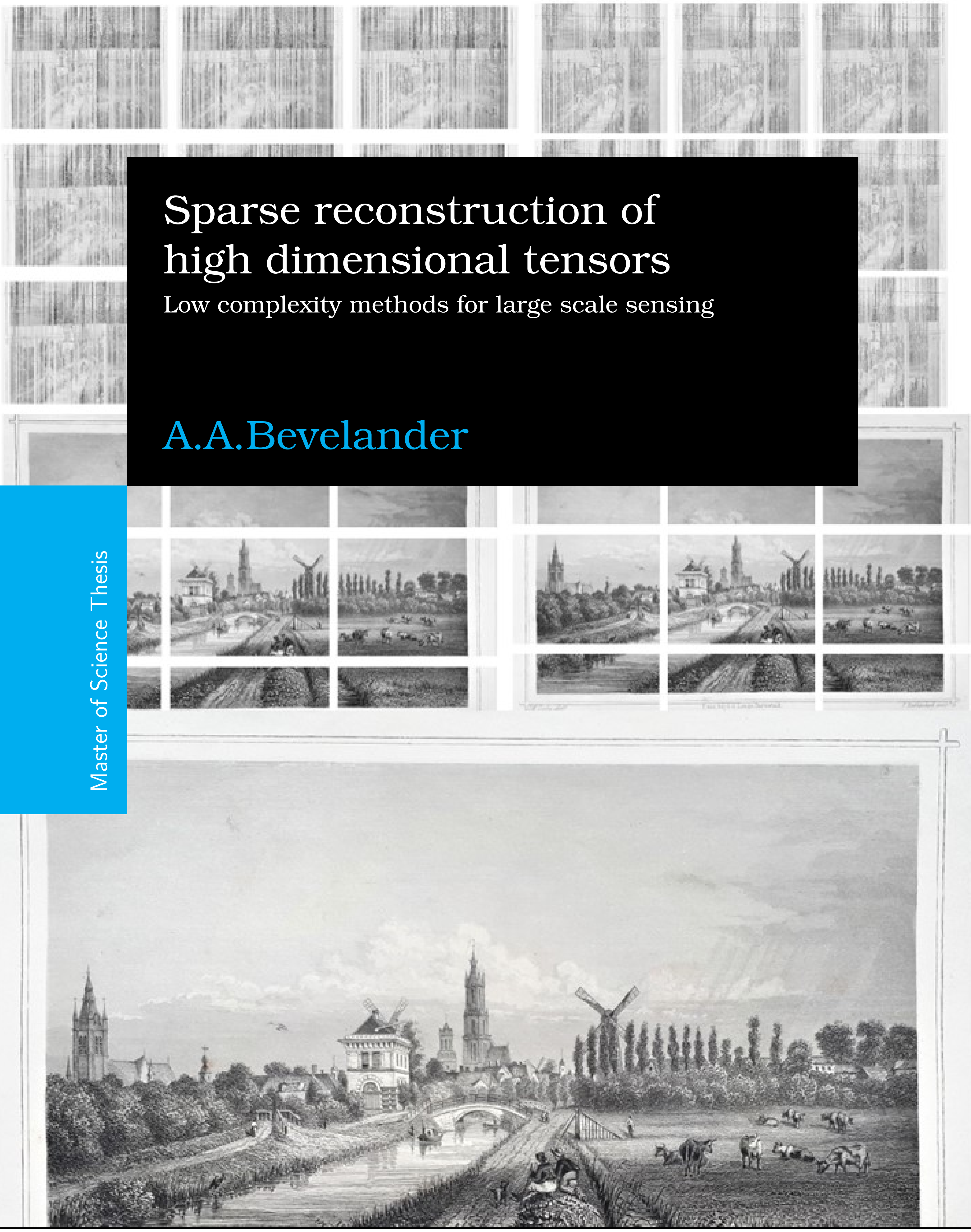
Sparse reconstruction of high dimensional tensors

Low complexity methods for large scale sensing

A.A.Bevelander

Master of Science Thesis



TU Delft Delft University of Technology

Delft Center for Systems and Control

Sparse reconstruction of high dimensional tensors

Low complexity methods for large scale sensing

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

A.A.Bevelander

January 28, 2024

Faculty of Mechanical Engineering · Delft University of Technology

Abstract

Compressed sensing is a framework in signal processing that enables the efficient acquisition and reconstruction of sparse signals. A widely-used class of algorithms that are used for this reconstruction, called greedy-algorithms, depend on non-convex optimization. With increasing signal size, these problems become computationally very hard. Block compressed sensing is a framework in compressed sensing that divides the compressed sensing problem into subproblems, gaining a better storage complexity. However, block compressed sensing has not yet been studied form a computational complexity perspective.

This thesis focuses on the application of block compressed sensing to signals of high dimension to gain insight into the relation between reconstruction performance and computational complexity. This is done by, first investigating how theoretical reconstruction guarantees change, when the problem is divided into smaller sub-problems and by doing a complexity analysis of the reconstruction itself. Each sub-problem solves for a portion of a signal, defined as a block. Next, experiments are conducted in order to get insight into the trade-off between computational complexity and quality of the reconstruction. It can be found that, by using this block-wise approach, the computational complexity of the reconstruction problem decreases, but at the same time, quality of the reconstruction deteriorates. Besides, a method to compensate for this performance loss is proposed. The key idea of this method is that, by propagating prior information among the different blocks, the reconstructions of the blocks can be improved. Finally, block compressed sensing and prior-aware block compressed sensing are analysed in a higher order tensor compressed sensing setting. Nevertheless, this setting was found to exhibit a less favourable complexity-performance trade-off than the normal one, as this setting resulted in both a more complex and a less accurate reconstruction than the normal one.

Table of Contents

Preface & Acknowledgements

This thesis explores a computational complexity approach to enhance signal reconstruction in Block Compressed Sensing (BCS). Focused on improving accuracy and efficiency of BCS, a novel strategy is proposed: the propagation of prior information among signal blocks. The synthesis in this thesis aims to advance the Compressed Sensing paradigm a tiny step further and to be helpful to researchers and engineers that are involved in this topic.

The cover of this thesis is based on an old drawing of the city of Delft called *'Van de Haagsche vaart gezien'* by *J.W. Cooke* made around 1858. On the bottom of the page, the original drawing is shown. The other parts show the image partitioned into blocks. In the middle row, the blocks are uniformly partitioned, while in the top row, the columns of the blocks are permuted such that they give rise to a more or less random pattern. This permutation operation is essential in the Block Compressed Sensing framework and the pictures give a clear interpretation of how this is done.

I would like to express my sincere gratitude to my supervisors Dr. Myers and Dr.Ir Batselier for their guidance along my thesis. I found myself highly motivated to work under their supervision. Not only because of their critical approach, resulting in a healthy demanding work environment, but mainly because of the inspiration I got from their profound expertise and interest.

Finally, besides that the cover page functions as a visualization of a step in the Block Compressed Sensing framework, it also visualizes the time, place and journey along my days as a student. The city of Delft and especially the bright building located alongside the canal, called *de Kogelgieterij*, are the places where I have become the person I am today, thanks to the amazing people that I lived with. To them I would like to dedicate this work and thank them for their unconditional support throughout the years.

Delft, University of Technology — A.A.Bevelander
January 28, 2024

Chapter 1

Introduction

The process of sensing, recording and storing signals in a digital manner has become of lasting importance in technical innovations over the last decades. As technology evolves, the demanded specifications in this domain increase, resulting in a need for more advanced hardware and software. On the hardware side, this problem can be addressed by building computer systems with more computing power and memory storage. However, merely ameliorating the systems in this way is not sustainable. For these innovations namely, more powerful computer systems are required which is associated with a need for more resources and having higher energy consumption than the previous ones. On the software side though, research can be done on how to better use these hardware components, such that the system as a whole becomes more efficient.

The question of how to acquire and retrieve signals using limited hardware resources is key in the domain of compressed sensing. This domain focuses on, for example, reducing the memory storage requirements of a signal or measuring a signal from only a limited number of measurements, due to hardware restrictions. This is done by exploiting the sparsity of a signal, i.e. the property that a signal consists mostly of values equal to zero. Formally in compressed sensing, a signal $\mathbf{x} \in \mathbb{R}^n$ is measured by:

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{1-1}$$

with measurements $\mathbf{y} \in \mathbb{R}^m$ for a measurement matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with typically $m \ll n$. The signal is $\mathbf{x}$ is sparse, implying that the number of non-zero entries in $\mathbf{x}$ is small with respect to its size, or formally: $||\mathbf{x}||_0 \ll n$. Here, the $|| \cdot ||_0$ operator denotes the ℓ_0-norm, returning the number of non-zero entries in its argument. It is formally defined in definition 2. Since the measurement matrix $\mathbf{A}$ has more columns than rows, reconstructing $\mathbf{x}$ from $\mathbf{y}$ and $\mathbf{A}$ is equivalent to solving an under-determined linear system of equations, allowing for multiple solutions. In compressed sensing, the sparsest possible solution to this system is of interest. Formally, the original signal can then be reconstructed using $\mathbf{y}$ and $\mathbf{A}$ by:

$$\begin{aligned} \hat{\mathbf{x}} = \min_{\mathbf{z} \in \mathbb{R}^n} ||\mathbf{z}||_0 \\ \text{Subject to: } \mathbf{A}\mathbf{z} = \mathbf{y}. \end{aligned} \tag{1-2}$$

From basic optimization theory, it is known that the computational complexity for optimizing non-convex objective functions, such as this ℓ_0-norm minimization problem, grows super-linearly with an increasing number of variables [1]. Various approaches to this problem exists. For example, under certain conditions the ℓ_0-norm can be relaxed to a convex ℓ_1-norm minimization problem, which is far less complex to solve. However, the relaxation also has downsides, such as additional constraints on the measurement matrix [2] and possible issues with signal support identification [3].

To maintain the advantages of the ℓ_0-norm at a reduced computational complexity, the reconstruction problem can be partitioned, i.e., cut up into sub-problems. The idea of partitioning in CS has received some research attention under the title of block-compressed sensing [4],[5]. In this framework, the sparse signal is sub-divided or partitioned into small *blocks* that allow for independent measuring and reconstruction. The available literature on block compressed sensing is tailored to specific signal processing applications. Besides, in [6, 7, 8], block compressed sensing is approached from a storage complexity perspective. In traditional CS, the storage complexity increases quadratically with the signal size, as the measurement matrix $\mathbf{A}$ has to be stored entirely. By partitioning, the total size of the measurement matrix is reduced. Based on this literature review it can be concluded that there is no work on a computational complexity study on block compressed sensing. Anyhow, this is a relevant topic as it considers the potential of BCS to handle signals of *large* and *high* dimension efficiently. Before going further, let's first define the terminology around dimensionality in a clear way.

Definition 1. *(Dimensionality)* A signal $\underline{\mathbf{X}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ has d dimensions.

- $\underline{\mathbf{X}}$ is high dimensional if d is larger than 2, i.e. $d \in \{3, 4, \dots\}$
- $\underline{\mathbf{X}}$ is of large dimension if the size of the dimensions are larger than 100, i.e. $n_i > 100$ for $i \in \{1, 2, \dots, d\}$

Moreover, $\underline{\mathbf{X}}$ is *large*, if the total number of signal entries is larger than 1000, i.e. $\Pi_{i=1}^{d} n_i > 1000$

1-1 Motivation

As mentioned, compressed sensing problems generally show a rapid increase in computational complexity with signal size. Applications such as millimeter wave radar and MRI often feature signals of large dimension. A signal size of $|\underline{\mathbf{X}}| = 10^6$ is no exception. With signals this large, CS tends to get time and energy consuming. Approaching the problem in parts, by reconstructing blocks within the signal, can reduce the computational complexity, but at what cost? A first take on answering this question, is by considering the *Welch bound* of the partitioned problem, which is further elaborated in subsection 2-2-2. This bound directly relates to the error bound on the reconstruction error relating to (1-2). This corresponds to a trade-off: computational complexity versus reconstruction performance. However, if there is a correlation across blocks and this correlation is known, it is possible to propagate information from a reconstructed block to the next block. With such a prior propagation strategy, the error bound could possibly be improved again.

These ideas are relevant for a number of cases.

1. **Multi-core processors**: Instead of solving the problems one-by-one, the independence of the different sub-problems can be exploited. The sub-problems can be run in parallel utilizing multiple cores. This can drastically reduce the computational time required for the reconstruction. The drawback of this approach is that the individual reconstruction problems for each partition will have to be independently solvable. So, no passing of information about correlation of the signal between the partitions is possible.

2. **Single-core processors**: With just one processor, partitioning a compressed sensing problem will still reduce the computational complexity. Furthermore, as the problems are solved in series, it is possible to pass information from one reconstructed partition to another and use it in the next partition. In this work, this is found to improve the overall reconstruction performance at high SNR.

To summarize, the motivation for the work in this thesis is to gain insight in how CS problems for large signals can still be executed with limited computational power. This is done by taking a partitioned approach to CS which can improve computational complexity of the reconstruction.

1-2 Objectives

In order to align with the motivation above, a number of research questions can be formulated. Main question:

- How does the partitioning of a compressed sensing problem affect its reconstruction performance and complexity?

To structure the answering process of this, still broad, question, the following corresponding sub-questions are formulated:

1. How does reconstruction performance change with the number of blocks in a BCS problem?

2. What is a proper strategy to partition a CS problem?

3. What is the change in computational complexity of the reconstruction when applying a BCS algorithm?

4. How can relevant prior information be propagated between different blocks in a BCS reconstruction and at what computational cost?

5. How can higher order CS algorithms be deployed within (prior-aware) BCS reconstruction?

Throughout this analysis, two and four dimensional data is used. Two-dimensional, as this allows for straightforward interpretation of the proposed methods and reconstruction results, such as the reconstruction of an image. Four-dimensional, as this clearly shows the behaviour of the algorithms in *higher order*. E.g. the *curse of dimensionality* [1] already applies in four dimensions: the number of entries in the data grows rapidly with the size of the dimensions.

1-3 Outline

Now that the aim of the thesis is clear, the proposed questions will be dealt with in the following structure:
Chapter 2 explains prior work and basic concepts that are required for the rest of the thesis. here, the compressed sensing framework and all of its relevant sub-domains are elaborated. Especially prior work on the topics of block compressed sensing and Prior-Aware compressed sensing is summarized and discussed when useful as these are the domains that lie closest to the contributions.
In chapter 3, the contribbutions of this thesis are made. Here, in section 3-1, an extensive description of the newly proposed prior-aware block compressed sensing framework is given. This involves detailed steps on the design synthesis of the pipeline that is proposed in the paper afterwards. Also, the choice for specific experiments and datasets are motivated and constructed. The main goal of this section is to prepare the reader for the more abstract and concise paper section that follows. After a short introduction, section 3-2 contains the conference paper which will be submitted to the European Signal Processing Conference (EUSIPCO) 2024. It contains all the principal findings and contributions in this thesis. The introduction section of the paper has a slight overlap with the chapter 2, however the paper is way more pithy. Next, the contributions that are relevant, but fall outside the scope of the paper are made in section 3-3. This section involves theoretical derivations about algorithm complexity of the tensor CS setting, but its focus is on showing numerical results of the experiments that were introduced in section 3-1. Finally, in chapter 4, the main conclusions of the thesis are summarized and recommendations for future research are made. Note again, that this Chapter has a slight overlap with the conclusions section of the paper in section 3-2.

1-4 Contributions

Within this thesis, the principal contributions are made in the paper in section 3-2. These are:

- Derivation of coherence-based reconstruction bounds for block compressed sensing.
- Analysis and experiments to motivate the use of block compressed sensing for the aim of decreasing computational complexity of the CS reconstruction problem.
- Synthesis of a pipeline and an algorithm that are able to propagate known correlation information among blocks in BCS reconstruction.

Besides, in section 3-3, some contributions are made that are outside the scope of the paper, but still relevant in the thesis. These are:

Name	**symbol**	**example**
Scalar	lowercase	c
Set	Uppercase	S
Vector	Bold Lowercase	$\mathbf{v}$
Matrix	Bold Uppercase	$\mathbf{A}$
Tensor	Underlined Bold Uppercase	$\underline{\mathbf{T}}$

Table 1-1: Writing convention for arrays

- Influence of comb-like partitioning strategy on performance of the BCS problem.
- Analysis on how the BCS can be extended to work with a higher order CS algorithm.
- Extension of the prior-aware BCS pipeline to higher order CS algorithms.

