Uditha Wijewardhana

SPARSE RECOVERY ALGORITHMS FOR STREAMING AND MULTIDIMENSIONAL SIGNALS
UDITHA WIJEWARDHANA

SPARSE RECOVERY ALGORITHMS
FOR STREAMING AND
MULTIDIMENSIONAL SIGNALS

Academic dissertation to be presented with the assent of the Doctoral Training Committee of Information Technology and Electrical Engineering of the University of Oulu for public defence in the OP auditorium (L10), Linnanmäa, on 28 February 2020, at 12 noon

UNIVERSITY OF OULU, OULU 2020
Wijewardhana, Uditha, Sparse recovery algorithms for streaming and multi-dimensional signals.
University of Oulu Graduate School; University of Oulu, Faculty of Technology; Centre for Wireless Communications
*Acta Univ. Oul. C 738, 2020*
University of Oulu, P.O. Box 8000, FI-90014 University of Oulu, Finland

**Abstract**

As the world is moving toward the era of big data, when a system will accumulate and process an immense amount of information, the cost and complexity of the acquisition and processing of high dimensional data is a critical issue to be addressed. In this respect, compressive sensing (CS), with its capability of utilizing sub-Nyquist sampling to recover a signal of interest, may play a vital role in addressing this problem. In this thesis, we develop sparse recovery algorithms to reconstruct streaming signals and multi-dimensional signals from compressive measurements.

First, we address the problem of reconstructing a streaming signal from compressive, streaming measurements. We utilize a lapped transform (LT) for the sparse representation of the streaming signal to reduce the blocking artifacts. To reconstruct the signal, we develop a progressive reconstruction algorithm based on sliding window processing. The recovery algorithm, which is based on sparse Bayesian learning (SBL), utilizes the preliminary information from the preceding processing window to improve the performance of signal reconstruction. Furthermore, we derive fast update formulae and propose a warm-start procedure to reduce the computational cost of the novel SBL algorithm. Next, we extend the solution method for a 2-dimensional scenario, in which we reconstruct an image from block compressive sensing (BCS) measurements. The results show that the novel SBL based sliding window method provides a more visually pleasing image without using any post-processing techniques.

We then address the problem of recovering an image from compressive measurements. Unlike the traditional image recovery algorithms, we utilize the total variation (TV) minimization and the sparse representation of the image in a given transform to improve the reconstruction performance of the image recovery algorithm. Later, we extend this work to a multiple bases scenario, as many natural signals may have compressible representations in different transforms rather than having a single sparse representation. For these scenarios, we modify the TV minimization problem by introducing an $l_1$-norm penalty. To solve these $l_1$-regularized TV minimization problems we propose customized interior-point methods by efficiently solving the Newton systems.

**Keywords:** compressive sensing, interior-point methods, sparse Bayesian learning, streaming signal recovery, total variation
Wijewardhana, Uditha, Harvat palautusalgoritmit suoratoisto- ja moniulotteisiin signaaleihin.
Oulun yliopiston tutkijakoulu; Oulun yliopisto, Teknillinen tiedekunta; Centre for Wireless Communications
*Acta Univ. Oul. C* 738, 2020
Oulun yliopisto, PL 8000, 90014 Oulun yliopisto

**Tiivistelmä**
Maailma on siirtymässä kohti aikakautta, jossa järjestelmät keräävät ja käsittelevät valtavan määrän tietoa. Tällöin tiedon hankkimisen kustannukset ja monimutkaisuus, sekä korkeaulotteisen datan käsittely tukevat kriittisiä reunaehdoiksi. Yksi ratkaisun tarjoaja on pakattu havainta (compressed sensing, CS), joka mahdollistaa alinäytteistetyn signaalin rekonstruktion. Tässä työssä kehitetään algoritmeja jatkuva-aikaisten ja moniulotteisten harvojen signaalien rekonstruktioon pakatuista mittauksista.


Seuraavaksi käsitetään pakatuista mittauksista. Työssä kehitetään uudenlainen total variation (TV) -minimointia ja kuvanmuunnoksen harvaa esiintyvää hyödynnävää rekonstruktioalgoritmi suorituskyvyn parantamiseksi. Koska luonnossa esiintyvät signaalit voivat usein esiintyä tehokkaasti eri muunnoksilla, algoritmi laajennetaan hyödyntämään useaa rinnakkaita signaalmuunnosta. Tässä mallissa TV-algoritmia muokataan lisäämällä $l_1$-normin perustyö säästämiseksi ja tehdään räätälöity menetelmän optimointi ongelman ratkaisemiseksi.

**Asiasanat:** harvojen signaalien Bayes-oppiminen, interior-point -menetelmä, jatkuva-aikaisten signaalien rekonstruktio, pakattu havainta, total variation -menetelmä
To My Wonderful Wife Achini, Dear Son Thinuga and Loving Parents.
Preface

The research work for this thesis was conducted at the Centre for Wireless Communications (CWC), at the University of Oulu, in Finland, between 2012 and 2016.

I would like to express my sincerest gratitude to Professor Matti Latva-aho for providing the opportunity to pursue doctoral studies in the Radio Access Technologies group at CWC. His continuous encouragement and guidance over the years have been invaluable, and I am very grateful to him.

I am deeply grateful to my principal supervisor, Adjunct Professor Marian Codreanu, for his advice and in-depth technical guidance, especially on optimization and compressed sensing. I am especially grateful for his immensely constructive comments which improved the quality of the articles. The countless friendly discussions on professional as well as personal matters with him allowed me to remain focused and positive, for which I am truly grateful.

I would also like to thank my follow-up group members, Professor Markku Juntti, Dr. Chathuranga Weeraddana from the University of Moratuwa, Sri Lanka, and especially Professor Nandana Rajatheva, for their insightful advice and discussion during my doctoral studies. I would also like to extend my gratitude to the pre-examiners, Professor Risto Wichman and Associate Professor Themistoklis Charalambous from Aalto University for their constructive feedback on my thesis, and Docent Mikko Vehkapera from the VTT Technical Research Centre of Finland for acting as the prestigious opponents in my doctoral thesis defense.

My special thanks go to Senior Professor Sampath Amaratunge, Vice Chancellor of the University of Sri Jayewardenepura (USJ), Sri Lanka, for providing me with the opportunity to work at this prestigious university. I am also grateful to all the academic staff members in the Faculty of Engineering (FOE) at USJ, including Dr. Konthesinghe (Dean FOE), Dr. Chamitha (Head, Department of Electrical and Electronic Engineering), Dr. Akila (Former Dean), Bandara, Dulini, Ganga, Nishan, Randima, and Udaya, as well as all the non-academic staff members for their continuous support.

I express my gratitude to the University of Oulu Graduate School, Nokia Foundation, TEKES, the Academy of Finland, and CWC projects, including LOCON, ECONET and ComingNets for their financial support. I was also privileged to receive a three-year
funded position in the Infotech Oulu Doctoral Program in 2013–2015. I would like to thank UniOGS and Infotech for funding several conference trips.

I would like to thank all my colleagues at the CWC, including Antti, Ayswarya, Bidushi, Evgeny, Fatih, Ganesh, Giang, Hirley, Ikram, Kalle, Kien, Markus, Mehdi, Mohammed, Satya, Tachporn, Upul and many others. They created a supportive and cheerful working environment, which I enjoyed over the years. I would like to thank all the administrative staff: Anu Niskanen, Eija Pajunen, Elina Komminaho, Hanna Saarela, Jari Sillanpaa, and Kirsi Ojutkangas for making my life a lot easier at CWC.

Life in Oulu was wonderful thanks to the beloved Sri Lankan community: Buddhika-Pawani, Chathu-Kanchana, Heshani, Madhusanka-Ruwanthi, Manosha-Dilani, Namal, Nuwan, Praneeth, Professor Rajatheva and family, Saliya-Bhagya, Sumudu-Inosha, Sandun-Chamari, Tharanga-Dimuthu and all the others who were in Oulu during my stay. My special thanks also go to the joyful little ones Ashini, Diliru, Senehas, Somnas, and Vinudi for making our gatherings so memorable.

I would like to thank my younger brother, Shashika, older brother Gihan and his family, and my in-laws for the support they have provided throughout my life. Last, but not least, I would like to express my humble gratitude to my loving family: my father Wijewardhana, my mother Kanthi, my wife Achini, and son Thinuga, for their love, care, encouragement, and support throughout my life and education.
**List of abbreviations**

**Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMP</td>
<td>Approximate message passing</td>
</tr>
<tr>
<td>BCS</td>
<td>Block Compressive sensing</td>
</tr>
<tr>
<td>BPDN</td>
<td>Basis pursuit denoising</td>
</tr>
<tr>
<td>CS</td>
<td>Compressive sensing</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete cosine transform</td>
</tr>
<tr>
<td>DS</td>
<td>Dantzig Selector</td>
</tr>
<tr>
<td>DWT</td>
<td>Discrete wavelet transform</td>
</tr>
<tr>
<td>EM</td>
<td>Expectation-maximization</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>LASSO</td>
<td>Least absolute shrinkage and selection operator</td>
</tr>
<tr>
<td>LOT</td>
<td>Lapped orthogonal transform</td>
</tr>
<tr>
<td>LT</td>
<td>Lapped transform</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum a-posterior</td>
</tr>
<tr>
<td>MRI</td>
<td>Magnetic resonance imaging</td>
</tr>
<tr>
<td>OMP</td>
<td>Orthogonal matching pursuit</td>
</tr>
<tr>
<td>RIP</td>
<td>Restricted isometry property</td>
</tr>
<tr>
<td>SBL</td>
<td>Sparse Bayesian learning</td>
</tr>
<tr>
<td>SOCP</td>
<td>Second order cone program</td>
</tr>
<tr>
<td>TV</td>
<td>Total variation</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless sensor network</td>
</tr>
</tbody>
</table>

**Roman-letter notations**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{B}$</td>
<td>A system matrix that models the composite effect of measurement and representation matrices</td>
</tr>
<tr>
<td>$\mathbf{B}_t$</td>
<td>The system matrix that models the composite effect of measurement and representation matrices at time instant $t$</td>
</tr>
<tr>
<td>$\hat{\mathbf{B}}_t$</td>
<td>The system matrix that models the composite effect of measurement and representation matrices up to time $t$</td>
</tr>
<tr>
<td>$\tilde{\mathbf{B}}_t$</td>
<td>The system matrix inside the active interval at time $t$</td>
</tr>
</tbody>
</table>
e A measurement noise vector
e_{i,j} The measurement error vector of y_{i,j}
e_{t} The measurement noise vector at time instant t
\hat{e}_{t} A measurement error vector at time t, i.e., \([e_{t}^{1} \ldots e_{t}^{T}]^T\)
e_{\tilde{t}} The measurement error vector inside the active interval at time t
\Gamma(\alpha|a, b) A random variable \alpha with a Gamma distribution with shape parameter a and rate parameter b
I_{N} The identity matrix of size \(N \times N\)
L The length of a basis vector in LOT
M A number of CS measurements
N A length of a signal vector
N(x|\mu, \Sigma) A Gaussian random vector x with mean \mu and covariance matrix \Sigma
p(x) The probability distribution of the random variable x
p(y|x) The conditional probability of y given x
\begin{bmatrix} P_{1} & P_{0} \end{bmatrix} The forward transform matrix of the LT
\begin{bmatrix} Q_{1} & Q_{0} \end{bmatrix} The inverse transform matrix of the LT
R Compression ratio
w_{t,i} The i-th element of the vector w_{t}
w A sparse transform domain coefficient vector
w_{i,j} The vectorized version of the LT coefficients \(W_{i,j}\), i.e., \(w_{i,j} = \text{vec}(W_{i,j})\)
w_{t} A vector of LOT coefficients obtained from \(X_{i-1,j-1}, X_{i-1,j}, X_{i,j-1}\) and \(X_{i,j}\)
\tilde{w}_{t} A sparse coefficient vector at time t, i.e., \(\tilde{w}_{t}^{1} \ldots \tilde{w}_{t}^{T}\)
\hat{w}_{t} The sparse coefficient vector inside the active interval at time t
\hat{w}_{t} The portion of \(\tilde{w}_{t}\) that is re-estimated at time t
\tilde{w}_{t} The portion of \(\tilde{w}_{t}\) that the posterior density remained unchanged from that obtained at time \(t-1\)
W_{i,j} A matrix of LT coefficients obtained from \(X_{i-1,j-1}, X_{i-1,j}, X_{i,j-1}\) and \(X_{i,j}\)
x_{i,j} The pixel in ith row and jth column of the image X
x(n) A discrete-time streaming signal
x A signal vector
\hat{x} An estimate of unknown x
The vectorized version of the \((i, j)\)-th block of the image \(X\), i.e., \(x_{i,j} = \text{vec}(X_{i,j})\).

The non-overlapping block of the signal \(x(n)\) at time instant \(t\).

The \((i, j)\)-th block of the image \(X\).

A measurement vector.

A vector of CS measurements obtained from the image block \(X_{i,j}\).

A vector of CS measurements acquired at time instant \(t\).

A measurement vector at time \(t\), i.e., \([y_1^T \ldots y_t^T]^T\).

The measurement vector inside the active interval at time \(t\).

A measurement vector containing measurements from time instant \(t'\) to \(t\), i.e., \([y_{t'+1}^T \ldots y_t^T]^T\).

**Greek-letter notations**

- \(\alpha_{\tau,i}\) The precision of \(w_{\tau,i}\).
- \(\mathbf{\alpha}_{\tau}\) The vector containing the hyperparameters associated with the vector \(w_{\tau}\), i.e., \(\mathbf{\alpha}_{\tau} = [\alpha_{\tau,1} \ldots \alpha_{\tau,N}]^T\).
- \(\mathbf{\alpha}_t\) A set of precision parameter vector at time \(t\), i.e., \(\mathbf{\alpha}_t = [\alpha_{t,1}^T \ldots \alpha_{t,N}^T]^T\).
- \(\delta(\mathbf{\alpha})\) The Delta function that is peaked at \(\mathbf{\alpha}\).
- \(\Phi\) A measurement matrix.
- \(\Psi\) A sparsifying basis.
- \(\sigma^2\) The variance of the Gaussian measurement noise.

**Mathematical-operator notations and symbols**

- \(\otimes\) The Kronecker product.
- \(|\cdot|_0\) The operator that counts the number of non-zero entries of a vector.
- \(|\cdot|_1\) The \(\ell_1\)-norm.
- \(|\cdot|_2\) The \(\ell_2\)-norm.
- \(|a|\) Absolute value of \(a\).
- \(\text{argmax } x\) The argument that maximizes the term \(x\).
- \(\text{argmin } x\) The argument that minimizes the term \(x\).
- \(\text{cov}(x)\) The covariance of \(x\).
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>diag(x)</td>
<td>A diagonal matrix with $x$ as diagonal</td>
</tr>
<tr>
<td>$E(x)$</td>
<td>The expectation of $x$</td>
</tr>
<tr>
<td>$\log x$</td>
<td>The logarithm of $x$</td>
</tr>
<tr>
<td>supp(w)</td>
<td>The support of $w$, i.e., ${i : w_i \neq 0}$</td>
</tr>
<tr>
<td>TV(X)</td>
<td>The total variation of image $X$</td>
</tr>
<tr>
<td>vec(X)</td>
<td>An operator that stacks the columns of matrix $X = [x_1 \ldots x_N]$ into the vector form $x = [x_1^T \ldots x_N^T]^T$</td>
</tr>
<tr>
<td>$\mathbb{R}$</td>
<td>The set of real numbers</td>
</tr>
<tr>
<td>$\nabla_w \psi$</td>
<td>The gradient of $\psi$ with respect to $w$</td>
</tr>
<tr>
<td>$X^T$</td>
<td>The transpose of matrix $X$</td>
</tr>
<tr>
<td>$X^{-1}$</td>
<td>The inverse of matrix $X$</td>
</tr>
</tbody>
</table>
Contents

Abstract
Tiivistelmä
Preface
List of abbreviations
Contents

1 Introduction

1.1 Motivation

1.2 Literature review

1.2.1 Sparse representation of signals

1.2.2 Fundamentals of compressive sensing

1.2.3 Streaming signal recovery from compressive measurements

1.3 Aims and the outline of the thesis

1.4 The author’s contribution to the publications

2 On-line recovery of streaming signals using sparse Bayesian learning

2.1 Introduction

2.2 System model and problem formulation

2.3 Sparse Bayesian learning-based recovery

2.3.1 Recovering the sparse coefficient vectors at time instant \( t \)

2.3.2 The effect of adding new measurements to the streaming system

2.3.3 The effect of removing old measurements from the streaming system

2.4 Recursive recovery of the signal using sliding windows

2.4.1 Processing window at time instant \( t \)

2.4.2 SBL algorithm for streaming signal recovery

2.4.3 Recovery algorithm with the warm-start

2.5 Numerical Results

2.5.1 Simulations with chirp signals

2.5.2 Simulations with real data sets

2.6 Summary and Discussion

15
3 Lapped transforms-based image recovery for block compressed sensing

3.1 Introduction ........................................................................... 55
3.2 Block compressive sensing model ........................................... 55
3.3 Recovery via lapped transforms ............................................. 56
   3.3.1 BCS with lapped transforms ........................................... 56
   3.3.2 Processing window ..................................................... 59
   3.3.3 Recovery algorithm based on SBL ................................. 61
3.4 Numerical results ............................................................... 63
3.5 Summary and discussion ..................................................... 65

4 Modified total variation exploiting transform domain sparsity

4.1 Introduction ........................................................................... 67
4.2 System model and problem formulation .................................... 67
4.3 A customized interior-point method ....................................... 69
   4.3.1 Interior-point method ................................................. 69
   4.3.2 Efficient solution of the Newton system ......................... 71
4.4 Numerical Results .............................................................. 73
4.5 Summary and discussion ..................................................... 77

5 Signal reconstruction using multiple sparsifying bases

5.1 Introduction ........................................................................... 79
5.2 Problem formulation ........................................................... 80
5.3 Signal reconstruction via multiple bases ................................. 80
5.4 Customized interior-point method for CS recovery via multiple bases .... 81
   5.4.1 Interior-point method ................................................. 82
   5.4.2 Efficient solution of the Newton system ......................... 83
5.5 Numerical results ............................................................... 86
5.6 Summary and discussion ..................................................... 88

6 Conclusions and Future Directions .......................................... 91
6.1 Conclusions ........................................................................... 91
6.2 Future directions ............................................................... 92

References .................................................................................. 95

1 The 3-stage hierarchical prior model

1.1 True prior on w ................................................................. 101

Appendices .................................................................................. 101
1 Introduction

1.1 Motivation

It is evident that the world is gradually moving into the era of big data, in which the amount of information that will be accumulated and processed by a system will be immense [1, 2]. Hence, the cost and complexity of the acquisition and processing of high dimensional data has become a critical area of interest for the research community [3]. For example, the Internet of Things (IoT) [4] applications will be one of the key data generating sources in the future. Generally, the IoT applications acquire and reconstruct streaming signals or high-dimensional data [5]. The key driving force behind these IoT applications is wireless sensor networks (WSNs), which consist of simple low-cost battery-powered sensors [6, 7]. Due to the limited computational power available, both acquisition and processing complexity should be reduced to facilitate the deployment of such WSNs.

Traditionally, the Shannon-Nyquist sampling theorem has inspired the acquisition and reconstruction of signals. It states that, a signal of interest can be perfectly reconstructed from its samples if the signal is sampled at the Nyquist rate, i.e., twice the bandwidth of the signal. However, this may not be feasible in some scenarios due to the high bandwidth of the signal, e.g., ultra-wideband communication signals, digital images, and magnetic resonance imaging (MRI). In situations such as WSNs, sampling a signal at a high rate may be restricted by the power constraint of the sensors [6].

In this thesis, we study several of the above mentioned problems, such as the compressed acquisition and reconstruction of signals without a clear beginning and/or end, and the reconstruction of multi-dimensional signals from sub-Nyquist samples. The key concept behind this work is compressive sensing (CS), which utilizes the sparse representation of the signals, i.e., a signal can be represented by few non-zero coefficients under a proper transformation. The work in this thesis is categorized into four major parts, and they are briefly described in Section 1.3.
1.2 Literature review

1.2.1 Sparse representation of signals

Following the famous Shanon-Nyquist sampling theorem, the introduction of CS has been a major breakthrough in the signal-processing community. The basic theories and the required mathematical foundation for the CS framework were developed in the early 2000s by Donoho, Candes, Romberg, and Tao [8, 9, 10, 11]. Essentially, CS is utilized to acquire either sparse or compressible signals.

A signal is specified to be sparse if all the information contained in the signal can be represented only with the help of a few significant components, compared to the total length of the signal. Let \( x \in \mathbb{R}^N \) be a real-valued signal that can be represented as \( x = \Psi w \), where \( \Psi \in \mathbb{R}^{N \times N} \) is the transformation basis and \( w = [w_1 \cdots w_N]^T \) is the transform domain coefficient vector. The signal \( x \) is said to be \( k \)-sparse in basis \( \Psi \) if \( w \) has at most \( k \ll N \) non-zero entries, i.e., \( \|w\|_0 \leq k \), where \( \|w\|_0 = |\text{supp}(w)| \) with the support of \( w \) is denoted by \( \text{supp}(w) = \{i : w_i \neq 0, i = 0, \ldots, N\} \) and \( \cdot \) denotes the cardinality for a discrete set.

Similarly, a signal is called compressible if the energy in \( w \) is concentrated in a small number of components, i.e., the ordered coefficients \( |w_{(1)}| \geq \cdots \geq |w_{(N)}| \) exhibits a power law decay satisfying \( |w_{(i)}| \leq C_q i^{-q} \) for \( i = 1, \ldots, N \) where \( q \geq 1 \) and \( C_q \) is a constant [10, 11, 12]. Generally, real-world signals are not exactly sparse; they are rather compressible in some well-known bases, i.e., they can be well-approximated by sparse signals. For example, natural images have sparse representations in known transforms, such as the discrete cosine transform (DCT) or discrete wavelet transform (DWT) [13], and audio signals are generally compressible in Fourier transforms.

1.2.2 Fundamentals of compressive sensing

According to the theory of CS, information contained in a sparse (or compressible) signal can be captured by a small number of random linear projections [8, 9, 10, 11, 14]. Thus, the CS acquisition model can be expressed mathematically as

\[
y = \Phi x + e, \tag{1}\end{equation}
\]

where \( x \) is an unknown \( N \)-dimensional sparse vector, \( \Phi \) is the \( M \times N \) measurement matrix, \( y \in \mathbb{R}^M \) is the random measurement vector and the measurement noise vector is denoted by \( e \in \mathbb{R}^M \), where each entry is assumed to be an independent Gaussian
random variable with zero mean and known variance $\sigma^2$, i.e., $e \sim \mathcal{N}(0, \sigma^2 I_M)$. Here, the number of measurements taken is much smaller than the dimension of the unknown vector, i.e., $M \ll N$. The goal of CS recovery is to estimate the sparse signal $x$ from this small number of random linear measurements $y$. This estimation problem is commonly referred to as a recovery or reconstruction problem. The theory of CS states that, under certain conditions (e.g., $\Phi$ satisfies the restricted isometry property (RIP)), the original sparse signal $x$ can be recovered from measurements $y$ by utilizing an appropriate recovery algorithm. Furthermore, the CS framework can be utilized even if the signal of interest $x$ is not itself sparse but has a sparse representation in a prior known basis, i.e., $x = \Psi w$, where $\Psi \in \mathbb{R}^{N \times N}$ is the sparsifying basis and $w$ is the sparse transform domain coefficient vector. In this case, the measurement matrix $\Phi$ should be sufficiently incoherent with $\Psi$.

A CS recovery algorithm should reconstruct the original signal from compressive measurements by solving (2), which is an underdetermined system of linear equations and has an infinite number of possible solutions. When the measurement error $e$ is neglected, and $B$ is such that no two $k$-sparse vectors are mapped to the same $y$, the unique solution can be obtained by posing the reconstruction problem as a combinatorial non-deterministic polynomial-time (NP) hard problem [18, Theorem 2.14]

$$\hat{w} := \arg\min ||w||_0 \text{ subject to } y = \Phi \Psi w,$$

where $\hat{w}$ is the estimate of $w$ resulting in the signal estimate $\hat{x} = \Psi \hat{w}$. The problem (3) searches for the sparsest $x$ consistent with the measurements $y$. Even for a medium-sized problem, searching for the solution of (3) by trying all possible $\binom{N}{k}$ combinations is computationally intractable. Hence, different approaches have been utilized by researchers to translate the $\ell_0$-minimization problem (3) into something more tractable and that can be solved in polynomial time. As examples, methods based on convex-relaxation [8, 21, 22, 23, 24, 25, 26], constructive greedy algorithms [27, 28, 29, 30], empirical Bayes methods [31, 32, 33, 34, 35, 36], methods based on approximate

$$y = \Phi \Psi w + e,$$
message passing (AMP) [37, 38], methods minimizing non-convex costs [39], and greedy-Bayesian hybrid methods [40] have been proposed in the literature.

\( \ell_1 \)-norm minimization

One of the most common avenues utilized to reduce the complexity of problem (3) is to replace the sparsity-promoting \( \ell_0 \)-term with its best convex approximation, the \( \ell_1 \)-norm. The resulting basis pursuit problem [9, 10, 11, 19, 41]

\[
\hat{w} := \text{argmin} \|w\|_1 \quad \text{subject to } y = \Phi \Psi w, \tag{4}
\]

is a convex problem, which searches for a solution having a minimum \( \ell_1 \)-norm, subject to the equality constraint. The main advantage of the \( \ell_1 \)-norm minimization problem (4) is that it can be cast as a linear program [12, 19, 26]. Thus, the solution can be obtained via solvers with polynomial-complexity. Some of the standard solvers available to solve \( \ell_1 \)-norm minimization problem (4) are \( \ell_1 \)-magic [42], NESTA [43] and YALL1 [44].

In practical settings, if the measurements are corrupted by noise, denoising can be achieved by relaxing the equality constraint in (4) to account for the measurement noise. The widely used convex programs for this purpose are least absolute shrinkage and selection operator (LASSO) [21] or basis pursuit denoising (BPDN) [22], and the Dantzig Selector (DS) [23]. The LASSO/BPDN poses the sparse estimation problem as

\[
\hat{w} := \text{argmin} \|w\|_1 \quad \text{subject to } \frac{1}{2} \|y - \Phi \Psi w\|_2^2 \leq \varepsilon, \tag{5}
\]

where the quadratic inequality constraint ensures that the squared \( \ell_2 \)-norm of the error between \( y \) and \( \Phi \Psi w \) should be in the order of measurement noise power. On the other hand, the data fidelity term in the Dantzig selector is \( \ell_\infty \)-norm, and it solves the problem as

\[
\hat{w} := \text{argmin} \|w\|_1 \quad \text{subject to } \| (\Phi \Psi)^T (\Phi \Psi w - y) \|_\infty \leq \tau, \tag{6}
\]

where \( \tau > 0 \) is a threshold parameter, which controls the tradeoff between the sparsity of the solution and its fidelity to the measurements. Since the data fidelity term in LASSO bounds the squared \( \ell_2 \)-norm of the error, it is generally suitable for scenarios where the noise norm \( \|e\|_2 \) is smaller. However, the Dantzig selector is suitable for scenarios with no outliers in the noise, and \( \| (\Phi \Psi)^T e \|_\infty \) is thus small [12, 23].
Total variation minimization

The Total variation (TV) norm is the $\ell_1$-norm of the derivatives of a signal. For a 2D digital signal, such as an image, the TV is defined as the sum of magnitudes of the discrete gradient at every point. Given an image $X \in \mathbb{R}^{N \times N}$, the TV can be expressed as

$$TV(X) = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \sqrt{|x_{i+1,j} - x_{i,j}|^2 + |x_{i,j+1} - x_{i,j}|^2},$$  \hspace{1cm} (7)$$

where $x_{i,j}$ denotes the pixel in $i$th row and $j$th column of the image $X$. Originally, the TV minimization method was proposed by Rudin et al. in [45] for image denoising. Later, the TV minimization was developed to be utilized in recovering images from compressive measurements [8, 10, 46]. It searches for an image $X$ with a sparse gradient by solving the recovery problem

\begin{align}
\text{minimize} & \quad TV(X) \\
\text{subject to} & \quad \|y - \Phi x\|_2 \leq \delta, \hspace{1cm} (8)
\end{align}

where $x$ stacks the columns of $X$ into a vector, i.e., $x = \text{vec}(X)$, $TV(\cdot)$ denotes the total variation of the image, and $\delta$ denotes a predefined threshold in the order of $\sqrt{M\sigma}$ [10]. The problem (8) is a convex optimization problem which can be solved using interior-point methods [47] very reliably and efficiently [42].

Sparse Bayesian learning

Sparse Bayesian learning (SBL), which is a method based on Bayesian inference, was originally introduced by Tipping [31, 32] to obtain sparse solutions to a variety of regression and classification problems. In [33], Wipf and Rao have adapted SBL to the problem of basis selection from overcomplete dictionaries and the application of SBL for the CS problem was proposed by Ji et al. in [34]. Henceforth, it was greatly enriched and extended by many researchers [35, 48, 49, 50]. The SBL can be considered an empirical Bayes method [48], where a type II likelihood (or evidence maximization) is performed to estimate the parameters of the prior distribution using the measured data. In [33, 48], it has been shown that under certain conditions the cost function of SBL (utilized to estimate the hyperparameters) retains a desirable property of the $l_0$-sparsity measure (i.e., the global minimum is achieved at the maximally sparse solution) while often having a smaller number of local minima. Further, the SBL algorithm not only has an improved performance over the traditional $l_1$-minimization approaches [48] but also...
offers certain advantages, such as providing flexibility to model correlation structures in signals [50], automatic estimation of model parameters, and a measure of the uncertainty of reconstruction [34].

In the SBL framework, we attempt to estimate the posterior density of the unknown vector $\mathbf{w}$ given the measurement vector $\mathbf{y}$ by employing a sparsity promoting prior on $\mathbf{w}$. In contrast, the widely used $l_1$-minimization framework can be viewed in a Bayesian setting as obtaining the maximum a-posteriori (MAP) estimation of $\mathbf{w}$ by utilizing the sparsity promoting Laplace prior [34]. However, the Laplace prior is not conjugate\(^1\) to the Gaussian likelihood function. Hence, the computation of the posterior becomes intractable. To address this issue, the SBL framework uses a hierarchical sparsity promoting prior model which partially preserves the conjugacy of the prior distribution of $\mathbf{w}$, $p(\mathbf{w})$.

In the literature, several hierarchical prior models have been utilized for $p(\mathbf{w})$ [31, 34, 35]. Among them, the standard 2-stage hierarchical prior model [31, 34] is the most common. Here, in the first stage, each element of the vector $\mathbf{w}$ is modeled as an independent Gaussian random variable with zero mean and individual precision parameters. In the second stage, each precision parameter is then modeled as an independent random variable with a Gamma distribution that has a common shape parameter and rate parameter. By marginalizing over the hyperparameters, the true prior over each element of $\mathbf{w}$ can be obtained analytically, and this corresponds to the density of a Student-$t$ distribution [31, 34]. Intuitively, this prior promotes sparsity, because it is sharply peaked at zero and consequently, it favors most of the elements of $\mathbf{w}$ as zero. In addition, a sparsity promoting 3-stage hierarchical prior model has been introduced in [35]. A more detailed explanation of the prior models is available in Section 2.3.1.

Having defined the priors, the SBL proceeds with the Bayesian inference to obtain the posteriors for the unknown sparse coefficients. This procedure is known as the evidence procedure (or the type-II maximum likelihood method) [31], which adopts a relevance vector machine (RVM) [52]. A more detailed description of the evidence procedure is available in Section 2.3.1.

