

RICE UNIVERSITY

**Compressive Sensing for Signal Ensembles**

by

**Marco F. Duarte**

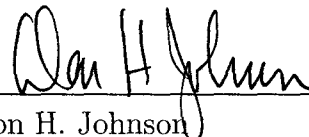
A THESIS SUBMITTED  
IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE

**Doctor of Philosophy**

APPROVED, THESIS COMMITTEE:



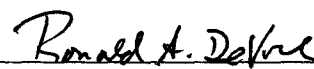
Richard G. Baraniuk, Chair  
Victor E. Cameron Professor of Electrical  
and Computer Engineering



Don H. Johnson  
J. S. Abercrombie Professor of Electrical  
and Computer Engineering



Wotao Yin  
Assistant Professor of Computational and  
Applied Mathematics



Ronald A. DeVore  
Walter E. Koss Professor of Mathematics  
Texas A&M University

Houston, Texas

July, 2009

UMI Number: 3421189

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3421189

Copyright 2010 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



ProQuest LLC  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106-1346

## ABSTRACT

### Compressive Sensing for Signal Ensembles

by

Marco F. Duarte

Compressive sensing (CS) is a new approach to simultaneous sensing and compression that enables a potentially large reduction in the sampling and computation costs for acquisition of signals having a sparse or compressible representation in some basis. The CS literature has focused almost exclusively on problems involving single signals in one or two dimensions. However, many important applications involve distributed networks or arrays of sensors. In other applications, the signal is inherently multidimensional and sensed progressively along a subset of its dimensions; examples include hyperspectral imaging and video acquisition. Initial work proposed joint sparsity models for signal ensembles that exploit both intra- and inter-signal correlation structures. Joint sparsity models enable a reduction in the total number of compressive measurements required by CS through the use of specially tailored recovery algorithms.

This thesis reviews several different models for sparsity and compressibility of

signal ensembles and multidimensional signals and proposes practical CS measurement schemes for these settings. For joint sparsity models, we evaluate the minimum number of measurements required under a recovery algorithm with combinatorial complexity. We also propose a framework for CS that uses a union-of-subspaces signal model. This framework leverages the structure present in certain sparse signals and can exploit both intra- and inter-signal correlations in signal ensembles. We formulate signal recovery algorithms that employ these new models to enable a reduction in the number of measurements required.

Additionally, we propose the use of Kronecker product matrices as sparsity or compressibility bases for signal ensembles and multidimensional signals to jointly model all types of correlation present in the signal when each type of correlation can be expressed using sparsity. We compare the performance of standard global measurement ensembles, which act on all of the signal samples; partitioned measurements, which act on a partition of the signal with a given measurement depending only on a piece of the signal; and Kronecker product measurements, which can be implemented in distributed measurement settings. The Kronecker product formulation in the sparsity and measurement settings enables the derivation of analytical bounds for transform coding compression of signal ensembles and multidimensional signals. We also provide new theoretical results for performance of CS recovery when Kronecker product matrices are used, which in turn motivates new design criteria for distributed CS measurement schemes.

### Acknowledgments

It has been a truly enlightening experience to work with so many talented and intelligent people here at Rice. The collaborative spirit that is fostered in the DSP group and the department is reflected in this work. Interacting with professors, fellow students, and external collaborators that know how to have fun and to enjoy the work we completed allowed me to better savor the successes and overcome the failures that always arise during research.

To begin with, I thank my collaborators whose significant input is reflected through this document: Dror Baron, Volkan Cevher, Chinmay Hegde, Shriram Sarvotham, and Michael Wakin. I thank my thesis committee, Don Johnson, Ronald DeVore, and Wotao Yin, for always providing useful perspective and comments to the work that I presented to them. I also thank additional collaborators whose work is not documented here: Petros Boufounos, Mark Davenport, Jason Laska, Justin Romberg, and Joel Tropp. My interactions with them helped me understand many technical details that improved my research. In particular, Kevin Kelly, Ting Sun, Dharmpal Takhar, and Yehia Massoud and his students helped us realize that compressive sensing was not just a neat mathematical theory. Finally, I thank my advisor, Richard Baraniuk, for providing an appropriate combination of encouragement, supervision, excitement, and support during my Ph.D. studies. My improvements as a researcher and technical writer in the past five years are without a doubt due to him. *Merci beaucoup.*

Many friends at Rice helped make the commute between home and work worth-

while: Albert, Andrea, Brian, Cansu, Chin, Chirag, Chris Rozell, Chris Steger, Courtney, David, Dharmpal, Dror, Eva, Fernando, Gareth, Hyeokho, Ilan, Jason, Jeff, Juan Pablo, Jyoti, Kevin, Kia, Kim, Laurent, Layla, Liz, Manjari, María, Marco, Mark, Marjan, Matthew, Melissa, Michael, Mike, Mona, Petros, Piotr, Prashant, Ray, Rich, Robin, Ron, Roy, Ryan, Sanda, Shri, Stephen, Tasos, Ting, Volkan, Wailam, William, and more. The friendliness and camaraderie between students, faculty, and staff is definitely one of the staples that make Rice unique!

Last but certainly not least, having the continuous support of my family and friends in Houston made things significantly easier throughout my stay: my parents, Jaime and Elizabeth, my siblings, Kike, Martha, and Laura (when she was able to visit); Douglas, Gloria, Jorge, Rafael, Roldán, and Sonia. Being surrounded by (medical) doctors and professionals gave me significant encouragement. You made sure that I took some time off work regularly, but were understanding and supportive when I had to decline your invitations. I also thank my uncle Miguel Duarte, and family friends Germán and Mildred Santamaría and Gonzalo Vargas for helping me through difficult times during my studies in the United States. *Gracias a todos.*

# Contents

Abstract	ii
List of Illustrations	xiii
List of Tables	xvii
<b>1 Introduction</b>	<b>1</b>
1.1 Compressive Sensing for Signal Ensembles and Multidimensional Signals	2
1.1.1 CS Measurement Matrices . . . . .	3
1.1.2 Sparsifying Matrices . . . . .	4
1.2 Scope . . . . .	6
1.3 Contributions . . . . .	9
1.4 Outline . . . . .	10
<b>2 Background</b>	<b>13</b>
2.1 Notation . . . . .	13
2.2 Low-Dimensional Signal Models . . . . .	13
2.2.1 Sparsity . . . . .	14
2.2.2 Finding Sparse Representations . . . . .	14
2.2.3 Transform Coding . . . . .	19
2.2.4 Sparse Approximation . . . . .	20

2.2.5	Compressibility . . . . .	23
2.2.6	Unions of Subspaces . . . . .	24
2.3	Compressive Sensing . . . . .	25
2.3.1	Restricted Isometry Property . . . . .	26
2.3.2	Restricted Isometry Property for Random Matrices . . . . .	27
2.3.3	Mutual Coherence . . . . .	28
2.3.4	Recovery Algorithms . . . . .	28
2.3.5	Performance Bounds on Signal Recovery . . . . .	30
2.4	Distributed Compressive Sensing . . . . .	35
2.4.1	Joint Sparsity Models . . . . .	35
2.4.2	Signal Ensemble Recovery Algorithms . . . . .	38
<b>3</b>	<b>Theoretical Measurement Bounds for Jointly Sparse Sig-</b>	
	<b>nals via Graphical Models</b>	<b>42</b>
3.1	Algebraic Framework . . . . .	43
3.1.1	Single Signal Case . . . . .	43
3.1.2	Multiple Signal Case . . . . .	44
3.2	Bound on Measurement Rates . . . . .	46
3.2.1	Graphical Model Framework . . . . .	47
3.2.2	Quantifying Redundancies . . . . .	48
3.2.3	Measurement Bounds . . . . .	50
3.2.4	Discussion . . . . .	52



<b>4</b>	<b>Distributed Compressive Sensing for Sensor Networks</b>	<b>54</b>
4.1	Related Work . . . . .	55
4.2	Distributed Sensing Using Random Projections . . . . .	56
4.2.1	Incoherent Measurements . . . . .	56
4.2.2	Communication to the Data Sink . . . . .	57
4.2.3	Joint Recovery . . . . .	58
4.3	Advantages of Distributed Compressive Sensing for Sensor Networks .	58
4.3.1	Simple, Universal Encoding . . . . .	58
4.3.2	Robustness, Progressivity, and Resiliency . . . . .	59
4.3.3	Security . . . . .	60
4.3.4	Fault Tolerance and Anomaly Detection . . . . .	60
4.3.5	Adaptivity to Channel Capacity . . . . .	61
4.3.6	Information Scalability . . . . .	61
4.4	Experiments . . . . .	63
4.4.1	Environmental Sensing . . . . .	64
4.4.2	Acoustic Sensing . . . . .	67
<b>5</b>	<b>Compressive Sensing for Wavelet-Sparse Signals</b>	<b>71</b>
5.1	The Structure of Multiscale Wavelet Transforms . . . . .	71
5.1.1	Deterministic Signal Models . . . . .	72
5.1.2	Probabilistic Signal Models . . . . .	74
5.2	Iterative Greedy Algorithms for Signal Recovery . . . . .	78
5.2.1	Tree Matching Pursuit and Tree Orthogonal Matching Pursuit	79

5.2.2	Experiments . . . . .	80
5.2.3	Extensions . . . . .	81
5.3	Optimization-Based Signal Recovery . . . . .	84
5.3.1	Iterative Reweighted $\ell_1$ -norm Minimization . . . . .	85
5.3.2	HMT-Based Weights for $\text{IR}\ell_1$ . . . . .	86
5.3.3	Experiments . . . . .	87
<b>6</b>	<b>Model-Based Compressive Sensing</b>	<b>91</b>
6.1	Structured Sparsity and Compressibility . . . . .	95
6.1.1	Structured Sparse Signals . . . . .	96
6.1.2	Model-Based RIP . . . . .	97
6.1.3	Structured Compressible Signals . . . . .	98
6.1.4	Nested Model Approximations and Residual Subspaces . . . . .	99
6.1.5	The Restricted Amplification Property (RAmP) . . . . .	100
6.2	Model-Based Signal Recovery Algorithms . . . . .	102
6.2.1	Model-based CoSaMP . . . . .	104
6.2.2	Performance of Structured Sparse Signal Recovery . . . . .	104
6.2.3	Performance of Structured Compressible Signal Recovery . . . . .	105
6.2.4	Robustness to Model Mismatch . . . . .	106
6.2.5	Computational Complexity of Model-Based Recovery . . . . .	108
6.3	Example: Wavelet Tree Model . . . . .	109
6.3.1	Tree-Sparse Signals . . . . .	109
6.3.2	Tree-Based Approximation . . . . .	110