1-5 Notation and Abbreviations

For the sake of clarity, Table 1-1 shows the writing conventions for number arrays that are used in this thesis. Next, throughout this thesis, various abbreviations are used. They are listed in Table 1-2.

Definition	**Abbreviation**
(Block) compressed sensing	(B)CS
Prior-aware	PA
Higher-Order	HO
(Orthogonal) Matching Pursuit	(O)MP
Performance-Complexity trade-off	PC trade-off

Table 1-2: List of abbreviations

Chapter 2

Background

This chapter functions as an introduction to the fundamental topics in the analysis, done in this thesis. Also, prior work is discussed and elaborated if relevant. First, some essential definitions are mentioned in section 2-1. Also, a necessary introduction to multi-linear algebra is given. Next, in section 2-2, the CS framework is introduced and some essential notions are explained. Then, in section 2-3 the prior-aware extension to CS is explained and existing literature is discussed. Next, in section 2-4 the BCS framework is introduced and finally, the higher order extensions of all the aforementioned frameworks are gone through in section 2-5.

2-1 Preliminaries

This sections presents some fundamental definitions used in CS and higher order algebra.

2-1-1 CS Definitions

Definition 2. *(ℓ_0-norm)* The ℓ_0-norm of an array returns its number of non-zero entries. The ℓ_0-norm of a vector $\mathbf{a} \in \mathbb{R}^n$ is given by $||\mathbf{a}||_0 = s \leq n$. Note that the ℓ_0-norm is a non-convex operator.

Definition 3. *(Sparsity)* A signal is sparse if the number of non-zero components is small compated to the signal length. The sparsity level indicates the fraction or percentage of a signal that is non-zero. Consider $\mathbf{x} \in \mathbb{R}^n$ with number of non-zero entries equal to $||\mathbf{x}||_0 = s$, then $\mathbf{x}$ is said to be sparse if $s \ll n$ with sparsity ratio s/n.

2-1-2 Multi-Linear Algebra

In the analysis and algorithms in both section 2-5 and section 3-3, a variety of multilinear notions and operations are used. These are now introduced formally [9].

Definition 4. *(n-way Tensor)* Any number array is an n-way tensor, where n represents the number of dimensions in that array. Hence, a vector and a matrix are respectively 1-way and 2-way tensors. An n-way tensor $\underline{\mathbf{A}}$ can be written as:

$$\underline{\mathbf{A}} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_n}.$$

Definition 5. *(mode-n product)* The mode-n product ($\times_n$) between a tensor $\underline{\mathbf{A}} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$ and a matrix $\mathbf{B} \in \mathbb{R}^{k \times m_n}$ is given by:

$$\underline{\mathbf{C}} = \underline{\mathbf{A}} \times_n \mathbf{B} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_{n-1} \times k \times m_{n+1} \times m_{n+2} \times \cdots \times m_d}.$$

In words, it is an operation that contracts the n-th mode, or dimension, of the tensor with the second mode of the matrix. Note that these modes must be of equal size. Also note, that the mode-n product is commutative over modes.

If $a_{i_1,i_2,\ldots,i_d}$, b_{i_k,i_m} and $c_{i_1,i_2,\ldots,i_d}$ denote the elements of $\underline{\mathbf{A}}$,$\underline{\mathbf{B}}$ and $\underline{\mathbf{C}}$ respectively, the elements of $\underline{\mathbf{C}}$ are computed by:

$$c_{i_1,i_2,\ldots,i_d} = \sum_{i_n}^{m_n} a_{i_1,i_2,\ldots,i_m,\ldots,i_d} \cdot b_{k,i_m}.$$

Definition 6. *(Kronecker product)* The Kronecker product ($\otimes$) between tensors $\underline{\mathbf{A}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$ is given by:

$$\underline{\mathbf{C}} = \underline{\mathbf{A}} \otimes \underline{\mathbf{B}} \in \mathbb{R}^{n_1 m_1 \times n_2 m_2 \times \cdots \times n_d m_d}.$$

In words, the Kronecker product between $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ constructs a new tensor by combining each element of $\underline{\mathbf{A}}$ with the entire tensor $\underline{\mathbf{B}}$. This results in a new tensor where each entry of $\underline{\mathbf{A}}$ is multiplied by the entire tensor $\underline{\mathbf{B}}$ with its dimensions equal to the product of the dimension sizes of $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$.

If $a_{j_1,j_2,\ldots,j_d}$, $b_{k_1,k_2,\ldots,k_d}$ and $c_{i_1,i_2,\ldots,i_d}$ denote the elements of $\underline{\mathbf{A}}$,$\underline{\mathbf{B}}$ and $\underline{\mathbf{C}}$ respectively, the elements of $\underline{\mathbf{C}}$ are computed by:

$$c_{i_1 i_2 \ldots i_d} = a_{j_1,j_2,\ldots,j_d} \cdot b_{k_1,k_2,\ldots,k_d},$$

with $i_p = j_p \cdot k_p \; \forall p \in \{1, 2, \ldots, d\}$

Definition 7. *(Khatri-Rao Product)* The Khatri-Rao product ($\odot$) two matrices $\mathbf{A} \in \mathbb{R}^{n_1 \times m}$ and $\mathbf{B} \in \mathbb{R}^{n_2 \times m}$ is given by:

$$\mathbf{C} = \mathbf{A} \odot \mathbf{B} \in \mathbb{R}^{n_1 n_2 \times m}.$$

In words it can be described as a column wise Kronecker product where each column of the resulting matrix is an element-wise outer product of the corresponding columns of $\mathbf{A}$ and $\mathbf{B}$. Note that it is only defined if the second modes of the matrices correspond in size.

If $\mathbf{a}_i$, $\mathbf{b}_i$ and $\mathbf{c}_i$ denote the i-th column of $\mathbf{A}$,$\mathbf{B}$ and $\mathbf{C}$ respectively, the i-th columns of $\mathbf{C}$ is then computed by:

$$\mathbf{c}_i = \mathbf{a}_i \otimes \mathbf{b}_i$$

Definition 8. *(Convolution)* A convolution ($\circledast$) between a tensor $\underline{\mathbf{A}} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$ and a kernel $\underline{\mathbf{\Phi}} \in \mathbb{R}^{k_1 \times k_2 \times \cdots \times k_d}$ is given by:

$$\underline{\mathbf{C}} = \underline{\mathbf{A}} \circledast \underline{\mathbf{\Phi}} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}.$$

At the edges, the convolution is chosen to be *valid* [10] with the use of *zero-padding*. The valid setting makes sure that the convolution is only applied at locations where the two tensors completely overlap. The zero padding adds zeros around the tensor, such that the edges are captured fairly as well. This results in a tensor of same size as the original one.

Definition 9. *(Vectorization)* The vectorization or flattening of a tensor is equivalent to making a vector out of a tensor. The vectorization of $\underline{\mathbf{A}} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$ is given by:

$$\text{vec}(\underline{\mathbf{A}}) = \mathbf{a} \in \mathbb{R}^{(m_1 \cdot m_2 \cdot \dots \cdot m_d) \times 1}.$$

Definition 10. *(Frobenius norm)* The Frobenius norm of tensor $\underline{\mathbf{A}} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$ is calculated by summing the squares of each element:

$$||\underline{\mathbf{A}}||_F = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \cdots \sum_{i_d=1}^{m_d} (a_{i_1,i_2,\dots,i_d})^2.$$

Note that it is equivalent to the inner product of the vectorization with itself:

$$||\underline{\mathbf{A}}||_F = \langle \text{vec}(\underline{\mathbf{A}})^T, \text{vec}(\underline{\mathbf{A}}) \rangle.$$

Definition 11. *(CP-decomposition)* A CP decomposition of a tensor can be thought of as a higher order extension of the Singular Value Decomposition. It is formally given as:

$$\underline{\mathbf{A}} = \underline{\mathbf{\Sigma}} \times_1 \mathbf{B}^{(1)} \times_2 \mathbf{B}^{(2)} \times_3 \cdots \times_d \mathbf{B}^{(d)} \tag{2-1}$$

where the matrices $\mathbf{B}^{(k)} \in \mathbb{R}^{n_k \times r}$ are called factor matrices for $k \in \{1, 2, \dots, d\}$ and with $\underline{\mathbf{\Sigma}} \in \mathbb{R}^{r^{(1)} \times r^{(2)} \times \cdots \times r^{(d)}}$ a super diagonal tensor containing the singular values in ascending magnitude. Formally, this is denoted as:

$$\sigma_{i_1,i_2,\dots,i_d} = \begin{cases} \lambda_b & \text{if } i_1 = i_2 = \dots i_d \text{ for } b \in \{1, 2, \dots, r\} \\ 0 & \text{else} \end{cases}$$

and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r$ where r is the rank of the decomposition. Note that the k-th column of each matrix relates to the k-th singular value in $\underline{\mathbf{\Sigma}}$.

This last observation allows us to rewrite (2-1) using the definitions of the vectorization and the Khatri-Rao product as:

$$\text{vec}(\underline{\mathbf{A}}) = (\mathbf{B}^{(d)} \odot \mathbf{B}^{(d-1)} \odot \cdots \odot \mathbf{B}^{(1)}) \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_r \end{bmatrix}. \tag{2-2}$$

2-2 Compressed Sensing Framework

In many real-world signal processing applications, a measured output $\mathbf{y}$ and a measurement model $\mathbf{y} = \mathbf{A}\mathbf{x}$ are used to retrieve information about the unknown signal state $\mathbf{x}$. In order to have a unique solution to the linear system of equations, $\mathbf{A}$ should be of full column rank and $\mathbf{y}$ and $\mathbf{x}$ should be of same size. Otherwise, if $\mathbf{y}$ is larger than $\mathbf{x}$, the system of equations is overdetermined and its solution can be found using least squares. In the CS framework, the case in which $\mathbf{x}$ only has a few non-zero elements is considered. Exploiting this information, the signal can be reconstructed with a number of measurements that is smaller than the original signal size. In order to enable more formal statements and explanations, first a few key mathematical concepts are defined:

2-2-1 The CS reconstruction problem

In CS, a measurement signal $\mathbf{y} \in \mathbb{R}^m$ is generally considered, originating from a measurement model:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{v} \tag{2-3}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the measurement matrix, or sensing matrix, $\mathbf{v} \in \mathbb{R}^m$ a noise vector and $\mathbf{x} \in \mathbb{R}^n$ the signal. Typically, in CS problems, the number of measurements is chosen to be $m \ll n$ such that the signal is compressed. In real-world applications, this can result in a decrease in the duration of measurement acquisition. Specifically, this can decrease the time a patient spends in a MRI machine or the time it takes for a radar to detect an object. Eventually, the original signal $\mathbf{x}$ can be reconstructed from the measurements $\mathbf{y}$ by solving the optimization problem [2]:

$$\begin{aligned} \hat{\mathbf{x}} &= \min_{\mathbf{z}} ||\mathbf{z}||_0 \\ \text{Subject to: } & ||\mathbf{A}\mathbf{z} - \mathbf{y}||_2 \leq \epsilon, \end{aligned} \tag{2-4}$$

for a small tolerance $\epsilon \in \mathbb{R}^+$. Various equivalent descriptions of this problem exist, all in order to comply with a specific optimization framework. Also various ways to tackle this optimization problem exist, but in this thesis, the problem is approached as formulated in (2-4), optimizing the non-convex ℓ_0-norm problem using a greedy approach. Additionally, let's consider measurement matrix $\mathbf{A}$. From (2-3) and (2-4), becomes clear that the reconstruction can only be successful when these matrices are carefully synthesized. Typically, the elements of $\mathbf{A}$ are chosen to be independently drawn from a random distribution. In [11], the performance of the reconstruction problem, is compared for measurement matrices that are sampled from different distributions. One of the key conclusions in [11] is that sampling the entries of $\mathbf{A}$ from a random partial Fourier matrix results outperforms the measurement matrices based on other random distributions such as Gaussian or Bernouli ones.

2-2-2 Reconstruction guarantees

The problem formulation in (2-4) relates to a condition on the number of measurements m. Namely, given the total length and sparsity of a signal, n and s respectively, the minimum number of measurement to be taken to achieve exact reconstruction is given by [2]:

$$m \geq c\, s \log(s/n) \tag{2-5}$$

with $c \in \mathbb{R}^+$ some positive constant, dependent on the data. Another take on reconstruction guarantees is by considering the notion of coherence.

Definition 12. *(Coherence)* Coherence is defined as the normalized order of similarity between the individual columns of the measurement matrix [2]:

$$\mu(\mathbf{A}) = \max_{1 \leq n_1 \neq n_2 \leq I_2} \left| \frac{\langle \mathbf{a}_{n_1}, \mathbf{a}_{n_2} \rangle}{||\mathbf{a}_{n_1}||_2 ||\mathbf{a}_{n_2}||_2} \right| \leq 1 \tag{2-6}$$

where $\mathbf{a}_i$ denotes the i-th column of $\mathbf{A}$.

The coherence is a measure for the quality of the measurement matrix. Intuitively, in the measurement model in (2-3), the elements of $\mathbf{x}$ take a linear combination of the columns of $\mathbf{A}$. So, if the columns of $\mathbf{A}$ are similar, it is harder to identify the reconstruct the signal. From this line of reasoning, a good measurement matrix is one that has a small coherence. The relevance of this property becomes even more obvious, when it is known that coherence directly relates to an upper-bound for the reconstruction error. This relation is different for each reconstruction algorithm. Moreover, in the extreme case of an optimal sensing matrix, the coherence is minimized. In this limiting case, the coherence is solely a function of the dimensions of the sensing matrix. If $\mathbf{A} \in \mathbb{R}^{m \times n}$ [2]:

$$\mu(\mathbf{A}) \geq \mu_{wb}(\mathbf{A}) = \sqrt{\frac{n-m}{m(n-1)}} \tag{2-7}$$

where $\mu_{wb}(\cdot)$ represents the Welch bound, or lower coherence bound for a matrix with certain dimensions. This lower bound is reached if $\mathbf{A}$ is an **Equiangular Tight Frame** or ETF. For further reading on the ETF, please consider [2]. In spite of the ETF achieving optimal coherence and therefore favourable reconstruction guarantees, in real-world applications, the measurement matrices are not always freely designable such that a system practically always operates above the Welch bound. However, the Welch Bound is related to the best performance a CS reconstruction can achieve, given the size of the measurement matrix.
Moreover, in [12] the authors derive reconstruction error bounds that explicitly depend on the coherence of the measurements matrix. These error bounds also depend on the type of algorithm used, the signal sparsity and SNR. Since these bounds are key in investigating the relation between complexity and performance, these will be discussed in depth in later sections.

2-2-3 Greedy Reconstruction Algorithms

In this section, the class of greedy algorithms will be discussed and one of the algorithms belonging to this class will be elaborated further. Greedy algorithms refer to those that iteratively find local optima for the reconstruction problem with the hope of finding a globally optimal solution [2]. These are *greedy* as they select the best available option at each iteration without considering the global consequences of that choice. An example of such an algorithm is the Orthogonal Matching Pursuit (OMP) algorithm. There are a few reasons that motivate elaborating this specific algorithm further. RHis algorithm has been used for a long time in a variety of applications, so a lot of research has already been done that proves the power of the algorithm. As a result, various modifications to the algorithm exist [13, 2, 14] and hence it offers the flexibility of choosing a modification that matches the demands of the specific application. Through this argumentation, the OMP algorithm is chosen to be used for experiments and modifications in the latter of this thesis. Now, the algorithm will be elaborated.