---

\(^1\)A class of prior probability distributions $p(\theta)$ is said to be conjugate to a class of likelihood functions $p(x|\theta)$ if the resulting posterior distributions $p(\theta|x)$ are in the same family as $p(\theta)$ [51].
1.2.3 Streaming signal recovery from compressive measurements

Most of the sparse recovery algorithms, including those discussed in Section 1.2.2, assume that the unknown signal is a finite length vector, for which a fixed set of linear measurements and a sparse representation basis are available. However, in a variety of signal processing applications (see [7, 50, 53, 54, 55, 56] and the references therein), we face the problem of estimating a streaming signal from compressive streaming measurements. A simple approach to tackle this problem is to divide the streaming signal into disjointed finite length blocks, and reconstruct each block assuming it has a sparse representation in a known representation basis. The work in [7, 49, 50, 53, 57, 58, 59, 60, 61, 62, 63] utilizes this traditional approach for reconstructing streaming signals from compressive measurements. However, it is well understood that such an approach can introduce significant blocking artifacts, which affect the sparsity structure of the signal, and hence the recovery performance may degrade [55]. Thus, different techniques have been introduced to improve the reconstruction performance. In [7, 49, 50], multiple consecutive blocks have been processed together to improve the reconstruction of the signal. On the other hand, [53, 57, 58, 59, 60, 61, 62, 63] utilize the estimation of the neighboring blocks to improve the reconstruction performance of the current signal block, assuming a signal model with slowly varying support.

1.3 Aims and the outline of the thesis

The aim of this thesis is to develop sparse signal recovery algorithms to reconstruct streaming and multi dimensional signals from compressive measurements. In particular, we develop an algorithm for recovering a streaming signal, which has a sparse representation only in a lapped transform. On the other hand, we consider the scenario where a signal may have compressible representations in multiple bases and develop an efficient algorithm to utilize those representations to improve the reconstruction performance.

In Chapter 2, we consider the problem of reconstructing a streaming signal from compressive streaming measurements. We develop a progressive reconstruction algorithm based on sliding window processing, where we reconstruct the streaming signal over small overlapping shifting intervals. Since the consecutive intervals share some common sparse signal vectors, the key idea of this work is to utilize the preliminary information from the preceding interval to improve the performance of the signal recovery algorithm. For this purpose, we propose a novel SBL algorithm which is highly efficient for
recovering streaming signals. One major advantage of SBL is that it provides a measure of the uncertainty of the reconstructed signal rather than computing only a point estimate. The proposed SBL algorithm utilizes the previous estimates and the correlations among the non-zero coefficients to improve the performance of the algorithm. Since the effect of the uncertainty of the reconstructed signal from the preceding interval is specifically taken into account in the recovery process of the current interval, the proposed algorithm is more robust to the error propagation. Further, we propose a warm-start procedure and derive fast update formulae to reduce the computational cost of the SBL algorithm. In addition, we discuss the properties of the signal and the underlying approximations which enable the progressive reconstruction of the streaming signal from compressive measurements. The results are presented in [64, 65, 66].

In Chapter 3, we extend our work in Chapter 2 to the 2-dimensional signals scenario. We consider the recovery of an image from compressed sensing measurements. We utilize the block compressive sensing (BCS) procedure to alleviate the huge computation and memory burdens associated with using a dense measurement matrix for the sensing and reconstruction processes of the image. Building on Chapter 2, we develop an iterative image reconstruction method, based on 2-D lapped transforms and SBL. Specifically, to suppress the blocking artifacts, we jointly process a small number of adjacent measurement blocks to recover an image block. Furthermore, we provide an analysis of the motive behind the joint processing of adjacent measurement blocks. This analysis is based on the graphical models [51]. We then discuss the set of measurement blocks that need to be processed together to recover an image block. The results are presented in [67, 68].

In Chapter 4, we examine the problem of recovering an image from compressive measurements from a different perspective. Generally, the reconstruction algorithms are developed either to exploit a sparse representation of the image in a given transform or to exploit the sparse gradient property of the images. Instead, here we exploit the sparsity of the image in a given transform while considering that it has a sparse gradient. Thus, the TV minimization problem is modified by introducing an $l_1$-norm penalty term. The main contribution of this work is the derivation of a customized interior-point method to solve the $l_1$-regularized TV minimization problem. The principle behind a customized interior-point method is the efficient solution of the Newton system by exploiting the structure of the Hessian. Such customized methods have been proposed to solve the TV minimization problem in [42] and the $l_1$-regularized least squares problem in [24]. However, these methods are not applicable to the $l_1$-regularized TV
minimization problem due to the variations in the structure of the Hessian. Thus, we derive a customized interior-point method to solve the $l_1$–regularized TV minimization problem by proposing an efficient method to solve the Newton system. The results are presented in [69].

In Chapter 5, we show that a performance improvement can be achieved in signal reconstruction from compressive samples by utilizing multiple sparsifying bases. This work is motivated by the fact that for many natural signals, such as images and video sequences, the specific basis in which the signal of interest is the sparsest is unknown. Generally, multiple bases may exist, leading to a compressible representation of such a signal. These sparsifying bases are typically utilized in conventional CS recovery algorithms. Naturally, the quality of the signal’s reconstruction varies, depending on the choice of basis. So far, most of the effort has been focused on finding a specific basis that provides the sparsest representation of the signal, e.g., via dictionary learning algorithms [70, 71], which are generally computationally expensive. Instead, in this chapter we develop a customized interior-point method for recovering 2-D signals (images) from compressive measurements which can utilize multiple sparsifying bases as well as the fact that the images usually have a sparse gradient. The results are presented in [72].

In Chapter 6, we conclude the thesis and discuss possible future directions.

1.4 The author’s contribution to the publications

The thesis is based on two journal papers [66, 69], and five related conference papers [64, 65, 67, 68, 72]. The author of this thesis had the ideas and main responsibility in carrying out the analysis, developing the MATLAB simulations, generating the numerical results, writing the papers. Other authors provided comments, constructive criticism, and support during the process.

In addition to the papers [64, 65, 66, 67, 68, 69, 72], during his doctoral studies the author has published a journal [73] and a conference paper [74] which are not included in this thesis to maintain the focus. The author has also contributed to the journal paper [75] and the conference papers [76, 77].
2 On-line recovery of streaming signals using sparse Bayesian learning

2.1 Introduction

In this chapter, we consider the problem of recovering a streaming signal from compressive streaming measurements. Such a reconstruction of streaming signals may arise in a variety of signal processing applications [5, 7, 50, 53, 54, 56, 78]. The traditional method for addressing this problem is to divide the streaming signal into disjointed finite-length blocks, and reconstruct each block assuming it has a sparse representation in a known representation basis [7, 49, 50, 53, 57, 58, 59, 60, 61, 62, 63]. However, the reconstruction performance of such an approach may degrade [55] due to the introduction of blocking artifacts, which affect the sparsity structure of the signal.

A method for overcoming this drawback which reduces blocking artifacts has been proposed in [55]. A lapped orthogonal transform (LOT) has been used in [55] for the sparse representation of the streaming signal. The main advantage of LOTs is that they can preserve the sparsity of a streaming signal by substantially reducing the blocking artifacts occurring in block transforms [13, 55, 79, 80]. However, the LOTs introduce coupling between consecutive blocks of the signal, and multiple blocks should therefore be processed together to recover the signal.

Inspired by [55], we use an LOT in this chapter for the sparse representation of the streaming signal in order to reduce the blocking artifacts. It is clear that, due to the coupling introduced by the LOTs, each signal block cannot be reconstructed independently using disjoint measurement blocks. Instead, we need to jointly estimate the streaming signal from the entire set of compressive measurements. By utilizing the fact that a banded matrix has a band-dominant inverse [81, 82, 83, 84, 85, 86], we first show that only a few nearby measurement vectors in the streaming system contain information on a certain sparse coefficient vector. We can therefore obtain an estimate of the sparse vector by processing only those nearby measurement vectors instead of processing the entire set of compressive measurements. To this end, we develop a progressive reconstruction algorithm based on sliding window processing, analogous to Kalman-like processing in the smoothing mode, where we reconstruct the sparse coefficient vectors (or equivalently the streaming signal) over small overlapping shifting
intervals. Since the consecutive intervals overlap each other, they share some common sparse vectors. Hence, as in Kalman-like processing, we can utilize the preliminary information from the preceding interval combined with the current measurement to refine the estimate of the sparse vector in the current interval. For this purpose, we develop a novel SBL algorithm which can be utilized for the reconstruction of a streaming signal that has a sparse representation in an LOT. The proposed SBL algorithm utilizes the previous estimates and the correlations among the non-zero coefficients of the sparse vectors to improve the performance of the sparse recovery algorithm. Furthermore, to reduce the computational cost of the reconstruction, we propose a warm-start procedure and fast update formulae for the SBL algorithm. Since the effect of uncertainty in the reconstructed signal from the preceding interval is specifically taken into account in the reconstruction process of the current active interval, the proposed algorithm is robust against error propagation.

The rest of the chapter is organized as follows. In Section 2.2, we describe the system model for streaming signal reconstruction from compressive streaming measurements. In Section 2.3.1, we provide a brief review of SBL, in which we jointly estimate the streaming signal from the entire set of compressive measurements. In Sections 2.3.2 and 2.3.3, we show that the considered streaming system can be decoupled and the sparse vectors can be reconstructed recursively over small overlapping intervals. Specifically, we obtain the set of measurement vectors that contains information on a certain sparse coefficient vector. This set of measurement vectors defines the size of the processing window and other design parameters in the proposed algorithm. For a given processing window size, we derive the recursive reconstruction algorithm based on SBL for recovering the streaming signal in Section 2.4. In Section 2.5, we provide simulation results to evaluate the performance of the proposed recovery algorithm, as well as the effect of processing window size and Section 2.6 summarizes the chapter.

2.2 System model and problem formulation

We consider the recovery of a discrete-time streaming signal \( x(n) \) from compressive measurements. We acquire the streaming signal \( x(n) \) utilizing the time-varying linear measurement model given by

\[
y_t = \Phi_t x_t + e_t, \quad t = 1, 2, \ldots
\]
where $x_t = [x(Nt - N + 1) \ldots x(Nt)] \in \mathbb{R}^N$ denotes the non-overlapping block of the signal $x(n)$ at time instant $t$; $y_t \in \mathbb{R}^M$ contains $M < N$ linear measurements of $x_t$, where $R = N/M > 1$ is called the compression ratio; the measurement matrix is denoted by $\Phi_t \in \mathbb{R}^{M \times N}$, and the measurement noise vector is denoted by $e_t \in \mathbb{R}^M$, where each entry is assumed to be an independent Gaussian random variable with zero mean and known variance $\sigma^2$, i.e., $e_t \sim \mathcal{N}(0, \sigma^2 I_M)$.

We use a lapped orthogonal transform (LOT) for the sparse representation of the streaming signal $x(n)$ [13, 55, 79, 80] to reduce the blocking artifacts that occur with traditional block transforms. For LOTs, the basis vectors have a length $L > N$ [80], i.e., they extend across the block boundaries. For clarity of presentation, we consider an LOT with $L = 2N$, but the extension to the general case of arbitrary $L$ is straightforward.

The vector of transform domain coefficients $w_t \in \mathbb{R}^N$, which we expected to be sparse or compressible, is given by $^Tw_t = \begin{bmatrix} P_1 & P_0 \end{bmatrix} \begin{bmatrix} x_{t-1}^T & x_t^T \end{bmatrix}^T$, where the LOT matrix is denoted by $\begin{bmatrix} P_1 & P_0 \end{bmatrix} \in \mathbb{R}^{N \times 2N}$. By using the perfect reconstruction property, the signal block $x_t$ can be synthesized as

$$x_t = \begin{bmatrix} P_0^T & P_1^T \end{bmatrix} ^Tw_t$$.

Thus, an arbitrary portion of the signal $x(n)$ can be represented using the transform domain coefficients as illustrated in Fig. 1a.

---

Fig. 1. System model for streaming signal recovery: (a) The linear relationship between the signal blocks $x_t$ and the transform domain coefficient vectors $w_t$. The streaming signal $x(n)$ has a sparse representation in an LOT with $L = 2N; (b)$ The linear relationship between the measurement vectors $y_t$ and the transform domain coefficient vectors $w_t$ (Reprinted by permission [66] ©2017 IEEE).
By substituting (10) in the measurement model (9), we can obtain the linear relationship between the measurement vector $y_t$ and the transform domain coefficient vectors at time instant $t$ as

$$y_t = B_t \begin{bmatrix} w_t \\ w_{t+1} \end{bmatrix} + e_t, \quad t = 1, 2, \ldots \quad (11)$$

where $B_t = \Phi_t \begin{bmatrix} P_0^T \\ P_1^T \end{bmatrix} \in \mathbb{R}^{M \times 2N}$ models the composite effect of the measurement and the representation matrices at time instant $t$. An example of such a streaming system is presented in Fig. 1b, which illustrates the relationship between the measurement vectors and the transform domain coefficient vectors.

Consider the problem of estimating the signal block $x_t$ at time instant $t$. From (10), we can see that $x_t$ can be synthesized with a knowledge of the sparse vectors $w_t$ and $w_{t+1}$. However, since the streaming system illustrated in Fig. 1b is not block diagonal, estimating $x_t$ independently from $y_t$ (by estimating $w_t$ and $w_{t+1}$ from $y_t$) is not optimal. Therefore, all sparse vectors $\{w_\tau\}_{\tau=1}^{t+1}$ must be jointly estimated from the measurement vectors $\{y_\tau\}_{\tau=1}^{t}$. For a large $t$, this approach introduces a large reconstruction delay and the recovery algorithm becomes exceedingly complex. To reduce the reconstruction delay, as well as the complexity of the recovery algorithm, we propose a progressive reconstruction algorithm based on sliding window processing, in which we reconstruct the sparse vectors over small overlapping shifting intervals. Since the consecutive intervals overlap, they share some common sparse vectors. Hence, the key idea of this work is to utilize the preliminary information from the preceding interval to improve the performance of the signal recovery algorithm. For this purpose, we develop a novel sparse Bayesian learning (SBL) algorithm which utilizes the previous estimates and the correlations among the non-zero coefficients to improve the performance of the algorithm. For completeness, prior to developing the progressive recovery algorithm, we provide a brief review of the SBL algorithm for recovering a block sparse signal. Further, we show that the considered streaming system decouples under the approximation that the inverse of a banded matrix is also banded [82, 83, 84, 85, 86]. Specifically, by using the above approximation, we show that only a few adjacent measurement vectors contain information about a certain sparse coefficient vector. Hence, we can estimate the sparse vector by processing only these measurement vectors, instead of processing the entire set of compressive measurements.
2.3 Sparse Bayesian learning-based recovery

In this section, we first provide a brief review of the SBL framework for recovering a block sparse signal. Then, by exploiting the structures of the involved matrices, we show that the considered streaming system decouples. Specifically, we show that all the information on a certain sparse vector \( \mathbf{w}_t \) is contained within a few neighboring blocks of measurement vectors around \( \mathbf{y}_t \). Hence, we can estimate \( \mathbf{w}_t \) by processing only those neighboring measurement vectors, which enables the progressive reconstruction of the signal and defines the necessary parameters for developing the progressive recovery algorithm.

2.3.1 Recovering the sparse coefficient vectors at time instant \( t \)

Consider the joint reconstruction of the sparse vectors \( \{ \mathbf{w}_\tau \}_{\tau=1}^{t+1} \) from the measurement vectors \( \{ \mathbf{y}_\tau \}_{\tau=1}^{t} \) using SBL at time instant \( t \). We can represent the measurement model at time instant \( t \) as

\[
\mathbf{y}_t = \mathbf{B}_t \mathbf{w}_t + \mathbf{e}_t, \tag{12}
\]

where the measurement vector is denoted by \( \mathbf{y}_t = \left[ \mathbf{y}_1^T \ldots \mathbf{y}_t^T \right]^T \in \mathbb{R}^{Mt} \), the system matrix \( \mathbf{B}_t \in \mathbb{R}^{Mt \times N(t+1)} \) models the composite effect of the measurement and the representation matrices, the sparse coefficient vector is denoted by \( \mathbf{w}_t = \left[ \mathbf{w}_1^T \ldots \mathbf{w}_{t+1}^T \right]^T \in \mathbb{R}^{N(t+1)} \), and the measurement error vector is denoted by \( \mathbf{e}_t = \left[ \mathbf{e}_1^T \ldots \mathbf{e}_t^T \right]^T \in \mathbb{R}^{Mt} \). Note that \( \mathbf{e}_t \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_{Mt}) \). The likelihood function therefore can be expressed as:

\[
p(\mathbf{y}_t | \mathbf{w}_t) = \mathcal{N}(\mathbf{y}_t | \mathbf{B}_t \mathbf{w}_t, \sigma^2 \mathbf{I}_{Mt}). \tag{13}
\]

In the SBL framework, we attempt to estimate the posterior density of the unknown vector \( \mathbf{w}_t \) given the measurement vector \( \mathbf{y}_t \) by employing a sparsity promoting prior on \( \mathbf{w}_t \). In contrast, the widely used \( l_1 \)-minimization framework can be viewed in a Bayesian setting as obtaining the maximum a-posterior (MAP) estimation of \( \mathbf{w}_t \) by utilizing the sparsity promoting Laplace prior [34]. However, the Laplace prior is not conjugate to the Gaussian likelihood function (13). Hence, the computation of the posterior becomes intractable. To address this issue, the SBL framework uses a hierarchical sparsity promoting prior model which partially preserves the conjugacy of \( p(\mathbf{w}_t) \).

\footnote{A class of prior probability distributions \( p(\theta) \) is said to be conjugate to a class of likelihood functions \( p(x|\theta) \) if the resulting posterior distributions \( p(\theta|x) \) are in the same family as \( p(\theta) \).}
In the literature, several hierarchical prior models have been utilized for $p(w_t)$ \cite{31, 34, 35}. In this work we utilize the standard 2-stage hierarchical prior model \cite{31, 34}. In the first stage, each element of the vector $w_t$ is modeled as an independent Gaussian random variable with zero mean and individual precision parameters, i.e.,

$$p(w_t) = \prod_{t=1}^{T+1} \prod_{i=1}^{N} \mathcal{N}(w_{t,i} | 0, \alpha_{t,i}^{-1}),$$

(14)

where $w_{t,i}$ denotes the $i$th element of the vector $w_t$ and $\alpha_{t,i}$ denotes its precision, which is the individual hyperparameter associated with $w_{t,i}$. We denote the vector containing the hyperparameters associated with the vector $w_t$ by $\alpha_t = [\alpha_{t,1} \ldots \alpha_{t,N}]^T \in \mathbb{R}^N$. Then, $\alpha_t = [\alpha_{t,}^T \ldots \alpha_{t,(t+1)}^T]^T$ is a vector containing $N(t+1)$ hyperparameters controlling the precision of each element of $w_t$.

In the second stage, hyperpriors over $\alpha_t$ are defined. Each precision parameter $\alpha_{t,i}$ is modeled as an independent random variable with a Gamma distribution \cite{31}, with a common shape parameter $a$ and rate parameter $b$, as

$$p(\alpha_t | a, b) = \prod_{t=1}^{T+1} \prod_{i=1}^{N} \text{Gamma}(\alpha_{t,i} | a, b).$$

(15)

By marginalizing over the hyperparameter $\alpha_{t,i}$, we can obtain the true prior over $w_{t,i}$ analytically, and it corresponds to the density of a Student-$t$ distribution \cite{31, 34}. Hence, the overall prior over the vector $w_t$ becomes the product of independent Student-$t$ distributions over $w_{t,i}$. For the appropriate choice of $a$ and $b$, the Student-$t$ distribution is highly peaked around $w_{t,i} = 0$, and hence this prior favors most of $w_{t,i}$ as zero promoting sparsity. For $a = b = 0$, the hyperpriors $p(\alpha_{t,i} | a, b)$ become non-informative\(^6\) (i.e., uniform hyperpriors over a logarithmic scale), and the resulting true prior over $w_{t,i}$ becomes an improper prior, $p(w_{t,i}) \propto 1/|w_{t,i}|$ \cite{31}. Intuitively, this prior promotes sparsity, because it is sharply peaked at zero and it therefore favors most of $w_{t,i}$ as zero. Additionally, in \cite{35}, a 3-stage hierarchical prior model has been introduced, and this prior also promotes sparsity. The true prior of this 3-stage hierarchical model is given in Appendix 1.

\(^5\)The Gamma prior is conjugate to the Gaussian likelihood with a known mean.

\(^6\)Making the prior over a scale parameter $\alpha$ non-informative is equivalent to setting the $p(\log(\alpha)) = \text{constant}$ \cite[sec.2.4.3]{51}.

32
Having defined the priors, the SBL proceeds with the Bayesian inference by utilizing the following decomposition of the posterior distribution\(^7\):

\[
p(w_t, \alpha_t | y_t) = p(w_t | \alpha_t, y_t)p(\alpha_t | y_t). \tag{16}
\]

The first term on the right-hand side of (16) represents the (conditional) posterior distribution over the sparse coefficient vector, and it can be analytically computed from the Bayes’ theorem for Gaussian variables [51, p.90] by noting that the likelihood function \(p(y_t | w_t)\) and the prior \(p(w_t | \alpha_t)\) are both Gaussian (see (13) and (14)), i.e.,

\[
p(w_t | \alpha_t, y_t) = \mathcal{N}(w_t; \mu^w_t, \Sigma^w_t), \tag{17}
\]

where

\[
\Sigma^w_t = A_t^{-1} - A_t^{-1}B_t^T \left( \sigma^2 I_{M_t} + B_tA_t^{-1}B_t^T \right)^{-1}B_tA_t^{-1}, \tag{18}
\]

\[
\mu^w_t = \sigma^{-2} \Sigma^w_t B_t^T y_t, \tag{19}
\]

with \(A_t = \text{diag}(\alpha_t)\), \(\mu^w_t \in \mathbb{R}^{N(t+1)}\) and \(\Sigma^w_t \in \mathbb{R}^{N(t+1) \times N(t+1)}\) denotes the posterior mean and covariance of \(w_t\) at time instant \(t\).

The second term on the right-hand side of (16) represents the posterior distribution of the hyperparameters \(p(\alpha_t | y_t)\), and it cannot be computed analytically. Hence, it is approximated to be sharply peaked around its mode and replaced by a delta function, i.e., \(p(\alpha_t | y_t) \approx \delta(\alpha^{MP}_t)\), where \(\alpha^{MP}_t\) denotes the mode of \(p(\alpha_t | y_t)\) [31]. Since, \(p(\alpha_t | y_t) \propto p(\alpha_t, y_t)\), the value of \(\alpha^{MP}_t\) is estimated by maximizing the joint distribution \(p(\alpha_t, y_t)\), or equivalently, its logarithm with respect to \(\alpha_t\), i.e.,

\[
\alpha^{MP}_t = \text{argmax } \mathcal{L}_t, \tag{20}
\]

where

\[
\mathcal{L}_t = \log p(y_t | \alpha_t) p(\alpha_t). \tag{21}
\]

This procedure is known as the evidence procedure or the type II maximum likelihood method [31]. Unfortunately, the value of \(\alpha^{MP}_t\) cannot be obtained in closed form. Thus, different iterative methods that guarantee finding a stationary point of \(\mathcal{L}_t\) have been derived in the literature, with all having their re-estimation formulae in the form:

\[
\alpha_{t,i}^{\text{new}} = f(\mu_{t,i}^{\tau}, \Sigma_{t,ii}^{\tau}), \quad i = 1, \ldots, N, \quad \tau = 1, \ldots, t + 1, \tag{22}
\]

\(^7\)Note that evaluating \(p(w_t, \alpha_t | y_t)\) from the standard decomposition, i.e., \(p(w_t, \alpha_t | y_t) = p(y_t | w_t)p(w_t, \alpha_t) / p(y_t)\), is intractable, since the the normalization factor \(p(y_t)\) cannot be evaluated (see [31] for details).
where $\mu^\tau_{t,i}$ is the $N(\tau - 1) + i$-th element of $\mu^\tau$ and $\Sigma^\tau_{t,ii}$ is the $N(\tau - 1) + i$-th diagonal element of $\Sigma^\tau$. For example, by taking the derivative of $L_t$ with respect to $\log \alpha^\tau_{t,i}$ and equating to zero results in the re-estimation formulae [31]:

$$\alpha^{\text{new}}_{t,i} = \frac{1 + 2a}{(\mu^\tau_{t,i})^2 + \Sigma^\tau_{t,ii} + 2b}, \quad i = 1, \ldots, N \quad \tau = 1, \ldots, t + 1.$$ 

This is equivalent to the expectation-maximization (EM) update, where $\bar{w}_t$ is treated as an unobserved latent or ‘hidden’ variable [31]. An alternative update formula can be obtained by substituting $\delta_{t,i} \equiv 1 - \alpha^\tau_{t,i} \Sigma^\tau_{t,ii}$ as in [87] in the derivative of $L_t$ with respect to $\log \alpha^\tau_{t,i}$ and equating to zero as [31, 32, 34]

$$\alpha^{\text{new}}_{t,i} = \frac{\delta_{t,i} + 2a}{(\mu^\tau_{t,i})^2 + 2b}, \quad i = 1, \ldots, N \quad \tau = 1, \ldots, t + 1.$$ 

Now, to obtain the posterior distribution of the sparse coefficient vector, i.e., $p(w_t|\bar{y}_t)$, it is required to integrate out the nuisance parameters $\bar{\alpha}_t$ from $p(w_t, \bar{\alpha}_t|\bar{y}_t)$ in (16). However, since we obtain a point estimate for $\bar{\alpha}_t$, the posterior density of $w_t$ can be approximated by $p(w_t|\bar{y}_t) \approx p(w_t|\bar{\alpha}_{t,\text{MP}}^\tau, \bar{y}_t)$ and computed by utilizing the following algorithm:

---

**Algorithm 1** SBL algorithm for block sparse recovery  

1. **Inputs:** $\bar{y}_t$, $B_t$.  
2. **Initialization:** Initialize $\bar{\alpha}_t$.  
3. while the convergence criteria are not met do  
4. Compute $\Sigma^\tau, \mu^\tau$ (use (18) and (19)).  
5. Set $\alpha^{\text{new}}_{t,i} = f(\mu^\tau_{t,i}, \Sigma^\tau_{t,ii})$ (see (22)).  
6. end while.  
7. **Outputs:** $\mu^\tau, \Sigma^\tau$ and $\alpha_t$.  

In practice, during this process, most of the $\alpha^\tau_{t,i}$ tend to infinity, i.e., they become numerically indistinguishable from infinity given the machine precision [31, 34]. According to (19) and (18), this implies that both the posterior mean and the variance of $w^\tau_{t,i}$ tend to zero. Hence, the posterior density of $w^\tau_{t,i}$ becomes highly peaked at zero, causing those $w^\tau_{t,i}$ to be zero, which realize the sparsity of $\bar{w}_t$.

In general, the posterior covariance $\Sigma^\tau$ is computed by inverting the precision matrix [31]

$$H_t = A_t + \sigma^2 B_t^T B_t.$$  

(23)
However, as $\alpha_{t,i}$ tends to infinity the random variable $\mathbf{w}_i$ becomes a degenerate Gaussian, and the corresponding $w_{t,i}$ become deterministic components of $\mathbf{w}_i$ [88]. Hence, the pseudo inverse of $\mathbf{H}_t$ can be utilized to obtain $\mathbf{H}_t^\dagger$, by pruning out those $\alpha_{t,i}$ that tend to infinity and the corresponding column vectors of $\mathbf{B}_t$ [31, App. B]. Since the above mentioned update rules depend on the re-computation of $\mathbf{H}_t^\dagger$, which involves an inverse operation of order $O(D^3)$ and memory of order $O(D^2)$, with $D$ denoting the number of columns in the model, the complexity of the recovery algorithm grows with $t$. At time instant $t$, the initial $D = N(t + 1)$, because the algorithm begins with $\mathbf{B}_t$ as the model, and as the algorithm progresses, $D$ reduces due to the pruning procedure. However, for a large $t$, this sparse recovery algorithm becomes exceedingly complex, since $D$ increases linearly with $t$. To overcome these drawbacks, we develop a recursive recovery algorithm based on sliding window processing, in which we reconstruct the sparse vectors over small overlapping intervals. The proposed algorithm recomputes only a small portion of the posterior covariance at each time instant $t$. Since the size of the recomputed portion of the posterior covariance does not increase with $t$, the complexity and the memory requirements of the algorithm remain tractable. We further reduce the complexity of the SBL algorithm by following an approach similar to [89], in which we initialize with an ‘empty’ model and sequentially add/remove columns from the model utilizing the fast update formulae.

Prior to deriving the recursive SBL algorithm, we discuss the properties of the matrices and underlying approximations which enable the utilization of a recursive reconstruction procedure for the considered streaming system in Section 2.2. The key idea behind the decoupling of the streaming system is as follows: the posterior covariance matrix of the sparse coefficient vector $\mathbf{w}_t$, i.e., $\Sigma_t^\mathbf{w}$, can be approximated by a block-banded matrix at any time instant $t$ by using the approximation that the inverse of a banded matrix is also banded [82]. Since the posterior density of $\mathbf{w}_t$ is jointly Gaussian, it is well understood that this implies that only a few measurement vectors adjacent to $\mathbf{y}_t$ contain information about a certain sparse coefficient vector $\mathbf{w}_t$. Hence, we can estimate the sparse vector $\mathbf{w}_t$ by processing only those neighboring measurement vectors, instead of processing the entire set of compressive measurements $\mathbf{y}_1, \mathbf{y}_2, \ldots$. For clarity, in Sections 2.3.2 and 2.3.3 we obtain this set of measurement vectors that contains information on the sparse vector $\mathbf{w}_t$, which we need to know before developing the proposed progressive recovery algorithm in Section 2.4.
2.3.2 The effect of adding new measurements to the streaming system

First, we consider the effect of obtaining a new measurement vector on the estimation of the posterior density of the sparse vectors. As we obtain a new measurement vector $y_{t+1}$ at time $t+1$, in general, we need to re-estimate the hyperparameters $\bar{\alpha}_{t+1}$, as well as the posterior density of the sparse coefficient vector $\bar{w}_{t+1}$.

To obtain a stationary point for $L_{t+1}$, we need to recompute the posterior covariance of $\bar{w}_{t+1}$, which has a complexity of $O(D^3)$ with $D = N(t+2)$.

However, by exploiting the structure of the system matrix $B_{t+1}$ and $\Sigma^w_{t+1}$, we show that a stationary point for $L_{t+1}$ can be obtained by re-estimating just a small portion of $\bar{\alpha}_{t+1}$, $\mu_{t+1}$, and $\Sigma^w_{t+1}$, where $\mu_{t+1}$ and $\Sigma^w_{t+1}$ denotes the posterior mean and covariance of $\bar{w}_{t+1}$ at time instant $t+1$. We do this by utilizing the following proposition which ensures that the posterior mean and covariance of $w_1, \ldots, w_{t-d}$ do not change from those we obtained at time instant $t$, although at time instant $t+1$ we also take into account the new measurement vector $y_{t+1}$.

**Proposition 1** The posterior estimation of $w_1, \ldots, w_{t-d}$ that can be obtained by processing the entire set of measurement vectors $y_1, \ldots, y_t, y_{t+1}, \ldots$ is equal to that we obtained at time instant $t$, i.e., $w_t \sim N(\mu_t(t), \Sigma_t(t))$, where $\mu_t(t)$ and $\Sigma_t(t)$ denote the mean and covariance of $w_t$ obtained at time instant $t$ for $t = 1, \ldots, t-d$. In other words, we can consider that $y_t$ is the last measurement vector that contains information on the sparse vectors $w_1, \ldots, w_{t-d}$.

**Proof 1** The proof is presented in Appendix 3.

Note that in the proposed algorithm $d$ is a design parameter, and the effect of this parameter on the estimation of the unknown signal is discussed in Section 2.5. For example, if $d = 1$, $t = 2$, then the posterior mean and covariance of $w_1$ obtained by processing the measurements $y_1, y_2$ will not change even if we take into account the measurement $y_3$.