6.3.3	Tree-Compressible Signals . . . . .	112
6.3.4	Stable Tree-Based Recovery From Compressive Measurements	113
6.3.5	Experiments . . . . .	114
6.4	Example: Block-Sparse Signals and Signal Ensembles . . . . .	121
6.4.1	Block-Sparse Signals . . . . .	122
6.4.2	Block-Based Approximation . . . . .	123
6.4.3	Block-Compressible Signals . . . . .	124
6.4.4	Stable Block-Based Recovery From Compressive Measurements	125
6.4.5	Experiments . . . . .	126
<b>7</b>	<b>Kronecker Product Compressive Sensing</b>	<b>136</b>
7.1	Stylized Applications . . . . .	138
7.1.1	Hyperspectral Imaging . . . . .	138
7.1.2	Video Acquisition . . . . .	138
7.1.3	Source Localization . . . . .	139
7.2	Background . . . . .	140
7.2.1	Tensor and Kronecker Products . . . . .	140
7.2.2	Signal Ensembles . . . . .	141
7.3	Kronecker Product Matrices for Multidimensional Compressive Sensing	142
7.3.1	Kronecker Product Sparsity Bases . . . . .	142
7.3.2	Kronecker Product Measurement Matrices . . . . .	143
7.3.3	Compressive Sensing Performance for Kronecker Product Matrices . . . . .	144

7.3.4	Extensions to multidimensional settings . . . . .	147
7.4	Case Study: CS with Multidimensional Wavelet Bases . . . . .	148
7.4.1	Isotropic, Anisotropic, and Hyperbolic Wavelets . . . . .	148
7.4.2	Isotropic Besov Spaces . . . . .	150
7.4.3	Anisotropic Besov Spaces . . . . .	152
7.4.4	Performance of Kronecker Product CS Recovery with Multidimensional Wavelet Bases . . . . .	155
7.5	Experimental Results . . . . .	157
7.5.1	Performance of Kronecker CS . . . . .	157
7.5.2	Hyperspectral Data . . . . .	158
7.5.3	Video Data . . . . .	161
7.5.4	Single-Pixel Hyperspectral Camera . . . . .	163
<b>8</b>	<b>Conclusions and Future Work</b>	<b>170</b>
8.1	Conclusions . . . . .	170
8.2	Future Work . . . . .	172
<b>A</b>	<b>Proof of Theorem 3.1</b>	<b>176</b>
<b>B</b>	<b>Proof of Lemma A.1</b>	<b>182</b>
<b>C</b>	<b>Proof of Theorem 3.2</b>	<b>185</b>
<b>D</b>	<b>Proof of Theorem 3.3</b>	<b>187</b>

<b>E Proof of Theorem 6.2</b>	<b>189</b>
<b>F Proof of Theorem 6.3</b>	<b>191</b>
<b>G Model-based Iterative Hard Thresholding</b>	<b>193</b>
<b>H Proof of Theorem 6.4</b>	<b>195</b>
<b>I Proof of Proposition 6.1</b>	<b>200</b>
<b>J Proof of Proposition 6.3</b>	<b>203</b>
<b>K Proof of Theorem 7.2</b>	<b>204</b>
<b>Bibliography</b>	<b>206</b>

## Illustrations

2.1	Example of transform coding . . . . .	20
2.2	Example of a compressible signal . . . . .	24
2.3	Set of sparse signals as a union of subspaces . . . . .	25
2.4	Joint recovery of synthetic JSM-2 signals . . . . .	40
3.1	Bipartite graphs for distributed compressive sensing . . . . .	49
4.1	Recovery of light intensity signals . . . . .	65
4.2	Recovery of humidity signals . . . . .	66
4.3	Recovery of temperature signals . . . . .	67
4.4	Quality of jointly sparse approximation of light intensity signals . . .	68
4.5	Recovery of example temperature signal . . . . .	69
4.6	Average SNR of recovered temperature signals . . . . .	69
4.7	Quality of jointly sparse approximation of vehicle audio signals . . .	70
4.8	Vehicle audio signals recovered via DCS . . . . .	70
5.1	Binary wavelet tree for a 1-D signal . . . . .	73

5.2	CS recovery of <i>Blocks</i> signal using several different algorithms . . . .	82
5.3	Performance of $\text{IR}\ell_1$ algorithm . . . . .	88
5.4	Example outputs for the recovery algorithms . . . . .	88
6.1	Example performance of model-based signal recovery for a piecewise smooth signal . . . . .	116
6.2	Performance of CoSaMP vs. tree-based recovery on a class of piecewise cubic signals . . . . .	117
6.3	Required overmeasuring factor $M/K$ to achieve a target recovery error for standard and tree-based recovery of piecewise cubic signals .	118
6.4	Robustness to measurement noise for standard and tree-based CS recovery algorithms . . . . .	119
6.5	Example performance of standard and quadtree-based recovery on images . . . . .	120
6.6	Example performance of model-based signal recovery for a block-sparse signal . . . . .	127
6.7	Example performance of model-based signal recovery for a block-compressible signal . . . . .	128
6.8	Performance of CoSaMP and model-based recovery on a class of block-sparse signals . . . . .	129
6.9	Example recovery of light intensity signal . . . . .	130

6.10	Performance of CoSaMP, DCS-SOMP and block-based recovery on a group of light signals . . . . .	132
6.11	Performance of CoSaMP, DCS-SOMP and block-based recovery on a group of humidity signals . . . . .	133
6.12	Performance of CoSaMP, DCS-SOMP and block-based recovery on a group of temperature signals . . . . .	133
6.13	Performance of CoSaMP and model-based distributed CoSaMP on a class of signals with common sparse supports . . . . .	135
7.1	Example capture from a single-pixel hyperspectral camera . . . . .	139
7.2	Example basis elements from 2-D wavelet bases . . . . .	150
7.3	Performance of Kronecker product CS measurements . . . . .	158
7.4	Examples of transform coding of a hyperspectral datacube . . . . .	164
7.5	Performance of Kronecker product sparsity basis in hyperspectral imaging . . . . .	165
7.6	Performance of Kronecker product sparsity and measurements matrices for hyperspectral imaging . . . . .	166
7.7	Performance of Kronecker product sparsity basis for transform coding of video sequences . . . . .	167
7.8	Performance of Kronecker product sparsity and measurements matrices for the <i>Foreman</i> video sequence . . . . .	168



7.9	Performance of Kronecker product sparsity and measurements	
	matrices for the <i>Akiyo</i> video sequence . . . . .	169
G.1	Example performance of model-based IHT . . . . .	194

## Tables

5.1	Computational complexity of CS algorithms . . . . .	80
6.1	Performance comparison for standard and model-based CoSaMP recovery on environmental sensing signals . . . . .	131

*Dedicada a mis abuelos, Marco, Dolores, Luis Felipe, y Paulina, y a mi padrino  
Aníbal con motivo de su septuagésimo cumpleaños.*

## Chapter 1

### Introduction

We are in the midst of a digital revolution that is driving the development and deployment of new sensing systems with ever increasing fidelity and resolution. The theoretical foundation of this revolution is the Shannon/Nyquist sampling theorem, which states that signals, images, videos, and other data can be exactly recovered from a set of uniformly-spaced samples taken at the Nyquist rate [1].

Unfortunately, in many important and emerging applications, the resulting Nyquist rate is so high that we end up with too many samples and must compress in order to store, process, or transmit them. To address this issue, we rely on *signal compression*, which aims to find the smallest representation of a signal that is able to achieve a target level of acceptable distortion, i.e., the lowest number of bits that describe the data contained in the signal. One of the most popular techniques for signal compression is known as *transform coding*, and relies on finding bases or frames that provide *sparse* or *compressible* representations for the class of signals of interest [2]. By a sparse representation, we mean that the signal has only  $K$  out of  $N$  nonzero coefficients, with  $K \ll N$ ; by a compressible representation, we mean that the magnitude of the signal coefficients, when sorted, have a power law decay with exponent  $-1/p$ ,  $p \leq 1$ . Both sparse and compressible signals can be compressed to high fidelity by preserving only the values and locations of the largest coefficients of the signal; in fact, many

data compression schemes like JPEG [3] and JPEG2000 [4] exploit signal sparsity and compressibility.

Leveraging the concept of transform coding, *compressive sensing* (CS) has emerged as a new framework for signal acquisition and sensor design that enables a potentially large reduction in the sampling and computation costs for sensing signals that have a sparse or compressible representation. CS builds on the work of Candès, Romberg, and Tao [5] and Donoho [6], who showed that a signal having a sparse or compressible representation in one basis can be recovered from projections onto a small set of *measurement vectors* that are *incoherent* with the sparsity basis, meaning that the representation of the measurement vectors in this basis is not sparse. CS acquisition devices perform *multiplexing* of the signal entries to calculate these inner products and obtain a compressed representation of the signal. *Random vectors* play a central role as a *universal* measurements in the sense that they are incoherent with any fixed basis with high probability.

## 1.1 Compressive Sensing for Signal Ensembles and Multidimensional Signals

The CS literature has mostly focused on problems involving single sensors, one-dimensional (1-D) signals, or 2-D images; however, many important applications that hold significant promise for CS involve signals that are inherently multidimensional. The coordinates of these signals may span several physical, temporal, or spectral dimensions. Additionally, the signals are often captured in a progressive fashion,

consisting of a sequence of captures corresponding to fixed values for a subset of the coordinates and are usually sensed progressively along a subset of the dimensions explored. Examples include hyperspectral imaging (with spatial and spectral dimensions), video acquisition (with spatial and temporal dimensions), and synthetic aperture radar imaging (with progressive acquisition in the spatial dimensions). Another class of promising applications for CS features distributed networks or arrays of sensors, including environmental sensors, microphone arrays, and camera arrays.