Assume an estimate $\tilde{\mathbf{x}} \in \mathbb{R}^n$ for the reconstruction in (2-4) that is iteratively updated,a measurement matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and noise varianve σ^2. In that case the likelihood of observing $\mathbf{y}$ under hypothesis $\mathbf{x} = \tilde{\mathbf{x}}$ is given by [15]:

$$\mathcal{L}(\tilde{\mathbf{x}}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{||\mathbf{y}-\mathbf{A}\tilde{\mathbf{x}}||^2}{2\sigma^2}},$$

the corresponding log-likelihood by:

$$\log(\mathcal{L}(\tilde{\mathbf{x}})) = -\log(\sqrt{2\pi\sigma^2}) - \frac{||\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}}||^2}{2\sigma^2}.$$

In order to find the maximum, its derivative over $d\tilde{\mathbf{x}}$ is taken and set equal to zero:

$$\frac{d}{d\tilde{\mathbf{x}}}\log(\mathcal{L}(\tilde{\mathbf{x}}) = -\frac{1}{\sigma^2}\mathbf{A}^T(\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}}) = 0,$$

hence the maximum likelihood for $\tilde{\mathbf{x}}$, under the assumption that $\tilde{\mathbf{x}}$ is s-sparse, is given by:

$$\tilde{\mathbf{x}}_{ML} = \arg\min_{\tilde{\mathbf{x}}} |\mathbf{A}^T(\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}})| \tag{2-8}$$

where $\tilde{\mathbf{x}}_{ML}$ is the maximum likelihood estimator of $\mathbf{x}$. In other words, for the current estimate $\tilde{\mathbf{x}}$, the expression $|\mathbf{A}^T(\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}})|$ contains a measure of distance to the optimal estimate $\hat{\mathbf{x}}$ for each element. In the same line of reasoning, the element of $\tilde{\mathbf{x}}$ that is positioned the furthest from $\hat{\mathbf{x}}$ is given by:

$$\arg\max_{j} |\mathbf{A}^T(\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}})|_j. \tag{2-9}$$

The OMP algorithm uses exactly this reasoning. The full algorithm is shown in Algorithm 1[2]:

Algorithm 1 OMP Algorithm

Require: Measurement vector $\mathbf{y} \in \mathbb{R}^m$, CS matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, desired error bound ϵ
1: Initialize iteration parameter $i = 0$, estimate $\hat{\mathbf{x}}_0 = \mathbf{0}$, empty set $S_0 = \varnothing$ and residual $\mathbf{r} \in \mathbb{R}^m = \mathbf{y}$
2: **while** $||\mathbf{r}||_2 > \epsilon$ **and** $i < m$ **do**
3: $i = i + 1$
4: $j_i = \text{argmax}_j |\mathbf{A}^T\mathbf{r}|_j$
5: $S_i = S_{i-1} \cup j_i$
6: $\tilde{\mathbf{x}}_i = \arg\min_{\mathbf{z}}(||\mathbf{y} - \mathbf{A}\mathbf{z}||_2, \text{supp}(\mathbf{z}) \subset S_i)$
7: $\mathbf{r} = \mathbf{y} - \mathbf{A}\tilde{\mathbf{x}}$
8: **end while**
9: **return** $\hat{\mathbf{x}} = \tilde{\mathbf{x}}_i$

In line 4, the algorithm executes the proposed step by selecting the index that is most likely to result in the smallest residual $\mathbf{r}$. Subsequently, in line 5 this index is added to the support set S and finally in line 6, the estimation is updated by optimizing only over the indices within the support set. Once the algorithm has converged, the final estimate is returned. Note that, since in each iteration, there is only a single index added to the support set, the final estimate $\hat{\mathbf{x}}$ can have at most i non-zeros entries.
The computational complexity of the algorithm is decomposed step by step in table Table 2-1 [16].

step	symbol	complexity
4	$\|\mathbf{A}^T\mathbf{r}\|$	$\mathcal{O}(n \times m)$
6	$\arg\min_{\mathbf{z}} \|\|\mathbf{y} - \mathbf{A}\mathbf{z}\|\|_2, \text{supp}(\mathbf{z}) \subset S_n$	$\mathcal{O}(i^3 + 2 \cdot i^2 \times m + m \times i)$
7	$\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}}$	$\mathcal{O}(n \times m + m)$

Table 2-1: Computational complexity of the steps in the i-th iteration of OMP

The complexities of step 4 and 7 consist merely of multiplications. The complexity in line 6 is equivalent to that of the Least-Squares solution:

$$(\mathbf{A}_{S_i}^T \mathbf{A}_{S_i})^{-1} \mathbf{A}_{S_i}^T \mathbf{y}$$

where $\mathbf{A}_{S_i} \in m \times i$ grows one column per iteration. For a reasonable number of iterations i, the cost is dominated by that of the matrix inverse that requires $i^3 + i^2 \cdot m$ multiplications, with the worst case at $i = m$. Hence, by conclusion the complexity of the OMP algorithm is of $\mathcal{O}(i^3)$.
Next, as proposed in subsection 2-2-2, different algorithms have a coherence-based reconstruction guarantee. In [12], the guarantee of OMP is derived. It states that for $\mathbf{v} \sim N(0, \sigma^2 \mathbf{I})$ and sparsity s, given the measurement model:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{v}$$

if it holds that:

$$|\mathbf{x}_{\min}| - (2s - 1)\mu|\mathbf{x}_{\min}| \geq 2\sigma\sqrt{2(1+\alpha)\log m} \tag{2-10}$$

with some positive constant $\alpha \in \mathbb{R}^+$ and $|\mathbf{x}_{\min}|$ is the minimal signal amplitude, then we can define a probability:

$$p(\Lambda_{\mathbf{x}}) \geq 1 - \frac{1}{m^\alpha \sqrt{\pi(1+\alpha)\log m}}, \tag{2-11}$$

where $p(\Lambda_{\mathbf{x}})$ is the probability that the OMP algorithm both identifies the correct support $\Lambda_{\mathbf{x}}$ and respects the error bound:

$$||\hat{\mathbf{x}} - \mathbf{x}||_2^2 \leq \frac{2(1+\alpha)}{(1-(s-1)\mu)^2} s\sigma^2 \log m. \tag{2-12}$$

Note that the condition in (2-10) depends on measurement matrix $\mathbf{A}$ and the signal $\mathbf{x}$ itself. The fact that $|\mathbf{x}_{\min}|$ is used, is to make sure that the signal has sufficient amplitude to overcome under noise. Also note that, since $\mu_{wb} \leq \mu \leq 1$, conclusions about coherence can be drawn based on (2-10) and (2-12) the performance bound increases with increasing μ. Namely, (2-10) is more easily fulfilled if μ is small. (2-12) likewise shows that an increasing μ relates to an increasing performance bound. Hence, as expected, a small coherence is desirable as it achieves tight bounds on the reconstruction quality, this coherence can be as small as the Welch bound in the best case.

2-3 Prior-aware CS

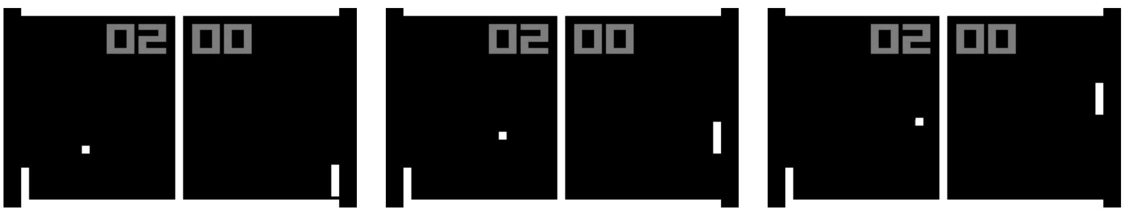
In CS, the sparse property of a signal is exploited, but what if a signal varies slowly along a certain dimension? In that case, is useful to propagate this correlation information to signals alongside this dimension, which might result in an improved reconstruction performance. In this section, some prior-aware CS algorithms from existing literature are discussed.

2-3-1 Dynamical Compressed Sensing

If the signal is temporally correlated, it helps to incorporate some dynamic model that describes the evolution of the signal over time within the algorithm. This is the idea in the Dynamical CS (DCS) framework. A formal definition for DCS does not yet exist as thus far, algorithms are tailored to applications. However, in an attempt to generalize this framework, it can be noted that DCS commonly consists of a normal CS reconstruction problem with the addition of a dynamical prior. A formal description is shown in (2-13)

$$
\begin{aligned}
\hat{\mathbf{x}}_t &= \min_{\mathbf{z}} ||\mathbf{z}||_0 \\
\text{Subject to:} & \\
\mathbf{A}\mathbf{z} &= \mathbf{y} \\
\mathbf{z} &= f(\hat{\mathbf{x}}_{t-1})
\end{aligned}
\tag{2-13}
$$

where the operator $f(\cdot)$ describes some dynamic model. To illustrate this visually, consider Figure 2-1, which shows subsequent snapshots of the classical *Poing* game. The game consists



(a) $t = 0$ **(b)** $t = 1$ **(c)** $t = 2$

Figure 2-1: Subsequent frames of the Poing game

of a static background, two dynamic bats and a dynamic ball. For the dynamic components in the game, a dynamic model can be made that predicts the position and velocity of these components in the next frame. In the figure for instance, it is clear that the ball moves towards the right corner and the right bat moves upwards. If now these frames are now sensed in a compressed way, the dynamics of the bat and the ball can be used to improve the reconstruction of subsequent signal frames.

Numerous publications consider this dynamical problem [17, 18, 19], but these are beyond the scope of this thesis due to their level of tailoredness. The problem as formulated in the research questions is of different origin. Instead of propagating prior information over a temporal dimension, the prior is propagated between different blocks within the signal. Until now, little research has been conducted on this topic.

2-3-2 Logit Weighted OMP

The incorporation of prior information that is not per se time-dependent into a CS problem, is described in [20]. This paper, derives information theoretic measurement bounds, such as in (2-5), based on the amount of prior information available. Besides, it proposes a modification to the OMP Algorithm 1, that directly incorporates this prior information into the algorithm to result in the Logit Weighted OMP (LW-OMP) Algorithm, visualized in Algorithm 2. In this algorithm, the vector $\mathbf{p} \in \mathbb{R}^n$ describes prior probabilities for each element in $\mathbf{x} \in \mathbb{R}^n$ being non-zero. The difference with the original OMP algorithm is visible in lines 4 and 5. In line 4, a weighting factor c_k is calculated. This factor is dependent on the desired sparsity of the solution κ, the iteration number k, a user defined positive constant g and the measurement noise σ:

$$c_k = \frac{g}{2}\left(2(\kappa - k) - 1 + 2\left(\frac{\sigma}{g}\right)^2\right),$$

where the term $\frac{g}{2}$ relates to a heuristically chosen weight factor, $(\kappa - k)$ relates to the number of non-zero components that yet have to be found, and $(\frac{\sigma}{g})^2$ is chosen to be increasing with σ, which imposes more weight on the prior when measurements are noisy and less when they are clear. In line 5, c_k is multiplied with the term $\bar{\mathbf{p}}$ that originates from the column stacking of $\log\left(\frac{p_i}{1-p_1}\right)$ for $i\{1, 2, \ldots, n\}$. This term, that increases with increasing amount of prior information, realizes a contribution of the prior information on the support identification step in line 5. Other weight terms than the c_k in the algorithm are investigated in the paper, but the one listed here was found to be the best among considered. The full derivation of this weight term can be found in the appendix of [20].

Algorithm 2 LW-OMP Algorithm [20]

Require: measurement vector $\mathbf{y} \in \mathbb{R}^m$, matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, desired sparsity level κ, average amplitude of the non-zero components g, prior vector $\mathbf{p} \in \mathbb{R}^n$ that describes the probability of each entry of $\mathbf{x}$ being non-zero.

1: Initialize iteration parameter $k = 0$, estimate $\tilde{\mathbf{x}}_0 = \mathbf{0}$, empty set $S_0 = \varnothing$ and calculate $\bar{\mathbf{p}} \in \mathbb{R}^n$ as the column stacking of $\log\left(\frac{p_i}{1-p_1}\right)$ for $i\{1, 2, \ldots, n\}$
2: **while** $k < \kappa$ **do**
3: $\quad k = k + 1$
4: $\quad c_k = \frac{g}{2}\left(2(\kappa - k) - 1 + 2\left(\frac{\sigma}{g}\right)^2\right)$
5: $\quad j_k = \text{argmax}_j \left(\left||\mathbf{A}^T(\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}})| + c_k\bar{\mathbf{p}}\right|\right)_j$
6: $\quad S_k = S_{k-1} \cup j_k$
7: $\quad \tilde{\mathbf{x}}_k = \arg\min_{\mathbf{z}}(||\mathbf{y} - \mathbf{A}\mathbf{z}||_2, \text{supp}(\mathbf{z}) \subset S_k)$
8: **end while**
9: **return** $\hat{\mathbf{x}} = \tilde{\mathbf{x}}_k$

To illustrate the LW-OMP algorithm, again consider the snapshots from the Poing game in Figure 2-1. Now instead of a dynamical model, a prior can be constructed, for instance by defining a probability distribution based on distance to the ball. The further away from the ball, the less likely that in the subsequent frame, the ball is there. Hence, a prior distribution over the whole signal is constructed that can be taken into account in the support identification of the LW-OMP algorithm and so, information from the reconstruction of one frame is used as prior information for the reconstruction problem of the next frame.

In this thesis, the prior-aware CS problem is solved using LW-OMP, a modification of OMP. The increase in computational complexity by shifting from OMP to LW-OMP is equal to $\mathcal{O}(n + \log n)$, as line 4 requires 5 multiplications, which is negligible, and line 5 has complexity $\mathcal{O}(n + \log(n))$. In the limiting case that the sparsity level close to zero, there is barely any prior information due to which the prior term $\bar{p}$ will be close to zero as well. However, when the values of $\bar{p}$ tend to get larger, this term possibly dominate the support selection in line 5, as the log function may return a very large argument. Also, the parameter g is user-determined and requires a heuristic tweaking procedure in order to find a value that works for the specific application.

2-4 Block Compressed Sensing

The motivation for the synthesis and use of BCS is to reduce memory storage of the traditional CS problem when signals are large. In this section, the general BCS framework is introduced, the storage complexity is analysed and relevant literature is discussed.

2-4-1 BCS framework

Instead of the traditional CS framework, as presented in (2-3) and (2-4), in BCS, the measurements are taken in a block-wise fashion [6, 7, 8]. Formally, consider a signal $\mathbf{x} \in \mathbb{R}^n$, measurements $\mathbf{y} \in \mathbb{R}^m$, noise $\mathbf{v} \in \mathbb{R}^m$, and sensing matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ in the traditional CS framework. In BCS, the problem is divided into β blocks, resulting in β CS sub-matrices, β signal blocks, β measurement blocks and β noise blocks, for the b-th block of the signal, the CS matrix is denoted by $\mathbf{A}_b \in \mathbb{R}^{\frac{m}{\beta} \times \frac{n}{\beta}}$, the b-th signal block by $\mathbf{x}_b \in \mathbb{R}^{\frac{n}{\beta}}$, the b-th measurement block and noise block by $\mathbf{y}_b \mathbf{v}_b \in \mathbb{R}^{\frac{m}{\beta}}$ where $b \in \{1, 2, \ldots, \beta\}$, in BCS, the standard CS measurement model, as in (2-3), can be written as:

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & 0 & \dots & 0 \\ 0 & \mathbf{A}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{A}_\beta \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_\beta \end{bmatrix} + \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_\beta \end{bmatrix}. \tag{2-14}$$

As briefly mentioned, the motivation for development of this framework was to reduce storage complexity. The CS measurement matrix in BCS is block-diagonal, as seen from (2-14).
Note that this property of the $\mathbf{A}$ matrix introduces independence of $\mathbf{y}_b$ on $\mathbf{x}_p$ where $p \neq b$. In other words, the measurement model of (2-14) can be written as:

$$\mathbf{y}_b = \mathbf{A}_b \mathbf{x}_b + \mathbf{v}_b \tag{2-15}$$

for $b \in \{1, 2, \ldots, \beta\}$.