2.3.3 The effect of removing old measurements from the streaming system

In the previous sub-section, we showed that with the addition of a new measurement vector, we only need to re-estimate the posterior density of a small portion of the sparse...
coefficient vectors. However, for this re-estimation procedure, we may require the utilization of the entire set of obtained measurements $\{y_t\}_{t=1}^\tau$, which prohibits the use of sliding window processing. In contrast, we show in this sub-section that we only need to retain some measurement vectors from the past for the re-estimation procedure. For this purpose, in this sub-section, we consider the effect of removing the old measurements (forgetting the past) from the streaming system. Specifically, we consider that at time instant $t$ we remove the old measurement vectors $y_1, \ldots, y_{t'}$ from the streaming system and estimate the posterior distribution of $w_t$ using the remaining measurements $y_{t-t'} = [y_{t+1}^T \ldots y_{t}^T]^T \in \mathbb{R}^{M(t-t')}$. The following proposition ensures that the posterior density of $w_t$ obtained by processing only measurements $y_{t-t'}$ is same as that obtained in section 2.3.1 by processing the entire set of the measurements at time instant $t$, i.e., $\{y_t\}_{t=1}^\tau$.

**Proposition 2** The posterior density of sparse vectors $w_{t'+d+2}, \ldots, w_{t+1}$ at time instant $t$ by processing only the measurements $y_{t'+1}, \ldots, y_t$ is equal to that which we obtained by processing the entire set of measurements at time $t$, i.e., $y_1, \ldots, y_t$. In other words, we can consider that $y_{t-d-1}$ is the oldest measurement vector that contains information on the sparse vector $w_t$.

**Proof 2** The proof is presented in Appendix 4.

For example, let $d = 1, t = 3$ and $t' = 1$. Then, the posterior mean and covariance of $w_4$ obtained by processing only measurements $y_2, y_3$ will be equivalent to that we can obtain by processing measurements $y_1, y_2, y_3$.

### 2.4 Recursive recovery of the signal using sliding windows

Sections 2.3.2 and 2.3.3 mean we can consider that all the information on a sparse vector $w_t$ in the considered system model is contained within a small number of measurement vectors adjacent to $y_t$. Thus, we can estimate the posterior density of the sparse vector $w_t$ by processing only those measurement vectors that contain information on $w_t$, rather than processing the entire set of measurements. Hence, in this section we investigate the sparse recovery problem in terms of sliding window processing and propose a recovery algorithm based on SBL to recover the sparse coefficient vectors $w_t$. Since the consecutive processing windows overlap, they share some common sparse vectors. We utilize the preliminary information from the preceding interval to improve the performance of the signal recovery algorithm. To make the recovery process more
robust to the error propagation, we utilize the covariance of the reconstructed sparse vectors from the preceding interval and specifically take into account its effect on the reconstruction process of the current interval.

### 2.4.1 Processing window at time instant $t$

According to Proposition 1 and Proposition 2, only the measurement vectors $\mathbf{y}_{t-2d-1}, \ldots, \mathbf{y}_t$ contain information on the sparse vector $\mathbf{w}_{t-d}$. Thus, we design the processing window (or the active interval) at time instant $t$ with the goal of outputing an estimation for the sparse vector $\mathbf{w}_{t-d}$, denoted as $\hat{\mathbf{w}}_{t-d}(t)$. The measurement vector $\hat{\mathbf{y}}_t$ consists of the current measurement $\mathbf{y}_t$ and $2d+1$ previous measurements. Hence, the linear measurement model for the active interval at time instant $t$ can be expressed as

$$\hat{\mathbf{y}}_t = \hat{\mathbf{B}}_t \hat{\mathbf{w}}_t + \hat{\mathbf{e}}_t,$$

(24)

where $\hat{\mathbf{B}}_t \in \mathbb{R}^{M(2d+2) \times N(2d+3)}$ denotes the portion of the system matrix inside the current active interval (see Fig. 2), $\hat{\mathbf{w}}_t = \begin{bmatrix} \mathbf{w}_{t-2d-1}^T \cdots \mathbf{w}_{t+1}^T \end{bmatrix}^T \in \mathbb{R}^{N(2d+3)}$ denotes the sparse coefficients vector inside the active interval, and the measurement noise vector is denoted by $\hat{\mathbf{e}}_t = \begin{bmatrix} \mathbf{e}_{t-2d-1}^T \cdots \mathbf{e}_{t}^T \end{bmatrix}^T \in \mathbb{R}^{M(2d+2)}$.

It follows from Section 2.3.2 that only the posterior distribution over $\hat{\mathbf{w}}_t$, where $\hat{\mathbf{w}}_t = \begin{bmatrix} \mathbf{w}_{t-d}^T \cdots \mathbf{w}_{t-1}^T \end{bmatrix}^T \in \mathbb{R}^{N(d+2)}$, changes with the addition of the new measurement vector $\mathbf{y}_t$ (see (125) and (126)). Hence, to single out the contribution of $\hat{\mathbf{w}}_t$ to $\hat{\mathbf{y}}_t$, we express the measurement model in (24) as

$$\hat{\mathbf{y}}_t = \mathbf{\hat{B}}_t \hat{\mathbf{w}}_t + \hat{\mathbf{B}}_t \hat{\mathbf{w}}_t + \hat{\mathbf{e}}_t,$$

(25)

where we partition the system matrix as $\mathbf{B}_t = \begin{bmatrix} \mathbf{\hat{B}}_t & \mathbf{\hat{B}}_t \end{bmatrix}$: $\mathbf{\hat{B}}_t$ denotes the submatrix of $\mathbf{B}_t$ obtained by keeping the first $N(d+1)$ columns of $\mathbf{\hat{B}}_t$, $\mathbf{\hat{B}}_t$ denotes the submatrix of $\mathbf{\hat{B}}_t$ obtained by removing the first $N(d+1)$ columns of $\mathbf{\hat{B}}_t$, and $\hat{\mathbf{w}}_t = \begin{bmatrix} \mathbf{w}_{t-2d-1}^T \cdots \mathbf{w}_{t-1}^T \end{bmatrix}^T \in \mathbb{R}^{N(d+1)}$. For convenience, we have presented all the notations used in the algorithm development in Fig. 2.

### 2.4.2 SBL algorithm for streaming signal recovery

According to Section 2.3.2, the posterior density over $\hat{\mathbf{w}}_t$ should not change from the estimation obtained at time instant $t - 1$ due to the addition of measurement vector
To derive the SBL algorithm, we first model the sparsity of the vector $\tilde{w}_t$ using the 2-stage hierarchical prior model introduced in Section 2.3.1, i.e,

$$p(\tilde{w}_t | \alpha) = \prod_{t=d-1}^{t+1} \prod_{\tau=1}^{N} \mathcal{N}(w_{t,\tau} | 0, \alpha_{t,\tau}^{-1}).$$  \hspace{1cm} (26)$$

and

$$p(\alpha | a, b) = \prod_{t=d-1}^{t+1} \prod_{\tau=1}^{N} \text{Gamma}(\alpha_{t,\tau} | a, b).$$ \hspace{1cm} (27)$$

Fig. 2. System matrix at time instant $t$. The shaded area shows the system matrix $\tilde{B}_t$, inside the processing interval at time instant $t$ for $d = 1$. We have divided the system matrix $\tilde{B}_t$ into two parts, i.e., $\tilde{B}_t = [\hat{B}_t \ \bar{B}_t]$, to single out the contribution of $\bar{B}_t$, on $\bar{y}_t$. $\hat{B}_t$ denotes the submatrix of $\tilde{B}_t$, obtained by keeping the first $N(d+1)$ columns of $\tilde{B}_t$, $\bar{B}_t$ denotes the submatrix of $\tilde{B}_t$, obtained by removing the first $N(d+1)$ columns of $\tilde{B}_t$.

(Reprinted by permission [66] ©2017 IEEE).

$y_t$, i.e., $p(w_t | y_{t-1}) = p(w_t | \tilde{y}_t)$ for $\tau = t-2d-1, \ldots, t-d-1$ (see (128)). Hence, we keep the posterior mean and covariance of $\tilde{w}_t$ fixed at those obtained at time instant $t-1$ throughout the process of estimating the posterior over the sparse vector $\tilde{w}_t$. Note that, Sections 2.3.2 and 2.3.3 suggest this procedure ensures that $p(\tilde{w}_t | \tilde{y}_t) = p(\tilde{w}_t | y_t)$ for any time instant $t$.

In this sub-section, we derive a fast and computationally efficient SBL algorithm for estimating the posterior density of $\tilde{w}_t$ from the measurements $\tilde{y}_t$. At time instant $t$, instead of re-estimating the hyperparameters and the posterior density of $\tilde{w}_t$ from the ground up, we utilize the estimates from the preceding interval and refine them using the new measurements. Since we utilize only a small portion of the measurements and we need to incorporate the preliminary information from the preceding interval, the SBL procedure does not directly follow from Section 2.3.1.
We set \(a = b = 0\) to make the hyperpriors non-informative. Having defined the priors, the Bayesian inference proceeds by calculating the posterior density \(p(\tilde{w}_t, \tilde{\alpha}_t | \tilde{y}_t)\) by utilizing the following decomposition:

\[
p(\tilde{w}_t, \tilde{\alpha}_t | \tilde{y}_t) = p(\tilde{w}_t | \tilde{\alpha}_t, \tilde{y}_t) p(\tilde{\alpha}_t | \tilde{y}_t). \tag{28}
\]

Now, the first term on the right-hand side of (28) can be expressed as

\[
p(\tilde{w}_t | \tilde{\alpha}_t, \tilde{y}_t) = \frac{p(\tilde{y}_t | \tilde{w}_t) p(\tilde{w}_t | \tilde{\alpha}_t)}{p(\tilde{y}_t | \tilde{\alpha}_t)}. \tag{29}
\]

Further, by using (25), we can evaluate the likelihood function \(p(\tilde{y}_t | \tilde{w}_t) = \mathcal{N}(\tilde{y}_t | \eta_t, \Omega_t)\) with

\[
\eta_t = \tilde{B}_t E(\tilde{w}_t | w_t) + \tilde{B}_t \tilde{w}_t,
\]

\[
\Omega_t = \sigma^2 I_{M(2d+2)} + \tilde{B}_t \text{cov}(\tilde{w}_t | w_t) \tilde{B}_t^T. \tag{30}
\]

To compute the conditional expectation \(E(\tilde{w}_t | \tilde{w}_t)\) and covariance \(\text{cov}(\tilde{w}_t | \tilde{w}_t)\), we utilize the preliminary information available on \(\tilde{w}_t = [\tilde{w}_t^T \tilde{w}_t^T]^T\), i.e., the estimation of the posterior density of \(\tilde{w}_t\) at time instant \(t - 1\) given by \(p(\tilde{w}_t | \tilde{y}_{t-1}) = \mathcal{N}(\tilde{w}_t | \mu^w_{t-1}, \Sigma^w_{t-1})\).

For notational convenience, we partition \(\mu^w_{t-1}\) and \(\Sigma^w_{t-1}\) as

\[
\mu^w_{t-1} = \begin{bmatrix} \mu^a_{t-1} \\ \mu^b_{t-1} \end{bmatrix}, \quad \Sigma^w_{t-1} = \begin{bmatrix} \Sigma^{aa}_{t-1} & \Sigma^{ab}_{t-1} \\ \Sigma^{ba}_{t-1} & \Sigma^{bb}_{t-1} \end{bmatrix}, \tag{32}
\]

where \(\mu^a_{t-1} \in \mathbb{R}^{N(d+1)}, \Sigma^{aa}_{t-1} \in \mathbb{R}^{N(d+1) \times N(d+1)}\) are the mean and covariance of \(\tilde{w}_t\) computed at time instant \(t - 1\); similarly \(\mu^b_{t-1} \in \mathbb{R}^{N(d+2)}, \Sigma^{bb}_{t-1} \in \mathbb{R}^{N(d+2) \times N(d+2)}\) are the mean and covariance of \(\tilde{w}_t\) computed at time instant \(t - 1\).\(^8\) Now, with the help of (32), the conditional expectation and covariance of \(\tilde{w}_t\) can be evaluated\(^9\) as [51, p. 87]

\[
E(\tilde{w}_t | \tilde{w}_t) = \mu^w_{t-1} + \Sigma^{aa}_{t-1} \left(\Sigma^{bb}_{t-1}\right)^{-1} \left(\tilde{w}_t - \mu^b_{t-1}\right), \tag{33}
\]

\[
\text{cov}(\tilde{w}_t | \tilde{w}_t) = \Sigma^{aa}_{t-1} - \Sigma^{aa}_{t-1} \left(\Sigma^{bb}_{t-1}\right)^{-1} \Sigma^{ba}_{t-1}. \tag{34}
\]

Consequently, we can rewrite the mean, \(\eta_t\), of the likelihood function by substituting (33) in (30) as

\[
\eta_t = \eta_t + \tilde{U}_t \tilde{w}_t, \tag{35}
\]

where

\[
\eta_t = \tilde{B}_t (\mu^{t-1}_{a} - \Sigma^{aa}_{t-1} \left(\Sigma^{bb}_{t-1}\right)^{-1} \mu^{t-1}_{b}), \tag{36}
\]

\(^8\)Note that \(w_{t+1}\) has not been estimated at time instant \(t - 1\).

\(^9\)Since \(\Sigma^w_{t-1}\) is a positive semi-definite matrix, we consider the pseudo inverse of the matrix.
\[ \tilde{U}_i = \tilde{B}_i + \tilde{B}_i \Sigma_{ab}^{-1} (\Sigma_{bb}^{-1})^T. \]  

Now, we can evaluate the posterior density of \( \tilde{w}_i \) for given CS measurements \( \tilde{y}_i \), analytically [51, p.90] as \( p(\tilde{w}_i | \alpha, \tilde{y}_i) = \mathcal{N}(\tilde{w}_i | \mu_w, \Sigma_w) \) with

\[ \mu_w = \tilde{A}_i^{-1} \tilde{U}_i^T (\Omega_i + \tilde{U}_i \tilde{A}_i^{-1} \tilde{U}_i^T)^{-1} \tilde{y}_i, \]

\[ \Sigma_w = \tilde{A}_i^{-1} - \tilde{A}_i^{-1} \tilde{U}_i^T (\Omega_i + \tilde{U}_i \tilde{A}_i^{-1} \tilde{U}_i^T)^{-1} \tilde{U}_i \tilde{A}_i^{-1}, \]

where \( \tilde{A}_i = \text{diag}(\alpha_i) \) and \( \tilde{y}_i = \tilde{y}_i - \tilde{\eta}_i \).

The second term on the right-hand side of (28), posterior density of the hyperparameters \( p(\alpha_i | \tilde{y}_i) \), cannot be evaluated analytically. Hence, we utilize the evidence procedure and approximate that \( p(\alpha_i | \tilde{y}_i) \) is sharply peaked at its mode, i.e., \( p(\alpha_i | \tilde{y}_i) \approx \delta(\alpha_i^{MP}) \), where \( \alpha_i^{MP} \) denotes the mode of \( p(\alpha_i | \tilde{y}_i) \). We can then approximate the posterior density of \( \tilde{w}_i \) by \( p(\tilde{w}_i | \alpha_i^{MP}, \tilde{y}_i) \), where \( \alpha_i^{MP} = \arg \max \hat{L}_i \), with

\[ \hat{L}_i = -\frac{1}{2} \log |\tilde{C}_i| - \frac{1}{2} \tilde{y}_i^T \tilde{C}_i^{-1} \tilde{y}_i, \]

where \( \tilde{C}_i = \Omega_i + \tilde{U}_i \tilde{A}_i^{-1} \tilde{U}_i^T \). Following a similar approach to the one we used in Section 2.3.1, we can obtain a relationship equivalent to (22), where \( \mu_w^\gamma \) and \( \Sigma_w^\gamma \) is replaced by \( \mu_w^\gamma \) and \( \Sigma_w^\gamma \) respectively, at a stationary point of \( \hat{L}_i \). The iterative procedure in Algorithm 1 can then be adopted to obtain the posterior density of \( \tilde{w}_i \).

Here, we derive an alternative algorithm, which is fast and computationally efficient, that maximizes \( \hat{L}_i \) based on the following decomposition:

\[ \tilde{C}_i = \Omega_i + \sum_{i=1, i \neq n}^{N(d+2)} \alpha_n^{-1} u_n u_n^T + \alpha_n^{-2} u_n u_n^T, \]

\[ = \tilde{C}_{-n} + \alpha_n^{-1} u_n u_n^T, \]  

where, for notational convenience, we have considered that, \( \alpha_n \) is the \( n \)th element\(^\text{10}\) of \( \alpha_i \), \( u_n \) is the \( n \)th column of \( U_i \), and \( \tilde{C}_{-n} \) is \( \tilde{C}_i \) with the contribution of vector \( u_n \) removed. By substituting (41) in (40), and then equating the derivative of \( \hat{L}_i \) with respect to \( \alpha_n \) to zero, we can show that the maximum of \( \hat{L}_i \), w.r.t. \( \alpha_n \) when all the others are fixed, is achieved [89] when

\[ \alpha_n = \begin{cases} 
\frac{s_n^2}{q_n^2 - s_n} & \text{if } q_n^2 - s_n > 0 \\
\infty & \text{otherwise}
\end{cases} \]

\(^{10}\)Let \( n = jN + k \), then \( \alpha_n = \alpha_{i-d+k} \).
where \( s_n = u_n^T \hat{C}_n^{-1} u_n \) and \( q_n = u_n^T \hat{C}_n^{-1} \hat{y}_n \). Note that in the case of \( \alpha_n = \infty \), the column \( u_n \) is pruned out of the model, and the corresponding \( \mu_n^w \) (where \( \mu_n^w \) denotes the \( n \)th element of the posterior mean \( \mu^w \)) is set to zero.

Thus, by noting that (42) provides a systematic method for deciding which columns of \( \hat{U} \) should be included in the model, and which should be excluded, we can develop an iterative algorithm to estimate \( \alpha_n^{MP}, \mu_n^w \) and \( \Sigma_n^w \) as follows. First, we choose a candidate \( u_n \) and add/remove it to/from the model or re-estimate the value of \( \alpha_n \) according to (42). Then, the quantities \( \mu_n^w \) and \( \Sigma_n^w \) can be computed\(^{12}\) by (38) and (39) respectively. However, for a more efficient computation of \( \mu_n^w \) and \( \Sigma_n^w \), we implement the fast update formulae given in Appendix 2 for add, remove and re-estimation operations. Finally, we calculate the quantities \( s_n \) and \( q_n \) for all \( u_n \). For this purpose, we maintain and update the quantities

\[
S_n = u_n^T \hat{C}_n^{-1} u_n = u_n^T \Omega_n^{-1} u_n - u_n^T \Omega_n^{-1} \hat{U}_t \Sigma_n^w \hat{U}_t^T \Omega_n^{-1} u_n, \tag{43}
\]

\[
Q_n = u_n^T \hat{C}_n^{-1} \hat{y}_n = u_n^T \Omega_n^{-1} \hat{y}_n - u_n^T \Omega_n^{-1} \hat{U}_t \Sigma_n^w \hat{U}_t^T \Omega_n^{-1} \hat{y}_n, \tag{44}
\]

for all \( u_n \). We utilize the corresponding update formulae provided in Appendix 2 to update \( S_n \) and \( Q_n \) for each add, remove, or re-estimation operation. The new values of \( s_n \) and \( q_n \) then follow from \(^{52}\)

\[
s_n = \frac{\alpha_n S_n}{\alpha_n - S_n}, \quad q_n = \frac{\alpha_n Q_n}{\alpha_n - S_n}. \tag{45}
\]

Using the new values of \( s_n \) and \( q_n \), once more for a candidate \( u_n \), the hyperparameter \( \alpha_n \) is updated according to (42). This iterative process is carried out until the increment in \( \mathcal{L}_t \) for each potential update\(^{13}\), denoted as \( \triangle \mathcal{L}_n \) for all \( u_n \), is smaller than a predefined threshold \( \varepsilon \). Following the convergence, we set \( p(\hat{w}_t | \hat{y}_t) = \mathcal{N}(\hat{w}_t | \mu^w, \Sigma^w) \) and output \( \hat{w}_{t-d}(t) = \mu_{t-d}(t) \), where \( \mu_{t-d}(t) \) denotes the mean of \( w_{t-d} \) obtained at time instant \( t \) (i.e., first \( N \) elements of \( \mu^w \)), as the posterior estimation of \( \hat{w}_{t-d} \).

### 2.4.3 Recovery algorithm with the warm-start

We reduce the computational cost of the recovery algorithm by providing a warm-start to the SBL algorithm. We initialize the SBL algorithm using the estimated

\(^{11}\) We start with an empty model and \( u_n \), with the largest \( ||u_n^T \hat{y}_n||^2/||u_n||^2 \) value included in the model in the first iteration.

\(^{12}\) Note that only the \( \alpha_n \neq \infty \) values and corresponding \( u_n \) are used in computing \( \mu_n^w \) and \( \Sigma_n^w \).

\(^{13}\) An efficient method for computing the increment in \( \mathcal{L}_t \) for different changes in the model is presented in Appendix 2.
support of $\bar{w}$ from the previous interval, i.e., $\mathcal{S}_t = \text{supp}(\mu_{t-1}^{\bar{w}})$, where the function $\text{supp}(w) = \{i : w_i \neq 0\}$, and the corresponding hyperparameters $\bar{\alpha}_{\mathcal{S}_t} = \bar{\alpha}_{\mathcal{S}_t}^{-1}$. The proposed warm-start procedure significantly reduces the number of add, delete and re-estimation operations of the SBL algorithm that are required for the estimation of the posterior density of $\bar{w}$. We can summarize the SBL algorithm for the current active interval as follows:

Algorithm 2  SBL based recursive recovery algorithm
1. **Inputs:** $\tilde{y}_t, \tilde{B}_t, \mu_{t-1}^{\bar{w}}, \Sigma_{t-1}^{\bar{w}}$ and $\bar{\alpha}_{t-1}$.
2. **Initialization:**
   a) Compute $\bar{U}_t, \bar{\eta}_t$ and $\Omega_t$, using (37), (36) and (31) respectively.
   b) $\mathcal{S}_t = \text{supp}(\mu_{t-1}^{\bar{w}})$.
   c) $\bar{\alpha}_{\mathcal{S}_t} = \bar{\alpha}_{\mathcal{S}_t}^{-1}$, all the other $\alpha_n$ are notionally set to infinity.
   d) Compute $\Sigma_{\bar{w}}, \mu_{\bar{w}}$, using (39) and (38) with $\bar{U}_t = \bar{U}_{\mathcal{S}_t}$.
   e) Compute $s_n$ and $q_n$, using (43), (44) and (45) for all $u_n$.
3. **while** $\triangle L_n \not\leq \epsilon$ for all $u_n$ **do**
4. Select a candidate vector $u_n$ (or equivalently choose a $\alpha_n$)
5. Compute $\theta_n = q_n^2 - s_n$.
6. **if** $\theta_n > 0$ AND $\alpha_n < \infty$ (i.e., $u_n$ is in model) **then**
   a) Re-estimate $\alpha_n$, using (42).
   b) Update $\Sigma_n$ and $\mu_n$, using (110) and (111).
   c) Update $S_m, Q_m$ for all $u_m$, using (112) and (113)
7. **else if** $\theta_n > 0$ AND $\alpha_n = \infty$ **then**
   a) Add $u_n$ to the model, update $\alpha_n$ using (42).
   b) Update $\Sigma_n$ and $\mu_n$, using (105) and (106).
   c) Update $S_m, Q_m$ for all $u_m$, using (107) and (108)
8. **else if** $\theta_n \leq 0$ AND $\alpha_n < \infty$ **then**
   a) Prune $u_n$ from the model, set $\alpha_n = \infty$.
   b) Update $\Sigma_n$ and $\mu_n$, using (115) and (116).
   c) Update $S_m, Q_m$ for all $u_m$, using (117) and (118)
9. **end if**
10. Update $s_n, q_n$ for all $u_n$, using (45).
11. **end while**
12. **Outputs:** $\mu_n^{\bar{w}}, \Sigma_n^{\bar{w}}$ and $\bar{\alpha}_t$. 

43
We derived fast update formulae for add, delete, and re-estimate operations in Algorithm 2 as given in Appendix 2. Further, in Appendix 2, we provide formulae for computing $\triangle \mathcal{L}_n$ for each potential update for all $u_n$. Thus, in Step 4 in Algorithm 2, rather than selecting $u_n$ randomly, we choose the $u_n$ with the largest $\triangle \mathcal{L}_n$ for fast convergence [52].

At time instant $t$, following the convergence of the SBL algorithm, i.e., obtaining an estimation for the posterior density of $\hat{w}_t$, we reconstruct the signal block $x_{t-d-1}$, using (10) as

$$\hat{x}_{t-d-1} = \begin{bmatrix} P_0^T & P_1^T \end{bmatrix} \begin{bmatrix} \hat{w}_{t-d-1}(t-1) \\ \hat{w}_{t-d}(t) \end{bmatrix},$$

(46)

where $\hat{w}_{t-d-1}(t-1)$ is the output at time instant $t-1$. We then shift the active interval by removing the oldest measurement $y_{t-2d-1}$ and adding the new one $y_{t+1}$, i.e., $\tilde{y}_{t+1} = [y_{t-2d}^T \ldots y_{t+1}^T]^T$. We follow this procedure to recover the entire streaming signal.

Remarks: As shown in [33, 48], the cost function (21) (or (40)) of the SBL retains a desirable property of the sparsity (i.e., the global minimum is achieved at the maximally sparse solution), which provides improved performance in the SBL algorithm when utilized to reconstruct a sparse signal. Since we utilized the fast update formulae based on a suboptimal solution of the cost function of SBL, this performance gain can be reduced [34]. In addition, the major performance improvement of the proposed algorithm compared to [55] is the robustness to error propagation especially in the case of sliding window based processing, where the signal reconstructed in the previous window is used to cancel out its contribution in the measurement vector processed during the current window. The $l_1$-homotopy algorithm proposed in [55] does not take into account the reconstruction errors of the recovered sparse vectors $w_t$. The authors assume that the recovered sparse vectors are exact, and their effect is directly removed from the measurements (see (13) in [55]), which makes the algorithm prone to error propagation. In contrast, we utilized the covariance of the reconstructed sparse vectors from the preceding interval and specifically took into account its effect in the reconstruction process of the current active interval. This prevents the propagation of the reconstruction error through the signal recovery process. Another key advantage of the proposed algorithm is the lack of tuning parameters. In contrast, the performance of the algorithm in [55] critically depends on the choice of parameters used for different procedures in the algorithm, such as regularization, re-weighting, and signal prediction.
2.5 Numerical Results

In this section, we present simulation results to demonstrate the performance of the proposed recursive SBL algorithm and compare its performance with the state of the art $l_1$-homotopy algorithm in [55]. For a fair comparison, we use a simulation set-up similar to the one presented in [55].

2.5.1 Simulations with chirp signals

A chirp is a signal in which the frequency increases and/or decreases with time. They are commonly used in applications such as sonar, radar [90], and spread spectrum communications [91], and they can be observed naturally in phenomena such as biological signals and machine vibrations [90]. In the simulations, we first use some chirp signals available in the Wavelab toolbox [92] for the discrete-time signal, namely Chirps, LinChirp, and TwoChirp. We generate $2^{15}$ samples of the discrete-time signal from the Wavelab toolbox and prepend with $N = 256$ zeros. We used an LOT, designed using modified cosine-IV basis functions that were multiplied by smooth windows [13, 55], with length $L = 2N$ for the sparse representation of the signal $x(n)$. We follow the linear measurement model (9), $y_t = \Phi_t x_t + e_t$, to acquire the signal $x(n)$. We generate a set of measurement vectors $y_t \in \mathbb{R}^M$ from non-overlapping blocks $x_t$ containing $N$ samples of the signal $x(n)$ at a compression ratio $R = M/N$. Each entry of the measurement matrix $\Phi_t$ is set to $\pm 1/\sqrt{M}$ independently at random with equal probability. We add a Gaussian noise to the measurements where the measurement noise vector $e_t$ follows the distribution $e_t \sim \mathcal{N}(0, \sigma^2 I_M)$. The noise variance $\sigma^2$ is selected such that the expected SNR with respect to the measurements $\Phi_t x_t$ is $\rho$ dB, i.e., $\rho = 10\log \left( \frac{||\Phi_t x_t||_2^2}{M\sigma^2} \right)$.

We recover the signal $x(n)$ using the proposed SBL based algorithm (Algorithm 2) and the $l_1$-homotopy algorithm proposed in [55]. For the $l_1$-homotopy algorithm\footnote{We used the MATLAB implementation provided in \url{http://users.ece.gatech.edu/sasif/homotopy/index.html} for the $l_1$-homotopy algorithm and the LOTs.}, we followed the system set-up given in [55], and the parameters are chosen according to the guidelines given in [55] to obtain the best performance\footnote{It is worth mentioning that the choice of the parameters is crucial in the $l_1$-homotopy method. During the simulations we observed that, for some parameter set-ups, the $l_1$-homotopy method failed to reconstruct the signal. We removed such unsuccessful reconstructions in presenting the results. In contrast, the proposed SBL method performed well across all parameter set-ups.}. For all the simulations, we

\footnote{Recall that $x_0$ is an all zero vector.}

14

15

16
performed 100 independent trials for the reconstruction of the streaming signal from compressive measurements, and the results are averaged over all trials.

Fig. 3a presents the quality of the reconstructed signal for different values of $d$ in terms of the average signal-to-error ratio (SER) in dB defined as

$$\text{SER} = -10 \log \frac{\| x - \hat{x} \|^2}{\| x \|^2},$$

where $x$ denotes the original signal, and $\hat{x}$ denotes the reconstructed signal. We choose $R = 4$ as the compression ratio. The measurement noise is generated with $\text{SNR} = 35$ dB and the stopping threshold of the SBL algorithm is set to $\epsilon = 10^{-5}$. We can see that the SER performance increases with $d$ and then saturates for all signal types recovered from both SBL (except for the LinChirp signal) and $l_1$-homotopy algorithms. Further, the figure demonstrates that the most significant performance gain is achieved when $d$ increases from 0 to 1. Hence, we use $d = 1$ for the rest of the simulations.

algorithm does not have any tuning parameters and we did not observe any failed signal reconstructions during our simulations.

The LinChirp signal has very few non-zero elements compared to the number of basis functions. In such scenarios, the fast-SBL algorithm is sometimes more likely to be trapped in a local maximum [93]. Note that the $l_1$-homotopy method also shows a slight performance degradation when $d$ increases from 2 to 3.
Fig. 3b illustrates the average number of iterations required for the reconstruction of a signal $x(n)$, using the SBL algorithm with and without the warm-start procedure for different signal types. We consider an add, delete, or re-estimate operation in Algorithm 2 as an iteration. The update formulae provided in Appendix 2 were utilized to perform these operations in both scenarios. Hence, the cost for an iteration is approximately given by $O(d^2NM + d^2MS + d^2S^2 + d^3SM^2 + dN)$ flops, assuming a support size of $S$ for all sparse vectors $w_t$. For the algorithm without warm-start (named cold-start), we initialized the recovery algorithm for each active interval following the initialization procedure in [52]. The results show that the proposed warm-start procedure significantly reduces the amount of iterations required for the reconstruction of the streaming signal.