These properties of multidimensional data and the corresponding acquisition hardware complicate the design of both the measurement matrix  $\Phi$  and the sparsifying basis  $\Psi$  to achieve maximum efficiency in CS, as measured by the number of measurements needed to achieve a target amount of distortion.

### 1.1.1 CS Measurement Matrices

For signals and signal ensembles of any dimension, *global* CS measurements that multiplex all the entries of all signals are optimal, as they allow for the largest degree of randomization. However, such measurements require the use of multiple accumulators along all data dimensions for multidimensional signals, or a considerable amount of communication among the sensors for signal ensembles. In many multidimensional signal settings it can be difficult to implement such accumulators due to the large dimensionality of the signals and the partial availability of the data during acquisition. For example, each frame in a video sequence is available only for a limited time; therefore, a device that calculates global CS measurements would have to store the sum of the  $M$  partial inner products from each of the frames. Similarly, global CS

measurements of a hyperspectral datacube would require simultaneous multiplexing in the spectral and spatial dimensions. However, existing systems rely on separate spatial multiplexing using optical modulators [7] and spectral multiplexing using optical devices like prisms; this separate multiplexing nature limits the structure of the measurements obtained [8].

These limitations naturally point us in the direction of measurements that depend only on a subset of the entries of the multidimensional signal or, correspondingly, a single signal from the signal ensemble that we aim to acquire. In other words, we must use *partitioned measurements* obtained by processing only a portion of the multidimensional signal or a single signal from the ensemble at a time. For multidimensional signals, each portion usually corresponds to a snapshot of the signal along a given dimension, such as one frame of a video signal or the image of one spectral band out of a hyperspectral datacube.

### 1.1.2 Sparsifying Matrices

For multidimensional signals, we can often characterize multiple types of structures corresponding to different dimensions or coordinates. Therefore, there are many possible choices of sparsity or compressibility bases for this type of signals, as each different structure present can usually be captured using a representation in a corresponding basis. For example, each frame of a video sequence is compressible in a wavelet bases, as they correspond to images obtained at different time instants. Simultaneously, the structure of each pixel in the video sequence along the time dimension is often smooth and piecewise smooth, due to camera movement, object

motion and occlusion, illumination changes, etc. A similar situation is observed in hyperspectral signals, where we acquire the reflectivity values of a 2-D area under a range of spectral frequencies. The reflectivity values at a given frequency correspond to an image, with known structure; additionally, the spectral signature of a given pixel is usually smooth or piecewise smooth, depending on the spectral range and materials present in the observed area.

Initial work on sparsity and compressibility of multidimensional signals and signal ensembles [9–19] has provided new sparsity and compressibility models for multidimensional signals. These models consider sections of the multidimensional data (i.e., cuts corresponding to a fixed value for a subset of the coordinates) as separate signals, and pose correlation models between the values and locations of their sparse representations. The resulting models are rather limited in the types of structures admitted. For almost all of these models, theoretical guarantees on signal recovery using these models have only been provided either for strictly sparse signals, for noiseless measurement settings, or in asymptotic regimes. Additionally, almost all of these models are tied to ad-hoc recovery procedures: the algorithms has to be specifically tailored to the structure assumed. Clearly, it is necessary to pose more generic models for sparse and compressible multidimensional signal that allow us to leverage the CS framework to a higher degree of effective compression.

Fortunately, there are other immediately evident ways in which inter-signal correlations can be encoded. Ideally, we would formulate a sparsity or compressibility basis for the entire multidimensional signal that *simultaneously* accounts for all the



types of structure present in the data.

## 1.2 Scope

The goal of this thesis is to develop and study new models for sparsity and compressibility of signal ensembles and multidimensional signals. We elaborate on initial work on distributed CS that focuses on joint sparsity models, which encode simple correlation structures between the values and locations of sparse signal ensembles [9].

The distributed CS framework has been advocated mainly for sensor and camera network settings, where several signals that correspond to a single physical event are simultaneously recorded. Because all the signals describe the same event, we expect them to be correlated according to the physics that rule the signal dissemination. The signal ensemble setting can be generalized to a two-dimensional signal simply by arranging the signals into a matrix, where each column of the matrix corresponds to one of the recorded signals.

Algorithms for signal ensemble recovery from CS measurements in sensor networks have been proposed, but they often require significant communication between sensors. These algorithms obtain measurements at each sensor that depend on its sensed signal, which are then shared between the sensors using standard communication techniques in order to calculate measurements for the entire signal ensemble. Such techniques include intersensor gossiping [20] and random sensor probing [21]. Further study has been devoted to the sensing capacity of a sensor network under this sensing and compression model [22, 23]. On the other hand, Bajwa et al. [24] exploit CS for

joint measurement of a spatial sensor field at a single time instant. This approach uses matched source-channel communication [25] to significantly reduce the required power. Unfortunately, these algorithms neglect intra-sensor correlations – those between the samples of each signal – by performing compression of the data for each time instance separately. Furthermore, [24] requires both the deployment of sensors on a regular grid and a potentially complicated time and power synchronization of wireless transmitters among the nodes.

Limited prior work exists for CS of multidimensional signals, where the focus is on hyperspectral imaging data [10], video sequences [11–16], and confocal microscopy [17]. These formulations employ CS acquisition schemes that distribute the measurements among a set of pieces of the signal, with the signal partitioning corresponding to different values for one of the dimensions spanned by the signal. This setup is immediately applicable to several architectures for compressive sensors, including single-pixel video cameras [7, 11, 26] and compressive hyperspectral imagers, such as the coded aperture spectral snapshot imager [8] and the single-pixel hyperspectral camera [27]. While global measurements that depend on the entire set of data have been proposed [8, 11, 17], practical architectures that provide such measurements are rare [8].

Several frameworks have been proposed to encode the structure of multidimensional signals using sparsity. The most significant class of structures link the signals through overlap of nonzero coefficient values and locations [13, 15]. These types of matrices are very rigid in the kinds of structures that can be represented. Standard

sparsity bases for CS, such as multidimensional isotropic wavelets, suffice only for very specific classes of signals [11, 17]. In other cases, specialized compression bases are combined with specially tailored recovery algorithms [10, 13, 16].

For both signal ensemble and multidimensional signal applications, we propose in this thesis the use of sparsity bases and CS measurement matrices that can be expressed as Kronecker products. Kronecker product bases for compressibility enable the simultaneous expression of different structures on each dimension spanned by the signal, while Kronecker product CS measurement matrices are well suited for distributed sensing due to the resulting two-stage implementation, as detailed in the sequel.

Kronecker product matrices have been proposed as an alternative sparsity and compressibility basis in specific cases for spatiotemporal signals [12, 14]. Kronecker product representations have also been proposed for transform coding compression of hyperspectral datacubes, although they have relied on linear approximations using principal component analysis and Karhunen-Loève transforms rather than sparse representations [28, 29]. Thus, the approaches are data-dependent and difficult to generalize among different datasets.

There have been very recent initial studies on the properties of Kronecker product matrices for CS [30, 31]. A study of their coherence properties [30] is repeated in this thesis, with a more intuitive proof formulation. Additionally, while [31] provides a lower bound for their restricted isometry constants, we provide in this thesis a tighter lower bound and a new upper bound for the restricted isometry constants based on

the properties of the eigendecomposition of their submatrices. Kronecker product matrices have also been proposed for CS due to their computational efficiency [30].

### 1.3 Contributions

The contributions of this thesis include:

- the *formulation of new theoretical bounds* on the minimum number of measurements required per signal for signal ensemble recovery from distributed CS measurements;
- the *design of new recovery algorithms* for structured sparse signals and jointly sparse signal ensembles that rely on the union-of-subspaces model formalism;
- the analysis of these new structured sparse signal recovery algorithms to *provide performance guarantees*, as related to the distortion of the recovered signal and the number of randomized measurements required for recovery;
- the *formulation of Kronecker product matrices* as bases to achieve sparse and compressible representations of multidimensional signals and as measurement matrices that can be easily implemented in distributed CS settings; and
- the *analysis of CS performance metrics* when Kronecker product matrices are used to obtain sparse or compressible representations, and to obtain CS measurement matrices that are easily implementable in distributed CS settings.

## 1.4 Outline

This thesis is organized as follows.

**Chapter 2** introduces notation and overviews concepts in low-dimensional signal models including sparsity, compressibility, and unions of subspaces. We review applications of sparsity in signal compression and processing. We also cover algorithms for sparse approximation and existing performance guarantees. Additionally, we give a basic review of CS, including quality metrics for measurement matrices, signal recovery algorithms, and their corresponding guarantees. We end this chapter with a brief review of distributed CS, including joint sparsity models and the corresponding ad-hoc signal recovery algorithms.

**Chapter 3** describes an analysis on the theoretical bounds of distributed CS; we describe an extension of the sparsest representation of a signal to the jointly sparsest representation of a signal ensemble. We then provide a theoretical result on the smallest number of randomized measurements per sensor that suffices for recovery of the signal ensemble when a combinatorially complex algorithm is used. Our recovery algorithm closely resembles  $\ell_0$ -norm minimization, which features the theoretically lowest bounds on number of measurements required for signal recovery.

**Chapter 4** provides detail on a practical implementation of distributed CS in a sensor network where measurements are calculated independently by each sensor and then sent to a central processing unit that performs signal recovery. We highlight the desirable properties that distributed CS has for this application and provide a set of experimental results that uses real-world data to demonstrate these advantages. We

also elaborate on applications of distributed CS to some distributed signal processing tasks involving linear operations.