Now assume n to be the number of signal entries and κ determines the compression ratio, such that $m = \kappa n$ is the total number of measurements taken. Then the storage complexity of the ordinary CS matrix is given by $\mathcal{O}(\kappa n^2)$,
i.e., the number of elements in $\mathbf{A}$. The BCS problem requires the storage of β sub-matrices, having a storage complexity of

$$\mathcal{O}(\beta \cdot \kappa \left(\frac{n}{\beta}\right)^2) = \mathcal{O}\left(\kappa \frac{n^2}{\beta}\right)$$

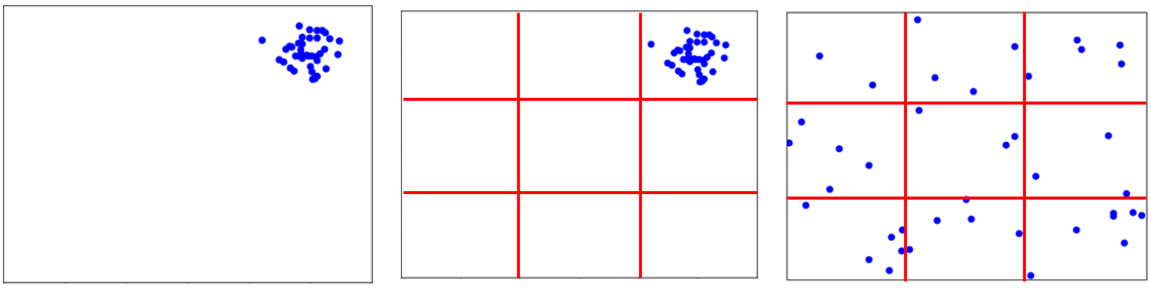
which is decreased by a factor β when compared to the standard CS matrix. The BCS reconstruction problem is then given by

$$\hat{\mathbf{x}}_b = \min_{\mathbf{z}\in\mathbb{R}^n} ||\mathbf{z}||_0 \quad \text{Subject to: } ||\mathbf{y}_b - \mathbf{A}_b\mathbf{z}_b||_2 \leq \epsilon_b \text{ for } b \in \{1, 2, \ldots, \beta\}. \tag{2-16}$$

The reconstruction is executed in block-wise fashion. In [21], the authors already note the advantage in computational complexity of the reconstruction for the BCS framework. However, until now there is no available literature discussing the computational complexity of BCS and the relating reconstruction performance.

2-4-2 The asymmetry problem

In prior work on BCS, the application is often in the field of image processing [6, 8]. However, if sparse components in the image are clustered together, the uniformly partitioned structure of (2-14) will yield asymmetric distribution of the sparse components across the signal partitions. Such a distribution is shown in Figure 2-2.



(a) $\mathbf{X}$ **(b)** Partitions over $\mathbf{X}$ **(c)** Partitions over $\mathbf{\Pi X}$

Figure 2-2: Partitioning in BCS for a sparse image

In this case, all the non-zero components lie within a single block, all the other blocks do not contain any non-zero components. The measurement relating to these "empty" blocks hence do not contribute to the reconstruction problem as a whole. In order to address this problem, the authors of [22] propose an adaptive measurement procedure. This means that based on the original signal, the different signal blocks are measured by taking a different number of measurements. This relates to (2-14) with each $\mathbf{y}_b$ a different number of elements. Hence, more measurements can be dedicated to partitions that contain more non-zero components and vice versa. The advantage of this method, is that in this way, the measurements can be properly distributed over the partitions. However, this method requires extensive prior knowledge of the signal. In image reconstruction, this might be the case, but in real-time measurement acquisition, this assumption cannot be made. Another way to tackle the asymmetry problem is by making sure that each block contains more or less an equal number of sparse components. This idea is presented in [23]. The corresponding framework, relating to (2-14), is given by:

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & 0 & \ldots & 0 \\ 0 & \mathbf{A}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \mathbf{A}_\beta \end{bmatrix} \mathbf{\Pi x} + \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_\beta \end{bmatrix} \tag{2-17}$$

where $\mathbf{\Pi} \in \mathbb{R}^{n\times n}$ is a permutation matrix that accommodates for an arbitrary partitioning structure. By choosing $\mathbf{\Pi}$ at random, $\mathbf{\Pi X}$ is a randomly permuted revision of $\mathbf{X}$. Such permutations *randomly* distribute the non-zero components across different partitions. In [23], the authors propose a strategy, applied on 2-D image data, on how to symetrize the BCS problem. This is done by permuting columns of each 2D block, such that each permuted block contains an equal number of sparse component. Let $\mathbf{X} \in \mathbb{R}^{n_1\times n_2}$ be a signal and $\beta \in \mathbf{N}^+\backslash 0$ the number of partitions such that $\sqrt{\beta} \in \mathbf{N}$. The permutation method proposed in [23], equivalent to $\mathbf{\Pi X}$ in (2-17), can then be explained by:

1. Define initial blocks: $\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2, \dots, \tilde{\mathbf{X}}_\beta \in \mathbb{R}^{\frac{n_1}{\sqrt{\beta}}\times\frac{n_2}{\sqrt{\beta}}}$
2. Create block-flat matrix by concatenating over the first dimension: $\tilde{\mathbf{X}}_{\text{flat}} = [\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2, \dots, \tilde{\mathbf{X}}_\beta] \in \mathbb{R}^{\frac{n_1}{\sqrt{\beta}}\times n_2\sqrt{\beta}}$
3. Derive permutation:
   (a) Initialization: Let there be β number of empty sets and score parameters:
   $$J_1 = J_2 = \cdots = J_\beta = \varnothing,\ s_1 = s_2 = \cdots = s_\beta = 0$$
   (b) ℓ_0-norm sorting: Sort the columns of $\tilde{\mathbf{X}}_{\text{flat}}$ according to their ℓ_0-norm. Save the sorted indices in $\mathbf{k}$ and the sorted signal vector in $\mathbf{b}$:
   $$\begin{aligned}\mathbf{k} &= \arg\text{sort}_j((||\tilde{\mathbf{x}}_{\text{flat}}||_0)_j)\\ \mathbf{b} &= \text{sort}_j((||\tilde{\mathbf{x}}_{\text{flat}}||_0)_j)\end{aligned}$$
   (c) Sort the score variables $s_{w_1} \le s_{w_2} \le \cdots \le s_{w_p}$
   (d) Distribute the first β entries of k to the smallest score parameters and add the corresponding indices to the related sets:
   $$\begin{aligned}s_{w_i} &= s_{w_i} + b_i\ \forall i \in \{1, 2, \dots, \beta\}\\ J_{w_i} &= J_{w_i} \cup k_i; \forall i \in \{1, 2, \dots, \beta\}\end{aligned}$$
   (e) Remove the first β elements from $\mathbf{b}$ and $\mathbf{k}$
   (f) Repeat step 3-5. until $\mathbf{k}$ and $\mathbf{b}$ are empty.
   (g) J_i contains the column indices that are in the i-th partition

The execution of this pseudo-code gives rise to an equal distribution of the sparse components over the different measurement blocks. However, this method has a few drawbacks. First of all, prior information about the signal is required to design a proper permutation strategy. Secondly, this *column-wise* partitioning strategy only works properly if there are enough columns containing non-zero elements. In the limiting case, where one single column contains all the sparse components, a column-wise permutation strategy will not work. Hence the application of this method requires some assumptions on the signal.

2-5 Extension to higher order settings

The standard CS framework and corresponding algorithms work with 1-dimensional signals. In essence, if a signal is of higher dimension, it could be flattened before feeding it through the CS pipeline, but with a higher number of dimensions, the problem size might grow significantly. Another way to approach this, is by extending the CS algorithm to higher orders. In this section, a multi-dimensional description of the already discussed CS frameworks and relevant algorithms are elaborated.

2-5-1 Higher order framework

Reconsider the measurement model and reconstruction problem in (2-3) and (2-4) respectively. Now instead of having a one dimensional signal, consider a signal $\underline{\mathbf{X}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$. In that case the measurement model reads:

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \cdots \times_d \mathbf{A}^{(d)}. \tag{2-18}$$

In which $A^{(k)} \in \mathbb{R}^{n_k \times m_k}\ \forall k \in \{1, 2, \ldots, d\}$ are factor matrices and $\underline{\mathbf{Y}} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$ is the compressed measurements tensor. Using this, the reconstruction problem for higher orders can be defined as:

$$\begin{aligned} \hat{\underline{\mathbf{X}}} &= \min_{\mathbf{Z}} ||\mathbf{Z}||_0 \\ \text{Subject to: } \underline{\mathbf{Y}} &= \underline{\mathbf{Z}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \cdots \times_d \mathbf{A}^{(d)}. \end{aligned} \tag{2-19}$$

Greedy methods to approach this problem are discussed in subsection 2-5-2.

2-5-2 Higher order CS algorithms

To solve the problem in (2-19), various approaches exist. Before, the OMP algorithm was explained in subsection 2-2-3, so in order to keep consistency, a higher order extension to this algorithm is now presented. First of all, recall the definition of the CP-decomposition from subsection 2-1-2. Now assume that a tensor $\underline{\mathbf{X}}$ is sparse, then a CP-decomposition can be fitted on $\underline{\mathbf{X}}$, such that the super-diagonal tensor of the decomposition contains all the sparse components of $\underline{\mathbf{X}}$. In that case, under the assumption that the factor matrices $\mathbf{A}^{(i)}$ remain unchanged, the measurement model can be rewritten using the Khatri-Rao product rule:

$$\begin{aligned} \underline{\mathbf{Y}} &= \underline{\mathbf{\Lambda}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \cdots \times_d \mathbf{A}^{(d)} \\ \Longrightarrow \text{vec}(\underline{\mathbf{Y}}) &= (\mathbf{A}^{(d)} \odot \mathbf{A}^{(d-1)} \odot \cdots \odot \mathbf{A}^{(1)})\text{vec}(\underline{\mathbf{\Lambda}}) \end{aligned} \tag{2-20}$$

where $\underline{\mathbf{\Lambda}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ with $\Lambda_{i_1, i_2, \ldots, i_d} = \begin{cases} \lambda_k & k = i_1 = i_2 = \cdots = i_d \\ 0 & \text{else} \end{cases}$. Using these equalities, a higher order OMP algorithm can be derived. Analogously with the maximum likelihood derivation in subsection 2-2-3, an equivalent description for the identification of the maximum component Equation 2-9 can be extended as:

$$\arg \max_{j_1, j_2, \ldots, j_d} |\underline{\mathbf{R}} \times_1 \mathbf{A}^{(1)T} \times_2 A^{(2)T} \times_3 \cdots \times_d \mathbf{A}^{(d)T}|_{j_1, j_2, \ldots, j_d} \tag{2-21}$$

with $\underline{\mathbf{R}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ the residual corresponding to the current estimate $\tilde{\underline{\mathbf{X}}}$ such that:

$$\underline{\mathbf{R}} = \underline{\mathbf{Y}} - (\underline{\mathbf{X}} \times_1 \mathbf{A}^{(1)} \times_2 A^{(2)} \times_3 \cdots \times_d \mathbf{A}^{(d)}).$$

Henceforward, (2-21) outputs the indices of the component that is most likely to be non-zero. Using (2-20), now the problem can be written towards a least squares problem. The algorithm as a whole is presented in Algorithm 3 [24].

Algorithm 3 K-OMP algorithm [24]

Require: Compressed measurements $\mathbf{Y} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times_d m_d}$, normalized factor matrices $\mathbf{A}^{(k)} \in \mathbb{R}^{m_k \times n_k}$, $k \in \{1, 2, \ldots, d\}$, desired error bound ϵ

Initialize iteration parameter $i = 0$, estimate $\hat{\mathbf{x}}_0 = \mathbf{0}$, empty set $S_0 = \varnothing$ and residual $\underline{\mathbf{R}} = \underline{\mathbf{Y}}$

while $||\underline{\mathbf{R}}||_F^2 > \epsilon$ **and** $i < \Pi_{k=1}^d m_k$ **do**

  $i = i + 1$

  $j_1^{(i)}, j_2^{(i)}, \ldots, j_d^{(i)} = \arg\max_{j_1, j_2, \ldots, j_d} |\underline{\mathbf{R}} \times_1 \mathbf{A}^{(1)T} \times_2 A^{(2)T} \times_3 \cdots \times_d \mathbf{A}^{(d)T}|_{j_1, j_2, \ldots, j_d}$

  $S_k^{(i)} = S_k^{(i-1)} \cup j_k^{(i)} \; \forall k \in \{1, 2, \ldots, d\}$

  $\mathbf{W}^{(k)}(: .i) = \mathbf{A}^{(k)}(:, j_k^{(i)}) \; \forall k \in \{1, 2, \ldots, d\}$

  $\tilde{\mathbf{x}}_i = \arg\min_{\mathbf{z}} ||\text{vec}(\underline{\mathbf{Y}}) - (\mathbf{W}^{(d)} \odot \mathbf{W}^{(d-1)} \odot \cdots \odot \mathbf{W}^{(1)})\mathbf{z}||_F^2$

  $\text{vec}(\underline{\mathbf{R}}) = \text{vec}(\underline{\mathbf{Y}}) - (\mathbf{W}^{(d)} \odot \mathbf{W}^{(d-1)} \odot \cdots \odot \mathbf{W}^{(1)})\mathbf{z}$

end while

return $\{S_1, S_2, \ldots, S_d\}, \mathbf{z}$

The algorithm outputs a vector $\mathbf{z} \in \mathbb{R}^{i_{\text{end}}}$ and index sets $S_k \; \forall k \in \{1, 2, \ldots, d\}$, such that $|S_k| = i_{\text{end}}$ where i_{end} is the iteration at which the algorithm stopped. Now the original signal $\underline{\mathbf{X}}$ can be reconstructed by initializing an empty tensor $\hat{\underline{\mathbf{X}}}$ and fill it in using:

$$\hat{\underline{\mathbf{X}}}_{s_1^{(i)}, s_2^{(i)}, \ldots, s_d^{(i)}} = \mathbf{z}_i$$

where $s_k^{(i)}$ is the i-th element of S_k, $i \in [1, i_{\text{end}}]$ and $k \in \{1, 2, \ldots, d\}$. The computational complexity of this algorithm is decomposed into Table 2-2.

step	symbol	complexity
identification	$\lvert\underline{\mathbf{R}} \times_1 \mathbf{A}_1^T \times_2 A_2^T \times_3 \cdots \times_d \mathbf{A}_d^T\rvert$	$\mathcal{O}(dm^{d-1}n)$
Least-Squares	$\arg\min_{\mathbf{z}} \lvert\lvert\text{vec}(\underline{\mathbf{Y}}) - (\mathbf{W}_d \odot \mathbf{W}_{d-1} \odot \cdots \odot \mathbf{W}_1)\mathbf{z}\rvert\rvert_F^2$	$i^3 + 2 \cdot i^2 \times m^d + m^d \times i$
Residual	$\text{vec}(\underline{\mathbf{R}}) = \text{vec}(\underline{\mathbf{Y}}) - (\mathbf{W}_d \odot \mathbf{W}_{d-1} \odot \cdots \odot \mathbf{W}_1)\mathbf{z}$	m^d

Table 2-2: Decomposed Computational Complexity K-OMP

2-5-3 Higher order prior-aware Block Compressed Sensing

To the best of our knowledge, no prior work exists on combining a higher order CS reconstruction algorithm with a prior-aware approach. Note that, although in DCS, signals are often of higher order, in the reconstruction step they are flattened to fit the 1-D reconstruction algorithm [17, 19, 18]. The same goes for BCS of high dimensional signals, where all the

blocks are flattened before reconstruction [6, 21, 23].
Nevertheless, with the potential of both frameworks clearly shown, it would be interesting to investigate the performance of a higher-order prior-aware BCS algorithm.

Chapter 3

Experiments and result

This chapter contains the contribution in this thesis. section 3-1 gives a summary of the experiments upfront, such that it is clear to which research question the experiments are assigned. Next, section 3-2 contains the main contributions in the form of a preprint of the conference paper. Lastly, section 3-3 discussed some experiments that relate to the research questions, but which did not within the scope of the paper. These are mainly about motivating the choice for partitioning strategies and about the higher order CS setting.

3-1 Block Compressed Sensing experiments setup

The aim of this section is to formally introduce an approach to analyse the computational complexity Block Compressed Sensing and its extensions. The paper in section 3-2, considers some topics only very briefly, these are elaborated here. Also, experiments regarding higher order CS methods, which are not present in the paper, are motivated and synthesized here. The results to these experiments are given in section 3-3.

This section is built-up, by first clarifying the problem that is central in the analysis and experiments in this thesis. Next, subsection 3-1-2 gives an overview of all the experiments that are conducted in this thesis. They are subdivided into those presented in the paper and those presented in section 3-3. The research questions to which the experiments apply are repeated. Lastly, the datasets that are used in the experiments are explained and motivated in subsection 3-1-3.

3-1-1 Problem description

In chapter 2, the trade-off between the complexity and quality of the reconstruction is mentioned for various frameworks. However, the idea that defining a larger number of blocks relates to a decrease in both complexity and quality of the reconstruction hasn't been studied before. The aim in this work is to study this issue by performing theoretical analysis and designing experiments for BCS and its prior-aware a higher order extensions, such that the reconstruction quality can be studied in case computing power or time is restricted. The theoretical analysis is performed in the paper and reconsiders the notion of coherence (2-6) to derive error bounds for BCS. The experiments validating this analysis are discussed next.

3-1-2 Experiments

To properly investigate the properties of BCS, a number of experiments can be designed. These are now presented, divided in those presented in the paper in section 3-2 and those presented in section 3-3.