Next, we compare the performance of the proposed recursive SBL algorithm with the $l_1$-homotopy algorithm for different signal types. First, we consider the signal $x(n)$ to be the Chirps signal, and Fig. 4 shows a snapshot of the signal (first 2560 samples), its LOT coefficients, and the reconstruction error of the signal recovered from compressive streaming measurements with a compression ratio $R = 4$ at SNR = 35 dB. The performance comparison of the two algorithms, SBL and $l_1$-homotopy algorithms, is presented in Fig. 5 for recovering the Chirps signal. The stopping threshold of the SBL algorithm is set to $\varepsilon = 10^{-7}$ for $R = 2$ and $\varepsilon = 10^{-5}$ for other cases. The quality of the reconstructed signal for different compression ratios ($R$) is illustrated in Fig. 5a in terms of the average SER. The figure shows that the SBL algorithm has a slightly better SER performance compared to the $l_1$-homotopy algorithm except at $R = 8$, where both the algorithms have poor SER performance. The average MATLAB execution time for recovering a signal $x(n)$ is presented in Fig. 5b. The results show that the proposed SBL algorithm clearly outperformed the $l_1$-homotopy algorithm in terms of computational complexity.

Next, we present the results for the LinChirp signal. A snapshot of the LinChirp signal, its LOT coefficients, and the reconstruction error of the signal recovered from compressive measurements with a compression ratio $R = 4$ at SNR = 35 dB is presented in Fig. 6. The two plots in Fig. 7 present the performance comparison of the two algorithms, the proposed SBL algorithm, and the $l_1$-homotopy algorithm for recovering the LinChirp signal from the compressive measurements. The stopping threshold of the SBL algorithm is set to $\varepsilon = 10^{-5}$. Fig. 7a compares the average SER for the two algorithms. The figure shows that the proposed SBL algorithm has a better SER performance at SNR = 35 dB. In the case of SNR = 30 dB, both algorithms have an almost similar SER performance. The average MATLAB execution time for recovering
Fig. 4. Top left: Snapshot of the Chirps signal. Bottom left: LOT coefficients of the Chirps signal. Top right: Reconstruction error of the signal using SBL algorithm ($R = 4$, SNR = 35 dB). Bottom right: Error in the reconstructed LOT coefficients ($R = 4$, SNR = 35 dB) (Reprinted by permission [66] ©2017 IEEE).

Fig. 5. Performance comparison of the reconstruction of the Chirps signal for the SBL algorithm and $l_1$-homotopy algorithm ($d = 1$). (a) Average SER of the reconstructed signal $x(n)$ for different compression ratios $R$. (b) Average MATLAB execution time for the reconstruction of a signal $x(n)$ (Reprinted by permission [66] ©2017 IEEE).

Fig. 7. Performance comparison of the reconstruction of the LinChirp signal for SBL algorithm and $l_1$-homotopy algorithm ($l = 1$). (a) Average SER of the reconstructed signal $x(n)$ for different compression ratios $R$. (b) Average MATLAB execution time for the reconstruction of a signal $x(n)$ (Reprinted by permission [66] ©2017 IEEE).
a signal $x(n)$ from compressive measurements is presented in Fig. 7b. The results show that the $l_1$-homotopy algorithm has a better MATLAB execution time at compression ratios of $R = 2, 4$, and the proposed SBL algorithm performs better at high compression ratios of $R = 6, 8$. However, we can see that both algorithms have average MATLAB execution times of the same order of magnitude. Further, we have also presented the results using the SBL method with a 3-stage hierarchical prior model [35] (named Laplace). For this purpose, we modified the algorithm presented in [35], using a similar approach as in Section 2.4 to estimate the posterior density of $\hat{\bar{w}}$ from the measurements $\tilde{y}$. The results show that the performance improvement due to the 3-stage hierarchical prior model was negligible in the case of streaming signal recovery.

To demonstrate the advantage of LOT-based reconstruction compared to a block transform-based reconstruction, we repeat the same experiments by replacing the LOT with the DCT for signal representation. We present the average SER results in Fig. 5a and Fig. 7a, utilizing the LASSO solver in the SPAMS toolbox\textsuperscript{18} ($l_1$-DCT) and DCS-AMP method [59]\textsuperscript{19}. We can see a significant improvement in the results for the LOT-based reconstruction compared to the results for the DCT-based reconstruction.

### 2.5.2 Simulations with real data sets

Next, we evaluate the performance of the proposed recursive recovery algorithm using some real data-sets. For this purpose, we use a selected electroencephalogram (EEG) signal from the ‘chbmit’ database in [94] and ‘Summer crickets chirping’ audio signal from [95] as the discrete-time signal $x(n)$. Snapshots of these signals and their LOT coefficients are presented in Fig. 8. We reconstruct the first $2^{15}$ samples of these discrete signals following a simulation set-up similar to the previous experiments. We select $N = 1024$ for the EEG signal and $N = 512$ for the audio signal. We use an LOT, designed using modified cosine-IV basis functions multiplied by smooth windows, with a length of $L = 2N$ for the sparse representation of the signal $x(n)$. We acquire the signal $x(n)$ following the linear measurement model (9). At each time instant $t$, we generate the measurements $y_t$ by sub-sampling the non-overlapping blocks $x_t$, containing $N$ samples of the signal $x(n)$ at a compression ratio of $R = M/N$.

\textsuperscript{18}Available online: http://spams-devel.gforge.inria.fr/

\textsuperscript{19}An alternative recovery algorithm could be developed by modifying the AMP based algorithm in [59] by taking into account the signal structure enforced by LOTs, which is beyond the scope of this thesis. It would be an interesting area of future research especially due to the low complex implementation of the AMP.
Table 1 presents the average SER of the reconstructed signal and the average MATLAB execution time for recovering a signal $x(n)$ from the compressive measurements for different signal types. We performed 50 independent trails for the reconstruction of the signal for all simulations, and the results are averaged over all trails. The results show that both SBL and $l_1$-homotopy algorithms could recover the streaming signal, and the SER performances are almost the same. However, the MATLAB execution time of the $l_1$-homotopy method for the audio signal is significantly larger than that of the SBL algorithm. For completeness, we also present the results for recovering the streaming signal using the greedy algorithm orthogonal matching pursuit (OMP). We develop the OMP-based recovery method by following the implementation in [55] with LOTs and replacing the $l_1$-homotopy based method with OMP to recover the sparse coefficients in each processing window. To reduce the reconstruction errors, we start the OMP algorithm for each processing window with an empty model, which led to a longer runtime, instead of using the estimated support from the preceding interval. We utilize the optimal stopping criterion proposed in [96, Theorem 8] with the maximum allowable none zero elements equal to the number of measurements inside the processing window. The results show that both the SBL and $l_1$-homotopy methods outperform the OMP counterpart in almost all scenarios.
<table>
<thead>
<tr>
<th>SNR = 35 dB</th>
<th>SNR = 30 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal Compression</td>
<td>SBL</td>
</tr>
<tr>
<td>SER runtime</td>
<td>SER runtime</td>
</tr>
<tr>
<td>Crickets</td>
<td></td>
</tr>
<tr>
<td>R = 8</td>
<td>14.28</td>
</tr>
<tr>
<td>R = 6</td>
<td>11.05</td>
</tr>
<tr>
<td>R = 4</td>
<td>7.95</td>
</tr>
<tr>
<td>R = 2</td>
<td>3.0</td>
</tr>
</tbody>
</table>

| EEG |
| R = 8  | 16.36  | 17.07  |
| R = 6  | 11.78  | 12.14  |
| R = 4  | 4.18   | 4.87   |
| R = 2  | 2.26   | 3.03   |

<table>
<thead>
<tr>
<th>SER runtime</th>
<th>SER runtime</th>
<th>SER runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.30</td>
<td>13.02</td>
<td>20.31</td>
</tr>
<tr>
<td>4.95</td>
<td>7.95</td>
<td>12.06</td>
</tr>
<tr>
<td>1.16</td>
<td>1.87</td>
<td>2.58</td>
</tr>
<tr>
<td>2.46</td>
<td>3.09</td>
<td>4.56</td>
</tr>
</tbody>
</table>

Table 1. Comparison of the average SER of the reconstructed signal (in dB) and average runtime (in seconds) for different real datasets.
2.6 Summary and Discussion

This chapter addressed the problem of recovering a streaming signal from compressive, streaming measurements. An LOT was utilized for the sparse representation of the streaming signal to reduce the blocking artifacts. Due to the coupling introduced by the LOTs, each signal block cannot be reconstructed independently using disjoint measurement blocks. Thus, a progressive reconstruction algorithm based on sliding window processing was developed, where the streaming signal was reconstructed over small overlapping shifting intervals. Since the consecutive intervals shared some common sparse vectors, we utilized the preliminary information from the preceding interval to improve the performance of the signal recovery algorithm. For this purpose, we developed a novel SBL algorithm, which is highly efficient for recovering streaming signals. The proposed SBL algorithm utilized the previous estimates as well as the correlations among the non-zero coefficients to improve the performance of the algorithm. Further, a warm-start procedure was proposed, and fast update formulae were derived to reduce the computational cost of the SBL algorithm.

The proposed algorithm can be utilized in a variety of applications such as environmental monitoring and audio/video streaming, where the signal of interest does not have a clear beginning or end. Furthermore, the ability to make a trade-off between the performance and complexity via the window size makes the algorithm versatile for applications with different requirements for computation power and reconstruction performance.
3 Lapped transforms-based image recovery for block compressed sensing

3.1 Introduction

The theory of CS [8, 9] asserts that under certain conditions, a high dimensional signal can be reconstructed from a small number of random linear projections by utilizing computationally efficient recovery algorithms. Generally, these recovery algorithms process the entire set of measurements of the image at once. Although such a method is feasible for a relatively small image, as the image size increases, this quickly becomes prohibitively complex [97, 98]. Block compressed sensing (BCS) [97, 98, 99] is a method usually utilized to reduce this complexity for larger images. In BCS, the image is divided into small disjoint blocks, and each block is acquired independently. However, recovering the image by independently reconstructing each block introduces significant blocking artifacts in the reconstructed image [55, 97, 98, 99, 100].

This blocking effect has been studied in the case of one-dimensional (1-D) signal recovery in [55, 64, 66]. The authors have proposed computationally efficient solutions, based on lapped transforms, to mitigate the blocking artifacts. Building on these results, we develop here an iterative image reconstruction method, based on 2-D lapped transforms and sparse Bayesian learning (SBL) [31]. Specifically, to suppress the blocking artifacts, we jointly process a small number of adjacent measurement blocks to recover an image block. Furthermore, we provide an analysis of the motive behind the joint processing of adjacent measurement blocks, the set of measurement blocks that need to be processed together, and introduce a method for finding the processing order based on the graphical models [51].

3.2 Block compressive sensing model

We consider the recovery of an image \( X \) from compressed sensing measurements. We utilize the block compressive sensing (BCS) procedure to alleviate the huge computation and memory burdens associated with using a dense measurement matrix for sensing and reconstruction processes of the image. Specifically, we divide the image \( X \) into \( H \times W \) non-overlapping blocks with each block having size \( N \times N \). Suppose that \( X_{i,j} \in \mathbb{R}^{N \times N} \)
denotes the \((i, j)\)-th block of the image \(X\). The linear acquisition model is then given by

\[
y_{i,j} = \Phi_{i,j}x_{i,j} + e_{i,j}, \quad i = 1, \ldots, H \quad j = 1, \ldots, W
\]

(47)

where \(x_{i,j} = \text{vec}(X_{i,j}) \in \mathbb{R}^{N^2}\), \(y_{i,j} \in \mathbb{R}^{M}\) contains \(M < N^2\) linear measurements of \(X_{i,j}\), the measurement matrix is denoted by \(\Phi_{i,j} \in \mathbb{R}^{M \times N^2}\), and the measurement noise vector is denoted by \(e_{i,j} \in \mathbb{R}^{M}\), where each entry is assumed to be an independent Gaussian random variable with zero mean and known variance \(\sigma^2\), i.e., \(e_{i,j} \sim \mathcal{N}(0, \sigma^2 I_M)\). This is equivalent to acquisition of the whole image using a block diagonal measurement matrix with diagonal entries \(\Phi_{i,j}\) for \(i = 1, \ldots, H\) and \(j = 1, \ldots, W\).

Our goal is to obtain an estimation of the image \(X\) utilizing the BCS measurements \(y_{i,j}\) for \(i = 1, \ldots, H\) and \(j = 1, \ldots, W\). However, if the image is recovered by reconstructing each image block \(x_{i,j}\) from the measurement \(y_{i,j}\) independently, significant blocking artifacts are introduced in the reconstructed image [55, 97, 98, 99, 100]. Hence, we develop an image reconstruction algorithm based on sliding window processing, where a small number of adjacent measurement blocks are jointly processed for recovering an image block.

3.3 Recovery via lapped transforms

In this section, we analyze the image recovery problem with 2-D lapped transforms based on the graphical models and provide the motive behind joint processing of adjacent measurement blocks. We then identify the set of measurement blocks that need to be processed together and derive an iterative algorithm based on sparse Bayesian learning (SBL) to recover an image using BCS measurements.

3.3.1 BCS with lapped transforms

A lapped transform (LT) is a type of linear discrete block transformation developed to eliminate the artificial discontinuities between the adjacent blocks in traditional block transform coding of images [13, 79, 101, 102]. While the basis functions of such a transformation extend beyond the block boundaries, creating an overlap to eliminate the blocking effect, the transform will maintain orthogonality. Thus, we use an LT for the sparse representation of the image \(X\) to reduce the blocking artifacts that

\[\text{The LT satisfy the perfect reconstruction property if } P_1Q_1^T + P_0Q_0^T = I \text{ and } P_1Q_0^T = P_0Q_1^T = 0, \text{ where } [P_1, P_0] \text{ is the forward transform matrix, and } [Q_1, Q_0] \text{ is the inverse transform matrix [13, 79, 101, 102].}\]
occur due to the segmentation in BCS in this work. By utilizing the synthesis (or inverse) transform matrix denoted by \( \begin{bmatrix} Q_1 & Q_0 \end{bmatrix} \in \mathbb{R}^{N \times 2N} \), the image block \( X_{i,j} \) can be synthesized as [101, 102]

\[
X_{i,j} = Q_1 W_{i,j} Q_1^T + Q_0 W_{i,j+1} Q_0^T + Q_1 W_{i+1,j} Q_1^T + Q_0 W_{i+1,j+1} Q_0^T.
\]

(49)

Then utilizing vec(ABC) = (C^T \otimes A)vec(B), we obtain

\[
x_{i,j} = \Psi_{11} w_{i,j} + \Psi_{10} w_{i+1,j} + \Psi_{01} w_{i,j+1} + \Psi_{00} w_{i+1,j+1},
\]

(50)

where \( w_{i,j} = \text{vec}(W_{i,j}) \in \mathbb{R}^{N^2} \) and \( \Psi_{kl} = Q_k \otimes Q_l \in \mathbb{R}^{N^2 \times N^2} \) for \( k, l = 0, 1 \). Now, by substituting (50) in the measurement model (47), we obtain the linear relationship between the measurement vector \( y_{i,j} \) and the transform domain coefficient vectors \( w_{i,j} \) as

\[
y_{i,j} = \Phi_{i,j} \begin{bmatrix} \Psi_{11} & \Psi_{10} & \Psi_{01} & \Psi_{00} \end{bmatrix} \begin{bmatrix} w_{i,j} \\
W_{i+1,j} \\
W_{i,j+1} \\
W_{i+1,j+1} \end{bmatrix} + e_{i,j}.
\]

(51)
Consider the problem of estimating $x_{i,j}$ (or equivalently the image block $X_{i,j}$) using the measurement vectors $y_{i,j}$ for $i = 1, \ldots, H$ and $j = 1, \ldots, W$. It is clear from (50) that $x_{i,j}$ can be synthesized with the knowledge of the sparse vectors $w_{i,j}$, $w_{i+1,j}$, $w_{i,j+1}$, and $w_{i+1,j+1}$. Thus, our goal is to estimate $w_{i,j}$ for $i = 1, \ldots, H + 1$ and $j = 1, \ldots, W + 1$ given the measurements $y_{i,j}$ for $i = 1, \ldots, H$ and $j = 1, \ldots, W$. In other words, we need to compute the posterior distribution $p(w_{1,1}, \ldots, w_{H+1,W+1} | y_{1,1}, \ldots, y_{H,W})$. The directed graph associated with this posterior distribution is illustrated in Fig. 10. The empty circles represent the random variables, while the observed variables are indicated with shaded circles. According to the $d$–separation criteria [51, pg. 378], all $w_{i,j}$ are statistically dependent given $y_{i,j}$, since the nodes $y_{i,j}$ have converging arrows, and they are observed. Thus, the optimal solution would be to jointly estimate all sparse vectors $w_{i,j}$ for $i = 1, \ldots, H + 1$ and $j = 1, \ldots, W + 1$ from the measurement vectors $y_{i,j}$ for $i = 1, \ldots, H$ and $j = 1, \ldots, W$. For a large image, the memory requirement and the computational complexity of the recovery algorithm become exceedingly high for such an approach.

Building on the concept utilized in [55, 64, 66], we consider that all the information on a transform domain coefficient block $w_{i,j}$ is contained within a small number of measurement blocks adjacent to $y_{i,j}$. Thus, to reduce the memory burden and the complexity of the recovery algorithm, we propose an iterative reconstruction algorithm, based on sliding window processing. We jointly process a small number of adjacent measurement blocks $y_{i,j}$ (inside a processing interval) to estimate each sparse transform domain coefficient block $w_{i,j}$ (or equivalently the image blocks). Since multiple intervals overlap, they share some common transform domain coefficient blocks $w_{i,j}$. Hence, we
propose a Kalman-like implementation, where the preliminary information obtained in the preceding processing intervals is efficiently reused to refine the estimation of the transform domain coefficient blocks $w_{i,j}$ in the current estimation interval.

### 3.3.2 Processing window

We consider that only the measurement vectors $y_{k,l}$ for $k = i - d, \ldots, i + d - 1, l = j - d, \ldots, j + d - 1$ contain information on the transform domain coefficient block $w_{i,j}$ (see Fig. 11). Thus, we design the processing window (or the active interval) with the goal of outputting an estimation for $w_{i,j}$, denoted as $\hat{w}_{i,j}$, by including only the measurement vectors $y_{k,l}$ for $k = i - d, \ldots, i + d - 1, l = j - d, \ldots, j + d - 1$. Note that, in the proposed algorithm, $d$ is a design parameter, and the effect of $d$ on the reconstruction of an image is discussed in Section 3.4. We denote the measurement vector inside the active interval by $\tilde{y}_{i,j} \in \mathbb{R}^{4d^2M}$. The vector $\tilde{y}_{i,j}$ consists of $y_{k,l}$ for $k = i - d, \ldots, i + d - 1, l = j - d, \ldots, j + d - 1$ (see Fig. 11), which are stacked according to the scanning pattern in Fig. 12. Hence, the linear measurement model for the active interval can be expressed...
as
\[
\tilde{y}_{i,j} = \tilde{B}_{i,j} \tilde{w}_{i,j} + \tilde{e}_{i,j},
\] (52)

where \(\tilde{B}_{i,j} \in \mathbb{R}^{4d^2M \times (2d+1)^2N^2}\) models the composite effect of the measurement and representation matrices inside the active interval, \(\tilde{w}_{i,j} \in \mathbb{R}^{(2d+1)^2N^2}\) denotes the sparse vector inside the active interval, and the measurement noise vector is denoted by \(\tilde{e}_{i,j} \in \mathbb{R}^{4d^2M}\). The sparse vector \(\tilde{w}_{i,j}\) stacks the transform domain coefficient blocks \(w_{k,l}\) for \(k = i-d, \ldots, i+d, l = j-d, \ldots, j+d\) (see Fig. 11) and \(\tilde{e}_{i,j}\) stacks the error vectors \(e_{k,l} = \tilde{e}_{i,j}\) in the same scanning pattern as \(\tilde{y}_{i,j}\).

Subject to the utilized scanning pattern, we may have a subset of the transform domain coefficient blocks in \(\tilde{w}_{i,j}\), which have already been committed to the output (see Fig. 12). We denote the sub-vector of \(\tilde{w}_{i,j}\) containing these transform domain coefficient blocks by \(\hat{\tilde{w}}_{i,j}\). To single out the contribution of \(\hat{\tilde{w}}_{i,j}\) on \(\tilde{y}_{i,j}\), we express the measurement model in (52) as
\[
\tilde{y}_{i,j} = \hat{\tilde{B}}_{i,j} \hat{\tilde{w}}_{i,j} + \tilde{B}_{i,j} \tilde{e}_{i,j} + \tilde{e}_{i,j},
\] (53)

where we partitioned \(\hat{\tilde{B}}_{i,j} = [\tilde{B}_{i,j} \hat{\tilde{B}}_{i,j}]\), \(\hat{\tilde{B}}_{i,j}\) denotes the submatrix of \(\tilde{B}_{i,j}\) obtained by keeping the columns of \(\tilde{B}_{i,j}\) corresponding to the indices of \(\hat{\tilde{w}}_{i,j}\), \(\tilde{B}_{i,j}\) denotes the submatrix of \(\tilde{B}_{i,j}\) obtained by removing the columns of \(\hat{\tilde{B}}_{i,j}\) corresponding to the indices of \(\hat{\tilde{w}}_{i,j}\) and we have partitioned the sparse vector \(\hat{\tilde{w}}_{i,j} = [\tilde{w}_{i,j}^T \tilde{w}_{i,j}^T]^T\).

The goal of the recovery algorithm is to obtain an estimation for the sparse vector \(\hat{\tilde{w}}_{i,j}\) by utilizing the measurement vector \(y_{i,j}\) while removing the effect of the estimate of \(\hat{\tilde{w}}_{i,j}\) on \(\tilde{y}_{i,j}\). Note that if we have a point estimate of \(\hat{\tilde{w}}_{i,j}\), we can use \(y_{i,j} - \hat{\tilde{B}}_{i,j} \hat{\tilde{w}}_{i,j}\) as the new measurement vector and utilize a sparse signal recovery algorithm for the estimation of \(\hat{\tilde{w}}_{i,j}\). However, such an approach is prone to error propagation. Hence, we next derive a recovery algorithm based on sparse Bayesian learning (SBL), which takes into account the uncertainty of the reconstruction of \(\hat{\tilde{w}}_{i,j}\).
3.3.3 Recovery algorithm based on SBL

In the SBL framework, we attempt to estimate the posterior density of the unknown vector $\mathbf{w}_{i,j}$ given the measurement vector $\mathbf{y}_{i,j}$ by employing a sparsity promoting prior on $\mathbf{w}_{i,j}$. Since, we committed the estimate of $\mathbf{w}_{i,j}$ to the output, we keep the marginal density of $\mathbf{w}_{i,j}$ unchanged during the process of estimating the posterior over the sparse vector $\mathbf{w}_{i,j}$.

To derive the SBL algorithm, we first model the sparsity of the vector $\mathbf{w}_{i,j}$, utilizing a hierarchical sparsity promoting prior model. In the literature, several hierarchical prior models have been utilized for $p(\mathbf{w}_{i,j})$ [31, 34]. We utilize the standard 2-stage hierarchical prior model [31, 34] in the sequel. For the notational convenience, let $\mathbf{w}_{i,j} \in \mathbb{R}^K$, i.e., $K = cN^2$, where $c$ is the number of transform domain coefficient blocks in $\mathbf{w}_{i,j}$, and label the elements of $\mathbf{w}_{i,j}$ as $w_1, \ldots, w_K$. In the first stage, we model each element of the vector $\mathbf{w}_{i,j}$ as an independent Gaussian random variable with zero mean and individual precision parameters, i.e.,

$$p(\mathbf{w}_i | \mathbf{a}_i) = \prod_{k=1}^{K} N(w_k | 0, \alpha_k^{-1}),$$  \hspace{1cm} (54)

where $\alpha_k$ denotes the precision of $w_k$, which is the individual hyperparameter associated with $w_k$; and $\mathbf{a}_{i,j} = [\alpha_1 \ldots \alpha_K]^T$ is a vector containing $K$ hyperparameters controlling the precision of each element of $\mathbf{w}_{i,j}$.

In the second stage, hyperpriors over $\mathbf{a}_{i,j}$ are defined. Each precision parameter $\alpha_k$ is modeled as an independent random variable with a Gamma distribution\(^21\) [31], with a common shape parameter $a$ and rate parameter $b$, as $p(\mathbf{a}_k | a, b) = \prod_{k=1}^{K} \text{Gamma}(\alpha_k | a, b)$. By marginalizing over the hyperparameter $\alpha_k$, we can obtain the true prior over $w_k$ analytically, and it corresponds to the density of a Student-$t$ distribution [31, 34]. Hence, the overall prior over the vector $\mathbf{w}_{i,j}$ becomes the product of independent Student-$t$ distributions over $w_k$. For appropriate choice of $a$ and $b$, the Student-$t$ distribution is highly peaked around $w_k = 0$, and hence, this prior favors most of $w_k$ as zero promoting sparsity.

Having defined the priors, we proceed with the Bayesian inference by calculating the posterior density $p(\mathbf{w}_{i,j}, \mathbf{a}_{i,j} | \mathbf{y}_{i,j})$ by utilizing the following decomposition\(^22\)

$$p(\mathbf{w}_{i,j}, \mathbf{a}_{i,j} | \mathbf{y}_{i,j}) = p(\mathbf{w}_{i,j} | \mathbf{a}_{i,j}, \mathbf{y}_{i,j}) p(\mathbf{a}_{i,j} | \mathbf{y}_{i,j}).$$  \hspace{1cm} (55)

\(^21\)Gamma prior is conjugate to the Gaussian likelihood with a known mean.
\(^22\)Note that evaluating $p(\mathbf{w}_{i,j}, \mathbf{a}_{i,j} | \mathbf{y}_{i,j})$ from the standard decomposition, i.e., $p(\mathbf{w}_{i,j}, \mathbf{a}_{i,j} | \mathbf{y}_{i,j}) = p(\mathbf{y}_{i,j} | \mathbf{w}_{i,j}) p(\mathbf{w}_{i,j}, \mathbf{a}_{i,j}) / p(\mathbf{y}_{i,j})$, is intractable, since the normalization factor $p(\mathbf{y}_{i,j})$ cannot be evaluated.
The first term on the right-hand side of (55) represents the (conditional) posterior distribution over the sparse coefficient vector \( \tilde{\mathbf{w}}_{i,j} \), and it can be expressed as

\[
p(\tilde{\mathbf{w}}_{i,j} | \hat{\mathbf{a}}_{i,j}, \tilde{\mathbf{y}}_{i,j}) = \frac{p(\tilde{\mathbf{y}}_{i,j} | \tilde{\mathbf{w}}_{i,j}) p(\tilde{\mathbf{w}}_{i,j} | \hat{\mathbf{a}}_{i,j})}{p(\tilde{\mathbf{y}}_{i,j} | \hat{\mathbf{a}}_{i,j})}.
\] (56)

Now, by using the measurement model (53), we can evaluate the likelihood function

\[
p(\tilde{\mathbf{y}}_{i,j} | \tilde{\mathbf{w}}_{i,j}) = \mathcal{N}(\tilde{\mathbf{y}}_{i,j} | \eta_{i,j}, \Omega_{i,j})
\] (57)

with

\[
\eta_{i,j} = \tilde{\eta}_{i,j} + \tilde{U}_{i,j} \tilde{w}_{i,j},
\]

\[
\Omega_{i,j} = \sigma^2 \mathbf{I}_{4d^2M} + \tilde{B}_{i,j} (\Sigma_{aa} - \Sigma_{ab} (\Sigma_{bb})^{-1} \Sigma_{ba}) \tilde{B}_{i,j}^T,
\]

where,

\[
\tilde{\eta}_{i,j} = \tilde{B}_{i,j} (\mu_a - \Sigma_{ab} (\Sigma_{bb})^{-1} \mu_b),
\]

\[
\tilde{U}_{i,j} = \tilde{B}_{i,j} + \tilde{B}_{i,j} \Sigma_{ab} (\Sigma_{bb})^{-1},
\]

and \( \mu_a, \Sigma_{aa} \) are the estimated mean and covariance of \( \tilde{w}_{i,j} \); similarly \( \mu_b, \Sigma_{bb} \) are the estimated mean and covariance of \( \tilde{w}_{i,j} \).

Now, we can evaluate the (conditional) posterior distribution over \( \tilde{w}_{i,j} \) for given CS measurements \( \tilde{\mathbf{y}}_{i,j} \) in (56) analytically using the Bayes’ theorem for Gaussian variables [51, pp.90] by noting that the likelihood function \( p(\tilde{\mathbf{y}}_{i,j} | \tilde{\mathbf{w}}_{i,j}) \) and the prior \( p(\tilde{\mathbf{w}}_{i,j} | \hat{\mathbf{a}}_{i,j}) \) are both Gaussian (see (57) and (54)) as \( p(\tilde{\mathbf{w}}_{i,j} | \hat{\mathbf{a}}_{i,j}, \tilde{\mathbf{y}}_{i,j}) = \mathcal{N}(\tilde{\mathbf{w}}_{i,j} | \mu_{w}^p, \Sigma_{w}^p) \) with

\[
\mu_{w}^p = \tilde{A}_{i,j}^{-1} \tilde{U}_{i,j}^T (\Omega_{i,j} + \tilde{U}_{i,j} \tilde{A}_{i,j}^{-1} \tilde{U}_{i,j}^T)^{-1} \tilde{\mathbf{y}}_{i,j},
\]

\[
\Sigma_{w}^p = \tilde{A}_{i,j}^{-1} - \tilde{A}_{i,j}^{-1} \tilde{U}_{i,j}^T (\Omega_{i,j} + \tilde{U}_{i,j} \tilde{A}_{i,j}^{-1} \tilde{U}_{i,j}^T)^{-1} \tilde{U}_{i,j} \tilde{A}_{i,j}^{-1},
\]

where \( \tilde{A}_{i,j} = \text{diag}(\tilde{\mathbf{a}}_{i,j}) \) and \( \tilde{\mathbf{y}}_{i,j} = \tilde{\mathbf{y}}_{i,j} - \tilde{\eta}_{i,j} \).

The second term on the right-hand side of (55), the posterior density of the hyperparameters \( p(\hat{\mathbf{a}}_{i,j} | \tilde{\mathbf{y}}_{i,j}) \), cannot be evaluated analytically. Hence, we utilize the evidence procedure [31, 34] and obtain a point estimate for \( \hat{\mathbf{a}}_{i,j} \) by computing the mode of the posterior density \( p(\hat{\mathbf{a}}_{i,j} | \tilde{\mathbf{y}}_{i,j}) \), i.e., \( \hat{\mathbf{a}}_{i,j}^{MP} = \text{argmax} p(\hat{\mathbf{a}}_{i,j} | \tilde{\mathbf{y}}_{i,j}) \). By noting that, \( p(\hat{\mathbf{a}}_{i,j} | \tilde{\mathbf{y}}_{i,j}) \propto p(\hat{\mathbf{a}}_{i,j} | \tilde{\mathbf{y}}_{i,j}) \), we estimate the value of \( \hat{\mathbf{a}}_{i,j}^{MP} \) by maximizing the joint distribution \( p(\hat{\mathbf{a}}_{i,j}, \tilde{\mathbf{y}}_{i,j}) \), or equivalently, its logarithm with respect to \( \hat{\mathbf{a}}_{i,j} \), i.e.,

\[
\hat{\mathbf{a}}_{i,j}^{MP} = \text{argmax} \hat{\mathcal{L}}_{i,j},
\] (64)

where
\[
\mathcal{L}_{i,j} = \log p(\tilde{y}_{i,j}|\tilde{\alpha}_{i,j}) + \log p(\tilde{\alpha}_{i,j}),
\]

\[
= -\frac{1}{2} \log |\tilde{C}_{i,j}| - \frac{1}{2} \tilde{y}_{i,j}^T \tilde{C}_{i,j}^{-1} \tilde{y}_{i,j} + \log p(\tilde{\alpha}_{i,j}),
\]

with \(\tilde{C}_{i,j} = \Omega_{i,j} + \bar{U}_{i,j} \bar{A}_{i,j}^{-1} \bar{C}_{i,j}^T\).