**Chapter 5** describes initial work on the use of structure in a sparse representation to improve the performance of CS through the design of specially tailored recovery algorithms. Our focus is on the class of piecewise smooth signals, which have a very succinct structure for the values and locations of the nonzero coefficients in a suitable wavelet transform. We provide algorithms that exploit both deterministic and probabilistic models for the signal coefficients and present experimental evidence of the advantages afforded by the use of this augmented signal model.

**Chapter 6** builds on the work in Chapter 5 by presenting a theoretical and algorithmic framework for the use of structured sparse signal models in CS, which relies on a union-of-subspaces formalism. The union-of-subspaces model for a signal can capture a variety of structures for the locations of a signal's nonzero coefficients, reducing the number of randomized measurements required for signal recovery. We formulate recovery algorithms that exploit the structure, together with guarantees on the quality of the recovery and bounds on the number of measurements required by these guarantees. While we apply this new framework to the piecewise smooth signals of Chapter 5, we also extend the framework to signal ensembles with common sparse supports; we present experimental results that validate the advantages of these models both for synthetic datasets and for the real-world data used in Chapter 4.

**Chapter 7** addresses extensions of sparsity and compressibility concepts for multidimensional signals. We propose the use of Kronecker products of individual bases

that achieve sparsity or compressibility of sections of the multidimensional signal across a subset of its dimensions. We compare the performance of Kronecker bases for multidimensional compression and standard bases for lower-dimensional, partitioned compression of signals that are sparse or compressible in a wavelet basis. We also show that the distributed measurement setups advocated in Chapters 3 and 4 can be expressed as Kronecker products, and provide results on the metrics of CS measurement matrices that are obtained using a Kronecker product formulation. Additionally, we provide experimental evidence that shows the tradeoffs that arise when Kronecker product matrices are used in transform coding and CS.

Finally, we conclude with a summary of our findings and a discussion of ongoing work in **Chapter 8**.

The research consigned in this thesis is the result of several intensive collaborations. The first page of each chapter contains a footnote identifying the collaborators that share credit for the respective work.

## Chapter 2

### Background

#### 2.1 Notation

We denote vectors by bold lower case letters ( $\mathbf{x}$ ,  $\mathbf{y}$ ), with the vector entries listed as  $\mathbf{x} = [\mathbf{x}(1) \ \mathbf{x}(2) \ \dots]$ . Matrices are denoted by bold upper case letters ( $\mathbf{W}$ ) and their entries are indexed similarly as  $\mathbf{W}(i, j)$ . Scalar quantities are denoted by upper case letters ( $K$ ,  $M$ ,  $N$ ), and running indices are denoted by the corresponding lower case letters ( $k$ ,  $m$ ,  $n$ ). Most calligraphic letters denote sets ( $\mathcal{M}$ ,  $\mathcal{T}$ ), and most double-barred letters denote operators acting on vectors ( $\mathbb{M}$ ,  $\mathbb{T}$ ). To keep the notation interesting, we let greek letters denote vectors, matrices, constants, and other structures.

#### 2.2 Low-Dimensional Signal Models

Through this thesis, we will use a series of models for finite-dimensional signals  $\mathbf{x} \in \mathbb{R}^N$ . These models are inspired by compression applications, where we desire that most of the signal's energy be captured in a representation of small size.

Given a basis  $\{\psi_i\}_{i=1}^N$  for  $\mathbb{R}^N$ , we can represent every signal  $\mathbf{x} \in \mathbb{R}^N$  in terms of  $N$  coefficients  $\{\theta_i\}_{i=1}^N$  as  $\mathbf{x} = \sum_{i=1}^N \psi_i \theta_i$ ; arranging the  $\psi_i$  as columns into the  $N \times N$  matrix  $\Psi$  and the coefficients  $\theta_i$  into the  $N \times 1$  *coefficient vector*  $\theta$ , we can



write succinctly that  $\mathbf{x} = \Psi\theta$ , with  $\theta \in \mathbb{R}^N$ . Similarly, if we use a full-rank frame  $\Psi$  containing  $N$  column vectors of length  $L$  with  $L < N$  (i.e.,  $\Psi \in \mathbb{R}^{L \times N}$ ), then for any vector  $\mathbf{x} \in \mathbb{R}^L$  there exist infinitely many decompositions  $\theta \in \mathbb{R}^N$  such that  $\mathbf{x} = \Psi\theta$ .

### 2.2.1 Sparsity

We say a signal  $\mathbf{x}$  is *K-sparse* in the basis or frame  $\Psi$  if there exists a vector  $\theta \in \mathbb{R}^N$  with only  $K \ll N$  nonzero entries such that  $\mathbf{x} = \Psi\theta$ . We call the set of indices corresponding to the nonzero entries the *support* of  $\theta$  and denote it by  $\text{supp}(\theta)$ .

The use of sparsity as a model for signal processing dates back to Donoho and Johnstone's initial works in the early 1990s [32–34], where wavelet-sparse signals and images were denoised by assuming that the noiseless version of the signal is sparse (see Section 2.2.4 for a review). Since then, many mathematicians, applied mathematicians, and statisticians have employed sparse signal models for applications that include signal enhancement and superresolution, signal deconvolution, and signal denoising [2]. The foundations of this work are detailed in the next few subsections.

### 2.2.2 Finding Sparse Representations

It is useful to determine whether a signal has a sparse representation in a given basis or frame. If an orthonormal basis  $\Psi$  is used, then a signal  $\mathbf{x}$  has a unique representation  $\theta = \Psi^{-1}\mathbf{x}$  and we can learn whether  $\mathbf{x}$  is *K-sparse* in  $\theta$  simply by inspecting this vector. When  $\Psi$  is a frame, however, there are infinitely many representations  $\theta$  for  $\mathbf{x}$ , making it more difficult to answer this question. Several algorithms have been proposed to obtain sparse representations for a signal  $\mathbf{x}$  in a frame  $\Psi$ .

### $\ell_0$ -norm Minimization

The most intuitive algorithm proceeds by finding the sparsest representation of a signal  $\mathbf{x}$  in a frame  $\Psi$ . It can be formalized by employing the  $\ell_0$  “norm”<sup>1</sup>, defined as the number of nonzero entries of the vector it operates on. Then the aforementioned algorithm can be expressed as

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^N} \|\theta\|_0 \text{ subject to } \mathbf{x} = \Psi\theta. \quad (2.1)$$

While this algorithm will – by construction – find the sparsest representation of the signal  $\mathbf{x}$  in the frame  $\Psi$ , its computational complexity is combinatorial; it must search whether the signal  $\mathbf{x}$  is in the span of any of the columns of  $\Psi$ , then whether it is in the span of any pair of columns of  $\Psi$ , then repeat for any set of three columns, etc., until a combination of columns for which  $\mathbf{x}$  is in their span is found.

### $\ell_1$ -norm Minimization

As a convex relaxation of (2.1), Chen, Donoho and Saunders [35] proposed the use of the  $\ell_1$  norm, defined as  $\|\theta\|_1 = \sum_{n=1}^N |\theta(n)|$ . This relaxation, known as *basis pursuit* (BP), is formally defined as

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^N} \|\theta\|_1 \text{ subject to } \mathbf{x} = \Psi\theta. \quad (2.2)$$

Thanks to the convex relaxation, this algorithm can be implemented as a linear program, making its computational complexity polynomial in the signal length.

---

<sup>1</sup>Although this is known as the  $\ell_0$  “norm”, it is not a real norm as it does not have the subadditivity property.

### Greedy Algorithms

As an alternative to optimization-based algorithms, there exist greedy algorithms that can find sparse representations. These algorithms are iterative in nature and select columns of  $\Psi$  according to their correlation with the relevant signal  $\mathbf{x}$ , as measured by the inner product.

The matching pursuit algorithm (MP) [36] proceeds by finding the column of  $\Psi$  most correlated to the signal residual, which is obtained by subtracting the contribution of previously selected columns from the original signal. The algorithm is formally defined as Algorithm 1, where  $\mathfrak{T}(\theta, K)$  denotes a *thresholding* operator on  $\theta$  that sets all but the  $K$  entries of  $\theta$  with the largest magnitudes to zero. The halting criterion used to find sparse representations consists of checking whether  $\mathbf{x} = \Psi\theta$ .

While the MP algorithm is computationally efficient and often features good performance, there are specific cases in which frames  $\Psi$  can be constructed that defeat the algorithm [37] by preventing convergence. Such a flaw is manifested, for example, when the algorithm selects a cycle of columns of  $\Psi$  that are highly coherent to correct for an overcompensation made by a certain column.

As an alternative, the orthogonal matching pursuit algorithm (OMP) [37, 38] has been proposed. The algorithm is modified as shown in Algorithm 2, where we let  $\Psi_\Omega$  denote the restriction of the matrix  $\Psi$  to the columns corresponding to the index set  $\Omega \subseteq \{1, \dots, N\}$ . The residual is obtained by subtracting the projection of the signal  $\mathbf{x}$  into the span of the previously selected columns. While OMP does not suffer the aforementioned flaw, it is penalized in its computational complexity by the calculation

---

**Algorithm 1** Matching Pursuit
 

---

Inputs: Sparsifying frame  $\Psi$ , signal  $\mathbf{x}$

Outputs: Sparse representation  $\hat{\theta}$

initialize:  $\hat{\theta}_0 = 0$ ,  $\mathbf{r} = \mathbf{x}$ ,  $i = 0$ .

**while** halting criterion false **do**

1.  $i \leftarrow i + 1$

2.  $\mathbf{b} \leftarrow \Psi^T \mathbf{r}$       {form residual signal estimate}

3.  $\hat{\theta}_i \leftarrow \hat{\theta}_{i-1} + \mathfrak{T}(\mathbf{b}, 1)$     {update largest magnitude coefficient}

4.  $\mathbf{r} \leftarrow \mathbf{r} - \Psi \mathfrak{T}(\mathbf{b}, 1)$     {update measurement residual}

**end while**

return  $\hat{\theta} \leftarrow \hat{\theta}_i$

---

of the pseudoinverse, defined and denoted as  $\mathbf{W}^\dagger = \mathbf{W}^T(\mathbf{W}\mathbf{W}^T)^{-1}$ .