Within the paper

The experiments in the paper in section 3-2 focus on the trade-off between complexity and performance of the reconstruction in BCS and investigate if this can be improved by incorporating prior information. The latter is done by using the LW-OMP algorithm and designing a pipeline on handling prior information in the BCS framework. The experiments in the paper can be motivated by a recap to the research questions of this thesis:

- *How does reconstruction performance change with the number of blocks in a BCS problem?* - (Sub-question 1)
- *What is the change in computational complexity of the reconstruction when applying a BCS algorithm?* - (Sub-question 3)
- *How can relevant prior information be propagated between different blocks in a BCS reconstruction and at what computational cost?* - (Sub-question 4)

These will be investigated by performing the following experiments:

- Compare the reconstruction performance of a BCS problem with a varying number of blocks, given a fixed SNR, measurement ratio and signal.
- Compare the execution time of a BCS problem with a varying number of blocks, given a fixed SNR, measurement ratio and signal.
- Compare the performance of BCS problems, executed with either traditional CS, or a prior-aware extension.
- Compare the reconstruction execution time of a BCS problem, executed with either traditional CS, or a prior-aware extension.

Auxiliary experiments

As stated in section 1-4, some research conducted in this thesis does not fit in the scope of the paper. Therefore, some analysis and experiments are discussed in section 3-3. These aim to respond to the following research questions:

- *What is a proper strategy to partition a CS problem?* - (Sub-question 2)
- *How does the prior-aware BCS framework extend to higher order CS algorithms?* - (Sub-question 5)

These will be investigated by performing the following experiments:

- Compare the reconstruction performance of a BCS problem with different permutation strategies, given a fixed number of blocks, SNR, measurement ratio and signal.
- Analyse whether it is possible to extend the BCS procedure and LW-OMP algorithm to a higher order setting

3-1-3 Data set

For all the experiments in the paper, a synthetic 4D nearfield communication dataset is used. It is retrieved from [25]. This dataset is suitable for the experiments as it has higher order features, it has a sparse 4D-DFT and the non-zero components are typically clustered together. The latter property is useful, as it emphasizes the weakness in traditional BCS towards clustered signals, as described in section 2-4. The exact signal model and dynamics are outside the scope of this thesis. For experiments in section 3-3, another synthetic dataset is used that creates a simple 2D sparse signal that has its non-zero entries clustered in a circle at a random position.

3-2 Paper

This section presents a preprint of our conference paper that will be submitted to the European Signal Processing Conference (EUSIPCO) 2024. EUSIPCO is a well-established yearly conference in signal-processing related topics. The analysis on the tradeoff between reconstruction quality and the computational complexity is novel and therefore it would make a suitable submission. As mentioned before, the paper makes contributions on studying theoretical reconstruction bounds for BCS and the design of a prior-aware extension to the BCS framework.

A divide-and-conquer approach for Compressed Sensing of large signals

Aron Bevelander*, Kim Batselier*, Nitin Jonathan Myers*
*Delft Center for Systems and Control, Delft University of Technology, The Netherlands
Email: {A.A.Bevelander@student. , K.Batselier@, N.J.Myers@}tudelft.nl

***Abstract*—For high dimensional signals, traditional Compressed Sensing (CS) reconstruction becomes computationally expensive, as algorithms often grow rapidly with the number of signal entries. In Block Compressed Sensing (BCS), a Compressed Sensing (CS) problem is partitioned into several independent sub-problems. However, until now, BCS has merely be approached from a storage complexity perspective. In this paper, theoretical reconstruction bounds are considered for the BCS problem, dependent on the number of blocks and it was found that an increasing number of blocks theoretically decreases the reconstruction performance. Next, the reconstruction performance of the BCS setting is investigated by experiment and finally, a prior-aware method is synthesized that is able to propagate correlation information among the blocks to compensate for the reconstruction loss.**

***Index Terms*—Divide-and-Conquer, Block Compressed Sensing, Prior Aware, Coherence, large arrays**

I. Introduction

Sparse signals can efficiently be stored and reconstructed using compressed sensing (CS) techniques. These are widely used in signal processing applications such as imaging and radar. Due to the use of large arrays in advanced imaging and radar systems, the signals in these applications are high dimensional. With state-of-the-art CS methods growing rapidly in complexity with the number of signal entries, dealing with these signals soon becomes problematic in real-time recovery, as it imposes increased time and energy consumption.

This paper proposes a way to decrease computational complexity of the compressed sensing problem, by taking a divide-and-conquer approach, or taking advantage of the block compressed sensing (BCS) framework [1]. In this framework namely, the CS problems is partitioned into β sub-problems, with each sub-problem having a computational complexity less than $\frac{1}{\beta}$ times the initial complexity. Besides, this paper will demonstrate that the block structure of the CS matrices in BCS influences the theoretical reconstruction guarantees. Particularly, increasing the number of blocks in BCS relates to an expected decrease in reconstruction performance. The problem thus shows a trade-off between complexity and reconstruction quality: a decrease in complexity relates to a decrease in reconstruction performance. Furthermore, using the BCS framework allows for the parallel reconstruction of the sub problems as these are completely independent. In addition, if the sub-problems are solved in series, the reconstruction result of one sub-problem can be used as prior information for the subsequent one, in order to compensate for the performance loss.

The divide-and-conquer approach is naturally related to Block compressed sensing (BCS). Prior work on BCS is driven by the desire for storage complexity reduction [2]–[4]. Reconstruction in the BCS framework works well with the number of sparse components well distributed among the blocks, but when this can not be assumed, for instance in case of clustered data, the reconstruction performance is severely affected. In [5], this problem is addressed by determining the number of measurements taken per block on its expected sparsity level. Implementation of this method, however, is rather complex and requires prior information on the signal. On the other hand, [3] proposes a permutation of the block-columns of a 2-D signal. Here, the authors propose an algorithm that equally distributes the number of sparse components over the blocks. On prior-aware CS, prior work has been published under Dynamical compressed sensing, such as in [6]–[8], exploiting a temporal correlation alongside a signal dimension by incorporating a dynamical model in the reconstruction problem. Besides, in [9] the authors assume a prior that is not per se temporal correlated. Instead, a prior probability for each signal entry is used in the support identification. Lastly, [10] presents empirical evidence for the influence of the partitioning of a CS problem relating to a change in reconstruction performance. However, a more theoretical approach can be formulated, by explicitly considering the coherence of a CS problem versus the number of partitions.

This paper makes a number of contributions. Namely, in section 2B, a computational complexity analysis of the BCS framework is made. Next, in seciton 2C, the theoretical reconstruction performance for the BCS problem is presented in relation with the number of sub-problems and section 3 proposes a pipeline that combines BCS with a prior-aware CS algorithm, such that prior information can be propagated from one partition to another.

Notation: $a \in \mathbb{R}$ denotes a scalar, $\mathbf{a} \in \mathbb{R}^n$ a vector, $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$ a matrix and $\underline{\mathbf{A}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ a d-th order tensor. Indexing starts at 1 and is shown as subscript. The convolution of the tensors $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$ is denoted by $\underline{\mathbf{C}} = \underline{\mathbf{A}} \circledast \underline{\mathbf{B}} \in \mathbb{R}^{k_1 \times k_2 \times \cdots \times k_d}$ where $k_i = \max(n_i, m_i)$ $i \in \{1, 2, \ldots, d\}$, such that the elements of $\underline{\mathbf{C}}$ are given by $c_{i_1,i_2,\ldots,i_d} = \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \cdots \sum_{j_d=1}^{n_d} a_{j_1,j_2,\ldots,j_d} \cdot b_{i_1-j_1,i_2-j_2,\ldots,i_d-j_d}$. Lastly, the Frobenius norm of tensor $\underline{\mathbf{A}} \in \mathbb{R}^{n_1 \times n_2}$ is given by $||\underline{\mathbf{A}}||_F = \sum_{i_1}^{n_1} \sum_{i_2}^{n_2} \cdots \sum_{i_d}^{n_d} |a_{i_1,I_2,\ldots,i_d}|^2$.

II. Divide-and-conquer strategy

A. Motivation

In traditional compressed sensing, a signal $\mathbf{x} \in \mathbb{R}^n$ is measured using a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ under noise $\mathbf{v} \in \mathbb{R}^m$, obtaining measurements $\mathbf{y} \in \mathbb{R}^m$ using:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{v} \tag{1}$$

Afterwards, the original signal can be reconstructed using the compressed measurements:

$$\hat{\mathbf{x}} = \min_{\mathbf{z}\in\mathbb{R}^n} ||\mathbf{z}||_0 \quad \text{Subject to: } ||\mathbf{y} - \mathbf{A}\mathbf{z}||_2 \leq \epsilon \tag{2}$$

with a small positive number ϵ. Let's now consider the widely-used class of *greedy algorithms* for this reconstruction. These algorithms are known to have an order of complexity of at least $\mathcal{O}(m \times n)$, originating from an indispensable matrix-vector multiplication. In case of a high dimensional signal, n and m both increase rapidly, having a devastating influence on the computational complexity of the reconstruction and hence on the reconstruction time. This phenomenon makes CS reconstruction for high dimensional signals infeasible for real-time applications.

B. Computational Complexity for BCS

In this work, the complexity problem is addressed by dividing the problem into smaller ones. This divide-and-conquer approach naturally relates to the BCS framework. Until now, BCS has merely been approached from a storage complexity point of view, so let's go about its computational complexity properties. For a block-diagonal matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, a signal $\mathbf{x} \in \mathbb{R}^n$, measurements $\mathbf{y} \in \mathbb{R}^m$, noise $\mathbf{v} \in \mathbb{R}^m$ and the number of blocks β, the measurement model of (1) translates into the BCS measurement model:

$$\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \vdots\\ \mathbf{y}_\beta\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1 & 0 & \dots & 0\\ 0 & \mathbf{A}_2 & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0\\ 0 & \dots & 0 & \mathbf{A}_\beta\end{bmatrix} \mathbf{\Pi}\mathbf{x} + \begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\\ \vdots\\ \mathbf{v}_\beta\end{bmatrix} \tag{3}$$

where $\mathbf{A}_b \in \mathbb{R}^{\frac{m}{\beta}\times\frac{n}{\beta}}$, and $\mathbf{y}_b, \mathbf{v}_b \in \mathbb{R}^{\frac{m}{\beta}}$ denote the b-th blocks of the corresponding-arrays for $b \in \{1, 2, \dots, \beta\}$. The matrix $\mathbf{\Pi} \in \mathbb{R}^{n\times n}$ is a permutations matrix that arbitrarily $\mathbf{x}$. Note that in each row and each column of $\mathbf{\Pi}$, only a single 1 can be present, the rest of the matrix is zero. Where the storage complexity of the CS measurement model in (1) was $\mathrm{O}(m \times n)$, its BCS equivalent is decreased to $\beta \cdot \mathrm{O}(\frac{m}{\beta} \times \frac{n}{\beta})$, yielding a total decrease of a factor β. If next, the reconstruction problem for BCS, equivalent to (2), with $\bar{\mathbf{x}}_b \in \mathbb{R}^{\frac{n}{\beta}}$, $\mathbf{y}_b, \mathbf{v}_b \in \mathbb{R}^{\frac{m}{\beta}}$ and $\mathbf{A}_b \in \mathbb{R}^{\frac{m}{\beta}\times\frac{n}{\beta}}$ as the b-th permuted signal block, measurement block, noise block and measurement sub-matrix respectively, is defined as:

$$\hat{\bar{\mathbf{x}}} = \min_{\mathbf{z}\in\mathbb{R}^n\beta} ||\mathbf{z}||_0 \quad \text{Subject to: } ||\mathbf{y}_b - \mathbf{A}_b\mathbf{z}||_2 \leq \epsilon \tag{4}$$

Then, the computational complexity, again for the class of greedy algorithms, is equal to β times the complexity of the sub-problem, or $\beta \cdot \mathrm{O}(\frac{m}{\beta} \times \frac{n}{\beta})$. Compared to the computational complexity of (2), again a decrease of a factor β. Finally, as is clear from (4), the sub-problems are independent, enabling them to be reconstructed in parallel. In the best case, all β sub-problems can be handled in parallel, yielding a complexity decrease of another factor β, or β^2 with respect to the traditional CS reconstruction.

C. Coherence-based reconstruction guarantees

With the previous section showing that the computational complexity can be decreased using BCS, now the reconstruction guarantees for BCS are analysed using coherence as a tool. This guarantee relates directly to an upper bound of the reconstruction errors when BCS is executed Orthogonal Matching Pursuit (OMP) [11], which belongs to the class of greedy algorithms. First of all consider the definition of the mutual coherence of matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ [12]:

$$\mu(\mathbf{A}) = \max_{1\leq m\neq n\leq I_2} \left| \frac{\langle \mathbf{a}_m, \mathbf{a}_n\rangle}{||\mathbf{a}_m||_2 ||\mathbf{a}_n||_2} \right| \leq 1 \tag{5}$$

Note that it is lower bounded by the Welch [13] bound:

$$\mu(\mathbf{A}) \geq \sqrt{\frac{n-m}{m(n-1)}} \tag{6}$$

which applies in case $\mathbf{A}$ is an equiangular tight frame [14], or a collection of unit vectors where the pairwise inner products have equal magnitudes. A small coherence generally relates to a good choice for a sensing matrix and thus favourable reconstruction guarantees. Note that the Welch bound in (6) only depends on the size of the measurement matrix. Note that this bound relates to the traditional CS framework of (1) and (2). When applying the divide-and-conquer method, equivalent to (4), the new expression for the coherence upper bound becomes:

$$\begin{aligned}\mu(\mathbf{A}) &= \max_{i\in\{1,2,\dots,\beta\}} \mu(\mathbf{A}_b)\\ &\leq \sqrt{\frac{\frac{n}{\beta} - \frac{m}{\beta}}{\frac{m}{\beta}(\frac{n}{\beta} - 1)}}\end{aligned} \tag{7}$$

which only depends on β for fixed dimension sizes m and n. This implies the possibility to consider the coherence as a function of the number of blocks, which makes it clear that an increase in the number of blocks relates to a monotonic increase in the coherence lower bound. Using contributions from [11], this can be linked to a lower bound of the reconstruction performance of OMP. The NMSE error bound is given by:

$$||\hat{\mathbf{x}} - \mathbf{x}||_2^2 \leq \frac{2(1+\alpha)}{(1-(s-1)\mu)^2} s\sigma^2 \log m \tag{8}$$

given that the sparse components in $\mathbf{x}$ satisfy a condition on their magnitude, ensuring that they can be separated from the present noise. Hence, the relation between the NMSE upper bound and the number of blocks can be visualized for the OMP algorithm, as is done in Fig. 1. From this figure, it seems that reconstruction guarantees rapidly worsen with increasing number of blocks. Besides, it shows that, for a small number of measurements m, the bound is less affected by an increasing number of blocks.

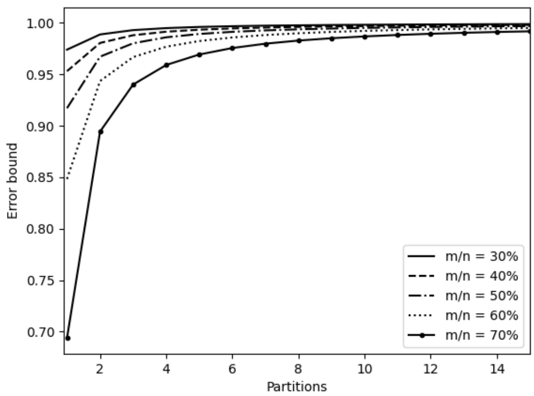


Fig. 1: NMSE upper bound in (7) as a function of the number of blocks for the OMP algorithm

III. PRIOR INCORPORATION

Reconsider Fig. 1 and the conclusion that the divide-and-conquer approach increases the reconstruction upper bound. To compensate for this loss in reconstruction quality, correlation in the sparse signal can be exploited and used in the reconstruction. This section aims to derive a framework that is able to perform BCS reconstruction in series, while propagating prior information from one sub-problem to another. To do so, a high level overview of the steps, used in the framework are mentioned and visualized.