### 3.4 Numerical results

In this section, we provide some numerical examples to illustrate the performance of the proposed recovery algorithm. First, we demonstrate the effect of \(d\) (size of the processing window) on the recovery performance of the proposed algorithm. For this purpose, we reconstructed two sample images named *Barbara* \((512 \times 512)\) and *Lily* \((1496 \times 1496)\) using the proposed algorithm. We choose \(N = 8\) as the block size, \(\delta = M/N^2 = 0.4\) as the measurement rate, and the sub-sampling operator\(^{23}\) as the measurement matrix \(\Phi\) in the acquisition process of the image. We added Gaussian noise to the measurements, where the measurement noise vector follows the distribution \(e_{i,j} \sim \mathcal{N}(0, \sigma^2 I_M)\). The noise variance \(\sigma^2\) is selected such that the expected SNR with respect to the measurements \(\Phi_{i,j} x_{i,j}\) is \(\rho = 45\) dB.

Table 2 presents the quality of reconstructed image for different values of \(d\) in terms of the average peak signal-to-noise ratio (PSNR) in dB for images *Barbara* and *Lily*. A variable length lapped bi-orthogonal transform (VLLBT) \(^{102}\), with length \(L = 16\) \((2N)\) and 4 long basis functions, is utilized for the sparse representation of the image. We perform 5 independent trials for the reconstruction of each image from compressive measurements and the results are averaged over all trials. As expected, the results show that PSNR performance increases with \(d\) for all images. However, the computational complexity and the memory requirement of the recovery algorithm also grow with \(d\). Hence, \(d\) should be chosen carefully to obtain a satisfactory performance while maintaining the computational complexity and the memory requirements at an acceptable level. Therefore, for the remaining numerical experiments we set \(d = 2\).

Next, we compare the performance of the proposed algorithm with two different recovery methods based on BCS, namely BCS-DCT, and BCS-TV. The BCS-DCT, reconstruct each \(N \times N\) image block independently from its measurements, assuming that the image block is sparse in 2D-DCT. We utilize the standard SBL algorithm in \([52]\) to recover the sparse coefficients in BCS-DCT. In BCS-TV, we utilize the total variation

\(^{23}\)A matrix containing random 1s and 0s.
Table 2. Average PSNR of the reconstructed image using the SBL algorithm for different values of \(d\) \((N = 8, \delta = 0.4, SNR = 45\text{dB})\). (Reprinted by permission [66] ©2016 IEEE)

<table>
<thead>
<tr>
<th>Image</th>
<th>Value of (d)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Barbara</td>
<td>23.94</td>
</tr>
<tr>
<td>Lily</td>
<td>38.35</td>
</tr>
</tbody>
</table>

Table 3. Comparison of the average PSNR (in dB) for Barbara image.(Reprinted by permission [66] ©2016 IEEE)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Measurement rate ((\delta))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.2</td>
</tr>
<tr>
<td>Barbara ((512 \times 512))</td>
<td></td>
</tr>
<tr>
<td>BCS-DCT</td>
<td>21.77</td>
</tr>
<tr>
<td>BCS-TV</td>
<td>22.75</td>
</tr>
<tr>
<td>BCS-SBL</td>
<td>22.73</td>
</tr>
<tr>
<td>Lily ((1496 \times 1496))</td>
<td></td>
</tr>
<tr>
<td>BCS-DCT</td>
<td>33.92</td>
</tr>
<tr>
<td>BCS-TV</td>
<td>33.00</td>
</tr>
<tr>
<td>BCS-SBL</td>
<td>35.92</td>
</tr>
</tbody>
</table>

(TV) minimization\(^{24}\) to reconstruct each \(N \times N\) image block independently from its measurements.

Table 3 compares PSNR for *Barbara* and *Lily* images at several measurement rates (\(\delta\)). We choose \(N = 16\) as the block size, and the rest of the parameters are chosen as the previous experiment. We utilize a VLLBT with \(L = 2N\), containing 8 long basis functions for the sparse representation of the images. The results show that the proposed BCS-SBL algorithm clearly outperforms the BCS-DCT algorithm, which only exploits the sparsity of the individual image blocks in the DCT domain. Further, the proposed BCS-SBL algorithm outperforms the BCS-TV algorithm. From Fig. 13, we can see that the reconstructed images based on BCS-DCT and BCS-TV contains blocking artifacts, while they have been mitigated in the reconstruction based on the proposed BCS-SBL algorithm.

\(^{24}\)We use the MATLAB implementation provided in \texttt{l1-magic} toolbox for the TV minimization [25].
3.5 Summary and discussion

In this chapter, the problem of recovering an image from block compressive sensing (BCS) measurements is considered. A lapped transform has been utilized for the sparse representation of the image to reduce the blocking artifacts. We developed an image recovery algorithm to employ an iterative image reconstruction method, where a small number of adjacent measurement blocks were jointly processed to recover an image.
block. The proposed iterative algorithm was based on sparse Bayesian learning and utilize a Kalman-like implementation. Furthermore, the results showed that the proposed method provided a more visually pleasing image without using any post-processing techniques. We believe that, the performance of the proposed BCS-SBL algorithm can be further improved by utilizing more complex LTs, such as the ones that exploit the images’ directional information. The proposed algorithm is more suitable for applications where BCS measurements are streamed and the full set of measurements is not simultaneously available for the reconstruction process.
4 Modified total variation exploiting transform domain sparsity

4.1 Introduction

In this chapter, we investigate the problem of reconstructing an image using compressive measurements. Conventional CS recovery algorithms exploit the different properties of the signal of interest to solve the reconstruction problem. Originally introduced in [45] for denoising applications, the TV minimization is commonly used to recover 2D signals, such as images, from compressive measurements [8, 46]. It searches for an image with a sparse gradient to solve the recovery problem [10]. Generally, the natural images have sparse representations in known transforms, such as the DCT or DWT. Exploiting this sparsity in the transform domain while searching for an image with a sparse gradient may improve the reconstruction performance of the recovery procedure [25, 103].

To recover the image from compressive measurements, we exploit the sparsity of the image in a given transform while considering that it has a sparse gradient. Thus, we modify the TV minimization problem by introducing an $l_1$–norm penalty term. The main contribution of this chapter is to derive a customized interior-point method to solve the $l_1$–regularized TV minimization problem. The principle behind a customized interior-point method is the efficient solution of the Newton system by exploiting the structure of the Hessian. Such customized methods have been proposed to solve the TV minimization problem in [42] and $l_1$–regularized least squares problem in [24]. However, these methods are not applicable to the $l_1$–regularized TV minimization problem, due to the variations in the structure of the Hessian. Thus, we derive a customized interior-point method for solving the $l_1$–regularized TV minimization problem by proposing an efficient method for solving the Newton system.

4.2 System model and problem formulation

We consider the reconstruction of an $N \times N$ image $X$ from compressive measurements. We acquire the image $X$ utilizing the linear measurement model given by

$$y = \Phi x + e,$$  \hspace{1cm} (67)
where \( x \in \mathbb{R}^{N^2} \) stacks the columns of \( X \) into a vector, i.e., \( x = \text{vec}(X) \); \( y \in \mathbb{R}^M \) contains \( M < N^2 \) linear measurements of \( X \); the measurement matrix is denoted by \( \Phi \in \mathbb{R}^{M \times N^2} \) and the measurement noise vector is denoted by \( e \in \mathbb{R}^M \), where \( e \sim \mathcal{N}(0, \sigma^2 I_M) \) with known variance \( \sigma^2 \).

Suppose that the image \( X \) (or equivalently \( x \)) has a sparse representation in a known basis, i.e., \( x = \Psi w \), where \( \Psi \in \mathbb{R}^{N^2 \times N^2} \) is the representation matrix and \( w \) is the sparse transform domain coefficient vector. We can then easily reconstruct the image \( X \) with the knowledge of \( w \). To estimate \( w \) from the measurements \( y \), we consider solving the following optimization problem:

\[
\begin{align*}
\text{minimize} \quad & \lambda \| w \|_1 + (1 - \lambda) \text{TV}(X) \\
\text{subject to} \quad & \| y - \Phi \Psi w \|_2 \leq \delta,
\end{align*}
\]

(68)

with the optimization variable \( w \). Here the \( p \)-norm of a vector is denoted by \( \| \cdot \|_p \), \( \text{TV}(\cdot) \) denotes the total variation of an image, the regularization parameter is denoted by \( \lambda \), and \( \delta \) denotes a predefined threshold in the order of \( \sqrt{M} \sigma \) [10]. The total variation of an image \( X \) is defined as the sum of the magnitudes of the discrete gradient at every point. Let \( x_{i,j} \) denote the pixel in \( i \)th row and \( j \)th column of the image \( X \). We define the vectors \( d^h_{ij} \in \mathbb{R}^{N^2} \) and \( d^v_{ij} \in \mathbb{R}^{N^2} \), such that

\[
(d^h_{ij})^T X = \begin{cases} x_{i,j+1} - x_{i,j} & j < N \\ 0 & j = N \end{cases},
\]

\[
(d^v_{ij})^T X = \begin{cases} x_{i+1,j} - x_{i,j} & i < N \\ 0 & i = N \end{cases}.
\]

The total variation of the image \( X \) can then be expressed as

\[
\text{TV}(X) = \sum_{i=1}^N \sum_{j=1}^N \| D_{ij}x \|_2,
\]

(69)

where \( D_{ij} = \begin{bmatrix} d^h_{ij} & d^v_{ij} \end{bmatrix}^T \in \mathbb{R}^{2 \times N^2} \).
4.3 A customized interior-point method

By introducing the auxiliary variables $u_n, n = 1, \ldots, N^2$ and $v_{ij}, i, j = 1, \ldots, N$, we can transform problem (68) into an equivalent second-order cone program (SOCP) as

$$\begin{align*}
&\text{minimize} & & \lambda \sum_{n=1}^{N^2} u_n + (1 - \lambda) \sum_{i=1}^{N} \sum_{j=1}^{N} v_{ij} \\
&\text{subject to} & & \|y - \Phi \Psi w\|_2 \leq \delta, \\
& & & \|D_{ij} \Psi w\|_2 \leq v_{ij}, \quad i, j = 1, \ldots, N \\
& & & -u_n \leq w_n \leq u_n, \quad n = 1, \ldots, N^2
\end{align*}$$

(70)

where $w \in \mathbb{R}^{N^2}$, $v = [v_{11}, \ldots, v_{N1}, \ldots, v_{NN}]^T \in \mathbb{R}^{N^2}$ and $u = [u_1, \ldots, u_{N^2}]^T \in \mathbb{R}^{N^2}$ are the variables. In this section, we derive a customized interior-point method for solving this equivalent SOCP (70).

4.3.1 Interior-point method

Let us first introduce the logarithmic barrier function for the inequality constraints in problem (70). It is given by,

$$\phi(w, v, u) = -\log(\delta^2 - \|y - \Phi \Psi w\|_2^2) - \sum_{i=1}^{N} \sum_{j=1}^{N} \log(v_{ij}^2 - \|D_{ij} \Psi w\|_2^2) - \sum_{n=1}^{N^2} \log(u_n + w_n) - \sum_{n=1}^{N^2} \log(u_n - w_n),$$

(71)

which is defined over the dom $\phi = \{(w, v, u) \in \mathbb{R}^{N^2} \times \mathbb{R}^{N^2} \times \mathbb{R}^{N^2} | \|y - \Phi \Psi w\|_2 < \delta, \|D_{ij} \Psi w\|_2 < v_{ij}, i, j = 1, \ldots, N, |w_n| < u_n, n = 1, \ldots, N^2\}$.

For notational convenience, we define $q = [w^T \ v^T \ u^T]^T$. Then, for $t > 0$, let $q^*_t$ be a minimizer of the convex function

$$\psi_t(q) = t \sum_{n=1}^{N^2} \lambda u_n + t \sum_{i=1}^{N} \sum_{j=1}^{N} (1 - \lambda) v_{ij} + \phi(q).$$

The central path associated with problem (70) is defined as the set of points $q^*_t$, $t > 0$, which are called central points. Specifically, $q^*_t$ is no more than $(3N^2 + 1)/t$-suboptimal [47, Sec. 11.2], and hence the central path leads to an optimal solution as $t \to \infty$.

In an interior-point method, we compute a set of central points $q^*_t$ for a sequence of increasing values of $t$ until $(3N^2 + 1)/t \leq \varepsilon$, which guarantees that we have an
ε-suboptimal solution of (70) [47, Sec. 11.3]. Thus, the proposed interior-point method can be summarized as follows:

**Algorithm 3** Interior-point method for $l_1$-regularized TV

given $t^0 > 0$, $\mu > 1$, $\epsilon > 0$ and strictly feasible $\mathbf{q}^0$.
initialize $\hat{\mathbf{q}} = \mathbf{q}^0$, $t = t^0$
repeat
1. Compute $\mathbf{q}^*_{t}$ by minimizing $\psi_t(\mathbf{q})$, starting at $\hat{\mathbf{q}}$.
2. Update. $\hat{\mathbf{q}} := \mathbf{q}^*_{t}$.
3. quit if $(3N^2 + 1)/t < \epsilon$ and output $\hat{\mathbf{q}}$.
4. Increase $t$. $t := \mu t$.

We compute the central point $\mathbf{q}^*_{t}$ at each iteration (except the first one) starting from the previously computed central point, and then increase $t$ by a factor $\mu > 1$. For this purpose, in Step 1, we minimize $\psi_t(\mathbf{q})$ for a fixed value of $t$ utilizing Newton’s method [47, Sec. 9.5]. Here the search direction (or the Newton step) $\Delta \mathbf{q} = [\Delta w^T \Delta v^T \Delta u^T]^T$ is computed as the solution to the Newton system [47, Sec. 9.7]

$$H \Delta \mathbf{q} = -\mathbf{g},$$

(72)

where $H = \nabla^2 \psi_t(\mathbf{w}, \mathbf{v}, \mathbf{u}) \in \mathbb{R}^{3N^2 \times 3N^2}$ is the Hessian and $\mathbf{g} = \nabla \psi_t(\mathbf{w}, \mathbf{v}, \mathbf{u}) \in \mathbb{R}^{3N^2}$ is the gradient of $\psi_t$ at $(\mathbf{w}, \mathbf{v}, \mathbf{u})$. Each subproblem is solved using Newton’s method:

**Algorithm 4** Newton’s method

given $\epsilon_{nt} > 0$ and staring point $\mathbf{q}^0 \in \text{dom } \psi_t(\mathbf{q})$.
initialize $\hat{\mathbf{q}} = \mathbf{q}^0$.
repeat
1. Compute the Newton step $\Delta \mathbf{q}$ by solving Newton system (72) and the decrement as $\eta^2 = -\nabla \psi_t(\hat{\mathbf{q}})^T \Delta \mathbf{q}$.
2. quit if $\eta^2/2 \leq \epsilon_{nt}$ and output $\hat{\mathbf{q}}$.
3. Choose step size $s$ by backtracking line search.
4. Update. $\hat{\mathbf{q}} := \hat{\mathbf{q}} + s \Delta \mathbf{q}$

Given the Newton step $\Delta \mathbf{q}$, the step size in the backtracking line search is taken as $s = \beta^k$, where $k \geq 0$ is the smallest integer that satisfies the Armijo linear search condition.
\[ \psi_t(q + \beta^t \Delta q) \leq \psi_t(q) + \alpha \beta^t \nabla \psi_t(q) \Delta q, \]

where \( \nabla \psi_t(q) \) is the gradient of \( \psi_t \) at \( q \), \( \alpha \in (0, 0.5) \) and \( \beta \in (0, 1) \) [47, Sec. 9.2].

### 4.3.2 Efficient solution of the Newton system

In this sub-section, we present an efficient method for solving the Newton system (72), which is the key to the effectiveness of the customized interior-point method. For this purpose, we first obtain the compact representations of the Hessian and gradient. The Hessian can be expressed as

\[
H = \begin{bmatrix}
F_1 + D_1 & F_2 & D_2 \\
F_2^T & D_3 & 0 \\
D_2 & 0 & D_1
\end{bmatrix} \in \mathbb{R}^{3N^2 \times 3N^2},
\]

where \( F_1, F_2 \) given by,

\[
F_1 = \frac{2A^T A}{\delta^2 - \|r\|^2_2} + \frac{4A^T r r^T A}{(\delta^2 - \|r\|^2_2)^2} + \sum_{i=1}^N \sum_{j=1}^N \left[ \frac{2B^T_i B_{ij}}{(v_j^2 - \|B_{ij}\|^2_2)} + \frac{4B^T_i (B_{ij}w) (B_{ij}w)^T B_{ij}}{(v_j^2 - \|B_{ij}\|^2_2)^2} \right] \in \mathbb{R}^{N^2 \times N^2},
\]

\[
F_2 = \begin{bmatrix}
-4u_1 w_1 \frac{B_{1i}^T B_{1i} w}{(v_1^2 - \|B_{1i}\|^2_2)^2} & \cdots & -4v_{NN} w_{NN} \frac{B_{NN}^T B_{NN} w}{(v_{NN}^2 - \|B_{NN}\|^2_2)^2} \\
(v_1^2 - \|B_{1i}\|^2_2)^2 & \cdots & (v_{NN}^2 - \|B_{NN}\|^2_2)^2
\end{bmatrix} \in \mathbb{R}^{N^2 \times N^2},
\]

and

\[
D_1 = \text{diag} \left( \frac{2(u_1^2 + w_1^2)}{(u_1^2 - w_1^2)^2}, \ldots, \frac{2(u_{N^2}^2 + w_{N^2}^2)}{(u_{N^2}^2 - w_{N^2}^2)^2} \right) \in \mathbb{R}^{N^2 \times N^2},
\]

\[
D_2 = \text{diag} \left( \frac{-4u_1 w_1}{(u_1^2 - w_1^2)^2}, \ldots, \frac{-4u_{N^2} w_{N^2}}{(u_{N^2}^2 - w_{N^2}^2)^2} \right) \in \mathbb{R}^{N^2 \times N^2},
\]

\[
D_3 = \text{diag} \left( \frac{2(v_1^2 + \|B_{1i}\|^2_2)}{(v_1^2 - \|B_{1i}\|^2_2)^2}, \ldots, \frac{2(v_{NN}^2 + \|B_{NN}\|^2_2)}{(v_{NN}^2 - \|B_{NN}\|^2_2)^2} \right) \in \mathbb{R}^{N^2 \times N^2},
\]

with \( A = \Phi \Psi \), \( B_{ij} = D_{ij} \Psi \) and \( r = y - Aw \). Here, we denote the diagonal matrix with diagonal elements \( x_1, \ldots, x_p \) by \( \text{diag}(x_1, \ldots, x_p) \). The gradient can be written as

\[
g = \begin{bmatrix}
g_1 \\
g_2 \\
g_3
\end{bmatrix}, \text{where } g_1 = \nabla_w \psi_t, g_2 = \nabla_y \psi_t, g_3 = \nabla_u \psi_t.
\]
with

\[
\begin{align*}
ge_1 &= -\frac{2A^T r}{\delta^2 - \|r\|^2} + \sum_{i=1}^{N} \sum_{j=1}^{N} 2B_{ij}^T B_{ij} w \begin{bmatrix} 2w_1/(u_1^2 - w_1^2) \\ \vdots \\ 2w_N/(u_N^2 - w_N^2) \end{bmatrix} \\
ge_2 &= (1 - \lambda) I - \begin{bmatrix} 2v_{11}/(v_{11}^2 - \|B_{11} w\|^2_2) \\ \vdots \\ 2v_{NN}/(v_{NN}^2 - \|B_{NN} w\|^2_2) \end{bmatrix} \in \mathbb{R}^{N^2} \\
ge_3 &= \lambda \lambda I - \begin{bmatrix} 2u_1/(u_1^2 - w_1^2) \\ \vdots \\ 2u_N/(u_N^2 - w_N^2) \end{bmatrix} \in \mathbb{R}^{N^2}.
\end{align*}
\]

Here, the symbol \( I \) denotes a vector in \( \mathbb{R}^{N^2} \) in which all its components are one.

To obtain the Newton step, we solve the Newton system (72) by exploiting the ‘arrow’ structure of the Hessian \( H \) (see (73)). In particular, we compute \( \Delta w, \Delta v \) and \( \Delta u \) efficiently by employing the block elimination procedure [47, App. C.4] instead of directly using the inverse of \( H \), which involves an operation of \( O(P^3) \) flops and memory of \( O(P^3) \) with \( P = 3N^2 \). For this purpose, we first form

\[
S = F_1 + D_1 - \begin{bmatrix} F_2 & D_2 \\ 0 & D_1 \end{bmatrix} \begin{bmatrix} D_2^-1 & 0 \\ 0 & D_1 \end{bmatrix}^{-1} \begin{bmatrix} F_2^-1 \\ D_2 \end{bmatrix} = F_1 + D_1 - F_2 D_2^-1 F_2^T - D_2 D_1^-1 D_2 \in \mathbb{R}^{N^2xN^2}.
\]

which is the Schur compliment of \( \begin{bmatrix} D_2 & 0 \\ 0 & D_1 \end{bmatrix} \). Let \( B_i \) be a matrix containing \( (d_i^j)^T \Psi \) as rows, and \( B_i \) contains \( (d_i^j)^T \Psi \) as rows. We can efficiently form \( S \) as

\[
S = D_1 - D_2 D_1^-1 D_2 + f_\delta^{-1} A^T A + f_\delta^{-2} A^T r r^T A + B_i^T (F_v^{-1} + \Sigma_{d_i^j}) B_i + B_i^T (F_v^{-1} + \Sigma_{d_j^i}) B_i + B_i^T (\Sigma_{d_j^i} \Sigma_{d_j^i}) B_i,
\]

where \( f_\delta = (\delta^2 - \|r\|^2_2)/2 \), \( F_v = \text{diag}\left((v_{11}^2 - \|B_{11} w\|^2_2)/2, \ldots, (v_{NN}^2 - \|B_{NN} w\|^2_2)/2\right) \), \( \Sigma_v = \text{diag}\left(-4/(v_{11}^2 - \|B_{11} w\|^2_2), \ldots, -4/(v_{NN}^2 - \|B_{NN} w\|^2_2)\right) \), \( \Sigma_{d_i^j} = \text{diag}(B_i^j w) \) and \( \Sigma_{d_j^i} \) (see Appendix 5.1 for detailed derivation).

Now, using the block elimination procedure, we can compute \( \Delta w \) by solving

\[
S \Delta w = z_1,
\]

\(72\)
where \( w_1 = -g_1 + F_2 \mathbf{D}_3^{-1} g_2 + \mathbf{D}_2 \mathbf{D}_1^{-1} g_3 \). Here the quantity \( F_2 \mathbf{D}_3^{-1} g_2 \) can be computed efficiently as

\[
F_2 \mathbf{D}_3^{-1} g_2 = (B_h \Sigma \partial h + B_v \Sigma \partial v) \mathbf{G} g_2,
\]

(79)

where \( \mathbf{G} = \text{diag}\left( -\frac{2v_{11}}{(v_{11}^2 + \| \mathbf{B}_1 w \|^2)} , \ldots , -\frac{2v_{NN}}{(v_{NN}^2 + \| \mathbf{B}_N w \|^2)} \right) \) (see Appendix 5.2 for detailed derivation). Then, by utilizing the value of \( \Delta w \), we can evaluate \( \Delta v \) and \( \Delta u \) as

\[
\begin{bmatrix}
\Delta v \\
\Delta u
\end{bmatrix} = - \begin{bmatrix}
\mathbf{D}_1 & 0 \\
0 & \mathbf{D}_1
\end{bmatrix}^{-1} \begin{bmatrix}
g_2 \\
g_3
\end{bmatrix} - \begin{bmatrix}
\mathbf{D}_3 & 0 \\
0 & \mathbf{D}_1
\end{bmatrix}^{-1} \begin{bmatrix}
F_2^T \mathbf{D}_2 \\
\mathbf{D}_2
\end{bmatrix} \Delta w.
\]

We can compute \( \Delta v \) and \( \Delta u \) efficiently by

\[
\Delta v = - \mathbf{D}_3^{-1} (g_2 + F_2^T \Delta w),
\]

(80)

\[
\Delta u = - \mathbf{D}_1^{-1} (g_3 + D_2^T \Delta w),
\]

(81)

where

\[
F_2^T \Delta w = H_v (\Sigma \partial h \mathbf{B}_h \Delta w + \Sigma \partial v \mathbf{B}_v \Delta w)
\]

(82)

with \( H_v = \text{diag}\left( -\frac{4v_{11}}{(v_{11}^2 + \| \mathbf{B}_1 w \|^2)^2} , \ldots , -\frac{4v_{NN}}{(v_{NN}^2 + \| \mathbf{B}_N w \|^2)^2} \right) \) (see Appendix 5.3 for detailed derivation).

The most computationally expensive operation for the block elimination procedure is solving (78) using the factorization of the matrix \( \mathbf{S} \). However, the cost of this operation is much less compared to that of inverting the matrix \( \mathbf{H} \) due to the smaller dimensions of \( \mathbf{S} \), i.e., \( N^2 \times N^2 \), compared to those of \( \mathbf{H} \), i.e., \( 3N^2 \times 3N^2 \) (see (76) and (73)).

### 4.4 Numerical Results

In this section, we provide some numerical examples to illustrate the performance of the proposed interior-point method. The algorithm parameters are taken as

\[
\alpha = 0.02, \ \beta = 0.3, \ \mu = 20, \ \varepsilon = 10^{-3}, \ \varepsilon_{\text{int}} = 10^{-3}.
\]

(83)

First, we compare the performance of the proposed algorithm with CVX [104], a standard package for convex programming. We consider the reconstruction of an \( N \times N \) block of the ‘Barbara’ image from compressive measurements. We acquire the image patch using the linear measurement model (67), \( y = \Phi \mathbf{x} + \mathbf{e} \), where \( \mathbf{x} \in \mathbb{R}^{N^2} \) is the vectorized version of the image block. The measurement matrix \( \Phi \in \mathbb{R}^{M \times N^2} \) is generated by selecting random \( M \) rows of the ‘noiselet transform’ matrix [20]. We add Gaussian noise to the measurements, where the measurement noise vector \( \mathbf{e} \) follows the
distribution \( e \sim \mathcal{N}(0, \sigma^2 I_M) \). The noise variance \( \sigma^2 \) is selected such that the expected SNR with respect to the measurements \( \Phi x \) is \( \rho \) dB, i.e., \( \rho = 10 \log \left( \frac{\| \Phi x \|_2^2}{M \sigma^2} \right) \).

We use DCT as the sparsifying transform, i.e., \( x = \Psi w \), where \( \Psi \in \mathbb{R}^{N^2 \times N^2} \) is the DCT matrix and \( w \) is the sparse transform domain coefficient vector. We solve the convex optimization problem (85), with \( \lambda = 0.5 \), using the proposed interior-point method \((\text{Algorithm 3})\) and CVX with the solver ‘MOSEK’\(^{25}\) to obtain an estimation of \( w \). The image block is then reconstructed utilizing the estimated \( w \). The results are summarized in Table 4. We chose a measurement rate \( M/N^2 = 0.4 \), and the measurement noise is generated with SNR = 45 dB. We performed 20 independent trials to reconstruct the image block from compressive measurements, and the results are averaged over all trials. We use the Peak Signal-to-Noise Ratio (PSNR) to evaluate the objective visual quality of the reconstructed image. The results show that both \text{Algorithm 3} and the CVX based method can recover the image block from compressive measurements, and they have the same PSNR performances. However, the MATLAB execution time of the proposed interior-point method is at least 20 times lower, due to the utilization of the special structure of the Hessian in solving the Newton system.

Next, we compare the recovery performance of the proposed algorithm with the TV minimization and \( l_1 \)-norm minimization. We utilize the interior-point method based MATLAB implementations in the \texttt{l1-MAGIC} \([42]\) package for both TV minimization and \( l_1 \)-norm minimization\(^{26}\). We consider the reconstruction of an image using block compressive sensing (BCS) \([97, 99, 106, 107, 108]\). In BCS, the image is divided into \( N \times N \) small blocks, and each block is acquired using the linear measurement model (67).

\(^{25}\) ‘MOSEK’ has the smallest runtime among the possible solvers ‘SDPT3’, ‘SeDuMi’ and ‘MOSEK’ (see \([105]\) for some numerical examples).

\(^{26}\) Note that, we can obtain exactly the same results by setting \( \lambda = 0 \) and \( \lambda = 1 \) in problem (68) (see \([105]\) for some numerical examples).
with the same measurement operator $\Phi \in \mathbb{R}^{M \times N^2}$. We generated the measurement matrix $\Phi$ by selecting random $M$ rows from the ‘noiselet transform’ matrix.

We choose $N = 16$, and use DCT as the sparsifying transform. We recover each image block utilizing Algorithm 5, TV minimization, and $l_1$–norm minimization. For Algorithm 5, we estimated the sparse transform domain coefficient vectors by solving problem (85) with the parameter $\lambda = 0.9$ for the ‘Barbara’ image and $\lambda = 0.1$ for the other cases. The parameters $\delta, \varepsilon$ and $\varepsilon_{nt}$ in all the interior-point solvers were set to the values given in (83). All the other parameters of TV minimization and $l_1$–norm minimization algorithms were chosen as stated in [42]. We perform simulations to recover test images ‘Mandrill’, ‘Barbara’, ‘Lenna’ and ‘Peppers’ with a size of $512 \times 512$ using compressive measurements. Fig. 14 illustrates a portion of the reconstructed ‘Barbara’ image at measurement rate $M/N^2 = 0.3$ using different recovery algorithms.
The recovered image via the proposed method provides a better subjective visual quality (see the stripe patterns on the dress) compared to the reconstructions via the TV minimization and $l_1$–norm minimization methods. We evaluate the objective visual quality of the reconstruction using the PSNR value. The results are summarized in Table 5. The results show that the proposed method achieves the best PSNR value in all the scenarios, out performing both TV minimization and $l_1$–norm minimization-based reconstructions. This is because the TV minimization recovers the signal of interest by searching only for a signal with a sparse gradient. On the other hand, $l_1$–norm minimization recovers the signal of interest by only exploiting the sparsity of the signal in the given transform. However, the proposed method can exploit both these properties of the signal to improve the quality of the reconstruction.