### Algorithmic Performance

To provide a guarantee for the performance of these algorithms, we define a metric of the frame  $\Psi$  known as *coherence*.

**Definition 2.1** [39, 40] *The coherence of a frame  $\Psi$ ,  $\mu(\Psi)$ , is the largest absolute inner product between any two columns of  $\Psi$ :*

$$\mu(\Psi) = \max_{1 \leq i, j \leq N} |\langle \psi_i, \psi_j \rangle|.$$

---

**Algorithm 2** Orthogonal Matching Pursuit

---

Inputs: Sparsifying frame  $\Psi$ , signal  $\mathbf{x}$

Outputs: Sparse representation  $\hat{\theta}$

Initialize:  $\hat{\theta}_0 = 0$ ,  $\mathbf{r} = \mathbf{x}$ ,  $\Omega = \emptyset$ ,  $i = 0$ .

**while** halting criterion false **do**

1.  $i \leftarrow i + 1$

2.  $\mathbf{b} \leftarrow \Psi^T \mathbf{r}$  {form residual signal estimate}

3.  $\Omega \leftarrow \Omega \cup \text{supp}(\mathfrak{T}(\mathbf{b}, 1))$  {add index of residual's largest magnitude entry  
to signal support}

4.  $\hat{\theta}_i|_{\Omega} \leftarrow \Psi_{\Omega}^{\dagger} \mathbf{x}$ ,  $\hat{\theta}_i|_{\Omega^c} \leftarrow 0$  {form signal estimate}

5.  $\mathbf{r} \leftarrow \mathbf{x} - \Psi \hat{\theta}_i$  {update measurement residual}

**end while**

return  $\hat{\theta} \leftarrow \hat{\theta}_i$

---

The coherence then dictates the maximum sparsity  $\|\theta\|_0$  for which the BP and OMP algorithms obtain the sparse representation of  $\mathbf{x} = \Psi\theta$ :

**Theorem 2.1** [39, 40] *The BP and OMP algorithms can obtain the sparse representation of any  $K$ -sparse signal in  $\Psi$  if*

$$K < \frac{1}{2} \left( \frac{1}{\mu(\Psi)} + 1 \right). \quad (2.3)$$

The success of these algorithms, however, depends on the existence of a unique sparsest representation. Uniqueness can be guaranteed by defining a relevant metric:

**Definition 2.2** [39] *The spark of a matrix  $\Psi$ ,  $\sigma = \text{spark}(\Psi)$ , is the smallest number  $\sigma$  such that there exists a set of  $\sigma$  columns of  $\Psi$  that are linearly dependent.*

We then obtain the following guarantee for uniqueness.

**Theorem 2.2** [39] *If a signal  $\mathbf{x}$  has a sparse representation  $\mathbf{x} = \Psi\theta$  with  $\|\theta\|_0 = K$  and*

$$K < \text{spark}(\Psi)/2, \quad (2.4)$$

*then  $\theta$  is the unique sparsest representation of  $\mathbf{x}$  in  $\Psi$ .*

### 2.2.3 Transform Coding

Sparse representations are the core tenet of compression algorithms based on *transform coding*. In transform coding, a sparse signal  $\mathbf{x}$  is compressed by obtaining its sparse representation  $\theta$  in a suitable basis or frame  $\Psi$  and encoding the values and locations of its nonzero coefficients. For a  $K$ -sparse signal, this type of compression requires  $\mathcal{O}(K \log N)$  bits. Transform coding is the foundation of most commercial compression algorithms; examples include the JPEG image compression algorithm, which uses the discrete cosine transform [3], and the JPEG2000 algorithm, which uses the discrete wavelet transform [4]. An example using wavelets on an image is shown in Figure 2.1.<sup>2</sup>

---

<sup>2</sup>We use the Daubechies-8 wavelet through this thesis, except when noted.

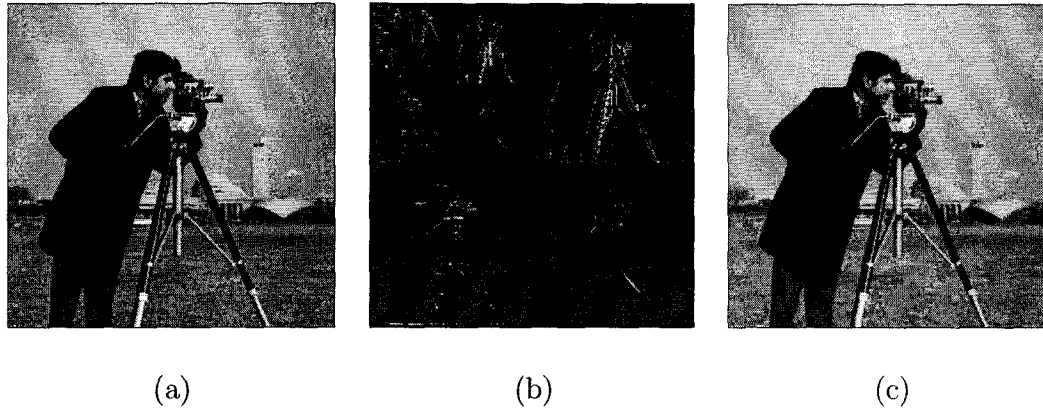


Figure 2.1 : *Example of transform coding. (a) Cameraman image,  $256 \times 256$  pixels. (b) Wavelet transform of (a); each pixel represents one wavelet coefficient. While all coefficients are nonzero, most are very small, represented by blue pixels. (c) Sparse approximation of (a) with only 6000 nonzero coefficients. Since most of the wavelet coefficients are small, the distortion of this sparse approximation is very low.*

#### 2.2.4 Sparse Approximation

While transform coding algorithms are able to encode sparse signals without distortion, the signals that we observe in nature are not exactly sparse. In other cases, our observations of the sparse signal are corrupted by additive noise. Therefore, we need to find the best sparse approximation to the signal in order to feed it to the transform coding compression algorithm. In this way, we achieve the lowest possible distortion for the compressed and/or denoised version of the signal. Our baseline for performance is the lowest distortion we can achieve by selecting  $K$  coefficients, which we call the *best  $K$ -term approximation error*:

$$\sigma_{\Psi}(\mathbf{x}, K) = \min_{\|\theta\|_0 \leq K} \|\mathbf{x} - \Psi\theta\|_2. \quad (2.5)$$

We say then that the optimal  $K$ -sparse approximation of  $\mathbf{x}$  in the basis or frame  $\Psi$  is the sparse vector  $\theta_K$  that achieves this distortion, i.e.,  $\|\mathbf{x} - \Psi\theta_K\|_2 = \sigma_\Psi(\mathbf{x}, K)$ .

When  $\Psi$  is an orthonormal basis, the optimal sparse approximation is obtained through thresholding, as defined in Section 2.2.2. When  $\Psi$  is a frame instead, we can immediately obtain two algorithms to find a suitable sparse approximation: the first one finds the sparsest approximation within a certain distortion level, and the second one finds the signal with lowest distortion that has a certain sparsity. These two formulations are formalized as

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^N} \|\theta\|_0 \text{ subject to } \|\mathbf{x} - \Psi\theta\|_2 \leq \epsilon \quad (2.6)$$

and

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^N} \|\mathbf{x} - \Psi\theta\|_2 \text{ subject to } \|\theta\|_0 \leq K. \quad (2.7)$$

These two minimizations become the same unconstrained optimization under a Lagrangian relaxation:

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^N} \|\theta\|_0 + \lambda \|\mathbf{x} - \Psi\theta\|_2. \quad (2.8)$$

Similarly to Section 2.2.2, these optimizations have convex relaxations using the  $\ell_1$  norm that result in some well-known algorithms; the first algorithm is relaxed as

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^N} \|\theta\|_1 \text{ subject to } \|\mathbf{x} - \Psi\theta\|_2 \leq \epsilon, \quad (2.9)$$

and we dub it *basis pursuit with inequality constraints* (BPIC). It can be solved by a quadratic or cone program. The second one, known as the *Lasso* [41], is formalized as

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^N} \|\mathbf{x} - \Psi\theta\|_2 \text{ subject to } \|\theta\|_1 \leq \delta, \quad (2.10)$$



for which fast solvers have been proposed when  $K$  is small [42]. The Lagrangian relaxation is known as *Basis Pursuit Denoising* (BPDN) [35]:

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^N} \|\theta\|_1 + \lambda \|\mathbf{x} - \Psi\theta\|_2; \quad (2.11)$$

it can be solved using iterative thresholding algorithms [43, 44]. The MP and OMP greedy algorithms can also be adapted to this setting, by changing their halting criterion to  $\|\mathbf{x} - \Psi\theta\|_2 \leq \epsilon$ .

We can also provide guarantees for the sparse approximations obtained from this algorithms. However, we will require a more complex metric for the frame.

**Definition 2.3** *The cumulative coherence of a frame  $\Psi$  is defined as*

$$\mu(\Psi, m) = \max_{1 \leq i \leq N, \Omega \subseteq \{1, \dots, N\}, |\Omega|=m} \sum_{j \in \Omega} |\langle \psi_i, \psi_j \rangle|.$$

We note that  $\mu(\Psi, 1) = \mu(\Psi)$  and  $\mu(\Psi, m) \leq m\mu(\Psi)$ . We then obtain the following guarantees.

**Theorem 2.3** [40] *If the frame  $\Psi$  has  $\mu(\Psi, K) \leq 1/3$ , then the OMP algorithm's approximation of the signal representation after  $K$  iterations  $\hat{\theta}_K$  to a signal  $\mathbf{x}$  using the frame  $\Psi$  obeys*

$$\|\mathbf{x} - \Psi\hat{\theta}_K\|_2 \leq \sqrt{1 + 6K}\sigma_\Psi(\mathbf{x}, K).$$

Algorithms that provide sparse approximation distortion bounds that are proportional to the distortion of the best sparse approximation are said to be *instance optimal*. For approximation of sparse signals embedded in noise, the guarantees depend on the constraint constants and the magnitude of the noise.