1) *Initialize*: Consider a signal $\mathbf{X} \in \mathbb{R}^{4\times 4}$ and a number of partitions $\beta = 4$.
2) *Partitioning*: A partitioning strategy is chosen, such that all four partitions contain 4 signal entries. In Fig. 2a, each square represents a single signal entry. The entries filled wit the same icon are assigned to the same partition.
3) *Measurement process*: For all partitions, compressed measurements are taken using the measurement model in (3).
4) *Reconstruct one partition* The entries belonging to an arbitrary first partition are reconstructed, visible in Fig. 2b. The information used for the reconstruction here, is merely the compressed measurements that relate to the signal entries in the specific partition,
5) *Collect prior information*: The current reconstruction is convoluted with a kernel describing statistical correlation of the signal. In Fig. 2, the correlation kernel is of size $\mathbb{R}^{3\times 3}$ and the gray-scale relates to the amount of prior information available.
6) *Reconstruct one partition using measurement and prior information*: Using the available prior information, first the partition that has most prior information available is chosen. Next, the reconstruction problem for this partition is solved. Sources of information are both measurements and the available prior information.
7) Repeat steps 5 and 6 until all partitions are reconstructed.

A. Development

To develop the above enumeration into a concrete framework, a couple of questions need answering. First of all, in bullet 2, a partitioning strategy is mentioned, but what strategy should be implemented, or how should matrix $\mathbf{\Pi}$ in (3) be designed? Next, in bullet 5, a correlation kernel is mentioned. How is such a kernel synthesized and why is it valid to be used? Finally, bullet 6 mentions a reconstruction algorithm to which both measurement and prior information are fed. What algorithm is well-suited for the execution of this task?

To start with the first question, finding a right permutation strategy is, as mentioned before, a well-known question within BCS. In this paper the permutations are done following a comb-like structure as is introduced in [15]. This permutation strategy is chosen as it doesn't require prior up-to-date information about the signal and works well in the case of a clustered signal. The comb-like structure is equivalent to that of Fig. 2a, where clearly the periodical *comb-like* pattern arises.

Also, the last question can be answered easily. The Logit-Weight OMP (LW-OMP) algorithm [9] namely, is a simple modification to OMP. Besides measurements, it takes prior information and a single hyper-parameter as inputs and is therefore a suitable candidate. The difference with OMP is only an additive term in the support identification step. For measurement matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{r}_k$ the residual at iteration k, $\mathbf{p} \in \mathbb{R}^n$ a vector containing the prior information and c_{k+1} a constant depending on the iteration number k, define $\bar{\mathbf{p}}$ to be the column stacking of $\log\left(\frac{p_i}{1-p_1}\right)$ for $i\{1, 2, \dots, n\}$. the LW-OMP support identification is given by:

$$j_{k+1} = \arg \max_{i\in\{1,2,\dots,n} \left| |\mathbf{A}^T \mathbf{r}_k| + c_{k+1}\bar{\mathbf{p}} \right|_i \tag{9}$$

Clearly, without the addition of $c_{k+1}\bar{\mathbf{p}}$, the support identification is equal to that of ordinary OMP.

Lastly, the question concerning the correlation kernel requires more elaborate reasoning. The kernel is created using a data-driven approach in a deterministic way. The creation of the kernel requires one, or a series of signals that are similar to those to be reconstructed. Its design process is summarized in the following pseudo-code: This simple algorithm calculates

Algorithm 1 Correlation Kernel creation

Require: Signal $\underline{\mathbf{X}} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ with non-zero entries having indices in set S.
Initialize empty kernel $\underline{\mathbf{\Theta}} = \underline{\mathbf{0}} \in \mathbb{R}^{3^{(1)}\times 3^{(2)}\times\cdots\times 3^{(d)}}$ and define its middle element as $\underline{\mathbf{\Theta}}[2, 2, \dots, 2]$.
for *Index* in S **do**
    Put $\underline{\mathbf{\Theta}}$ on the signal such that its middle element coincides with *Index*
    All entries in $\underline{\mathbf{\Theta}}$ that coincide with a non-zero in the signal are raised by 1, the middle element is kept at zero.
end for
return $\bar{\underline{\mathbf{\Theta}}} = \frac{\mathbf{\Theta}}{\text{card}(S)}$

the probability of having a neighbour in a certain direction, given that the current position is a non-zero element. This is done by calculating the number of neighbouring occurrences in line 5, and normalizing by the number of non-zeros in line 7. The algorithm can be extended to work with a series of signals, to obtain a more accurate representation.

B. Prior-Aware Block Compressed Sensing

Now that all questions are answered, the Prior-Aware block-compressed sensing (PA-BCS) framework can be formalized further.

Algorithm 2 PA-BCS algorithm

Require: Signal $\underline{\mathbf{X}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, call $N = \Pi_{i=1}^{d} n_i$ the total number of elements in $\underline{\mathbf{X}}$. a permutation strategy $\mathbf{\Pi} \in \mathbb{R}^{N \times N}$, a number of partitions β, a measurement rate γ, a correlation kernel $\underline{\mathbf{\Theta}} =\in \mathbb{R}^{3^{(1)} \times 3^{(2)} \times \cdots \times 3^{(d)}}$ and a sparsity level s.

1: Initialize: make $\underline{\mathbf{X}}$ a vector $\mathbf{x} \in \mathbb{R}^N$, define index set $K = \{1, 2, \ldots, \beta\}$ and prior space $\underline{\mathbf{P}} = s \cdot \underline{\mathbf{1}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ with its vector equivalent $\mathbf{p} \in \mathbb{R}^N$, set iteration parameter $k = 1$

2: Take measurements according to (3). Save measurements $[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_\beta]$ and matrices $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_\beta$. Define prior vector blocks as $[\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_\beta] = \mathbf{\Pi}\mathbf{p}$

3: **while** K > 0 **do**

4: Solve partition k using LW-OMP [9]: $\hat{\mathbf{x}}_k =$ LW-OMP$(\mathbf{y}_k, \mathbf{A}_k, \mathbf{p}_k)$

5: Update prior space $\underline{\mathbf{P}} = \underline{\mathbf{P}} + \underline{\mathbf{X}}_k \circledast \underline{\mathbf{\Theta}}$

6: remove k from set K

7: Calculate best prior match $k = \arg\max_{j \in K} \Sigma|\mathbf{p}_j|$

8: **end while**

9: **return** $\hat{x} = \mathbf{\Pi}^{-1}[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\beta]^T$

In line 2, the inputs are made ready for the BCS measurement process. Furthermore, a *prior space* is initialized. This is a tensor of equal size as $\underline{\mathbf{X}}$, that saves prior probabilities of each entry of $\underline{\mathbf{X}}$ being non-zero. The prior space is initialized at the global sparsity level for each element. In line 5, the partial solution $\hat{x}_k$ is used to update the prior space. This is done by applying a convolution with the correlation kernel $\underline{\mathbf{\Theta}}$. Next, in line 7, the next partition is selected. This is done by only considering the leftover partitions and look for the partition for which the most prior information is available. Finally, in line 9 the result is regained by multiplying the inverse of the permutation matrix with the stacked partial solutions.

The computational complexity of this prior aware method is different than that of the parallel, prior-less method. The difference is, on one hand, that the support identification in the LW-OMP algorithm requires the pointwise calculation of the logarithm function, as was shown in (9), or an increase of $\mathrm{O}(\log(\frac{n}{\beta}))$. On the other hand, the prior space has to be updated iteratively, with the convolution as its main contribution to the computational complexity. The prior space has to be updated $\beta - 2$ times, not for the first and last iteration. Each convolution is of complexity $\mathrm{O}(3^d \times \frac{N}{\beta})$, yielding a total complexity of $\mathrm{O}(3^d \times \frac{(\beta-2)N}{\beta})$, which increases linearly in the number of signal entries.

IV. Experimental results

For the numerical simulations, a 4th order tensor is used that is inspired on a nearfield communication signal processing application that is taken from [16]. The exact signal analysis behind it is beyond the scope of this paper. The signal has a clustered, sparse 4D-DFT and is thus useful to investigate the performance of BCS with respect to both reconstruction and computational performance.

Ihe simulations the measurement mode in (3) is applied where the noise term $\mathbf{v}$ is adjusted such that a certain SNR is obtained. As a benchmark, a $\beta = 1$ reconstruction is used, meaning that the signal is flattened to become a 1D array and afterwards the traditional OMP, is applied.

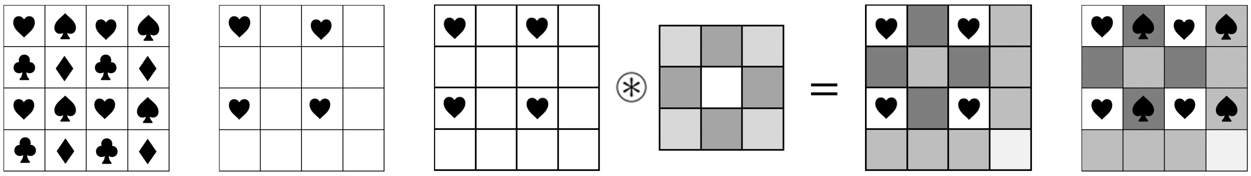
From Fig. 3, a number implications can be derived. First of all. for low SNR both the OMP and LW-OMP algorithms perform similar. In this SNR range, support identification is more probable to fail than in high SNR due to the presence of noise. This failure corrupts the prior space in Algorithm 2 line 5 as it is based on a falsely identified support. Hence, prior information cannot be effectively exploited. Next, in high SNR, the measurements already contain accurate information on the signal support and hence the prior information is not able to enhance this further. In the region in between however, the LW-OMP outperforms the traditional OMP by around 2 dB. In other words, when the LW-OMP algorithm is applied in this SNR domain, the power of the signal can be chosen a significant 25% smaller compared to the case in which the signal is reconstructed using traditional OMP.

Besides, Fig. 4 shows the reconstruction performance of the two algorithms for different measurement rations. This ratio μ is defined as the fraction of measurements of the number of signal entries, or in case measurements $\mathbf{y} \in \mathbb{R}^m$ and signal $\mathbf{x} \in \mathbb{R}^n$, then $\mu = \frac{m}{n}$. From this graph, it can be seen that, while the reconstruction improves for higher measurement ratios, the performance difference between the LW-OMP and the OMP algorithms decreases. Again, this is due to the fact that the prior is of less importance in this range, as the measurements result in an accurate reconstruction by themselves,

Finally, let's discuss the reconstruction time. Based on the statements about computational complexity in the previous chapter, the expected reconstruction time should increase superlinear with increased signal size. The results are shown in table 1. The traditional solution for $\beta = 1$, so $n = 16 \cdot 10^3$ using OMP took on average $18.2 \cdot 10^8$ ms or 30 minutes. The table shows that, indeed the computational time grows

Partitions	t_{OMP} $[ms]$	t_{LW-OMP} $[ms]$
16	5910	9827
64	476	1469

TABLE I: Simulation times for the two algorithms on a CS problem of size $\underline{\mathbf{X}} \in \mathbb{R}^{16 \times 16 \times 8 \times 8}$ with measurement fraction $\frac{m}{n} = 0.4$



(a) Comb-like partitioning strategy of a signal $\mathbf{X} \in \mathbb{R}^{4\times4}$

(b) Reconstruction of the first partition

(c) Creation of prior information by convolution of first partition result with a correlation kernel

(d) Reconstruction of the next partition based on amount of available prior information

Fig. 2: Prior-aware block compressed sensing pipeline

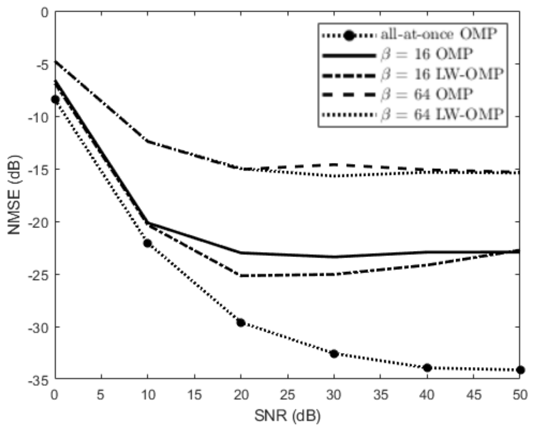


Fig. 3: NMSE for different Block sizes using traditional OMP and the newly proposed LW-OMP method, using signal $\underline{\mathbf{X}} \in \mathbb{R}^{16\times16\times8\times8}$ and measurement fraction $\frac{m}{n} = 0.4$

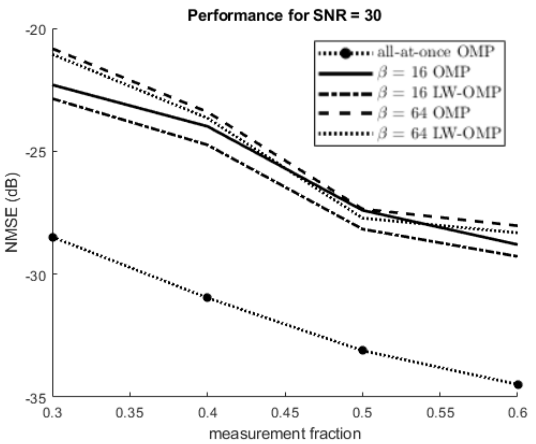


Fig. 4: Reconstruction performance for OMP and LW-OMP method for different measurement fractions

superlinear with the problem size. For the traditional OMP simulation times, the problem size grows wih a factor 4 and computational time with a factor around 12. Also note that, the larger β, the more convolutions are required to update the prior space. Hence, the computational complexity for LW-OMP decreases slower than that of OMP.

V. Conclusions

In this paper, the BCS framework is approached from a computation complexity point of view. This is done by deriving a theoretic relationship between the BCS problem and the reconstruction performances. Next, a pipeline is presented that solves the partitions within the BCS problem one by one, while propagating prior information along the partitions. This pipeline was shown to yield an increase in reconstruction quality for a specific SNR range. Follow-up research could be conducted tailoring a prior aware reconstruction method, such as the LW-OMP, to the task of propagating prior information within a BCS problem, such that the propagation of prior information among the partitions yields a larger performance increase.

References

[1] L. Gan, "Block compressed sensing of natural images," in *2007 15th International Conference on Digital Signal Processing*, 2007, pp. 403–406.

[2] J. CHEN, K.-X. SU, W.-X. WANG, and C.-D. LAN, "Residual distributed compressive video sensing based on double side information," *Acta Automatica Sinica*, vol. 40, no. 10, pp. 2316–2323, 2014.

[3] B. Zhang, Y. Liu, J. Zhuang, K. Wang, and Y. Cao, "Matrix permutation meets block compressed sensing," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 69–78, 2019.

[4] L. Zhu, H. Song, X. Zhang, M. Yan, T. Zhang, X. Wang, and J. Xu, "A robust meaningful image encryption scheme based on block compressive sensing and svd embedding," *Signal Processing*, vol. 175, p. 107629, 2020.

[5] J. Chen, X. Zhang, and H. Meng, "Self-adaptive sampling rate assignment and image reconstruction via combination of structured sparsity and non-local total variation priors," *Digital Signal Processing*, vol. 29, pp. 54–66, 2014.

[6] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5270–5284, 2013.

[7] M. S. Asif and J. Romberg, "Dynamic updating for sparse time varying signals," in *2009 43rd Annual Conference on Information Sciences and Systems*, 2009, pp. 3–8.

[8] J. Ji and T. Lang, "Dynamic mri with compressed sensing imaging using temporal correlations," in *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2008, pp. 1613–1616.

[9] J. Scarlett, J. Evans, and S. Dey, "Compressed sensing with prior information: Information-theoretic limits and practical decoders," *IEEE Transactions on Signal Processing*, vol. 61, p. 427, 01 2013.

[10] R. Pournaghshband and M. Modarres-Hashemi, "A novel block compressive sensing algorithm for sar image formation," *Signal Processing*, vol. 210, p. 109053, 2023.

[11] Z. Ben-Haim, Y. C. Eldar, and M. Elad, "Coherence-based performance guarantees for estimating a sparse vector under random noise," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5030–5043, 2010.

[12] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, 2013.

[13] P. D. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, June 1967.

[14] M. A. Sustik, J. A. Tropp, I. S. Dhillon, and R. W. Heath, "On the existence of equiangular tight frames," *Linear Algebra and its Applications*, vol. 426, no. 2, pp. 619–635, 2007.

[15] H. Masoumi, N. J. Myers, G. Leus, S. Wahls, and M. Verhaegen, "Structured sensing matrix design for in-sector compressed mmwave channel estimation," 2022.