<table>
<thead>
<tr>
<th>Measurement rate ($M/N^2$)</th>
<th>$l_1$-DCT</th>
<th>TV</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandrill (512 × 512)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>21.31</td>
<td>21.12</td>
<td>21.30</td>
</tr>
<tr>
<td>0.3</td>
<td>20.60</td>
<td>22.34</td>
<td>22.50</td>
</tr>
<tr>
<td>0.4</td>
<td>21.79</td>
<td>23.44</td>
<td>23.58</td>
</tr>
<tr>
<td>0.5</td>
<td>23.16</td>
<td>24.65</td>
<td>24.78</td>
</tr>
<tr>
<td>Barbara (512 × 512)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>22.76</td>
<td>23.24</td>
<td>23.97</td>
</tr>
<tr>
<td>0.3</td>
<td>25.22</td>
<td>24.24</td>
<td>25.94</td>
</tr>
<tr>
<td>0.4</td>
<td>27.23</td>
<td>25.07</td>
<td>27.61</td>
</tr>
<tr>
<td>0.5</td>
<td>29.47</td>
<td>26.17</td>
<td>29.68</td>
</tr>
<tr>
<td>Lenna (512 × 512)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>25.68</td>
<td>29.67</td>
<td>29.85</td>
</tr>
<tr>
<td>0.3</td>
<td>28.43</td>
<td>31.88</td>
<td>31.97</td>
</tr>
<tr>
<td>0.4</td>
<td>30.47</td>
<td>33.41</td>
<td>33.58</td>
</tr>
<tr>
<td>0.5</td>
<td>32.67</td>
<td>35.23</td>
<td>35.40</td>
</tr>
<tr>
<td>Peppers (512 × 512)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>25.19</td>
<td>30.36</td>
<td>30.53</td>
</tr>
<tr>
<td>0.3</td>
<td>28.23</td>
<td>32.69</td>
<td>32.83</td>
</tr>
<tr>
<td>0.4</td>
<td>30.15</td>
<td>34.11</td>
<td>34.23</td>
</tr>
<tr>
<td>0.5</td>
<td>32.27</td>
<td>35.60</td>
<td>35.72</td>
</tr>
</tbody>
</table>
4.5 Summary and discussion

This chapter addressed the problem of reconstructing an image using compressive measurements. To do this, we considered solving a modified total variation (TV) minimization problem by introducing an $l_1$–norm penalty term for exploiting the sparsity of the image in a given transform. We recast the $l_1$–regularized TV minimization problem in the form of a convex second-order cone program, and derived a specialized interior-point method to obtain the optimal solution. The results showed that an improved performance can be seen in the reconstructed image from the proposed algorithm compared to TV reconstruction and the image reconstructed based on the sparsifying basis. The developed MATLAB based solver can handle an averaged size (512 x 512) in a typical computer which is in par with the state-of-the-art solver 11-MAGIC [42] package.
5 Signal reconstruction using multiple sparsifying bases

5.1 Introduction

For many natural signals, such as images and video sequences, the specific basis \( \Psi \) in which the signal of interest is the sparsest is unknown. Generally, multiple bases may exist that lead to a compressible representation of such a signal, i.e., the energy of the signal is concentrated in a relatively small set of transform domain coefficients \([11, 12]\). Hereafter, we refer to them as sparsifying bases. These sparsifying bases are generally utilized in conventional CS recovery algorithms. In solving the CS recovery problem (4), the basis \( \Psi \) is chosen from a set of possible sparsifying bases, e.g., DWT, DCT. Naturally, the quality of the signal’s reconstruction varies, depending on the choice of basis \( \Psi \). Hence, to obtain a fairly accurate estimate, the choice of basis \( \Psi \) is of the utmost importance in traditional CS recovery algorithms.

Thus far, most of the research focus has been on finding a specific basis that provides the sparsest representation of the signal, e.g., via dictionary learning algorithms \([70, 71]\), which are generally computationally expensive. However, as pointed out in \([109]\) in a signal denoising application, significant performance improvement can be obtained by combining several medium-sparse representations of the signal, as compared with the case in which only the sparsest representation is utilized. In this chapter, we show that a similar performance improvement can also be achieved in the context of a signal reconstruction from compressive samples by utilizing multiple sparsifying bases. Further, we derive a customized interior-point method for recovering 2-D signals (images) from compressive measurements which can utilize multiple sparsifying bases, as well as the fact that the images usually have a sparse gradient. It is important to note that the signal reconstruction method proposed in this chapter differs fundamentally from the signal denoising method introduced in \([109]\). Although both methods rely on combining multiple sparse signal representations, the method introduced in \([109]\) uses a randomization technique to generate multiple sparse representations in a single overcomplete dictionary, while our method uses several distinct sparsifying bases to obtain multiple sparse representations of the signal of interest.
5.2 Problem formulation

We consider the recovery of a 2-dimensional signal from compressive measurements, where the sparsifying basis of the signal is unknown. Specifically, we obtain a set of compressive measurements \( y \in \mathbb{R}^M \) from the 2-D signal \( X \in \mathbb{R}^{N \times N} \).

5.3 Signal reconstruction via multiple bases

Let us suppose that the signal of interest \( x \) is reasonably compressible in a set of \( K \) a-priori known bases \( \mathcal{S} = \{ \Psi_1, \Psi_2, \ldots, \Psi_K \} \). We then perform the signal reconstruction, utilizing the idea illustrated in Figure 15. We first estimate the set of sparse coefficient vectors \( \hat{w} = \{ \hat{w}_1, \hat{w}_2, \ldots, \hat{w}_K \} \) from the measurements \( y \) by assuming \( x \) is sparse in each basis \( \Psi_k \in \mathcal{S} \). The signal \( x \) is then reconstructed as \( \hat{x} = f(\hat{w}, \mathcal{S}) \), where \( f(\cdot) \) is the fusion function. We can summarize the key idea as follows:

**Algorithm 5 Signal recovery via multiple sparsifying bases**

1. **Inputs:** \( y, \Phi, \mathcal{S} = \{ \Psi_1, \Psi_2, \ldots, \Psi_K \} \).
2. **Estimate:** \( \hat{w} = \{ \hat{w}_1, \hat{w}_2, \ldots, \hat{w}_K \} \).
3. **Output:** \( \hat{x} = f(\hat{w}, \mathcal{S}) \).

In step 2, \( \hat{w}_k \in \mathcal{W} \) can be estimated from measurements \( y \) by solving the \( l_1 \)-norm minimization problem for each basis \( \Psi_k \in \mathcal{S} \). Alternatively, we propose a novel method in the sequel which can utilize multiple sparsifying bases, as well as the fact that the images usually have a sparse gradient in order to estimate \( \hat{w}_k \in \mathcal{W} \). In step 3, for
example, we can utilize simple averaging as the fusion function \( f(W, S) \) to reconstruct the signal, i.e., \( \tilde{x} = 1/K \sum_{k=1}^{K} \Psi_k \hat{w}_k \). However, utilization of a more sophisticated fusion function, e.g., by taking into account the sparsity of each \( \hat{w}_k \in W \), may provide a better recovery performance. Finding the optimal fusion function would be an interesting research area, but it is beyond the scope of this thesis. However, the numerical results show that even a simple averaging of several estimates of the signal significantly improves the reconstruction performance compared to any single estimate.

In general, we need to estimate \( \hat{w}_k \in W \) in Step 2, such that it provides the best sparse representation of the signal \( x \) in \( \Psi_k \in \mathcal{S} \). To jointly estimate \( \hat{w}_k \in W \) in the case where the signal of interest is a natural image, we consider solving the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \sum_{k=1}^{K} \lambda_k \|w_k\|_1 + \text{TV}(\sum_{k=1}^{K} X_k) \\
\text{subject to} & \quad \|y - \Phi \Psi_k w_k\|_2 \leq \delta_k, \quad k = 1, \ldots, K
\end{align*}
\]

with the optimization variable \( w_k \) for \( k = 1, \ldots, K \). Specifically, we derive a customized interior-point method for solving the problem (84). Here, the \( p \)-norm of a vector is denoted by \( \| \cdot \|_p \), \( \text{vec}(X_k) = \Psi_k w_k \), and \( \text{TV}(\cdot) \) denotes the total variation of an image. Problem (84) not only exploits the sparsity of the image in given transforms but also considers that the image has a sparse gradient. The parameter \( \delta_k \) allows us to account for the case of the signal’s noisy acquisition. Further, by utilizing the ratio between the regularization parameters \( \lambda_k \) for \( k = 1, \ldots, K \), we can encode the belief on the sparsity of the signal in bases \( \Psi_k \in \mathcal{S} \).

### 5.4 Customized interior-point method for CS recovery via multiple bases

Customized methods have been proposed to solve the \( l_1 \)-norm minimization problem in [42, 43] and the TV minimization problem in [42]. However, these methods are not applicable to problem (84) due to the variations in the structure of the Hessian. Thus, in this section, we derive a customized interior-point method for solving problem (84) that efficiently computes the search direction of the Newton method by exploiting the specific structure of the Hessian. It is worth mentioning that the implementation of the proposed interior-point method in [110] is inspired by [42], and it can reconstruct medium-sized images.

81
\[ (d_{ij})^T = \begin{cases} 
  x_{i,j+1} - x_{i,j} & j < N \\
  0 & j = N 
\end{cases} \quad (d_{ij}^T)^T = \begin{cases} 
  x_{i+1,j} - x_{i,j} & i < N \\
  0 & i = N. 
\end{cases} \]

The total variation of \( X \) can then be expressed as \( TV(X) = \sum_{i=1}^{N} \sum_{j=1}^{N} ||D_{ij}x||_2 \), where \( D_{ij} = \begin{bmatrix} d_{ij} & d^T_{ij} \end{bmatrix} \in \mathbb{R}^{2 \times N} \) and \( x = \text{vec}(X) \). Now, by introducing the auxiliary variables \( u_{k,n} \) for \( k = 1, \ldots, K, \ n = 1, \ldots, N^2 \) and \( v_{i,j} \) for \( i, j = 1, \ldots, N \), we equivalently reformulate the problem (84) as

\[
\begin{align*}
\text{minimize} & \quad \sum_{k=1}^{K} \sum_{n=1}^{N^2} \lambda_k u_{k,n} + \sum_{i=1}^{N} \sum_{j=1}^{N} v_{i,j} \\
\text{subject to} & \quad ||y - A_k w_k||_2 \leq \delta_k, \quad k = 1, \ldots, K \\
& \quad ||D_{ij}x||_2 \leq v_{i,j}, \quad i, j = 1, \ldots, N \\
& \quad -u_{k,n} \leq w_{k,n} \leq u_{k,n}, \quad k = 1, \ldots, K, \ n = 1, \ldots, N^2
\end{align*}
\] (85)

where \( A_k = \Phi \Psi_k \) with the optimization variables \( w_k \in \mathbb{R}^{N^2}, \ u_k = [u_{k,1}, \ldots, u_{k,N^2}]^T \in \mathbb{R}^{N^2}, \) and \( v = [v_{1,1}, \ldots, v_{N,1}, \ldots, v_{N,N}]^T \in \mathbb{R}^{N^2} \). Next, we introduce the logarithmic barrier function for the inequality constraints in problem (85) as

\[
\phi(w,v,u) = -\sum_{k=1}^{K} \sum_{n=1}^{N^2} \lambda_k u_{k,n} - \sum_{i=1}^{N} \sum_{j=1}^{N} v_{i,j} \log(||B_{ij}w||_2) - \sum_{k=1}^{K} \sum_{n=1}^{N^2} \log(u_{k,n} + w_{k,n}) + \log(u_{k,n} - w_{k,n})
\] (86)

where \( w = [w_1^T \ldots w_k^T]^T, \ u = [u_1^T \ldots u_k^T]^T, \ r_k = y - A_k w_k, \) and \( B_{ij} = D_{ij} \Psi \) with \( \Psi = [\Psi_1, \ldots, \Psi_K] \). The log barrier function (86) is defined over \( \text{dom} \phi = \{ (w,v,u) \in \mathbb{R}^{KN^2} \times \mathbb{R}^{N^2} \times \mathbb{R}^{KN^2} | ||r_k||_2 \leq \delta_k, ||B_{ij}w||_2 \leq v_{i,j}, |w_{k,n}| < u_{k,n} \} \) for \( k = 1, \ldots, K, \ n = 1, \ldots, N^2, \) and \( i, j = 1, \ldots, N \).

For the notational convenience, we define \( q = [w^T v^T u^T]^T \). Then, for \( t > 0 \), let \( q^* \) minimize the convex function

\[
\psi_t(q) = t \left( \sum_{k=1}^{K} \sum_{n=1}^{N^2} \lambda_k u_{k,n} + \sum_{i=1}^{N} \sum_{j=1}^{N} v_{i,j} \right) + \phi(q).
\] (87)

The central path associated with problem (85) is defined as the set of central points \( q^*_t \) for \( t > 0 \). Specifically, the central path leads to an optimal solution of problem (85) as \( t \to \infty \), since \( q^*_t \) is no more than \( ((2K+1)N^2 + K)/t \)-suboptimal [47, Sec. 11.2].

5.4.1 Interior-point method

Let \( x_{i,j} \) denote the pixel in the \( i \)th row and \( j \)th column of an image \( X \). We define the vectors \( d_{ij}^k \in \mathbb{R}^{N^2} \) and \( d_{ij}^T \in \mathbb{R}^{N^2} \) such that

The log barrier function (86) is defined over \( \text{dom} \phi = \{ (w,v,u) \in \mathbb{R}^{KN^2} \times \mathbb{R}^{N^2} \times \mathbb{R}^{KN^2} | ||r_k||_2 \leq \delta_k, ||B_{ij}w||_2 \leq v_{i,j}, |w_{k,n}| < u_{k,n} \} \) for \( k = 1, \ldots, K, \ n = 1, \ldots, N^2, \) and \( i, j = 1, \ldots, N \).

For the notational convenience, we define \( q = [w^T v^T u^T]^T \). Then, for \( t > 0 \), let \( q^* \) minimize the convex function

\[
\psi_t(q) = t \left( \sum_{k=1}^{K} \sum_{n=1}^{N^2} \lambda_k u_{k,n} + \sum_{i=1}^{N} \sum_{j=1}^{N} v_{i,j} \right) + \phi(q).
\] (87)

The central path associated with problem (85) is defined as the set of central points \( q^*_t \) for \( t > 0 \). Specifically, the central path leads to an optimal solution of problem (85) as \( t \to \infty \), since \( q^*_t \) is no more than \( ((2K+1)N^2 + K)/t \)-suboptimal [47, Sec. 11.2].

82
In an interior-point method, we compute a set of central points $q^*_t$ for a sequence of increasing values of $t$ until $((2K + 1)N^2 + K)/t \leq \varepsilon$ to obtain an $\varepsilon$–suboptimal solution for problem (85) [47, Sec. 11.3]. We can thus summarize the interior-point method as follows:

**Algorithm 6** Interior-point method for $l_1$–regularized TV with multiple bases

**given** $t^0 > 0$, $\mu > 1$, $\varepsilon > 0$ and strictly feasible $q^0$.
**initialize** $\hat{q} = q^0$, $t = t^0$
**repeat**
1. Compute $q^*_t$ by minimizing $\psi_t(q)$, starting at $\hat{q}$.
2. Update. $\hat{q} := q^*_t$.
3. **Quit** if $((2K + 1)N^2 + K)/t < \varepsilon$ and output $\hat{q}$.
4. Increase $t$. $t := \mu t$.

At each iteration (except the first), we compute the central point $q^*_t$ starting from the previously computed central point. For this purpose, in Step 1, we utilize Newton’s method [47, Sec. 9.5] to minimize $\psi_t(q)$, for a fix value of $t$.

### 5.4.2 Efficient solution of the Newton system

In Newton’s method, the search direction $\Delta q = [\Delta w^T \Delta v^T \Delta u^T]^T$ is computed as the solution to the Newton system [47, Sec. 9.7]

$$H\Delta q = -g,$$

(88)

where $H = \nabla^2 \psi_t(w, v, u) \in \mathbb{R}^{(2K+1)N^2 \times (2K+1)N^2}$, and $g = \nabla \psi_t(w, v, u) \in \mathbb{R}^{(2K+1)N^2}$ denotes the Hessian and the gradient of $\psi_t$ at $(w, v, u)$ respectively. The principle behind a customized interior-point method is the efficient solution of the Newton system by exploiting the structure of the Hessian. In this sub-section, we present an efficient method for solving the Newton system (88), which is the key to the effectiveness of the customized interior-point method. Newton’s method, utilized to compute $q^*_t$ in step 1 of Algorithm 6, can be summarized as follows:
Algorithm 7 Newton’s method

given \( \epsilon_{\text{nt}} > 0 \) and staring point \( \mathbf{q}^0 \in \text{dom} \; \psi_i(\mathbf{q}) \).
initialize \( \hat{\mathbf{q}} = \mathbf{q}^0 \).
repeat
1. Compute the search direction \( \Delta \mathbf{q} \) by solving Newton system (88) and the decrement as \( \eta^2 = -\nabla \psi_i(\hat{\mathbf{q}})^T \Delta \mathbf{q} \).
2. quit if \( \eta^2 / 2 \leq \epsilon_{\text{nt}} \) and output \( \hat{\mathbf{q}} \).
3. Choose step size \( s \) by backtracking line search.
4. Update. \( \hat{\mathbf{q}} := \hat{\mathbf{q}} + s \Delta \mathbf{q} \)

Given the search direction \( \Delta \mathbf{q} \), the step size in the backtracking line search is taken as \( s = \beta^p \), where \( p \geq 0 \) is the smallest integer that satisfies
\[
\psi_i(\mathbf{q} + \beta^p \Delta \mathbf{q}) \leq \psi_i(\mathbf{q}) + \alpha \beta^p \nabla \psi_i(\mathbf{q})^T \Delta \mathbf{q},
\]
where \( \nabla \psi_i(\mathbf{q}) \) is the gradient of \( \psi_i \) at \( \mathbf{q}, \alpha \in (0, 0.5) \) and \( \beta \in (0, 1) \) [47, Sec. 9.2].

To derive an efficient solution to the Newton system (88), first we obtain compact representations of the Hessian and gradient. The Hessian can be expressed as
\[
H = \begin{bmatrix} F_1 + F_2 + D_1 & F_3 & D_2 \\ F_3^T & D_3 & 0 \\ D_2 & 0 & D_1 \end{bmatrix} \in \mathbb{R}^{(2K+1)N^2 \times (2K+1)N^2},
\]
where \( F_1 = \text{diag}(F_{1,1}, \ldots, F_{K,1}) \), \( D_1 = \text{diag}(D_{1,1}, \ldots, D_{K,1}) \), and \( D_2 = \text{diag}(D_{1,2}, \ldots, D_{K,2}) \) with
\[
F_{2,i} = \begin{bmatrix} 2A_i^T A_i + 4A_i^T r_i r_i^T A_i \\
\sum_{j=1}^{N} 2 B_{i,j} B_{i,j}^T + \frac{4 B_{i,j}^T (B_{i,j} w) (B_{i,j} w)^T B_{i,j}}{(v_{i,j} - \|B_{i,j} w\|_2^2)^2} \end{bmatrix} \in \mathbb{R}^{N^2 \times N^2},
\]
\[
F_{3,i} = \begin{bmatrix} -4V_{i,1} B_{i,1}^T B_{i,1} w \\
\sum_{j=1}^{N} -4V_{i,j} B_{i,j}^T B_{i,j} w \end{bmatrix} \in \mathbb{R}^{K N^2 \times N^2},
\]
\[
D_{i,1} = \begin{bmatrix} 2(u_{i,1}^2 + w_{i,1}^2) \\
\sum_{j=1}^{N} 2(u_{i,j}^2 + w_{i,j}^2) \end{bmatrix} \in \mathbb{R}^{N^2 \times N^2},
\]
\[
D_{i,2} = \begin{bmatrix} -4 u_{i,1} w_{i,1} \\
\sum_{j=1}^{N} -4 u_{i,j} w_{i,j} \end{bmatrix} \in \mathbb{R}^{N^2 \times N^2},
\]
\[
D_1 = \begin{bmatrix} 2(v_{1,1}^2 + \|B_{1,1} w\|_2^2) \\
\sum_{j=1}^{N} 2(v_{1,j}^2 + \|B_{1,j} w\|_2^2) \end{bmatrix} \in \mathbb{R}^{N^2 \times N^2}.
\]
Here, we denote the diagonal matrix with diagonal blocks $x_1, \ldots, x_p$ by $\text{diag}(x_1, \ldots, x_p)$.

The gradient can be written as $g = [g_1^T \; g_2^T \; g_3^T]^T \in \mathbb{R}^{(2K+1)N^2}$, where

$$g_1 = \begin{bmatrix}
-2A_1^T r_1 \\
\frac{\partial^2}{\partial t^2} - ||r||_2^2
\end{bmatrix} + \begin{bmatrix}
2w_{1,1}/(u_{1,1}^2 - w_{1,1}^2) \\
\vdots \\
2w_{1,N^2}/(u_{1,N^2}^2 - w_{1,N^2}^2)
\end{bmatrix} \quad + \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{2B_{ij}^T B_{ij} w}{(r_{ij}^2 - ||B_{ij} w||_2^2)} \in \mathbb{R}^{KN^2}
$$

$$g_2 = r1 - \begin{bmatrix}
-2A_K^T r_K \\
\frac{\partial^2}{\partial t^2} - ||r||_2^2
\end{bmatrix} + \begin{bmatrix}
2w_{K,1}/(u_{K,1}^2 - w_{K,1}^2) \\
\vdots \\
2w_{K,N^2}/(u_{K,N^2}^2 - w_{K,N^2}^2)
\end{bmatrix} \in \mathbb{R}^{KN^2}
$$

$$g_3 = \begin{bmatrix}
\lambda_1 r1 - \begin{bmatrix}
2u_{1,1}/(u_{1,1}^2 - w_{1,1}^2) \\
\vdots \\
2u_{1,N^2}/(u_{1,N^2}^2 - w_{1,N^2}^2)
\end{bmatrix} \\
\lambda_K r1 - \begin{bmatrix}
2u_{K,1}/(u_{K,1}^2 - w_{K,1}^2) \\
\vdots \\
2u_{K,N^2}/(u_{K,N^2}^2 - w_{K,N^2}^2)
\end{bmatrix}
\end{bmatrix} \in \mathbb{R}^{KN^2}.
$$

Next, to obtain the search direction $\Delta q$, we solve the Newton system (88) by exploiting the structure of the Hessian $H$ in (89). In particular, we efficiently compute $\Delta w, \Delta v$, and $\Delta u$ by employing the block elimination procedure [47, App. C.4] instead of directly using the inverse of $H$, which involves an operation of $O(P^3)$ flops and memory of $O(P^2)$ with $P = (2K+1)N^2$. 

85
For this purpose, we first express \( \Delta w \) by using the block elimination procedure as

\[
\Delta w = S^{-1} z_1,
\]

where

\[
S = F_1 + F_2 + D_1 - \begin{bmatrix} F_3 & D_2 \\ D_3 & 0 \end{bmatrix}^{-1} \begin{bmatrix} F_3 \\ D_2 \end{bmatrix} = F_1 + D_1 - D_2 D_1^{-1} D_2 + F_2 - F_1 D_1^{-1} F_3^T \in \mathbb{R}^{KN \times KN}
\]

\[
z_1 = -g_1 + \begin{bmatrix} F_3 & D_2 \\ D_3 & 0 \end{bmatrix}^{-1} \begin{bmatrix} g_2 \\ g_3 \end{bmatrix} = -g_1 + F_3 D_1^{-1} g_2 + D_2 D_1^{-1} g_3 \in \mathbb{R}^{KN}.
\]

Let \( S_1 = F_1 + D_1 - D_2 D_1^{-1} D_2 \). \( B_\Delta \) be a matrix containing \( (d_i^j)\) as rows and \( B_\Sigma \) contains \( (d_i^j)\) as rows. We can then express \( S \) as \( S = S_1 + B^T C B \), where

\[
B = \begin{bmatrix} B_\Delta \\ B_\Sigma \end{bmatrix} \in \mathbb{R}^{2N^2 \times KN^2}, \quad C = \begin{bmatrix} F_v^{-1} + \Sigma_v \Sigma_{\partial h} \Sigma_{\partial v} & \Sigma_v \Sigma_{\partial h} \Sigma_{\partial v} \\ \Sigma_v \Sigma_{\partial h} \Sigma_{\partial v} & F_v^{-1} + \Sigma_v \Sigma_{\partial h} \Sigma_{\partial v} \end{bmatrix} \in \mathbb{R}^{2N^2 \times 2N^2}
\]

with

\[
F_v = \text{diag} \left( \frac{(v_1^2 - \| B_{11} w \|^2_2)}{2}, \ldots, \frac{(v_{KN}^2 - \| B_{KN} w \|^2_2)}{2} \right), \quad \Sigma_v = \text{diag} \left( -4/(v_{11}^4 - \| B_{11} w \|^4_2), \ldots, -4/(v_{KN}^4 - \| B_{KN} w \|^4_2) \right), \quad \Sigma_{\partial h} = \text{diag}(B_{\partial h}) \quad \text{and} \quad \Sigma_{\partial v} = \text{diag}(B_{\partial v})
\]

(see Appendix 6 for detailed derivation). Now, utilizing the matrix inversion lemma we compute \( \Delta w \) as

\[
\Delta w = S_1^{-1} z_1 - S_1^{-1} B^T (C^{-1} + BS_1^{-1} B^T)^{-1} BS_1^{-1} z_1.
\]

Then, by utilizing the value of \( \Delta w \), we can compute \( \Delta v \) and \( \Delta u \) as

\[
\Delta v = -D_3^{-1} (g_2 + F_3^T \Delta w), \quad \Delta u = -D_1^{-1} (g_3 + D_2^T \Delta w).
\]

Since \( D_1, D_2 \) are diagonal, and \( F_1 \) is block diagonal, the matrix \( S_1 \) is a block diagonal matrix containing \( N^2 \times N^2 \) blocks. Thus, the most computationally expensive operation for the block elimination procedure is the inversion of \( C^{-1} + BS_1^{-1} B^T \in \mathbb{R}^{2N^2 \times 2N^2} \) in (93). Note that in the case of \( K = 1 \), since \( S \in \mathbb{R}^{N^2 \times N^2} \), it is efficient to inverse \( S \) in (91) instead of utilizing (93). The cost of the block elimination procedure is therefore \( O(P^3) \) flops with \( P = 2N^2 \) (\( P = N^2 \) for \( K = 1 \)), which is much less compared to inverting the matrix \( \mathbf{H} \in \mathbb{R}^{[2K+1]N^2 \times (2K+1)N^2} \).

### 5.5 Numerical results

In this section, we evaluate the performance of utilizing multiple bases for compressive sensing reconstruction. Figure 16 presents an example of the improved performance,
in terms of the visual quality, of utilizing multiple sparsifying bases in recovering an image from compressive measurements. The figure illustrates a portion of the ‘Peppers’ image $X$ reconstructed using different bases, as well as using Algorithm 5 from the compressive measurements $y = \Phi x$, where $x$ is the vectorized version of the 2-D image $X$. The measurement matrix $\Phi \in \mathbb{R}^{M \times N^2}$ is generated by selecting random $M$ rows of the ‘noiselet transform’ matrix [20] at sensing rate $M/N^2 = 0.15$. For each basis, we solve the $l_1$-norm minimization problem (68) using the NESTA toolbox [43] with the parameters $\mu_f = 10^{-7}$, NESTAmaxiter = 500, and $\delta = 0$. Further, in Step 3 of Algorithm 5, we utilized the simple fusion function $\tilde{x} = \frac{1}{K} \sum_{k=1}^{K} \Psi_k \hat{w}_k$. The figure shows that the reconstructions using only a single basis contain different artifacts which are specific to the given basis, e.g., different ringing artifacts for DWT bases and different blocking artifacts for the DCT bases. The fusion in Algorithm 5 eliminates these artifacts, and as a result it provides a better subjective visual quality compared to the reconstructions via a single basis.

Table 6 demonstrates the objective quality comparison via Peak Signal-to-Noise Ratio (PSNR) for some example images under different sensing rates. The results show that the basis which achieves the highest PSNR among the considered set of bases depends on the image as well as the sensing rate. For example, the basis DWT ‘sym6’ achieves the highest PSNR for the ‘Peppers’ image at sensing rate 0.4, while for the same sensing rate the basis DCT $32 \times 32$ achieves the highest PSNR for the ‘Barbara’ image. Similarly, for the same ‘Peppers’ image at sensing rate 0.25, the basis DCT $16 \times 16$ achieves the best PSNR, but at sensing rate 0.4, the basis DWT ‘sym6’ achieves...
the best PSNR. However, we can see that the reconstruction via Algorithm 5 utilizing multiple bases provides the best PSNR for all images and sensing rates.

Table 7 presents the reconstruction performance of fusion via the solution of problem (84) \((l_1 + TV\text{-fusion})\) in terms of PSNR. We utilize the proposed interior-point method to solve problem (84) with parameters \(\lambda_k = 1, \alpha = 0.02, \beta = 0.3, \mu = 20, \varepsilon = 10^{-3}, \varepsilon_{nt} = 10^{-3}\), and \(\hat{x} = \frac{1}{K} \sum_{k=1}^{K} \Psi_k \hat{w}_k\) as the fusion function. The results show that the reconstruction via the solution of problem (84) even out-performs the reconstruction based on the fusion using the estimates from \(l_1\text{-norm minimization (}l_1\text{-fusion)}\). This is because problem (84) also takes an image’s sparse gradient into account.

5.6 Summary and discussion

In this chapter, the reconstruction of a signal from compressive measurements when the basis in which the signal of interest is the sparsest is unknown was considered. We showed that a fusion of multiple estimates of the signal using different sparsifying bases led to a better signal recovery compared to reconstructions via any single basis. Further, we derived a customized interior-point method to jointly obtain such multiple estimates of an image from compressive measurements which utilized multiple sparsifying bases, as well as the fact that the images usually have a sparse gradient. The developed MATLAB based solver could handle an averaged size \((512 \times 512)\) in a typical computer, which is on par with the state-of-the-art solver \texttt{l1-MAGIC} [42] package.

<table>
<thead>
<tr>
<th>Sensing rate</th>
<th>DWT 'sym6'</th>
<th>DWT 'bior4.4'</th>
<th>DWT 'sym8'</th>
<th>DCT 16×16</th>
<th>DCT 32×32</th>
<th>DCT 64×64</th>
<th>DCT 128×128</th>
<th>DCT 256×256</th>
<th>Fused 256×256</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barbara (512×512)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.15</td>
<td>20.07</td>
<td>19.77</td>
<td>20.12</td>
<td>22.51</td>
<td>23.15</td>
<td>22.99</td>
<td>22.54</td>
<td>21.86</td>
<td>23.46</td>
</tr>
<tr>
<td>0.20</td>
<td>21.89</td>
<td>21.98</td>
<td>24.14</td>
<td>24.70</td>
<td>24.34</td>
<td>23.79</td>
<td>23.00</td>
<td>24.09</td>
<td>25.03</td>
</tr>
<tr>
<td>0.25</td>
<td>23.31</td>
<td>23.91</td>
<td>25.67</td>
<td>25.12</td>
<td>25.62</td>
<td>24.92</td>
<td>24.05</td>
<td>25.10</td>
<td>26.41</td>
</tr>
<tr>
<td>0.30</td>
<td>24.73</td>
<td>24.32</td>
<td>27.15</td>
<td>27.43</td>
<td>26.82</td>
<td>26.05</td>
<td>25.14</td>
<td>26.09</td>
<td>29.13</td>
</tr>
<tr>
<td>0.35</td>
<td>26.17</td>
<td>25.70</td>
<td>28.63</td>
<td>28.74</td>
<td>27.98</td>
<td>27.14</td>
<td>26.92</td>
<td>27.15</td>
<td>30.47</td>
</tr>
<tr>
<td>0.40</td>
<td>27.58</td>
<td>27.11</td>
<td>29.96</td>
<td>30.01</td>
<td>29.14</td>
<td>28.23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peppers (512×512)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.15</td>
<td>24.51</td>
<td>24.02</td>
<td>25.70</td>
<td>25.54</td>
<td>25.92</td>
<td>25.46</td>
<td>25.48</td>
<td>25.92</td>
<td>28.03</td>
</tr>
<tr>
<td>0.20</td>
<td>26.88</td>
<td>26.42</td>
<td>27.69</td>
<td>27.01</td>
<td>26.72</td>
<td>26.43</td>
<td>26.13</td>
<td>27.28</td>
<td>30.54</td>
</tr>
<tr>
<td>0.25</td>
<td>29.00</td>
<td>28.67</td>
<td>29.98</td>
<td>29.01</td>
<td>28.23</td>
<td>27.62</td>
<td>27.28</td>
<td>28.32</td>
<td>31.80</td>
</tr>
<tr>
<td>0.30</td>
<td>30.57</td>
<td>30.24</td>
<td>30.55</td>
<td>30.48</td>
<td>30.10</td>
<td>29.29</td>
<td>28.69</td>
<td>29.66</td>
<td>32.89</td>
</tr>
<tr>
<td>0.35</td>
<td>31.80</td>
<td>31.54</td>
<td>31.77</td>
<td>31.55</td>
<td>31.10</td>
<td>30.29</td>
<td>29.66</td>
<td>30.54</td>
<td>33.84</td>
</tr>
<tr>
<td>0.40</td>
<td>32.79</td>
<td>32.60</td>
<td>32.49</td>
<td>32.01</td>
<td>31.21</td>
<td>30.54</td>
<td>30.18</td>
<td>31.18</td>
<td></td>
</tr>
</tbody>
</table>
Table 7. Reconstruction performance comparison between the fusion via $l_1$-norm minimization and solution of problem (84) in average PSNR (in dB). (Reprinted by permission [72] ©2017 IEEE)

<table>
<thead>
<tr>
<th>Sensing rate</th>
<th>DWT <code>sym6</code></th>
<th>DCT $16 \times 16$</th>
<th>DCT $32 \times 32$</th>
<th>$l_1$-fusion</th>
<th>$l_1$+TV-fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Peppers (512 × 512)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>26.88</td>
<td>27.69</td>
<td>27.62</td>
<td>28.95</td>
<td>33.01</td>
</tr>
<tr>
<td>0.30</td>
<td>30.57</td>
<td>30.48</td>
<td>30.10</td>
<td>31.91</td>
<td>34.81</td>
</tr>
<tr>
<td>0.40</td>
<td>32.79</td>
<td>32.49</td>
<td>32.01</td>
<td>33.92</td>
<td>36.15</td>
</tr>
<tr>
<td>0.50</td>
<td>34.61</td>
<td>34.30</td>
<td>33.71</td>
<td>35.67</td>
<td>37.36</td>
</tr>
<tr>
<td><strong>Cameraman (256 × 256)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>21.97</td>
<td>22.88</td>
<td>22.54</td>
<td>23.84</td>
<td>27.94</td>
</tr>
<tr>
<td>0.30</td>
<td>24.84</td>
<td>25.44</td>
<td>24.82</td>
<td>26.48</td>
<td>30.80</td>
</tr>
<tr>
<td>0.40</td>
<td>27.86</td>
<td>27.78</td>
<td>27.11</td>
<td>29.09</td>
<td>33.50</td>
</tr>
<tr>
<td>0.50</td>
<td>30.65</td>
<td>30.28</td>
<td>29.22</td>
<td>31.59</td>
<td>36.23</td>
</tr>
</tbody>
</table>
6 Conclusions and Future Directions

6.1 Conclusions

The purpose of this thesis was to develop compressive sensing recovery algorithms to recover signals that may not have sparse representations in a known block transform - specifically, scenarios where the signal of interest may have a sparse representation in a lapped transform, or it may have multiple sparsifying bases.