**Theorem 2.4** [45] *If the observed signal  $\mathbf{x} = \Psi\theta$  is  $K$ -sparse in  $\Psi$ , with  $K$  obeying (2.3), and it is corrupted by a noise vector of magnitude  $\delta$ , then the BPIC algorithm's approximation  $\hat{\theta}$  obeys*

$$\|\theta - \hat{\theta}\|_2 \leq \frac{\delta + \epsilon}{\sqrt{1 - \mu(\Psi)(4K - 1)}},$$

*while the OMP algorithm's approximation obeys*

$$\|\theta - \hat{\theta}\|_2 \leq \frac{\delta}{\sqrt{1 - \mu(\Psi)(K - 1)}}$$

*provided that  $\delta = \epsilon \leq A(1 - \mu(\Psi)(2K - 1))/2$  for OMP.*

### 2.2.5 Compressibility

The amount of compression that we apply to a signal is dependent on the number of coefficients of  $\theta$  that we keep, i.e., to the  $\ell_0$  norm of the signal's sparse approximation. To that end, we want to quantify the benefit in reduced distortion to adding more coefficients to the compressed version of the signal.

Consider a signal  $\mathbf{x}$  whose coefficients  $\theta$ , when sorted in order of decreasing magnitude, decay according to the power law

$$|\theta(\mathcal{I}(n))| \leq S n^{-1/r}, \quad n = 1, \dots, N, \quad (2.12)$$

where  $\mathcal{I}$  indexes the sorted coefficients. Thanks to the rapid decay of their coefficients, such signals are well-approximated by  $K$ -sparse signals. The best  $K$ -term approximation error for such a signal obeys

$$\sigma_{\Psi}(\mathbf{x}, K) \leq (rs)^{-1/2} SK^{-s}, \quad (2.13)$$

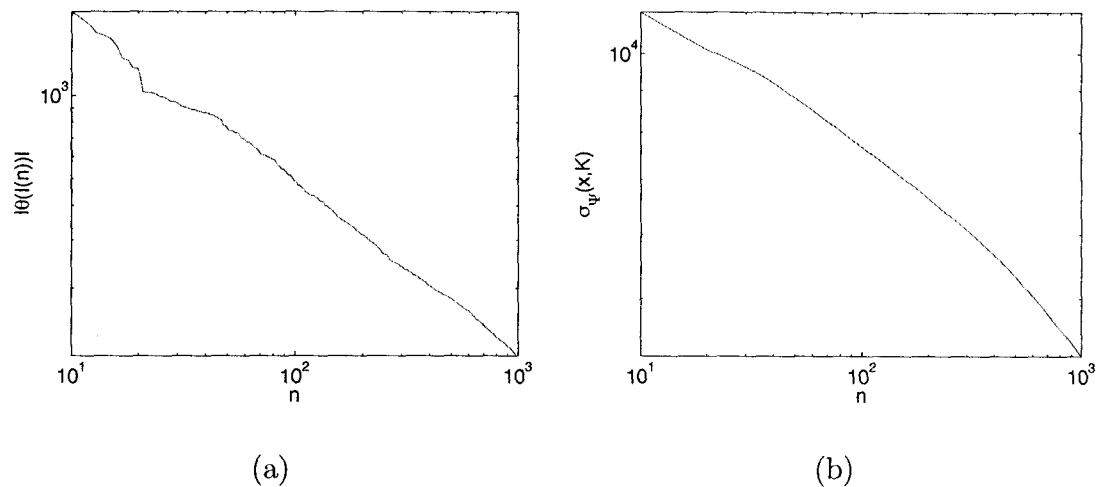


Figure 2.2 : Example of a compressible signal. (a) Wavelet coefficients of the Cameraman image from Figure 2.1(a) sorted by magnitude. The slope of this plot corresponds to the exponent of the decay of the coefficient magnitudes  $p$ . (b) Error of the optimal  $K$ -sparse approximation of the Cameraman image as a function of  $K$ . The slope of this plot corresponds to the exponent of the decay of the optimal sparse approximation error  $s$ .

with  $s = \frac{1}{r} - \frac{1}{2}$ . That is, the signal's best approximation error has a power-law decay with exponent  $s$  as  $K$  increases. We dub such a signal *s-compressible*. An example is shown in Figure 2.2.

### 2.2.6 Unions of Subspaces

There is a geometric interpretation for sparse signal representations. Each coefficient vector  $\theta \in \mathbb{R}^N$  corresponds to a point in  $N$ -dimensional space with coordinates given by the entries of  $\theta$ . Consider now all  $K$ -sparse representations that share the locations of the nonzero coefficients. The corresponding points form a  $K$ -dimensional subspace spanned by the canonical basis vectors for the  $K$  dimensions of the  $N$ -dimensional space corresponding to the support of  $\theta$ . Therefore, the set of all  $K$ -sparse signals

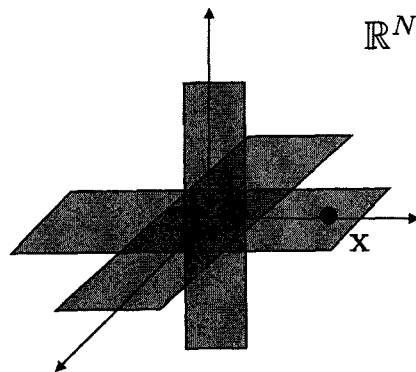


Figure 2.3 : The set  $\Sigma_K \subseteq \mathbb{R}^N$  contains all  $K$ -sparse signals and can be expressed as a union of subspaces. We illustrate an example with  $N = 3$  and  $K = 2$ .

can be described as the union of  $\binom{N}{K}$  orthogonal canonical subspaces of dimension  $K$ , each corresponding to a different possible supports for a  $K$ -sparse signals. We denote this union of subspaces by  $\Sigma_K$ , with an example illustrated in Figure 2.3.

When we use a non-canonical sparsifying basis  $\Psi$ , the points corresponding to  $K$ -sparse signals are contained in a union of orthogonal non-canonical subspaces that correspond to a rotation of the canonical case. When  $\Psi$  is a frame, then the points corresponding to  $K$ -sparse signals is a union of non-canonical non-orthogonal subspaces, with each subspace corresponding to the span of  $K$  column vectors from  $\Psi$ .

## 2.3 Compressive Sensing

While a widely accepted standard, the sample-then-compress ideology behind transform coding compression suffers from three inherent inefficiencies: First, we must start with a potentially large number of samples  $N$  even if the ultimate desired  $K$  is

small. Second, the encoder must compute all of the  $N$  transform coefficients  $\theta$ , even though it will discard all but  $K$  of them. Third, the encoder faces the overhead of encoding the locations of the large coefficients.

Compressive sensing (CS) integrates the signal acquisition and compression steps into a single process [5, 6, 46–53]. In CS we do not acquire  $\mathbf{x}$  directly but rather acquire  $M < N$  linear measurements  $\mathbf{y} = \Phi\mathbf{x} = \Phi\Psi\theta$  using an  $M \times N$  measurement matrix  $\Phi$ . We then recover  $\mathbf{x}$  by exploiting its sparsity or compressibility. Our goal is to push  $M$  as close as possible to  $K$  in order to perform as much signal “compression” during acquisition as possible. Clearly, the number of measurements  $M > K$ ; the combinatorially complex  $\ell_0$ -norm minimization algorithm (2.1) can achieve recovery in certain cases for  $M \geq K + 1$ . Our goal, therefore, is to find computationally feasible algorithms that can get as close to this bound as possible.

To recover the signal representation  $\theta$  from its measurements  $\mathbf{y}$ , we can exploit the fact that  $\mathbf{y}$  will be sparse in the frame  $\Phi\Psi$ . However, a distinguishing feature of CS is that we do not want to find just a sparse representation of  $\mathbf{y}$ , but rather we aim for the *correct representation*  $\theta$  that yields our data  $\mathbf{x} = \Psi\theta$ . Therefore, the requirements, guarantees, and algorithms relevant to CS signal recovery are slightly different from, although clearly based on, the sparse representation and approximation algorithms of Section 2.2. For brevity, we define the matrix product  $\Upsilon = \Phi\Psi$ , so that  $\mathbf{y} = \Upsilon\theta$ .

### 2.3.1 Restricted Isometry Property

In order to recover a good estimate of  $\theta$  (the  $K$   $\theta(n)$ ’s with largest magnitudes, for example) from the  $M$  compressive measurements, the matrix  $\Upsilon$  should satisfy the

restricted isometry property (RIP).

**Definition 2.4** [5] A matrix  $\Upsilon$  has the  $K$ -restricted isometry property ( $K$ -RIP) with constant  $\delta_K$  if, for all  $\theta \in \Sigma_K$ ,

$$(1 - \delta_K)\|\theta\|_2^2 \leq \|\Upsilon\theta\|_2^2 \leq (1 + \delta_K)\|\theta\|_2^2. \quad (2.14)$$

In words, the  $K$ -RIP ensures that all submatrices of  $\Upsilon$  of size  $M \times K$  are close to an isometry, and therefore distance (and information) preserving. Practical recovery algorithms typically require that  $\Upsilon$  have a slightly stronger  $2K$ -RIP,  $3K$ -RIP, or higher-order RIP in order to preserve distances between  $K$ -sparse vectors (which are  $2K$ -sparse in general), three-way sums of  $K$ -sparse vectors (which are  $3K$ -sparse in general), and other higher-order structures. In fact, the uniqueness requirement (2.4) is implied when the matrix has the  $2K$ -RIP with  $\delta_{2K} > 0$  as this implies that all sets of  $2K$  columns be linearly independent, putting  $\text{spark}(\Upsilon) > 2K$ .

### 2.3.2 Restricted Isometry Property for Random Matrices

While checking whether a measurement matrix  $\Phi$  satisfies the  $K$ -RIP is an NP-complete problem in general [5], random matrices whose entries are independent and identically distributed (i.i.d.) Gaussian, Rademacher ( $\pm 1$ ), or more generally subgaussian<sup>3</sup> work with high probability provided  $M = \mathcal{O}(K \log(N/K))$  [50, 55].