[16] E. Torkildson, U. Madhow, and M. Rodwell, "Indoor millimeter wave mimo: Feasibility and performance," *IEEE Transactions on Wireless Communications*, vol. 10, pp. 4150–4160, 12 2011.

3-3 Auxiliary Results & Analysis

In this section, analysis is performed that relate to contributions outside the scope of the paper in the previous section. These contributions are mainly about the influence of different partitioning strategies on the BCS reconstruction. This section is started by subsection 3-3-1, in which an experiment demonstrating the influence of partitioning strategies to the performance of BCS reconstruction. This is done by comparing different strategies for a range of SNR values with respect to reconstruction quality. These experiments are done using a 2D dataset, as this enables straightforward visual interpretation of the results.

Finally, in section 3-3-2 a higher order extension of LW-OMP to higher order is proposed and its relevance is discussed with respect to the BCS framework

3-3-1 Permutation strategies in BCS reconstruction

Recall the permuted BCS problem as explained in section 2-4, where the BCS measurement model was given by:

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & 0 & \dots & 0 \\ 0 & \mathbf{A}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{A}_\beta \end{bmatrix} \mathbf{\Pi x} + \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_\beta \end{bmatrix}. \tag{3-1}$$

The aim of the experiments in this subsection, is to show the influence of the choice of $\mathbf{\Pi}$ on the reconstruction performance. This is done by first, defining some permutation strategies, then discuss their expected reconstruction performance and finally by performing the experiments and interpreting the results. The scope of the analysis and results in this subsection is on 2D data, as this leads to interpretable results and doesn't complicate the permutation-based strategies too much. Note that for the experiments, 2D signals are used that have clustered non-zero components in a certain region of the signal. This is done to give a fair measure of how the strategies perform with respect to the asymmetry problem, as introduced in subsection 2-4-2.

Partitioning strategies

In Figure 2-2, the need for permuting a BCS problem is motivated visually, but how do different strategies relate to reconstruction performance? To answer this question, first a few strategies should be introduced. Note that all these strategies are based on a 2D signal $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ that has to be partitioned into $\beta \in \mathrm{N}^+$ blocks such that they relate to:

$$\mathbf{\Pi X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_\beta \end{bmatrix}.$$

Definition 13. *(Uniform partitioning)* In uniform partitioning, a signal is cut into pieces without having its columns permuted. The blocks $\mathbf{X}_b \in \mathbb{R}^{\frac{n_1}{\sqrt{\beta}} \times \frac{n_2}{\sqrt{\beta}}}$ can then be defined as:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_{\sqrt{\beta}} \\ \mathbf{X}_{\sqrt{\beta}+1} & \mathbf{X}_{\sqrt{\beta}+2} & \cdots & \mathbf{X}_{2\sqrt{\beta}} \\ \vdots & \vdots & & \vdots \\ \mathbf{X}_{\beta-\sqrt{\beta}+1} & \mathbf{X}_{\beta-\sqrt{\beta}+2} & \cdots & \mathbf{X}_{\beta} \end{bmatrix}. \tag{3-2}$$

Definition 14. *(Random permutation)* The random permutation strategy of a 2D signal is taken by performing uniform partitioning and afterwards permuting the columns of the blocks in a random fashion. This is done by first, defining a flat block-wise matrix, based on the uniform partitioning in (3-2):

$$\mathbf{X}_{\text{flat}} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_{\beta} \end{bmatrix} \Leftarrow \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_{\sqrt{\beta}} \\ \mathbf{X}_{\sqrt{\beta}+1} & \mathbf{X}_{\sqrt{\beta}+2} & \cdots & \mathbf{X}_{2\sqrt{\beta}} \\ \vdots & \vdots & & \vdots \\ \mathbf{X}_{\beta-\sqrt{\beta}+1} & \mathbf{X}_{\beta-\sqrt{\beta}+2} & \cdots & \mathbf{X}_{\beta} \end{bmatrix}.$$

Next, by randomly permuting the columns of $\mathbf{X}_{\text{flat}} \in \mathbb{R}^{\frac{n_1}{\sqrt{\beta}} \times n_2\sqrt{\beta}}$:

$$\bar{\mathbf{X}}_{\text{flat}} = \mathbf{X}_{\text{flat}}[:, \text{shuffle}(\{1, 2, \ldots, n_2\sqrt{\beta}\})]$$

And finally be redefining the blocks:

$$\begin{bmatrix} \bar{\mathbf{X}}_1 & \bar{\mathbf{X}}_2 & \cdots & \bar{\mathbf{X}}_{\beta} \end{bmatrix} = \bar{\mathbf{X}}_{\text{flat}}$$

such that $\bar{\mathbf{X}}_b$ $b \in \{1, 2, \ldots, \beta\}$ are the resulting permuted signal blocks.

Definition 15. *(Optimal permutation)* The optimal permutations strategy is already elaborated in subsection 2-4-2. To summarize, it is similar to the *random permutation* strategy, but instead of shuffling the columns of $\mathbf{X}_{\text{flat}}$ randomly, the blocks are created by assignment of the columns, based on the number of sparse components within each column.

Definition 16. *(Comb-like partitioning)* Inspired from [26], a signal can be permuted by taking elements via a comb-like pattern. To visualize, an toy example is shown in Figure 3-1, where all elements having the same colour are assigned to the same signal block. The application of the comb-like pattern has a number of advantages to it. First of all, it does not depend on prior knowledge of the signal, making it a suitable choice for real-time measuring. Next, the comb-like structure is naturally resilient to clustered data, as neighbouring signal entries are likely to be assigned to a different block. Also, the pattern is based on a real-world application, which makes is useful to investigate.

Experiment design

An experiment is designed to demonstrate the performance difference between the permutation strategies. The dataset used for this experiment is a clustered one in order to be able

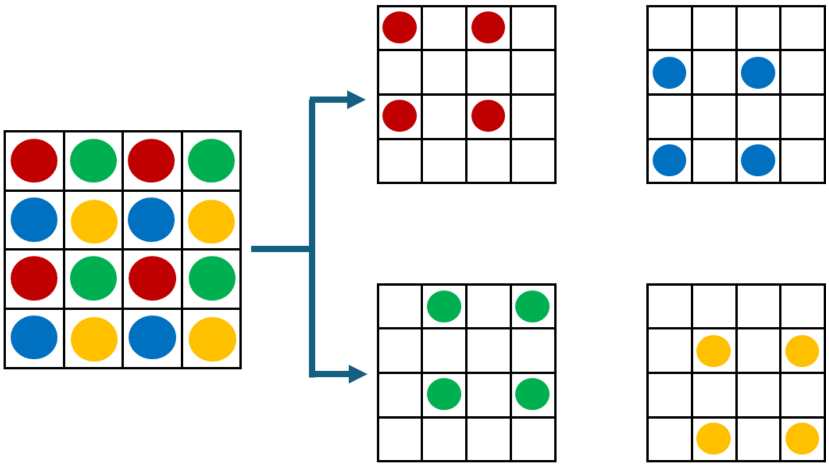


Figure 3-1: Comb-like pattern to define partitions

to see how different strategies cope with the asymetry problem. These experiments comprise executions of the BCS problem for different numbers of blocks and noise levels. Note that, in order to make a fair comparison, the *optimal permutation* strategy is based on a signal, but not the same signal used in the experiment.

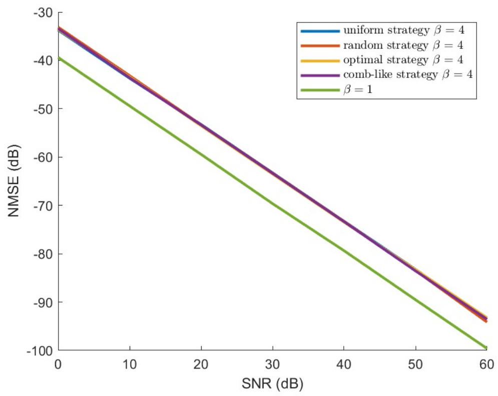
All experiment are performed with clustered sparse a signal $\mathbf{X} \in \mathbb{R}^{28 \times 28}$ and a measurement ratio of $\frac{m}{n} = 0.7$.

Results & Discussion

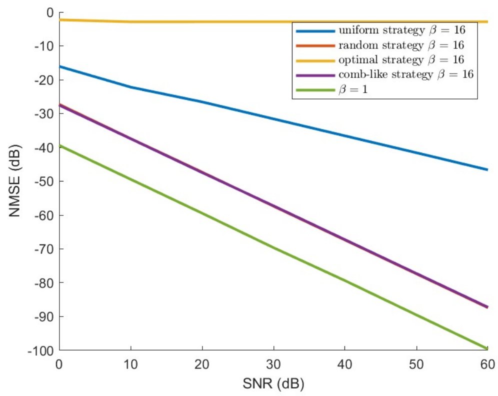
The experiment outcomes are presented in two different ways. In Figure 3-2 the reconstruction performance of classical BCS between is shown for different numbers of blocks. In Figure 3-3 the performance difference is shown for different partitioning strategies.

First of all, it strikes form Figure 3-2 that when the partitions are large, i.e. $\beta = 4$ and $n = 196$, each strategy is able to perform accurate reconstruction. When the partitions are very small, i.e. $\beta = 16$, only the comb-like permutation strategy is able to perform a somewhat accurate reconstruction. Furthermore, surprisingly, it seems from Figure 3-3 that the optimal permutation strategy does not yield a good reconstruction performance. This is probably because of the fact that the *optimal strategy*, for which detailed prior information about the signal is needed, was not supplied. Instead, the strategy was based on a similar signal and apparently, this doesn't supply the strategy with adequate information to create a decent strategy.

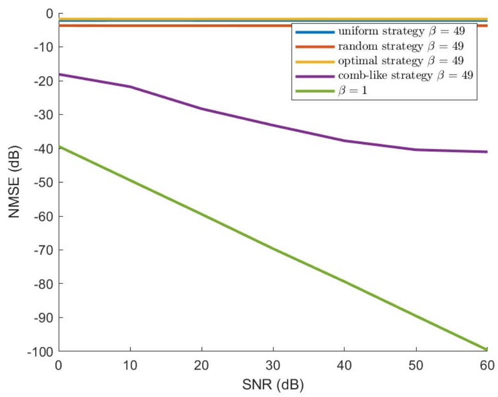
All together, it can be concluded that, compared to the other strategies listed in subsection 3-3-1, a comb-like permutation strategy yields the best performance in term of reconstruction performance. Besides, it has the advantage that it can cope with clustered data without having prior information about the signal itself. Also, Figure 3-2 and Figure 3-3 clearly confirm the need of a permuting a signal before feeding it into a BCS algorithm, skipping the permutation step would induce performance equivalent to that of the *uniform partitioning* strategy.



(a) $\beta = 4$ blocks. All strategies have sufficient information to yield an accurate reconstruction.



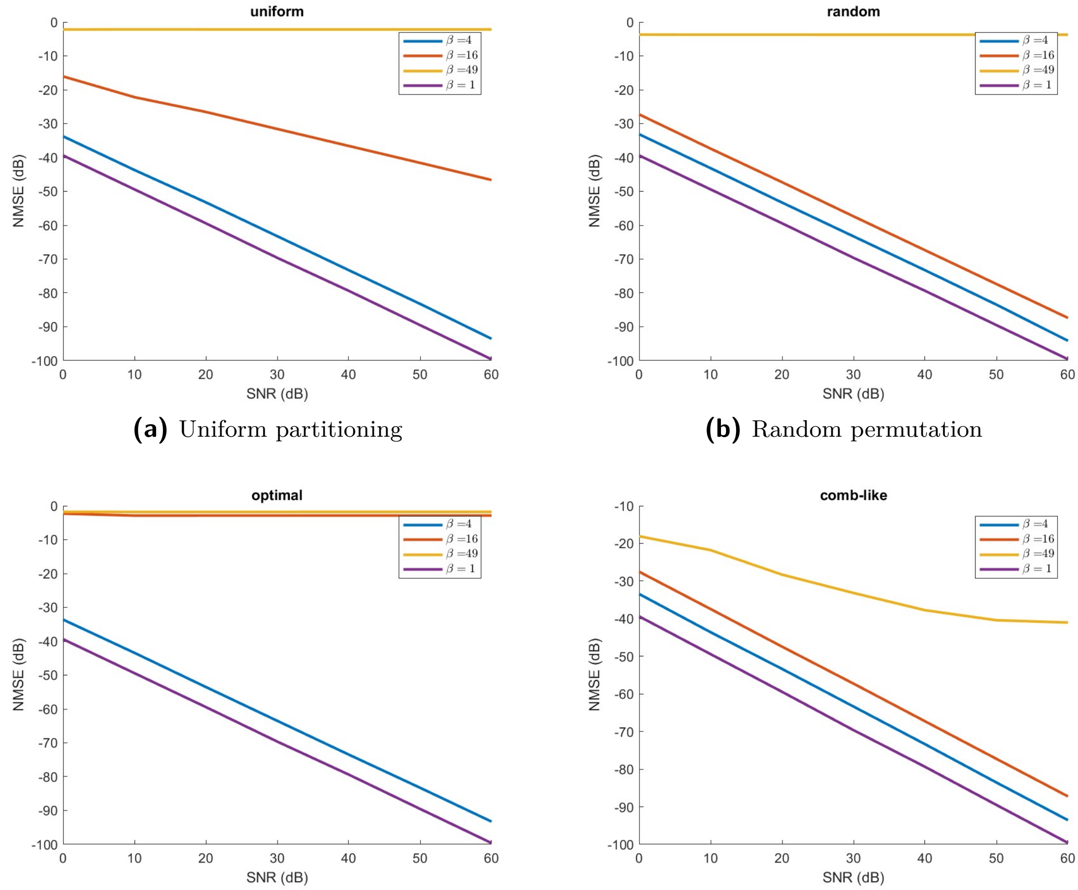
(b) $\beta = 16$ blocks. The comb-like strategy outperforms the others.



(c) $\beta = 64$ blocks. The comb-like strategy is the only strategy still yielding an accurate reconstruction.

Figure 3-2: Difference in reconstruction performance between partitioning strategies

Moreover, higher order extension of the strategies could be considered. The comb-like structure has a straightforward extension to higher order, maintaining all of its advantages. The other strategies on the other hand, are based on the permutation of columns. It was already mentioned in section 2-4 that this works well with the assumption that the sparse components are not grouped in just a few columns, as this will most certainly eliminate the advantages imposed by the permutation operations. In higher dimensions however, these strategies still only permute alongside a single direction. This increases the chance of ending up with few permutable slices that contain a large share of all the sparse components, again leading to a asymmetric distribution of the non-zero components over the partitions. To summarize, when working with higher order data, it is recommended to use the comb-like permutation strategy exclusively.



(a) Uniform partitioning

(b) Random permutation

(c) Optimal permutation

(d) Comb-like partitioning

Figure 3-3: Influence of the use of different partitioning strategies

3-3-2 Extension to higher order setting

BCS with K-OMP

Recalling the K-OMP algorithm in Algorithm 3, it is clear that, just as in BCS, the measurement matrices are already compressed in size in comparison to the traditional CS problem. In K-OMP, measurements are taken using one of the equivalent descriptions:

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \cdots \times_d \mathbf{A}^{(d)} + \underline{\mathbf{V}},$$
$$\mathrm{vec}(\underline{\mathbf{Y}}) = (\mathbf{A}^{(d)} \otimes \mathbf{A}^{(d-1)} \otimes \cdots \otimes \mathbf{A}^{(1)})\mathrm{vec}(\underline{\mathbf{X}}) + \mathrm{vec}(\underline{\mathbf{V}}).$$

With the measurement matrices already compressed, applying the BCS framework would relate to compressing these further.