Chapter 1 presented the motivation and relevant literature associated with the scope of this thesis. Chapter 2, addressed the problem of recovering a signal from streaming compressive measurements where the signal of interest does not have a spare representation in any known block transform basis. To reduce the blocking artifacts, an LOT was utilized for the sparse representation of the streaming signal. However, each block of the signal could not be reconstructed independently using disjoint measurement blocks due to the coupling introduced by the LOTs. A progressive reconstruction algorithm based on sliding window processing was therefore developed, where the streaming signal was reconstructed over small overlapping shifting intervals. The proposed algorithm utilized the preliminary information from the preceding interval to improve the performance of the signal recovery since the consecutive intervals shared some common sparse vectors. Specifically, the proposed SBL algorithm utilized the previous estimates and the correlations among the non-zero coefficients to improve the performance of the signal recovery. Further, a warm-start procedure was proposed, and fast update formulae were derived to reduce the computational cost of the SBL algorithm. The developed novel SBL algorithm was thus highly efficient for recovering streaming signals.

Chapter 3 investigated the problem of recovering an image from block compressive sensing (BCS) measurements. The BCS procedure was utilized to alleviate the huge computation and memory burdens associated with using a dense measurement matrix for the image’s sensing and reconstruction processes. Traditional BCS algorithms recovers each image block independently and utilizes post-processing methods to remove the blocking artifacts. In contrast, the proposed image recovery method in this chapter was free of post-processing, where we utilized a lapped transform (LT) for the sparse representation of the image to reduce the blocking artifacts. The developed iterative image reconstruction method, which built on the concepts of Chapter 2, was based on
2-D lapped transforms and SBL. The results showed that the proposed method provided a more visually pleasing image without using any post-processing techniques.

In Chapter 4, an CS image recovery algorithm which exploited the sparsity of the image in a given transform while considering the image has a sparse gradient was developed. For this purpose, we modified the total variation (TV) minimization problem by introducing an $l_1$-norm penalty term to exploit the sparsity of the image in a given transform. A customized interior-point method that provided the optimal solution based on SOCP was derived to solve the $l_1$-regularized TV minimization problem. The proposed algorithm solved the Newton system efficiently by exploiting the structure of the Hessian. The results showed that an improved performance could be seen in the reconstructed image from the proposed algorithm compared to TV reconstruction and the image reconstructed based on the sparsifying basis. The developed MATLAB-based solver could handle an averaged size (512 x 512) in a typical computer.

In Chapter 5, we addressed the problem of reconstructing a signal from compressive measurements when the basis in which the signal of interest is the sparsest was unknown. Indeed, this is applicable to many natural signals, such as images and video sequences, where the specific basis in which they are the sparsest is unknown. Generally, multiple bases which lead to a compressible representation of such signals may exist. Conventional CS recovery algorithms typically utilize these sparsifying bases, and the quality of reconstruction therefore varies, depending on the choice of basis. In this chapter, we showed that a performance improvement could be achieved in signal reconstruction from compressive samples by utilizing multiple sparsifying bases. We showed that a fusion of multiple estimates of the signal using different sparsifying bases led to a better signal recovery compared to reconstructions via any single basis. Furthermore, we derived a customized interior-point method in order to jointly obtain such estimates of an image from compressive measurements. The proposed algorithm utilized multiple sparsifying bases, as well as the fact that images usually have a sparse gradient. The developed MATLAB-based solver could handle an averaged size (512 x 512) in a typical computer.

6.2 Future directions

We utilized SBL to develop the algorithms in Chapters 2 and 3 to utilize the prior information available from the preceding intervals when sliding window processing is applied. An alternative recovery algorithm could be developed by modifying the
AMP based algorithm [38, 59] by taking into account the signal structure enforced by LOTs. It would be an interesting future work, especially due to the low complex implementation of the AMP.

Although in the recovery algorithms, the direction of the sliding window was obvious in Chapter 2, the optimal moving direction was not certain in the case of a 2-dimensional scenario in Chapter 3. It would therefore be an interesting research direction to explore the optimal sliding window direction in the case of a 2-dimensional scenario to minimize the memory burden of the algorithm.

Furthermore, TV minimization is one of the key concepts used in image reconstruction. Hence, another important research direction would be to solve the TV minimization problem utilizing a method such as SBL or AMP.

In Chapter 5, we utilized the TV and multiple sparsifying bases to recover an image from compressive measurements. Although we showed that fusing multiple estimates of the signal using TV and several sparsifying bases would improve the reconstruction performance, some unanswered questions remain:

1. What is the optimal fusing method?
2. What is the optimal number of sparsifying bases to be used?
3. What are the properties of the optimal sparsifying bases?

Investigating the solutions to the aforementioned questions would therefore present interesting and valuable future research work.
References


Appendix 1 The 3-stage hierarchical prior model

In [35], a 3-stage hierarchical prior model has been utilized to model the sparsity of a vector \( \mathbf{w} \in \mathbb{R}^N \). In the first stage of this hierarchical model, each element of \( \mathbf{w} \) can be modeled as an independent random variable with a zero-mean Gaussian distribution, with individual variance parameters, as

\[
p(\mathbf{w} | \bm{\gamma}) = \prod_{i=1}^{N} \mathcal{N}(w_i | 0, \gamma_i),
\]

(95)

where \( w_i \) denotes the \( i \)th element of \( \mathbf{w} \) and \( \bm{\gamma} \) is a vector containing \( N \) hyperparameters.

In the second stage, each variance parameter \( \gamma_i \) is modeled as an independent random variable with a Gamma distribution, with a shape parameter equal to 1 and a common rate parameter \( \lambda \), as

\[
p(\bm{\gamma} \mid \lambda) = \prod_{i=1}^{N} \operatorname{Gamma}(\gamma_i | 1, \lambda / 2), \lambda \geq 0.
\]

(96)

In the final stage, a Gamma hyperprior is employed on \( \lambda \) as

\[
p(\lambda \mid a, b) = \operatorname{Gamma}(\lambda | a, b).
\]

By marginalizing over the hyperparameters \( \bm{\gamma} \) and \( \lambda \), we can evaluate the true prior over \( \mathbf{w} \) as

\[
p(\mathbf{w}) = \int \int p(\mathbf{w}, \bm{\gamma}, \lambda) d\bm{\gamma} d\lambda = \int \int p(\mathbf{w} | \bm{\gamma}) p(\bm{\gamma} \mid \lambda) p(\lambda) d\bm{\gamma} d\lambda = \int p(\mathbf{w} | \lambda) p(\lambda) d\lambda
\]

(97)

where

\[
p(\mathbf{w} | \lambda) = \int p(\mathbf{w} | \bm{\gamma}) p(\bm{\gamma} \mid \lambda) d\bm{\gamma} = \prod_{i=1}^{N} \int p(\mathbf{w}_i | \gamma_i) p(\gamma_i \mid \lambda) d\gamma_i.
\]

(98)

Substituting \( p(\mathbf{w}_i | \gamma_i) = \mathcal{N}(w_i | 0, \gamma_i) \) and \( p(\gamma_i \mid \lambda) = \operatorname{Gamma}(\gamma_i | 1, \lambda / 2) \), we can express (98) as

\[
p(\mathbf{w} | \lambda) = \left( \frac{\lambda}{2} \right)^N \prod_{i=1}^{N} \int_{0}^{\infty} \exp \left[ -\frac{\lambda}{2} \gamma_i - \frac{w_i^2}{2 \gamma_i} \right] d\gamma_i.
\]

(99)
Fig. 17. The function $f(w) = 1/\|w\|_1^N$ in two dimensions.

By integrating (99) with the help of [111, eq.3.471.9] we obtain

$$p(w|\lambda) = \left(\frac{\lambda}{2}\right)^N \prod_{i=1}^{N} \frac{2}{\sqrt{2\pi}} \left(\frac{w_i^2}{\lambda}\right)^{\frac{1}{2}} K_{\frac{1}{2}}\left(\sqrt{\frac{\lambda}{2}} w_i^2\right),$$

(100)

where $K_{\nu}(\cdot)$ denotes the modified Bessel function of order $\nu$. Using $K_{\frac{1}{2}}(z) = \sqrt{\frac{\pi}{2z}} e^{-z}$ [111, eq.8.469.3] and mathematically simplifying, we obtain

$$p(w|\lambda) = \left(\frac{\lambda}{2}\right)^N \exp\left(-\sqrt{\lambda} \sum_{i=1}^{N} |w_i|\right).$$

(101)

Since $p(\lambda) \propto 1/\lambda$, we can evaluate the true prior over $w$ as

$$p(w) \propto \int_0^\infty p(w|\lambda) \frac{1}{\lambda} d\lambda = \frac{1}{2N} \int_0^\infty \lambda^{\frac{N}{2}-1} \left(-\sqrt{\lambda} \sum_{i=1}^{N} |w_i|\right) d\lambda.$$

(102)

By using the change of variables $z = \sqrt{\lambda}$, and evaluating the integral with the help of [111, eq.3.326.2], we can obtain the improper prior for $w$ as

$$p(w) \propto \frac{1}{2^{N-1}} \frac{\Gamma(N)}{\left(\sum_{i=1}^{N} |w_i|\right)^N} \propto \frac{1}{\|w\|_1^N},$$

(103)

where $\Gamma(\cdot)$ denotes the Gamma function. Intuitively, this prior promotes sparsity since it is sharply peaked at zero (see Fig. 17). Note that in [35], the true prior of $w$ is not evaluated. Rather, the authors have evaluated only the conditional distribution of the prior $w$ given $\lambda$ expressed in (101). Although (101) corresponds to a Laplace distribution, the true prior on $w$ is given by (103).
Appendix 2 Fast update formulae

In this Appendix, all the quantities are defined for a time instant \( t \). For simplicity of presentation, we have dropped the time index. The block diagonal structure of \( \Omega \) can be used for the efficient computation\(^{27} \) of \( \Omega^{-1} \). The following update formulae can be utilized to add, remove, and re-estimate operations in Algorithm 2. The updated values are denoted by a hat (e.g., \( \hat{w} \)).

2.1 Adding the vector \( u_n \) to the model

\[
2 \triangle L_n = (Q_n^2 - S_n)/S_n + \log(S_n/Q_n^2),
\]

\[
\hat{\Sigma} = \begin{bmatrix}
\Sigma + \Sigma_{nn} \Sigma_n^T \Sigma_n^{-1} u_n \Sigma_n^T \\
-\Sigma_{nn} \Sigma_n^T \Sigma_n^{-1} \Sigma_n
\end{bmatrix},
\]

\[
\hat{\mu} = \begin{bmatrix}
\mu - \mu_n \Sigma_n^T \Sigma_n^{-1} u_n \\
\mu_n
\end{bmatrix},
\]

\[
\hat{S}_m = S_m - \Sigma_{nm} (\Sigma_n^T \Sigma_n^{-1} z_m)^2, \text{ for all } u_m
\]

\[
\hat{Q}_m = Q_m - \mu_n \Sigma_n^T \Sigma_n^{-1} u_m,
\]

where \( \Sigma_{nn} = (\alpha_n + S_n)^{-1} \), \( \mu_n = \Sigma_{nn} Q_n \) and \( z_m = u_m - \bar{u} \Sigma \Sigma^T \Sigma_n^{-1} u_m \).

2.2 Re-estimating precision corresponds to the vector \( u_n \)

Let \( \kappa_n = (\Sigma_{nn} + (\bar{\alpha}_n - \alpha_n)^{-1})^{-1} \), and \( \Sigma_n \) be the \( n \)th column of \( \Sigma \).

\[
2 \triangle L_n = \frac{Q_n^2}{S_n + [\bar{\alpha}_n - \alpha_n^{-1}]^{-1}} - \log(1 + S_n[\bar{\alpha}_n - \alpha_n^{-1}]),
\]

\[
\hat{\Sigma} = \Sigma - \kappa_n \Sigma_n \Sigma_n^T,
\]

\[
\hat{\mu} = \mu - \kappa_n \mu_n \Sigma_n,
\]

\[
\hat{S}_m = S_m + \kappa_n (\Sigma_n^T \Sigma_n^{-1} u_m)^2, \text{ for all } u_m
\]

\[
\hat{Q}_m = Q_m + \kappa_n \mu_n (\Sigma_n^T \Sigma_n^{-1} u_m), \text{ for all } u_m.
\]

\(^{27}\) Note that, by construction \( \Omega = \sigma^2 I_{M(d+2)} + \text{cov}(\hat{w}|\hat{w}) \hat{B}^T \) is a block diagonal, since the bottom \((d+1)M\) rows of \( \hat{B} \) are all zero vectors. Thus, the most expensive operation in inverting \( \Omega \) is the inversion of top left \((d+1)M \times (d+1)M\) matrix of \( \Omega \).
2.3 Deleting the vector \( u_n \) from the model

\[
2\Delta \mathcal{L} = \frac{Q_n^2}{(S_n - \alpha_n)} - \log(1 - S_n / \alpha_n),
\]

(114)

\[
\hat{\Sigma} = \Sigma - \Sigma_n \Sigma_n^\top / \Sigma_{nn},
\]

(115)

\[
\hat{\mu} = \mu - \mu_n \Sigma_n / \Sigma_{nn},
\]

(116)

\[
\hat{S}_m = S_m + \left( \Sigma_n^\top \bar{U}^\top \Omega^{-1} u_m \right)^2 / \Sigma_{nn}, \quad \text{for all } u_m
\]

(117)

\[
\hat{Q}_m = Q_m + \mu_n (\Sigma_n^\top \bar{U}^\top \Omega^{-1} u_m) / \Sigma_{nn}, \quad \text{for all } u_m.
\]

(118)

After the update, the \( n \)th column and row of \( \hat{\Sigma} \), and \( n \)th element of \( \hat{\mu} \) are removed.
Appendix 3 Proof of Proposition 1 in Chapter 2

First, we approximate the posterior covariance $\Sigma_w^t = (A_t + \sigma^{-2}B_t^TB_t)^{-1}$ with a $d$-block banded matrix by utilizing the approximation given below.

**The $d$-block banded approximation:** Since $A_t$ is a diagonal matrix and $B_t$ is a banded matrix (see Fig. 18a), by construction $H_t = A_t + \sigma^{-2}B_t^TB_t$ is a block tridiagonal (1-block banded) matrix. It is well known that the inverse of a banded matrix is a 'band-dominant' matrix [82], where the entries of the inverse have an exponential decay rate [81, Theorem 2.4]29. This property can be generalized for block-banded matrices, and we can approximate the inverse of a block-banded matrix with another block-banded matrix [82, 83, 84, 85, 86]. The approximation is based on the component-wise magnitude. Since the magnitude of the elements in the inverse decay rapidly with the distance to the main diagonal [81, 82, 83, 84, 85, 86], we simply set them to zero if they are off the $d$-block band. Thus, we approximate the covariance matrix $\Sigma_w^t = (A_t + \sigma^{-2}B_t^TB_t)^{-1}$ with a $d$-block-banded matrix. Note that this approximation is asymptotically tight as $d$ increases. In the proposed algorithm, $d$ is a design parameter, and the effect of this parameter on the estimation of the unknown signal is discussed in Section 2.5.

To single out the contribution of $y_{t+1}$, let us express the system matrix $B_{t+1}$ as

$$B_{t+1} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \in \mathbb{R}^{M(t+1) \times N(t+2)}$$

(119)

where $B_1 = [B, 0_{M \times N}] \in \mathbb{R}^{M \times N(t+2)}$ and $B_2 = [0_{M \times N}, B_t+1] \in \mathbb{R}^{M \times N(t+2)}$ (see Fig. 18a). Similarly to Section 2.3.1, we approximate the posterior distribution over $w_{t+1}$ as $p(w_{t+1}|y_{t+1}) \approx p(w_{t+1} | \hat{\sigma}_{\tau+1}^M, \hat{\gamma}_{\tau+1}^y) = \mathcal{N}(w_{t+1} | \mu_w^t + \bar{\Sigma}_w^t, \Sigma_w^t)$ where

$$\Sigma_w^t = (A_{t+1} + \sigma^{-2}B_{t+1}^TB_{t+1})^{-1}$$

(120)

$$= (A_{t+1} + \sigma^{-2}B_t^TB_1 + \sigma^{-2}B_2^TB_2)^{-1}$$

(121)

$$= L^{-1} - L^{-1}B_1^T(\sigma^2I + B_2^TL^{-1}B_1L^{-1})^{-1}B_2L^{-1},$$

(122)

$$\mu_w^t = \sigma^{-2}\Sigma_w^t | B_{t+1}^T y_{t+1},$$

(123)

28Note that the precision matrix $H_t$ has a block tridiagonal structure due to the LOT with length $L = 2N$. For a general LOT with $L = LN$, the precision matrix $H_t$ becomes a $(l-1)$-block banded matrix.

29It is noteworthy that Theorem 2.4 in [81] is valid for the inverses and the pseudo inverses of the matrices.
with $\alpha_{t+1}^{MP} = \arg \max L_{t+1}$ and $L = A_{t+1} + \sigma^{-2}B_1^TB_1 = \begin{bmatrix} A_t + \sigma^{-2}B_1^TB_1 & 0_{N(t+1) \times N} \\ 0_{N \times N(t+1)} & \text{diag}(\alpha_{t+2}) \end{bmatrix}$

is a block diagonal matrix, where the first block diagonal entry $H_t = A_t + \sigma^{-2}B_1^TB_1$ is the precision matrix given in (23). Note that in (120), for simplicity of presentation, we have expressed $\Sigma_w^{(t+1)}$ as the inverse of the precision matrix $H_{t+1}$.

We obtained (121) by substituting (119) in (120), and (122) followed from the matrix inversion lemma.

Since the matrix $L$ is block diagonal and the inverse of its first block diagonal entry is $d$-block-banded, i.e., $\Sigma_w = (A_t + \sigma^{-2}B_1^TB_1)^{-1}$, we can see that the matrix $L^{-1}$ also becomes $d$-block-banded. Now, utilizing the block-banded structure of $L^{-1}$ and the structure of $B_2$ (see Fig. 18a), we can show that the second term in (122) has the following structure:

$$L^{-1}B_2^T(\sigma^2I_d + B_2^TL^{-1}B_2^{-1})^{-1}B_2L^{-1} = \begin{bmatrix} 0_{N(t-d) \times N(t-d)} & 0_{N(t-d) \times N(t+2)} \\ 0_{N(t+2) \times N(t-d)} & \Sigma_1 \end{bmatrix}$$

where $\Sigma_1$ is a $N(d + 2) \times N(d + 2)$ matrix.

---

50 Recall that as some of the $\alpha_t$ tend to infinity, the random variable $w_{t+1}$ becomes a degenerate Gaussian, and the pseudo inverse is utilized.

51 By exploiting the structure of $B_2$, we can see that the $N(t + 2) \times N(t + 2)$ matrix $B_2^T(\sigma^2I_d + B_2^TL^{-1}B_2^{-1})B_2$ has the structure $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, where $Z$ is a $2N \times 2N$ matrix. Now, it can be easily verified that the left and right multiplication of this matrix by the $d$-block-banded matrix $L^{-1}$ gives the structure in (124).
Using (122) and (124), we can express \( \Sigma_{t+1}^w \) as:

\[
\Sigma_{t+1}^w = \begin{bmatrix} \Sigma_t^w & 0_{N(t+1)\times N} \\ 0_{N\times N(t+1)} & \text{diag}(\alpha_{t+2}) \end{bmatrix} = \begin{bmatrix} 0_{N(t-d)\times N(t-d)} & 0_{N(t-d)\times N(d+2)} \\ 0_{N(d+2)\times N(t-d)} & Z_1 \end{bmatrix}.
\] (125)

From (125), we can see that the values of the \( N(t-d) \times N(t-d) \) block in the top-left corner of \( \Sigma_{t+1}^w \) are the same as those values of \( \Sigma_t^w \). In other words, the posterior covariance of the vectors \( w_1, \ldots, w_{t-d} \) obtained at time instant \( t + 1 \) is the same as that obtained at time instant \( t \) (see Fig. 18b). Hence, we can say that only the covariance of vectors \( w_{t+1-d}, \ldots, w_{t+2} \) changes from the covariance obtained at time instant \( t \) (i.e., \( \Sigma_t^w \)) due to the addition of the new measurement vector \( y_{t+1} \). Further, utilizing the structure of \( \Sigma_{t+1}^w \) and the structure of \( B_{t+1} \), we can show that \( \mu_{t+1}^w \) in (123) has the following structure\(^{32}\):

\[
\mu_{t+1}^w = \begin{bmatrix} \mu_t^w \\ \alpha_{t+2} \end{bmatrix} + \begin{bmatrix} 0_{N(t-d)\times 1} \\ Z_1 \end{bmatrix}
\] (126)

where \( z_1 \) is an \( N(d+2) \)-dimensional vector. From (126), we can see that the posterior mean of the vectors \( w_1, \ldots, w_{t-d} \) obtained at time instant \( t + 1 \) is the same as that obtained at time instant \( t \) (i.e., the first \( N(t-d) \) elements of \( \mu_{t+1}^w \) and \( \mu_t^w \) are equal). Since the posterior mean and covariance of \( w_1, \ldots, w_{t-d} \) obtained at time instant \( t + 1 \) are the same as those obtained at time instant \( t \) (see (126) and (125)), the \( \alpha_{t,i} \) for \( \tau = 1, \ldots, t-d, i = 1, \ldots, N \) obtained at time instant \( t \) still satisfies the stationary point condition (22). Hence, we can obtain a stationary point for \( Z_{t+1} \) by only optimizing over \( \alpha_t \) for \( \tau = t + 1 - d, \ldots, t + 2 \). Therefore, at the stationary point

\[
\alpha_{t}^{MP}(t + 1) = \alpha_{t}^{MP}(t), \quad \tau = 1, \ldots, t - d,
\] (127)

\[
p(w_1, \ldots, w_{t-d} | y_{t+1}) = p(w_1, \ldots, w_{t-d} | y_{t+1}),
\] (128)

where \( \alpha_{t}^{MP}(t) \) represents the estimated value of the hyperparameters \( \alpha_t \) at time instant \( t \). Roughly speaking, this implies that the measurement vectors succeeding \( y_t \), i.e., \( y_{t+1}, y_{t+2}, \ldots \), contain no information on the sparse vectors \( w_1, \ldots, w_{t-d} \). Therefore, the posterior estimation of \( w_1, \ldots, w_{t-d} \) that can be obtained by processing the entire set of measurement vectors \( y_1, \ldots, y_t, y_{t+1}, \ldots \) is equal to that which we obtained at time

\(^{32}\)Using (119), (122), and (124), we can express \( \alpha_{t+1}^v = \sigma^2 \left( \begin{bmatrix} I_N \times 0_{N(t-d)\times N(t-d)} \\ 0_{N\times N(t-d)} \end{bmatrix} \right) \), where \( \alpha_{t+1}^v \) is a vector in \( \mathbb{R}^{2N} \). Now, it can be easily seen that the multiplication of the first terms inside the brackets gives the first term in (126), and multiplication by \( z_t \) provides the structure of the second term in (126).
instant $t$, i.e., $\mathbf{w}_\tau \sim \mathcal{N}(\mathbf{\mu}_\tau(t), \mathbf{\Sigma}_\tau(t))$, where $\mathbf{\mu}_\tau(t)$ and $\mathbf{\Sigma}_\tau(t)$ denote the mean and covariance of $\mathbf{w}_\tau$ obtained at time instant $t$ for $\tau = 1, \ldots, t - d$. In other words, we can consider that $\mathbf{y}_t$ is the last measurement vector that contains information on the sparse vectors $\mathbf{w}_1, \ldots, \mathbf{w}_{t-d}$. 


Appendix 4 Proof of Proposition 2 in Chapter 2

To single out the contribution of \( y_{t}, w_{t} \), we express the system matrix at time instant \( t \) as

\[
B_{t} = \begin{bmatrix} B_{3} \\ B_{4} \end{bmatrix} \in \mathbb{R}^{M_{t} \times N(t+1)}
\]  

(129)

where \( B_{3} = [B_{3}, 0_{M_{t^{'}} \times N(t-t^{'})}] \in \mathbb{R}^{M_{t^{'}} \times N(t+1)} \) and \( B_{4} \in \mathbb{R}^{M(t-t^{'}) \times N(t+1)} \) (see Fig. 19a).

We approximate the posterior distribution over \( w \), as \( p(w | y_{t}, r_{t}) \approx p(w | A_{t}, y_{t}, r_{t}) = \mathcal{N}(\mu_{t}, \Sigma_{t}) \) with

\[
\mu_{t} = \sigma^{-2} \Sigma_{t} B_{3}^{T} y_{t},
\]

(132)

where \( A_{t} = \arg \max p(y_{t} | x_{t}, r_{t}) p(x_{t}) \). Note that we obtained (131) by substituting (129) in (23) and then taking the inverse of \( A_{t} + \sigma^{-2} B_{3} B_{4} \) utilizing the matrix inversion lemma.

Now, similarly to Appendix 3, by approximating \( \Sigma_{t} \) with a \( d \)-block-banded matrix and utilizing the matrix structure of \( B_{3} \) (see Fig. 19a), we can show that the second term in (131) has the following structure\(^{33} \):

\[
\Sigma_{t} B_{3}^{T} (\sigma^{-2} I_{M_{t^{'}}} + B_{3} \Sigma_{t} B_{3}^{T})^{-1} B_{3} \Sigma_{t} = \begin{bmatrix} Z_{2} & 0_{N(t+1) \times N(t+1)} \\ 0_{N(t+1) \times N(t+1)} & 0_{N(t+1) \times N(t+1)} \end{bmatrix}
\]

(133)

where \( Z_{2} \) is a \( N(t+1) \times N(t+1) \) matrix.

Using (131) and (133), we can express \( \Sigma_{t} \) as:

\[
\Sigma_{t} = \Sigma_{t} + \begin{bmatrix} Z_{2} & 0_{N(t+1) \times N(t+1)} \\ 0_{N(t+1) \times N(t+1)} & 0_{N(t+1) \times N(t+1)} \end{bmatrix}
\]

(134)

From (134), we can see that the posterior covariance of the vectors \( w_{t=1}, \ldots, w_{t+1} \) obtained by removing the measurements \( y_{1}, \ldots, y_{t} \) is the same as that obtained by

\(^{33}\)By exploiting the structure of \( B_{3} \), we can see that the \( N(t+1) \times N(t+1) \) matrix \( B_{3}^{T} (\sigma^{-2} I_{M_{t^{'}}} + B_{3} \Sigma_{t} B_{3}^{T})^{-1} B_{3} \) has the structure \( \begin{bmatrix} Z_{2} & 0_{N(t+1) \times N(t+1)} \\ 0_{N(t+1) \times N(t+1)} & 0_{N(t+1) \times N(t+1)} \end{bmatrix} \) where \( Z_{2} \) is a \( N(t+1) \times N(t+1) \) matrix. Now, it can be easily verified that left and right multiplication of this matrices by the \( d \)-block-banded matrix \( \Sigma_{t} \) gives the structure in (133).
considering all the measurements at time instant $t$ (i.e., values of the $N(t - t' - d) \times N(t - t' - d)$ block of the bottom right corner of $\Sigma_{w,t}$ and $\Sigma_{t}$ are the same). Hence, we can say that only the posterior covariance of vectors $w_{t}, \ldots, w_{t+1}$ differs from the posterior covariance obtained by considering all measurements at time instant $t$ (i.e., $\Sigma_{t}$), due to the removal of the measurement vectors $y_{t}, \ldots, y_{t'}$ (see Fig. 19b). Further, by utilizing the structure of $\Sigma_{t}$ and the structure of $B_{i}$ in (129), we can show that $\mu_{w,t'}^{w}$ in (132) has the following structure:

$$
\mu_{w,t'}^{w} = \mu_{t}^{w} - \left[ Z_{2} \right]^{-1} \left[ 0_{N(t-t'-d) \times 1} \right]
$$

(135)

where $Z_{2}$ is a $N(t' + d + 1)$-dimensional vector. Now, from (135), we can see that the mean of the vectors $w_{t'+d+2}, \ldots, w_{t+1}$ obtained by removing measurements $y_{t}, \ldots, y_{t'}$ is the same as that obtained by processing all the measurements at time instant $t$ (i.e., the last $N(t-t'-d)$ elements of $\mu_{w,t'}^{w}$ are equal). Since the posterior mean and covariance of the vectors $w_{t'+d+2}, \ldots, w_{t+1}$ obtained by removing the measurements $y_{t}, \ldots, y_{t'}$ are the same as those obtained by processing all the measurements at time $t$.