---

<sup>3</sup>A random variable  $X$  is called subgaussian if there exists  $c > 0$  such that  $\mathbb{E}(e^{Xt}) \leq e^{c^2 t^2/2}$  for all  $t \in \mathbb{R}$ . Examples include the Gaussian, Bernoulli, and Rademacher random variables, as well as any bounded random variable [54].

These random matrices also have a so-called *universality* property in that, for any choice of orthonormal basis  $\Psi$ ,  $\Phi\Psi$  has the  $K$ -RIP with high probability. This is useful when the signal is sparse in a non-canonical basis  $\Psi$ . A random measurement matrix  $\Phi$  corresponds to an intriguing data acquisition protocol in which each measurement  $\mathbf{y}(m)$  is a randomly weighted linear combination of the entries of  $\mathbf{x}$ .

### 2.3.3 Mutual Coherence

In particular cases, the choice of measurements that can be taken from the signal are limited to a transformation, such as the Fourier/Radon transform performed in magnetic resonant imaging. Thus, we can assume that a basis  $\Phi \in \mathbb{R}^{N \times N}$  is provided for measurement purposes, and we can choose a subset of the signal's coefficients in this transform as measurements. That is, let  $\bar{\Phi}$  be an  $N \times M$  submatrix of  $\Phi$  that preserves the basis vectors with indices  $\Gamma$  and  $\mathbf{y} = \bar{\Phi}^T \mathbf{x}$ . Under this setup, a different metric arises to evaluate the performance of CS.

**Definition 2.5** [56] *The mutual coherence of the  $N$ -dimensional orthonormal bases  $\Phi$  and  $\Psi$  is the maximum absolute value for the inner product between elements of the two bases:*

$$\mu(\Phi, \Psi) = \max_{1 \leq i, j \leq N} |\langle \phi_i, \psi_j \rangle|,$$

### 2.3.4 Recovery Algorithms

As mentioned earlier, the CS signal recovery process leverages the knowledge that the signal  $\mathbf{x}$  has a sparse representation by performing a sparse approximation of the

measurement vector  $\mathbf{y}$  in the frame  $\Upsilon$ . To that end, the Basis Pursuit (BP), Matching Pursuit and Orthogonal Matching Pursuit (OMP) algorithms are used to recover signals from noiseless measurements, while the BPIC, Lasso, BPDN, MP and OMP algorithms are used for recovery from noisy measurements [6, 46, 49, 57]. Furthermore, solvers for the optimization problems used in Lasso and BPDN that exploit the structure of the CS measurement matrices allow for fast and accurate recovery [58–61]. Additional algorithms have been proposed for the specific CS setting; we list several relevant examples below.

### Complexity-based Regularization and Iterative Hard Thresholding

Haupt and Nowak proposed an algorithm for recovery from noisy measurements [62]. The algorithm has a simple boundedness assumption on the entries of  $\theta$  ( $\theta(n) < B$ ,  $1 \leq n \leq N$ ) and employs two penalties: one measures the complexity of the signal:

$$c(\theta) = 2 \log N \|\theta\|_0.$$

while the other measures the goodness of fit to the measurements:

$$R(\theta) = \frac{1}{M} \sum_{m=1}^M \left( \mathbf{y}(m) - \sum_{n=1}^N \Upsilon(m, n) \mathbf{x}(n) \right)^2,$$

These penalties arise by posing a sparsity-promoting complexity measure on the coefficients of the signal representation  $\theta$  (in this specific case, assigning an i.i.d. Laplacian distribution to the coefficients) and an i.i.d. Gaussian prior on the noise added to each measurement. The recovery algorithm then consists of the optimization

$$\hat{\theta} = \arg \min_{\theta} R(\theta) + \frac{c(\theta) \log 2}{M\epsilon}, \quad (2.15)$$



where  $\epsilon$  is a constant. This optimization is equivalent to (2.8) and can be solved through iterative hard thresholding (IHT) [43, 44, 63–65]: starting with an initial estimate  $\hat{\theta}_0 = 0$ , IHT obtains a sequence of improving estimates using the iteration

$$\hat{\theta}_{i+1} = \mathfrak{T} \left( \hat{\theta}_i + \Upsilon^T(\mathbf{y} - \Upsilon \hat{\theta}_i), K \right).$$

### CoSaMP

The Compressive Sampling Matching Pursuit (CoSaMP) algorithm borrows concepts from greedy algorithms as well as solvers for the optimization-based CS signal recovery algorithms to achieve a high-performance, computationally efficient algorithm [66]. CoSaMP is an iterative algorithm that relies on two stages of sparse approximation: a first stage selects an enlarged candidate support set in a similar fashion to the OMP algorithm, while a second stage prunes down this initial approximation to the desired sparsity level. The algorithm is formally detailed as Algorithm 3. Subspace Pursuit (SP) [67], an independently proposed algorithm, features a very similar implementation.

#### 2.3.5 Performance Bounds on Signal Recovery

##### Instance Optimality Guarantees

Several CS signal recovery algorithms have similar guarantees on the signal estimate distortion. We collect a set of independent results in a single theorem.

**Theorem 2.5** [5, 65, 66] *The outputs  $\hat{\theta}$  of the CoSaMP, IHT, and BPIC algorithms*

---

**Algorithm 3** CoSaMP
 

---

Inputs: CS matrix  $\Upsilon$ , measurements  $\mathbf{y}$ , signal sparsity  $K$

Output:  $K$ -sparse approximation  $\hat{\theta}$  to true signal representation  $\theta$

Initialize:  $\hat{\theta}_0 = 0$ ,  $\mathbf{r} = \mathbf{y}$ ;  $i = 0$

**while** halting criterion false **do**

1.  $i \leftarrow i + 1$
2.  $\mathbf{e} \leftarrow \Upsilon^T \mathbf{r}$                       {form signal residual estimate}
3.  $\Omega \leftarrow \text{supp}(\mathfrak{T}(\mathbf{e}, 2K))$         {prune signal residual estimate}
4.  $T \leftarrow \Omega \cup \text{supp}(\hat{\theta}_{i-1})$         {merge supports}
5.  $\mathbf{b}|_T \leftarrow \Upsilon_T^\dagger \mathbf{y}$ ,  $\mathbf{b}|_{T^c}$         {form signal estimate}
6.  $\hat{\theta}_i \leftarrow \mathfrak{T}(\mathbf{b}, K)$                 {prune signal estimate}
7.  $\mathbf{r} \leftarrow \mathbf{y} - \Upsilon \hat{\theta}_i$                 {update measurement residual}

**end while**

return  $\hat{\theta} \leftarrow \hat{\theta}_i$

---

operating on  $\mathbf{y} = \Upsilon \theta + \mathbf{n}$  obey

$$\|\theta - \hat{\theta}\|_2 \leq C_1 \|\theta - \theta_K\|_2 + C_2 \frac{1}{\sqrt{K}} \|\theta - \theta_K\|_1 + C_3 \|\mathbf{n}\|_2. \quad (2.16)$$

For the CoSaMP algorithm, this guarantee requires  $\delta_{4K} \leq 0.1$  and set  $C_1 = C_2 = C_3 = 20$ . For the IHT algorithm, we require  $\delta_{3K} \leq 1/\sqrt{32}$  and set  $C_1 = C_2 = C_3 = 7$ .

For the BPIC algorithm, we require  $\delta_{2K} < \sqrt{2} - 1$  and  $\epsilon > \|\mathbf{n}\|_2$ , and set  $C_1 = 0$ ,  $C_2 = 4.2$  and  $C_3 = 8.5$  when  $\delta_{2K} = 0.2$ .

Theorem 2.5 states that these algorithms achieve provably *instance optimal* stable signal recovery (recall (2.5)).

For a  $K$ -sparse signal, these algorithms offer perfect recovery from noiseless measurements, meaning that the signal  $\hat{\mathbf{x}}$  recovered from the measurements  $\mathbf{y} = \Phi\mathbf{x}$  is exactly  $\hat{\mathbf{x}} = \mathbf{x}$ . Note also that in this case, we can apply the BPIC algorithm guarantee by setting the constant  $\epsilon = 0$ , turning in this case into the standard BP algorithm.

For a  $K$ -sparse signal  $\mathbf{x}$  whose measurements are corrupted by noise  $\mathbf{n}$  of bounded norm, the mean-squared error of the recovered signal  $\hat{\mathbf{x}}$  is bounded by

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2 \leq C\|\mathbf{n}\|_2, \quad (2.17)$$

with  $C$  a small constant.

For an  $s$ -compressible signal  $\mathbf{x}$  whose measurements are corrupted by noise  $\mathbf{n}$  of bounded norm, we can simplify this expression to

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2 \leq \frac{C_1SK^{-s}}{\sqrt{2s}} + \frac{C_2SK^{-s}}{s - 1/2} + C_3\|\mathbf{n}\|_2. \quad (2.18)$$

### Mutual Coherence-Dependent Guarantees

In some cases, the measurement matrix  $\bar{\Phi}$  corresponds to the transpose of a submatrix of an orthonormal basis  $\Phi$ , with the columns chosen randomly. In this case, we can formulate a guarantee for the distortion of the recovered signal that relies on the bases' mutual coherence.

**Theorem 2.6** [56] *Let  $\mathbf{x} = \Psi\theta$  be a  $K$ -sparse signal in  $\Psi$  with support  $\Omega \subset \{1, \dots, N\}$ ,  $|\Omega| = K$ , and with entries having signs chosen uniformly at random.*

Choose a subset  $\Gamma \subseteq \{1, \dots, N\}$  for the set of observed measurements, with  $M = |\Gamma|$ . Suppose that  $M \geq CKN\mu^2(\Phi, \Psi) \log(N/\delta)$  and  $M \geq C' \log^2(N/\delta)$  for fixed values of  $\delta < 1, C, C'$ . Then with probability at least  $1 - \delta$ ,  $\theta$  is the solution to (2.2).