With defining the total number of blocks $\bar{\beta} = \Pi_{i=1}^{d} \beta_i$:

$$\begin{aligned}\mathrm{superdiag}([\underline{\mathbf{Y}}_1, \underline{\mathbf{Y}}_2, \ldots, \underline{\mathbf{Y}}_{\bar{\beta}}]) =& \mathrm{superdiag}([\underline{\mathbf{X}}_1, \underline{\mathbf{X}}_2, \ldots, \underline{\mathbf{X}}_{\bar{\beta}}]) \times_1 \bar{\mathbf{A}}^{(1)} \times_2 \bar{\mathbf{A}}^{(2)} \times_3 \cdots \times_d \bar{\mathbf{A}}^{(d)} \\ &+ \mathrm{superdiag}([\underline{\mathbf{V}}_1, \underline{\mathbf{V}}_2, \ldots, \underline{\mathbf{V}}_{\bar{\beta}}])\end{aligned} \quad (3\text{-}3)$$

such that:

$$\bar{\mathbf{A}}^{(k)} = \begin{bmatrix} \mathbf{A}_1^{(k)} & 0 & \cdots & 0 \\ 0 & \mathbf{A}_2^{(k)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{A}_{\bar{\beta}}^{(k)} \end{bmatrix} \tag{3-4}$$

where superdiag$(\cdot)$ creates a block diagonal tensor with its argument on the block diagonal. Here, $\underline{\mathbf{V}}, \underline{\mathbf{Y}}_b \in \mathbb{R}^{\frac{m_1}{\beta_1} \times \frac{m_2}{\beta_2} \times \cdots \times \frac{m_d}{\beta_d}}$ denote the b-th blocks of noise and measurement, $\underline{\mathbf{X}}_b \in \mathbb{R}^{\frac{n_1}{\beta_1} \times \frac{n_2}{\beta_2} \times \cdots \times \frac{n_d}{\beta_d}}$ is the b-th signal block and $\bar{\mathbf{A}}^{(k)}$ is the factor matrix for mode $k \in \{1, 2, \ldots, d\}$, consisting of matrices $\mathbf{A}_b^{(k)} \in \mathbb{R}^{\frac{n_k}{\beta_k} \times \frac{m_k}{\beta_k}}$ on its diagonal for $b \in \{1, 2, \ldots, \bar{\beta}\}$

With the definition of the coherence limit in Equation 2-7, it can be deduced that taking a BCS approach to this higher order framework deteriorates the theoretical reconstruction bounds severely, as in each mode the measurement matrices are partitioned and therefore worsened.

For the reasons above, simulations of BCS applied to a Kronecker-based measuring strategy did not yield any results worth mentioning. These were either equal or worse than the trivial, all zeros, solution. Hence, further results on BCS are not discussed.

Prior-aware Kronecker Matching Pursuit

The final contribution in this thesis is about extending the prior-incorporation pipeline to a higher dimension CS setting. This is done by taking the pipeline as presented in section 3 of the conference paper and instead of performing the reconstruction using the LW-OMP algorithm, a prior-aware extensions to the K-OMP algorithm is derived based on LW-OMP. Just like in the paper, only the support identification of the K-OMP algorithm in Algorithm 3 is modified. For parameter c_k and $\underline{\bar{\mathbf{P}}}_{i_1,i_2,\ldots,i_d} = \log \frac{p(\underline{\mathbf{X}}_{i_1,i_2,\ldots,i_d})}{1-p(\underline{\mathbf{X}}_{i_1,i_2,\ldots,i_d})}$ depending on the iteration number, this reads:

$$j_1^{(i)}, j_2^{(i)}, \ldots, j_d^{(i)} = \arg \max_{j_1,j_2,\ldots,j_d} |\underline{\mathbf{R}} \times_1 \mathbf{A}^{(1)T} \times_2 A^{(2)T} \times_3 \cdots \times_d \mathbf{A}^{(d)T}|_{j_1,j_2,\ldots,j_d} + c_k \log \frac{p(\underline{\mathbf{X}})}{1 - p(\underline{\mathbf{X}})}.$$

Resulting in a higher-order extension to the LW-OMP algorithm: Validation of the benefits of this algorithm is lead for further research, as due to the properties listed in subsection 3-3-2, it is not relevant for usage in conjunction with a BCS structure and is therefore outside the scope of this thesis.

Algorithm 4 Logit-weighted Kronecker Orthogonal Matching Pursuit (LW-K-OMP) algorithm

Require: Compressed measurements $\mathbf{Y} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$, normalized factor matrices $\mathbf{A}^{(k)} \in \mathbb{R}^{m_k \times n_k},\ k \in \{1, 2, \ldots, d\}$, desired error bound ϵ, a probability distribution containing the prior probabilities for the support positions $p(\underline{\mathbf{X}}) \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, tuning parameter $g \in \mathbb{R}$ and noise level $\sigma \in \mathbb{R}^+$.

Initialize iteration parameter $i = 0$, estimate $\hat{\mathbf{x}}_0 = \mathbf{0}$, empty set $S_0 = \varnothing$. Put $\bar{\underline{\mathbf{P}}} = \log \frac{p(\underline{\mathbf{X}})}{1-p(\underline{\mathbf{X}})}$ and $\kappa = \Pi_{k=1}^{d} m_k$ the maximum number of iterations.

while $||\underline{\mathbf{R}}||_F^2 > \epsilon$ **and** $i < \kappa$ **do**

  $i = i + 1$

  $c_i = \frac{g}{2}\left(2(\kappa + 1 - i) - 1 + 2(\frac{\sigma}{g})^2\right)$

  $j_1^{(i)}, j_2^{(i)}, \ldots, j_d^{(i)} = \arg\max_{j_1, j_2, \ldots, j_d} |\underline{\mathbf{R}} \times_1 \mathbf{A}^{(1)T} \times_2 A^{(2)T} \times_3 \cdots \times_d \mathbf{A}^{(d)T}|_{j_1, j_2, \ldots, j_d} + c_i \bar{\underline{\mathbf{P}}}$

  $S_k^{(i)} = S_k^{(i-1)} \cup j_k^{(i)}\ \forall k \in \{1, 2, \ldots, d\}$

  $\mathbf{W}^{(k)}(: .i) = \mathbf{A}^{(k)}(:, j_k^{(i)})\ \forall k \in \{1, 2, \ldots, d\}$

  $\tilde{\mathbf{x}}_i = \arg\min_{\mathbf{z}} ||\text{vec}(\underline{\mathbf{Y}}) - (\mathbf{W}^{(d)} \odot \mathbf{W}^{(d-1)} \odot \cdots \odot \mathbf{W}^{(1)})\mathbf{z}||_F^2$

  $\text{vec}(\underline{\mathbf{R}}) = \text{vec}(\underline{\mathbf{Y}}) - (\mathbf{W}^{(d)} \odot \mathbf{W}^{(d-1)} \odot \cdots \odot \mathbf{W}^{(1)})\mathbf{z}$

end while

return $\{S_1, S_2, \ldots, S_d\}, \mathbf{z}$

Chapter 4

Conclusion

4-1 Summary

The aim of this thesis was to give insight in the matter that relates to the main research question:

- How does the partitioning of a Compressed Sensing problem affect its reconstruction performance and complexity?

The conclusions drawn from this thesis can be summarized in the following paragraphs:

In BCS, there is a trade-off between computational complexity and quality of the reconstructed signal. Theoretical bounds on how these two entities relate were derived. Using this theory, it was concluded that the computational resources can be drastically decreased for specific applications with less accurate reconstruction quality requirements. Also, prior information can be propagated among different blocks in BCS to achieve the goal of improving the reconstruction performance, while maintaining a favourable complexity in both storage and reconstruction performance.

In BCS, the problem of having a disequilibrium in the number of non-zero components in each block, i.e. asymmetry problem, can be overcome by permuting a signal, which distributes its sparse components more evenly over the different blocks. However, in high dimensions, sparse components are typically located within few slices, which disables a single permutation operation to be a solution to the asymmetry problem. Instead, a comb-like partitioning strategy can be chosen to overcome this problem. Not only is it applicable to clustered data and easily extendable to higher dimensions, it also carries advantageous properties in a prior-aware approach.

Finally, the computational complexity approach on BCS cannot be properly extended to work with Kronecker-based CS reconstruction, as the resulting BCS techniques are severely affected by the partitioning process.

4-2 Answers to sub-questions

For the sake of completeness, the sub-questions, as mentioned in the introduction are now repeated and answered one-by-one:

1. How does reconstruction performance change with the number of blocks in a BCS problem?
   - Increasing the number of blocks relates to a decrease in reconstruction performance. This was supported by a coherence-based analysis and experiments.
2. What is a proper strategy to partition a CS problem?
   - It was shown that the comb-like strategy is a good partitioning strategy as is can cope with the asymmetry problem, also it worked well in the prior construction process.
3. What is the change in computational complexity of the reconstruction when applying a BCS algorithm?
   - When applying the BCS, the computational complexity is decreased with a factor equal to the number of blocks in comparison to the computational complexity og the traditional CS problem.
4. How can relevant prior information be propagated between different blocks in a BCS reconstruction and at what computational cost?
   - Prior information can be collected based on statistics of the signal. The information can be propagated among blocks by using a prior-aware CS algorithm that solves the subsequent sub-problems. The computational cost of this prior propagating scheme is low in comparison to traditional CS methods.
5. How can higher order CS algorithms be deployed within (prior-aware) BCS reconstruction?
   - Although prior-aware higher order setting of the OMP algorithm is proposed in this thesis, it was found that combining BCS with this higher order setting does not yield good results. This is the result as the measurement procedure in such a settings is too restricted.

4-3 Limitations

For the execution of the experiments in this thesis, specific datasets were used. On one hand, this can be defended by saying that the data used is sparse and high dimensional, as required by the frameworks used, on the other hand the use of these frameworks is not validated for specific signal types such as very high dimensional data (e.g. 100 dimensional) or periodically sparse datasets. It will be hard to perform experiments using these signals. First, because it will be rather impossible to calculate the reference performance, since executing traditional

CS using this problem sizes is not feasible. Second, because in very high dimensions, a very large number of blocks is required to make the BCS problem feasible. With the comb-like partitioning strategy, this will increase the probability of having blocks containing few or no non-zero components, again causing asymmetry in the CS problem.

The construction of the prior in the proposed pipeline for prior-aware BCS, assumes that a signal entry is related to its neighbouring entries, such that for a d-dimensional signal, a $3^{(1)} \times 3^{(2)} \times \ldots 3^{(d)}$ kernel is used to find the prior probabilities. However, signals are not per se correlated in this fashion. If a signal is periodically correlated for example, this prior construction method is not able to properly capture the dynamics in the signal.

Although the comb-like strategy is known to be realizable in a real world application, for the other permutation strategies listed, this might not be the case. The freedom in synthesis of the measurement matrix namely is restricted by specifications and dynamics of the measurement application. Therefore, applying a comb-like measurement strategy in real world may not be a straightforward task.

4-4 Recommendations

The choice for working with the OMP algorithm in this work is partly because of its simplicity over other greedy methods and its LW-OMP extension. Future research can be performed on investigating whether the proposed prior incorporation strategy has potential in higher order extensions as well. Here it would be valuable to see whether this extensions can even have a more favourable performance-complexity trade-off.

Next, CS algorithms that depend on ℓ_1 norm relaxation exist. These are not considered in this thesis, but the application in prior-aware BCS of these algorithms could also be investigated. However, with these algorithms, e.g. ISTA relying on convex optimization, it is expected that the partitioning of a ℓ_1-norm relaxed CS problem has less potential to be improved by applying the BCS framework to it. On the other hand, for storage complexity reasons, it is still an attractive research direction. Also, it is interesting to know how the reconstruction guarantees and constrains on the CS problem change for an ℓ_1-norm generalized problem when used in BCS.

As mentioned in the limitations, the prior collection is based on a kernel, that is designed to work well for clustered signals. The influence of different shaped probability kernels on different signal types could be investigated as well. A kernel could for instance be designed to work well for periodically sparse signals or signals that exhibit some common probability distribution of its sparse components. Important to note, is that computational complexity difference between traditional CS and the prior-aware pipeline is dominated by the convolution of this kernel with an intermediate reconstruction estimate. When doing this research, enlarging this kernel is therefore probably not a good choice.

When applying the comb-like partitioning strategy in very high dimensions, the probability of having few or none sparse components within a block increases with the increase in the number of blocks. The comb-like strategy could be improved to address this problem. This is not an easy task, as a main advantage of this strategy is that it doesn't require prior information about the signal itself, namely by constructing the partitions by assigning the non-zero elements individually, similar to the *optimal* partitioning strategy. Nevertheless, a modification to the comb-like strategy preferably maintains the independence of prior knowledge.

Bibliography

[1] T. Van den Boom and B. De Schutter, *Lecture Notes for the Course SC42055, Optimization in Systems and Control*. 2018.

[2] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. 2013.

[3] E. J. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[4] C. Zhou, F. Liu, Z. Zhu, and B. Li, "A strategy of sar imaging based on 2-d block compressive sensing," in *2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013)*, pp. 1–4, 2013.

[5] R. Pournaghshband and M. Modarres-Hashemi, "A novel block compressive sensing algorithm for sar image formation," *Signal Processing*, vol. 210, p. 109053, 2023.

[6] L. Gan, "Block compressed sensing of natural images," in *2007 15th International Conference on Digital Signal Processing*, pp. 403–406, 2007.

[7] J. Chen, K.-X. Su, W.-X. Wang, and C.-D. Lan, "Residual distributed compressive video sensing based on double side information," *Acta Automatica Sinica*, vol. 40, no. 10, pp. 2316–2323, 2014.

[8] L. Zhu, H. Song, X. Zhang, M. Yan, T. Zhang, X. Wang, and J. Xu, "A robust meaningful image encryption scheme based on block compressive sensing and svd embedding," *Signal Processing*, vol. 175, p. 107629, 2020.

[9] A. Cichocki, N. Lee, I. Oseledets, A. Phan, Q. Zhao, and D. Mandic, "Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions," *Foundations and Trends® in Machine Learning*, vol. 9, no. 4-5, pp. 249–429, 2016.

[10] T. E. Oliphant *et al.*, *NumPy: Array manipulation routines*. NumPy Community, 2022. https://numpy.org/doc/stable/reference/generated/numpy.convolve.html.

[11] Y. Arjoune, N. Kaabouch, H. El Ghazi, and A. Tamtaoui, "A performance comparison of measurement matrices in compressive sensing," *International Journal of Communication Systems*, vol. 31, no. 10, p. e3576, 2018. e3576 IJCS-17-0570.R1.

[12] Z. Ben-Haim, Y. C. Eldar, and M. Elad, "Coherence-based performance guarantees for estimating a sparse vector under random noise," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5030–5043, 2010.

[13] Y. Arjoune, N. Kaabouch, H. El Ghazi, and A. Tamtaoui, "Compressive sensing: Performance comparison of sparse recovery algorithms," 2018.

[14] G. Xiao, Z.-J. Bai, and W.-K. Ching, "A modified orthogonal matching pursuit for construction of sparse probabilistic boolean networks," *Applied Mathematics and Computation*, vol. 424, p. 127041, 2022.

[15] C. Smith and M. Verhaegen, *Lecture Notes for the Course SC42150, Statistical Signal Processing:Stochastic Processes for Scientists and Engineers with Modern Applications.* 2020.

[16] S.-H. Hsieh, C.-S. Lu, and S.-C. Pei, "Fast omp: Reformulating omp via iteratively refining 2-norm solutions," pp. 189–192, 08 2012.

[17] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5270–5284, 2013.

[18] M. S. Asif and J. Romberg, "Dynamic updating for sparse time varying signals," in *2009 43rd Annual Conference on Information Sciences and Systems*, pp. 3–8, 2009.

[19] J. Ji and T. Lang, "Dynamic mri with compressed sensing imaging using temporal correlations," in *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1613–1616, 2008.

[20] J. Scarlett, J. Evans, and S. Dey, "Compressed sensing with prior information: Information-theoretic limits and practical decoders," *IEEE Transactions on Signal Processing*, vol. 61, p. 427, 01 2013.

[21] Z. Gao, C. Xiong, L. Ding, and C. Zhou, "Image representation using block compressive sensing for compression applications," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 885–894, 2013.

[22] J. Chen, X. Zhang, and H. Meng, "Self-adaptive sampling rate assignment and image reconstruction via combination of structured sparsity and non-local total variation priors," *Digital Signal Processing*, vol. 29, pp. 54–66, 2014.

[23] B. Zhang, Y. Liu, J. Zhuang, K. Wang, and Y. Cao, "Matrix permutation meets block compressed sensing," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 69–78, 2019.

[24] C. Caiafa and A. Cichocki, "Computing sparse representations of multidimensional signals using kronecker bases," *Neural computation*, vol. 25, 09 2012.

[25] E. Torkildson, U. Madhow, and M. Rodwell, “Indoor millimeter wave mimo: Feasibility and performance,” *IEEE Transactions on Wireless Communications*, vol. 10, pp. 4150–4160, 12 2011.

[26] H. Masoumi, N. J. Myers, G. Leus, S. Wahls, and M. Verhaegen, “Structured sensing matrix design for in-sector compressed mmwave channel estimation,” 2022.