34 Using (131), (133), and $B_{i}$ in (129), we can express $\mu_{w,t'}^{w} = \mu_{t}^{w} - \left[ z_{2} \right]^{-1} \left[ z_{2} \right]^{T} \left[ \begin{array}{c} z_{2} \\ e_{o} \end{array} \right] \left[ \begin{array}{c} w_{t'} \\ w_{t+1} \end{array} \right]$, where $w_{t'}$ is a vector in $\mathbb{R}^{N(t'-1)}$. Now, it can be easily seen that the multiplication of the first terms inside the brackets gives the first term in (135), and multiplication by $z_{2}$ provides the structure of the second term in (135).
instant \( t \) (see (135) and (134)), the estimated values for \( \alpha_{\tau,i} \) where \( \tau = t' + d + 2, \ldots, t + 1, i = 1, \ldots, N \) in Section 2.3.1 still satisfy the stationary point condition (22). Hence, at the stationary point

\[
\alpha_{\tau}^{\text{MP}}(t \setminus t') = \alpha_{\tau}^{\text{MP}}(t), \quad \tau = t' + d + 2, \ldots, t + 1,
\]

\[
p(w_{t'+d+2}, \ldots, w_{t+1} | y_{t', \ldots, t}) = p(w_{t'+d+2}, \ldots, w_{t+1} | y_{t}),
\]

where \( \alpha_{\tau}^{\text{MP}}(t \setminus t') \) represents the estimated value of \( \alpha_{\tau} \) obtained using measurements \( y_{t', \ldots, t} \). Roughly speaking, this implies that measurement vectors \( y_1, \ldots, y_{t'} \) provide no information concerning the sparse vectors \( w_{t'+d+2}, \ldots, w_{t+1} \). Hence, we can estimate the posterior density of those sparse vectors at time instant \( t \) by processing only the measurements \( y_{t'+1}, \ldots, y_t \). In other words, we can consider that \( y_{t'-d-1} \) is the oldest measurement vector that contains information on the sparse vector \( w_t \).
Appendix 5 Efficient computation of the Newton’s step in Chapter 4

In this Appendix, we discuss efficient methods which can be utilized to compute different quantities that are essential to implementing the solution of the Newton system in Chapter 4. Specifically, we provide the detailed derivation of the expressions (77), (79), and (82) that can be used to compute the quantities $S$, $F_2D_3^{-1}g_2$ and $F_2^T\Delta w$ respectively.

When implementing the Newton’s method (Algorithm 4), we compute the search direction (or Newton step) $\Delta q = [\Delta w^T \Delta v^T \Delta u^T]^T$ by solving the Newton system

$$H\Delta q = -g,$$  \hspace{1cm} (138)

where

$$H = \nabla^2 \psi_t(w,v,u) = \begin{bmatrix} F_1 + D_1 & F_2 & D_2 \\ F_2^T & D_3 & 0 \\ D_2 & 0 & D_1 \end{bmatrix},$$  \hspace{1cm} (139)

and

$$g = \begin{bmatrix} \nabla_w \psi_t \\ \nabla_v \psi_t \\ \nabla_u \psi_t \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}.$$  \hspace{1cm} (140)

We solve the Newton system (138) by exploiting the ‘arrow’ structure of the Hessian $H$ (see (139)). In particular, we efficiently compute $\Delta w$, $\Delta v$, and $\Delta u$ by employing the block elimination procedure [47, App. C.4] instead of directly using the inverse of $H$.

Specifically, we compute $\Delta w$ by solving

$$S\Delta w = z_1,$$  \hspace{1cm} (141)

where $z_1 = -g_1 + F_2D_3^{-1}g_2 + D_2D_1^{-1}g_3$ and $S$ is the Schur complement of $\begin{bmatrix} D_3 & 0 \\ 0 & D_1 \end{bmatrix}$.

Then, by utilizing the value of $\Delta w$, we evaluate $\Delta v$ and $\Delta u$ as

$$\begin{bmatrix} \Delta v \\ \Delta u \end{bmatrix} = \begin{bmatrix} D_3^{-1}g_2 \\ D_1^{-1}g_3 \end{bmatrix} - \begin{bmatrix} D_3 & 0 \\ 0 & D_1 \end{bmatrix}^{-1} \begin{bmatrix} F_2^T \\ D_2^T \end{bmatrix} \Delta w,$$

which can be computed efficiently by

$$\Delta v = -D_3^{-1}(g_2 + F_2^T\Delta w),$$  \hspace{1cm} (142)

$$\Delta u = -D_1^{-1}(g_3 + D_2^T\Delta w).$$  \hspace{1cm} (143)
5.1 Efficient computation of the Schur complement $S$

We need to generate the Schur compliment $S$ and the quantity $z_1$ to evaluate $\Delta w$ using (141). Here, we briefly discuss a method for efficiently implementing the Schur compliment $S$. We can write the Schur complement $S$ of

$$
S = \begin{bmatrix} D_3 & 0 \\ 0 & D_1 \end{bmatrix}^{-1} \begin{bmatrix} F^T_2 \\ D_2 \end{bmatrix}
$$

$$(144)$$

where

$$
F_1 = \frac{2A^TA}{\delta^2 - \|r\|^2_2^2} + \frac{4A^Trr^TA}{\delta^2 - \|r\|^2_2^2} + \sum_{i=1}^N \sum_{j=1}^N \left[ \begin{array}{c} 2B_i^T B_j \\ 2B_i^T B_j \end{array} \right] + \frac{4B_i^T (B_i w)(B_i w)^T B_i}{\|B_i w\|^2_2^2},
$$

$$(145)$$

$$
F_2 = \begin{bmatrix} -4v_{11} B_{11}^T B_{11} w \\ \vdots \\ -4v_{NN} B_{NN}^T B_{NN} w \end{bmatrix},
$$

$$(146)$$

$$
D_1 = \text{diag}\left( \frac{2(u_1^2 + w_1^2)}{(u_1^2 - w_1^2)^2}, \ldots, \frac{2(u_N^2 + w_N^2)}{(u_N^2 - w_N^2)^2} \right),
$$

$$(147)$$

$$
D_2 = \text{diag}\left( \frac{-4u_1 w_1}{(u_1^2 - w_1^2)^2}, \ldots, \frac{-4u_N w_N}{(u_N^2 - w_N^2)^2} \right),
$$

$$(148)$$

$$
D_3 = \text{diag}\left( \frac{2(v_{11}^2 + \|B_{11} w\|^2_2)}{(v_{11}^2 - \|B_{11} w\|^2_2)}, \ldots, \frac{2(v_{NN}^2 + \|B_{NN} w\|^2_2)}{(v_{NN}^2 - \|B_{NN} w\|^2_2)} \right).
$$

$$(149)$$

Here, $D_1, D_2, D_3$ are diagonal matrices. Thus, the most computationally expensive operation in generating $S$ is computing $F_1 - F_2 D_3^{-1} F_2^T$. Next, we explain in detail a method to generate the quantity $F_1 - F_2 D_3^{-1} F_2^T$ efficiently.

Let $f_\delta = (\delta^2 - \|r\|^2_2)/2$ and $B = \begin{bmatrix} B_{11}^T, \ldots, B_{NN}^T \end{bmatrix}^T$. Now, we can express $F_1$ in (145) as

$$
F_1 = F_{11} + F_{12},
$$

$$(150)$$

where

$$
F_{11} = f_\delta^{-1} A^T A + f_\delta^{-2} A^T r r^T A,
$$

$$(151)$$

$$
F_{12} = B^T \text{diag}\left( \frac{2I_2}{(v_{11}^2 - \|B_{11} w\|^2_2)}, \ldots, \frac{4(B_{11} w) (B_{11} w)^T}{(v_{11}^2 - \|B_{11} w\|^2_2)}, \ldots, \frac{2I_2}{(v_{NN}^2 - \|B_{NN} w\|^2_2)}, \ldots, \frac{4(B_{NN} w) (B_{NN} w)^T}{(v_{NN}^2 - \|B_{NN} w\|^2_2)} \right) B.
$$

$$(152)$$
Furthermore, we can express $F_2$ in (146) as
\begin{equation}
F_2 = B^T \text{diag } \left( -\frac{4v_1}{(v_1^2 - \|B_{11}w\|^2)^2}, \ldots, -\frac{4v_N}{(v_N^2 - \|B_{NN}w\|^2)^2} \right). \tag{153}
\end{equation}

By substituting (150) in (144), we can rewrite $S$ as
\begin{align*}
S &= F_{11} + F_{12} - F_2D_3^{-1}F_2^T + D_1 - D_2D_1^{-1}D_2 \\
&= F_{11} + S_1 + D_1 - D_2D_1^{-1}D_2, \tag{154}
\end{align*}
where $S_1 = F_{12} - F_2D_3^{-1}F_2^T$. By re-arranging elements in the matrices, we can express $S_1$ as (155), where $f_{ij} = 2/(v_{ij}^2 - \|B_{ij}w\|^2)$, $\sigma_{ij} = -4/(v_{ij}^4 - \|B_{ij}w\|^4)$, $B_h = \begin{bmatrix} b_{h1}^1 \\ \vdots \\ b_{hN}^N \end{bmatrix}$, and $\Sigma_{\delta h} = \text{diag } (B_h w)$.

Now, substituting (155) in (154), we can express $S$ as
\begin{align*}
S &= D_1 - D_2D_1^{-1}D_2 + f_{ij}^{-1}A^T A + f_{ij}^{-2}A^T r r^T A + \\
&\quad B_h^T (F_{11}^{-1} + \Sigma_{\delta h}^2) B_h + B_v^T (F_{11}^{-1} + \Sigma_v^2) B_v + \\
&\quad B_h^T (\Sigma_h \Sigma_{\delta h}^2 B_h) + B_v^T (\Sigma_v \Sigma_{\delta v}^2 B_v).
\end{align*} \tag{156}

Note that $D_1, D_2, F_1, \Sigma_h, \Sigma_{\delta h}$, and $\Sigma_{\delta v}$ are all diagonal matrices in (156). Hence, the most computationally expensive operation for generating $S$ is computing $A^T A$. However, this computation can be done one time and reused in each Newton step since $A = \Phi \Psi$. Thus, we can compute $S$ efficiently using (156).

### 5.2 Efficient computation of $F_2D_3^{-1}g_2$

To evaluate $\Delta w$ using (141), not only $S$ but also the quantity $z_1 = -g_1 + F_2D_3^{-1}g_2 + D_2D_1^{-1}g_3$ need to be computed efficiently. Here, $g_1, g_2$, and $g_3$ are column vectors and $D_1, D_2,$ and $D_3$ are diagonal matrices. Thus, the most computationally expensive operation in generating $z_1$ is computing $F_2D_3^{-1}g_2$. Next, we briefly discuss a method to generate the quantity $F_2D_3^{-1}g_2$ efficiently.
First, we express $F_2$ in (146) as

$$
F_2 = \begin{bmatrix}
  b_{11}^h \\
  b_{11}^v \\
  \vdots \\
  b_{NN}^h \\
  b_{NN}^v
\end{bmatrix}^T
\begin{bmatrix}
  \rho_{11} b_{11}^h w \\
  \rho_{11} b_{11}^v w \\
  \vdots \\
  \rho_{NN} b_{NN}^h w \\
  \rho_{NN} b_{NN}^v w
\end{bmatrix}
$$

(157)

where $\rho_{ij} = -4v_{11}/(v_{11}^2 - \|B_{11}w\|_2^2)$.

By re-arranging the elements in matrices, we can express $F_2$ in (157) as

$$
F_2 = \begin{bmatrix}
  b_{11}^h \\
  b_{11}^v \\
  \vdots \\
  b_{NN}^h \\
  b_{NN}^v
\end{bmatrix}^T
\begin{bmatrix}
  \rho_{11} b_{11}^h w \\
  \rho_{11} b_{11}^v w \\
  \rho_{NN} b_{NN}^h w \\
  \rho_{NN} b_{NN}^v w
\end{bmatrix}
$$

(158)

$$
= \begin{bmatrix}
  B_{\cdot h}^T & B_{\cdot v}^T
\end{bmatrix}
\begin{bmatrix}
  \Sigma_{\partial h} \\
  \Sigma_{\partial v}
\end{bmatrix}
H_w
$$

(159)

where $H_w = \text{diag}(\rho_{11}, \ldots, \rho_{NN})$. Thus, we can compute the quantity $F_2D_3^{-1}g_2$ as

$$
F_2D_3^{-1}g_2 = \begin{bmatrix}
  B_{\cdot h}^T & B_{\cdot v}^T
\end{bmatrix}
\begin{bmatrix}
  \Sigma_{\partial h} \\
  \Sigma_{\partial v}
\end{bmatrix}
H_wD_3^{-1}g_2,
$$

(160)

where $G_v = H_wD_3^{-1} = \text{diag}(-2v_{11}/(v_{11}^2 + \|B_{11}w\|_2^2), \ldots, -2v_{NN}/(v_{NN}^2 + \|B_{NN}w\|_2^2))$. Note that $G_v, \Sigma_{\partial h}$, and $\Sigma_{\partial v}$ are all diagonal matrices in (160). Thus, we can compute $F_2D_3^{-1}g_2$ efficiently using (160).

### 5.3 Efficient computation of $F_2^T\Delta w$

With the evaluated value of $\Delta w$, we can compute $\Delta v$ and $\Delta u$ by using (142) and (143) respectively. Here, $\Delta w, g_2$, and $g_3$ are column vectors and $D_1, D_2$, and $D_3$ are diagonal matrices. Thus, the most computationally expensive operation in computing $\Delta v$ and $\Delta u$ is evaluating $F_2^T\Delta w$. 

117
By using the expression (159) for $F_2$, we can express the quantity $F_2^T \Delta w$ as

$$F_2^T \Delta w = H_v \left[ \begin{array}{c} \Sigma_{\theta h} \\ \Sigma_{\theta v} \end{array} \right]^T \left[ \begin{array}{cc} B_h^T & B_v^T \end{array} \right]^T \Delta w,$$

$$= H_v (\Sigma_{\theta h} B_h \Delta w + \Sigma_{\theta v} B_v \Delta w).$$

(161)

Note that $\Delta w$ is a column vector and $H_v, \Sigma_{\theta h},$ and $\Sigma_{\theta v}$ are diagonal matrices in (161). Thus, we can compute $F_2^T \Delta w$ efficiently using (161).
Appendix 6 Efficient computation of the Newtons step in Chapter 5

In this Appendix, we discuss efficient methods which can be utilized to compute different quantities essential to implement the solution of the Newton system in Chapter 5.

When implementing Newton’s method (Algorithm 7), we compute the search direction (or Newton step) \( \Delta q = [\Delta w^T \Delta v^T \Delta u^T]^T \) by solving the Newton system

\[
H \Delta q = -g,
\]

where

\[
H = \begin{bmatrix}
F_1 + F_2 + D_1 & F_3 & D_2 \\
F_3^T & D_3 & 0 \\
D_2 & 0 & D_1
\end{bmatrix},
\]

and

\[
g = \begin{bmatrix}
\nabla_w \psi_t \\
\nabla_v \psi_t \\
\nabla_u \psi_t
\end{bmatrix} = \begin{bmatrix}
g_1 \\
g_2 \\
g_3
\end{bmatrix}.
\]

We solve the Newton system (162) by exploiting the ‘arrow’ structure of the Hessian \( H \) (see (163)). In particular, we efficiently compute \( \Delta w, \Delta v, \) and \( \Delta u \) by employing the block elimination procedure [47, App. C.4] instead of directly using the inverse of \( H \).

Specifically, we compute \( \Delta w \) as

\[
\Delta w = S^{-1} z_1,
\]

where \( z_1 = -g_1 + F_3 D_3^{-1} g_2 + D_2 D_1^{-1} g_3 \), and \( S \) is the Schur complement of \( \begin{bmatrix} D_3 & 0 \\ 0 & D_1 \end{bmatrix} \).

Then, by utilizing the value of \( \Delta w \), we evaluate \( \Delta v \) and \( \Delta u \) as

\[
\begin{bmatrix}
\Delta v \\
\Delta u
\end{bmatrix} = -\begin{bmatrix} D_3^{-1} g_2 \\ D_1^{-1} g_3 \end{bmatrix} - \begin{bmatrix} D_3 & 0 \\ 0 & D_1 \end{bmatrix}^{-1} \begin{bmatrix} F_3 \\ D_2 \end{bmatrix} \Delta w,
\]

which can be computed efficiently by

\[
\Delta v = -D_3^{-1}(g_2 + F_3^T \Delta w),
\]

\[
\Delta u = -D_1^{-1}(g_3 + D_2^T \Delta w).
\]
6.1 Efficient computation of $S^{-1}$

To evaluate $\Delta w$ using (165), we need to generate the inverse of Schur compliment $S$ and the quantity $z_1$. Here, we briefly discuss a method to efficiently implement the inverse of Schur compliment $S$.

First, we express the Schur complement $S$ of $\begin{bmatrix} D_3 & 0 \\ 0 & D_1 \end{bmatrix}$ in (163) as

$$S = F_1 + F_2 + D_1 - \begin{bmatrix} F_3 & D_2 \\ 0 & F_1 \end{bmatrix} \begin{bmatrix} F_3 & 0 \\ 0 & D_1 \end{bmatrix}^{-1} \begin{bmatrix} F_3^T \\ D_2^T \end{bmatrix},$$

$$= F_1 + F_2 + D_1 - F_3 D_3^{-1} F_3^T - D_2 D_1^{-1} D_2,$$

$$= S_1 + S_2,$$  \hspace{1cm} (168)

where $S_1 = F_1 + D_1 - D_2 D_1^{-1} D_2$ and $S_2 = F_2 - F_3 D_3^{-1} F_3^T$, with

$$F_1 = \text{diag}(F_{1,1}, \ldots, F_{K,1})$$

$$F_{k,1} = \frac{2A_k^T A_k}{\delta^2 - \|r_k\|^2} + \frac{4A_k^T r_k r_k^T A_k}{\delta^2 - 3\|r_k\|^2} \in \mathbb{R}^{N^2 \times N^2},$$

$$D_1 = \text{diag}(D_{1,1}, \ldots, D_{K,1})$$

$$D_{k,1} = \text{diag} \left( \frac{2(u^2_1 + w^2_1 k)}{(u^2_1 - w^2_1 k)^2}, \ldots, \frac{2(u^2_4 + w^2_4 k)}{(u^2_4 - w^2_4 k)^2} \right) \in \mathbb{R}^{N^2 \times N^2},$$

$$D_2 = \text{diag}(D_{1,2}, \ldots, D_{K,2})$$

$$D_{k,2} = \text{diag} \left( \frac{-4u_k w_k}{(u_k^2 - w_k^2 k)^2}, \ldots, \frac{-4u_k w_k}{(u_k^2 - w_k^2 k)^2} \right) \in \mathbb{R}^{N^2 \times N^2},$$

$$F_3 = \sum_{i=1}^{N} \sum_{j=1}^{N} \left[ \frac{2B_i^T B_j}{(v_i^2 - \|B_i\|^2)^2} + \frac{4B_i^T (B_j^T w) (B_j^T w)^T B_j}{(v_i^2 - \|B_j\|^2)^2} \right] \in \mathbb{R}^{KN^2 \times KN^2},$$

$$F_1 = \left[ \frac{-4v_i B_i^T B_i w}{(v_i^2 - \|B_i\|^2)^2} \ldots, \frac{-4v_i B_i^T B_i w}{(v_i^2 - \|B_i\|^2)^2} \right] \in \mathbb{R}^{KN^2 \times N^2},$$

$$D_1 = \text{diag} \left( \frac{2v^2_1 + \|B_1\|^2}{(v_1^2 - \|B_1\|^2)^2}, \ldots, \frac{2v^2_4 + \|B_4\|^2}{(v_4^2 - \|B_4\|^2)^2} \right) \in \mathbb{R}^{N^2 \times N^2},$$

(169)

Now, we can express $S_1 \in \mathbb{R}^{KN^2 \times KN^2}$ as

$$S_1 = F_1 + D_1 - D_2 D_1^{-1} D_2,$$

$$= \text{diag}(f^{-1}_1 A_1^T A_1, \ldots, f^{-1}_K A_K^T A_K) + \text{diag}(f^{-2}_1 A_1^T r_1 r_1^T A_1, \ldots, f^{-2}_K A_K^T r_K r_K^T A_K) + D_1 - D_2 D_1^{-1} D_2,$$  \hspace{1cm} (170)
where \( f_{\delta} = (\delta^2 - \|r_k\|_2^2)/2 \). It is clear from (170) that \( S_1 \) is a block diagonal matrix containing \( N^2 \times N^2 \) blocks, since \( D_1 \) and \( D_2 \) are diagonal matrices and \( A_k \) for \( k = 1, \ldots, K \) are \( M \times N^2 \) matrices.

Let \( f_{vj} = (v_{jv}^2 - \|B_v w\|_2^2)/2 \). We can then express the matrix \( F_2 \) as

\[
F_2 = f_{v11}^{-1} B_{11}^T B_{11} + f_{v12}^{-2} (B_{11} w)(B_{11} w)^T + \ldots + f_{vNN}^{-1} B_{NN}^T B_{NN} + f_{vNN}^{-2} (B_{NN} w)(B_{NN} w)^T
= B^T \text{diag}(f_{v11}^{-1} 1_2 + f_{v12}^{-2} (B_{11} w)(B_{11} w)^T, \ldots, f_{vNN}^{-1} 1_2 + f_{vNN}^{-2} (B_{NN} w)(B_{NN} w)^T) B,
\]

(171)

where \( B = \begin{bmatrix} B_{11}^T, \ldots, B_{NN}^T \end{bmatrix} \in \mathbb{R}^{2N^2 \times KN^2} \). Then, by letting \( \rho_{ij} = -4v_{ji}/(v_{ji}^2 - \|B_{ij} w\|_2^2) \), we can express \( F_3 \) as

\[
F_3 = \begin{bmatrix} \rho_{11} B_{11}^T B_{11} w & \cdots & \rho_{NN} B_{NN}^T B_{NN} w \end{bmatrix},
= B^T \text{diag}(\rho_{11} B_{11} w, \ldots, \rho_{NN} B_{NN} w),
\]

(172)

which can be utilized to rewrite the term \( F_3 D_3^{-1} F_3^T \) as

\[
F_3 D_3^{-1} F_3^T = B^T \text{diag}(\rho_{11} B_{11} w, \ldots, \rho_{NN} B_{NN} w) \text{diag}(1/\gamma_{i1}, \ldots, 1/\gamma_{NN})
\]

\[
\text{diag}(\rho_{11} B_{11} w, \ldots, \rho_{NN} B_{NN} w) B,
\]

(173)

where \( \gamma_{i1} = 2(v_{i1}^2 + \|B_{ij} w\|_2^2)/(v_{i1}^2 - \|B_{ij} w\|_2^2) \).

Now, by using (171) and (173) we can rewrite \( S_2 = F_2 - F_3 D_3^{-1} F_3^T \) as

\[
S_2 = B^T \text{diag}(f_{v11}^{-1} 1_2 + \sigma_{i1} (B_{ij} w)(B_{ij} w)^T, \ldots, f_{vNN}^{-1} 1_2 + \sigma_{NN} (B_{NN} w)(B_{NN} w)^T) B,
\]

(174)

where \( \sigma_{ij} = f_{vij}^{-2} - \rho_{ij}^2/\gamma_{ij} = -4/(v_{ij}^2 - \|B_{ij} w\|_2^2) \). We can illustrate \( S_2 \) in (174) as given in (175). Now, by re-arranging elements in the matrices, we can express \( S_2 \) in (175) as

\[
S_2 = \tilde{B}^T \tilde{S}_2 \tilde{B}
\]

(176)

where \( \tilde{B} = \begin{bmatrix} B_x \vline B_v \end{bmatrix} \in \mathbb{R}^{2N^2 \times KN^2} \) and \( \tilde{S}_2 = \begin{bmatrix} F_v^{-1} + A_{i1} A_{i1}^\top & \Sigma_i \Sigma G_{i1} \Sigma_i \Sigma G_{i1}^\top \\ \Sigma_i \Sigma G_{i1} \Sigma_i \Sigma G_{i1}^\top & F_v^{-1} + \Sigma_i \Sigma G_{i1} \Sigma_i \Sigma G_{i1}^\top \end{bmatrix} \in \mathbb{R}^{2N^2 \times 2N^2} \),

with \( B_x = \begin{bmatrix} b_{i1}^x \\ \vdots \\ b_{iN}^x \end{bmatrix} \), \( B_v = \begin{bmatrix} b_{i1}^v \\ \vdots \\ b_{iN}^v \end{bmatrix} \), \( F_v = \text{diag}(f_{v11}, \ldots, f_{vNN}) \), \( \Sigma_i = \text{diag}(\sigma_{i1}, \ldots, \sigma_{iN}) \),

121
\[
\begin{pmatrix}
\varepsilon(W_q^{N}) & \varepsilon(W_q^{N}) + \frac{N}{2}f \\
\vdots & \vdots \\
\varepsilon(W_q^{N}) & \varepsilon(W_q^{N}) + \frac{N}{2}f \\
\end{pmatrix}

= \varepsilon S
\]
\( \Sigma_{\partial h} = \text{diag}(B_h w) \) and \( \Sigma_{\partial v} = \text{diag}(B_v w) \). Thus, using (170) and (176), we can express the Schur compliment \( S \) in (168) as

\[
S = S_1 + \tilde{B}^T \tilde{S}_2 \tilde{B},
\]

where \( S_1 \in \mathbb{R}^{KN^2 \times KN^2} \) is a block diagonal matrix containing \( N^2 \times N^2 \) blocks, \( \tilde{B} \in \mathbb{R}^{2N^2 \times KN^2} \) is a full matrix, and \( \tilde{S}_2 \in \mathbb{R}^{2N^2 \times 2N^2} \) consists of four \( N^2 \times N^2 \) diagonal matrices. Let \( \tilde{S}_2 = \begin{bmatrix} Z_1 & Z_2 \\ Z_2 & Z_4 \end{bmatrix} \), where \( Z_1 = F_r^{-1} + \Sigma_v \Sigma_{\partial h} \), \( Z_2 = \Sigma_v \Sigma_{\partial h} \), and \( Z_4 = F_r^{-1} + \Sigma_v \Sigma_{\partial v} \) are \( N^2 \times N^2 \) diagonal matrices. Now, utilizing the matrix inversion lemma we can compute \( S^{-1} \) efficiently as

\[
S^{-1} = S_1^{-1} - S_1^{-1} \tilde{B}^T (\tilde{S}_2^{-1} + \tilde{B} S_1^{-1} \tilde{B})^{-1} \tilde{B} S_1^{-1},
\]

where \( (\tilde{S}_2^{-1} + \tilde{B} S_1^{-1} \tilde{B}) \in \mathbb{R}^{2N^2 \times 2N^2} \). Furthermore, we can evaluate the inverse of \( \tilde{S}_2 \) efficiently as

\[
\tilde{S}_2^{-1} = \begin{bmatrix} (Z_4 Z_4 - Z_2^2)^{-1} Z_4 & -(Z_4 Z_4 - Z_2^2)^{-1} Z_2 \\ -(Z_4 Z_4 - Z_2^2)^{-1} Z_2 & (Z_4 Z_4 - Z_2^2)^{-1} Z_1 \end{bmatrix}.
\]

Since \( S_1 \) is a block diagonal matrix containing \( N^2 \times N^2 \) blocks, the most computationally expensive operation is the inversion of \( (\tilde{S}_2^{-1} + \tilde{B} S_1^{-1} \tilde{B}) \), which costs \( \mathcal{O}(P^3) \) flops with \( P = 2N^2 \). Note that for the case of \( K = 1, 2 \), it is efficient to inverse \( S \) instead of using (178) since \( S \in \mathbb{R}^{KN^2 \times KN^2} \).

### 6.2 Efficient computation of \( F_3 D_3^{-1} g_2 \)

To evaluate \( \Delta w \) using (165), both the inverse of \( S \) and the quantity \( z_1 = -g_1 + F_3 D_3^{-1} g_2 + D_2 D_1^{-1} g_1 \) need to be computed efficiently. Here, \( g_1, g_2, \) and \( g_3 \) are column vectors and \( D_1, D_2, \) and \( D_3 \) are diagonal matrices. Thus, the most computationally expensive operation in generating \( z_1 \) is computing \( F_3 D_3^{-1} g_2 \). Next, we briefly discuss a method to generate the quantity \( F_3 D_3^{-1} g_2 \) efficiently.
Using (172), we can express $F_3$ as

$$F_3 = B^T \text{diag}(\rho_{11}B_{11}w, \ldots, \rho_{NN}B_{NN}w),$$

$$= \begin{bmatrix} b_{h11}^T & \rho_{11}b_{h11}w \\
\vdots & \ddots \\
b_{hNN}^T & \rho_{NN}b_{hNN}w \\
b_{v11}^T & \rho_{11}b_{v11}w \\
\vdots & \ddots \\
b_{vNN}^T & \rho_{NN}b_{vNN}w \end{bmatrix}$$

(180)

where $\rho_{ij} = -4v_{ij}/(v_{ij}^2 - \|B_{ij}w\|_2^2)$. By re-arranging the elements in matrices, we can express $F_3$ in (180) as

$$F_3 = \begin{bmatrix} b_{h11}^T & \rho_{11}b_{h11}w \\
\vdots & \ddots \\
b_{hNN}^T & \rho_{NN}b_{hNN}w \\
b_{v11}^T & \rho_{11}b_{v11}w \\
\vdots & \ddots \\
b_{vNN}^T & \rho_{NN}b_{vNN}w \end{bmatrix} \left[ \begin{bmatrix} \rho_{11}b_{h11}w \\
\vdots \\
\rho_{NN}b_{hNN}w \\
\rho_{11}b_{v11}w \\
\vdots \\
\rho_{NN}b_{vNN}w \end{bmatrix} \right]$$

(181)

where $H_v = \text{diag}(\rho_{11}, \ldots, \rho_{NN})$. Thus, we can compute the quantity $F_3D_3^{-1}g_2$ as

$$F_3D_3^{-1}g_2 = \begin{bmatrix} B_h^T & B_v^T \end{bmatrix} \begin{bmatrix} \Sigma_{\partial h} \\
\Sigma_{\partial v} \end{bmatrix} H_v$$

(182)

where

$$G_v = H_vD_3^{-1} = \text{diag}(-2v_{11}/(v_{11}^2 + \|B_{11}w\|_2^2), \ldots, -2v_{NN}/(v_{NN}^2 + \|B_{NN}w\|_2^2)).$$

Note that, $G_v, \Sigma_{\partial h}$ and $\Sigma_{\partial v}$ are all diagonal matrices in (182). Thus, we can compute $F_3D_3^{-1}g_2$ efficiently using (182).

### 6.3 Efficient computation of $F_3^T \Delta w$

With the evaluated value of $\Delta w$, we can compute $\Delta v$ and $\Delta u$ by using (166) and (167) respectively. Here, $\Delta w, g_2,$ and $g_3$ are column vectors, and $D_1, D_2,$ and $D_3$ are diagonal matrices.
matrices. Thus, the most computationally expensive operation in computing $\Delta v$ and $\Delta u$ is evaluating $F_3^T \Delta w$.

By using the expression (181) for $F_3$, we can express the quantity $F_3^T \Delta w$ as

$$
F_3^T \Delta w = H_v \left[ \Sigma_{\partial h} \right]^T \left[ B_h^T \begin{bmatrix} B_h^T & B_v^T \end{bmatrix}^T \Delta w, \\
= H_v \left( \Sigma_{\partial h} B_h \Delta w + \Sigma_{\partial v} B_v \Delta w \right). \tag{183}
$$

Note that $\Delta w$ is a column vector, and $H_v, \Sigma_{\partial h},$ and $\Sigma_{\partial v}$ are diagonal matrices in (183). Thus, we can compute $F_3^T \Delta w$ efficiently using (183).
724. Aho, Pekka (2019) Automated state model extraction, testing and change detection through graphical user interface
726. Nuottila, Jouko (2019) Flexibility in agile projects: contracting practices and organisational arrangements
729. Saastamoinen, Ari (2020) Processing and microstructure of direct-quenched and tempered ultra-high strength steels
734. Mahmood, Shahriare (2020) The substantive approach for managing demand and supply sustainably in fashion industry
735. Lukić, Aleksa (2020) The effects of microalloying on the scale formation of AISI 304 stainless steel in walking beam furnace conditions
736. Ali, Samad (2020) Learning-based predictive resource allocation for machine type communications
737. Vuoksalu, Jarkko (2020) Energy efficient rendering of extensible 3D virtual environments on mobile devices

Book orders:
Virtual book store
http://verkkokauppa.juvenesprint.fi
Uditha Wijewardhana

SPARSE RECOVERY ALGORITHMS FOR STREAMING AND MULTIDIMENSIONAL SIGNALS