The range of possible coherence values  $\mu(\Phi, \Psi)$  is  $[N^{-1/2}, 1]$ . Thus, the number of measurements required by Theorem 2.6 ranges from  $O(K \log(N))$  to  $O(N)$ . It is possible to expand the guarantee of Theorem 2.6 to compressible signals by adapting an argument of Rudelson and Vershynin in [68] to link coherence and restricted isometry constants.

**Theorem 2.7** [68, Remark 3.5.3] Choose a subset  $\Gamma \subseteq \{1, \dots, N\}$  for the set of observed measurements, with  $M = |\Gamma|$ . Suppose that

$$M \geq CK\sqrt{Nt}\mu(\Phi, \Psi) \log(tK \log N) \log^2 K \quad (2.19)$$

for a fixed value of  $C$ . Then with probability at least  $1 - 5e^{-t}$  the resulting matrix  $\Phi^T \Psi$  has the RIP with constant  $\delta_{2K} \leq 1/2$ .

Using this theorem, we obtain the guarantee of Theorem 2.5 for compressible signals with the number of measurements  $M$  dictated by the coherence value  $\mu(\Phi, \Psi)$ .

### Probabilistic Guarantees

We can provide probabilistic guarantees for the complexity-based regularization algorithm when we observe exactly sparse signals in the canonical basis  $\Psi = \mathbf{I}$  and use random measurement matrices  $\Phi$  with i.i.d. normalized Rademacher entries.

**Theorem 2.8** [62] For  $\mathbf{x} \in \Sigma_K$ , the output of the complexity-based regularization (2.15) obeys

$$\mathbb{E} \left[ \frac{\|\theta - \hat{\theta}\|^2}{N} \right] \leq \frac{CK \log n}{M},$$

where  $C$  is a constant dependent on  $B$  and  $\sigma$ .

For  $s$ -compressible signals, a similar guarantee is obtained.

**Theorem 2.9** [62] If  $\mathbf{x}$  is an  $s$ -compressible signal in the basis  $\Psi = \mathbf{I}$ , then

$$\mathbb{E} \left[ \frac{\|\theta - \hat{\theta}\|^2}{N} \right] \leq \left( \frac{CK \log n}{M} \right)^{1 + \frac{1}{2s}},$$

where  $C$  is a constant dependent on  $B$  and  $\sigma$ .

Similarly, there is a probabilistic guarantee for the OMP algorithm when random matrices with i.i.d. normalized Gaussian entries (with any basis  $\Psi$ ) or Rademacher entries (with the canonical basis  $\Psi = \mathbf{I}$ ) are used.

**Theorem 2.10** [57] Let  $M \geq CK \log(N/\delta)$  for some  $\delta \in (0, 0.36)$  and  $\mathbf{x} \in \Sigma_K$ . Then with probability  $1 - \delta$  the output of OMP obeys  $\hat{\theta} = \theta$ . Furthermore, if  $M > 2K$  and OMP yields a residual  $\mathbf{r} = 0$  after  $K$  iterations, then  $\hat{\theta} = \theta$  with probability 1; otherwise, OMP fails to recover the signal.

Unfortunately, no guarantees for recovery from noisy measurements or for compressible signals have been proven for OMP-based recovery in CS.

## 2.4 Distributed Compressive Sensing

In this section, we generalize the notion of a signal being sparse in some basis to the notion of an ensemble of signals being *jointly sparse* [9]. A *joint sparsity model* (JSM) encodes the correlation between the values and locations of the coefficients for a group of sparse signals. As we will show later, joint sparsity is applicable to cases where multiple sparse signals are generated by a single event. In most of these cases, we either favor or it is our only choice to obtain independent measurements for each sparse signal, resulting in a set of measurement vectors  $\mathbf{y}_j = \Phi_j \mathbf{x}_j$ ,  $1 \leq j \leq J$ . Since the measurements are independent, we use joint sparsity models in order to exploit the correlations between the signals in the ensemble to improve the performance of CS recovery.

### 2.4.1 Joint Sparsity Models

We consider three different JSMs that are inspired by different real world situations. In the first two models, each signal is itself sparse, and so we could use the CS framework from above to encode and decode each one separately, yet there also exists a framework wherein a *joint representation* for the ensemble uses fewer total vectors. In the third model, no signal is itself sparse, yet there still exists a joint sparsity among the signals that allows recovery with significantly fewer measurements per sensor. We note that for different real world settings, different models for sparsity-based signal ensemble structure can be posed, together with appropriate recovery algorithms.

We use the following notation for signal ensembles. Denote the *signals* in the

ensemble by  $\mathbf{x}_j$ ,  $j = 1, 2, \dots, J$  where each  $\mathbf{x}_j \in \mathbb{R}^N$ . We assume that there exists a known basis or frame  $\Psi$  for  $\mathbb{R}^N$  in which  $\mathbf{x}_j$  can be sparsely represented.

### JSM-1: Sparse Common Component + Innovations

In this model, all signals share a *common* sparse component while each individual signal contains a sparse *innovations* component:

$$\mathbf{x}_j = \mathbf{z}_C + \mathbf{z}_j, \quad j \in \{1, 2, \dots, J\}$$

with

$$\mathbf{z}_C = \Psi\theta_C, \quad \|\theta_C\|_0 = K,$$

$$\mathbf{z}_j = \Psi\theta_j, \quad \|\theta_j\|_0 = K_j.$$

Thus, the signal  $\mathbf{z}_C$  is common to all of the  $\mathbf{x}_j$  and has sparsity  $K$  in basis  $\Psi$ . The signals  $\mathbf{z}_j$  are the unique portions of the  $\mathbf{x}_j$  and have sparsity  $K_j$  in the same basis.

A practical situation well-modeled by JSM-1 is a group of sensors measuring temperatures at a number of locations throughout the day. The temperature readings  $\mathbf{x}_j$  have both temporal (intra-signal) and spatial (inter-signal) correlations. Global factors, such as the sun and prevailing winds, could have an effect  $\mathbf{z}_C$  that is both common to all sensors and structured enough to permit sparse representation. More local factors, such as shade, water, or animals, could contribute localized innovations  $\mathbf{z}_j$  that are also structured (and hence sparse). A similar scenario could be imagined for a sensor network recording light intensities, air pressure, or other phenomena. All of these scenarios correspond to measuring properties of physical processes that change smoothly in time and in space and thus are highly correlated.

### JSM-2: Common Sparse Supports

In this model, all signals are constructed from the same sparse index set of basis vectors, but with different coefficients:

$$\mathbf{x}_j = \Psi\theta_j, \quad j \in \{1, 2, \dots, J\},$$

where each  $\theta_j$  is supported only on the same  $\Omega \subset \{1, 2, \dots, N\}$  with  $|\Omega| = K$ . Hence, all signals are  $K$ -sparse, and all are constructed from the same  $K$  elements of  $\Psi$ , but with arbitrarily different coefficients. This model can be viewed as a special case of JSM-1 (with  $K_C = 0$  and  $K_j = K$  for all  $j$ ) but features additional correlation structure that suggests distinct recovery algorithms.

A practical situation well-modeled by JSM-2 is where multiple sensors acquire the same Fourier-sparse signal but with phase shifts and attenuations caused by signal propagation. In many cases it is critical to recover each one of the sensed signals, such as in many acoustic localization and array processing algorithms. Another application for JSM-2 is MIMO communication [69]. Section 4.4 presents a series of experiments applying JSM-2 to environmental and acoustic data.

### JSM-3: Nonsparse Common + Sparse Innovations

This model extends JSM-1 so that the common component need no longer be sparse in any basis; that is,

$$\mathbf{x}_j = \mathbf{z}_C + \mathbf{z}_j, \quad j \in \{1, 2, \dots, J\}$$

with

$$\mathbf{z}_C = \Psi\theta_C \quad \text{and} \quad \mathbf{z}_j = \Psi\theta_j, \quad \|\theta_j\|_0 = K_j,$$



but  $\mathbf{z}_C$  is not necessarily sparse in the basis  $\Psi$ . We also consider the case where the supports of the innovations are shared for all signals, which extends JSM-2.

A practical situation well-modeled by JSM-3 is where several sources are recorded by different sensors together with a background signal that is not sparse in any basis. Consider, for example, a computer vision-based verification system in a device production plant. Cameras acquire snapshots of components in the production line; a computer system then checks for failures in the devices for quality control purposes. While each image could be extremely complicated, the ensemble of images will be highly correlated, since each camera observes the same device with minor (sparse) variations.

#### 2.4.2 Signal Ensemble Recovery Algorithms

The algorithm used for joint signal recovery depends on the relevant JSM for the signals observed. We briefly overview proposed recovery techniques for each JSM; more details on the algorithms (and the theoretical requirements on the measurement rates  $M_j$ ) can be found in [9].

For JSM-1, there exists an analytical framework inspired by principles of information theory. This allows us to characterize the measurement rates  $M_j$  required to *jointly* recover the signals  $x_j$ . The measurement rates relate directly to the signals' *conditional sparsities*, in parallel with the Slepian-Wolf theory. The recovery technique is based on a single execution of a weighted linear program that seeks the sparsest components  $[\mathbf{z}_C; \mathbf{z}_1; \dots \mathbf{z}_J]$  that account for the observed measurements. Theoretical analysis and numerical experiments confirm that the rates  $M_j$  required

for joint CS recovery are well below those required for independent CS recovery of each signal  $\mathbf{x}_j$  [9].

For JSM-2, there exist algorithms inspired by conventional greedy algorithms (such as OMP) that can substantially reduce the number of measurements when compared with independent recovery. In the single-signal case, OMP iteratively constructs the sparse support set  $\Omega$ ; decisions are based on inner products between the columns of  $\Phi\Psi$  and a residual. In the multi-signal case, there are more clues available for determining the elements of  $\Omega$ . An example algorithm is DCS-SOMP, a simple variant of Simultaneous Orthogonal Matching Pursuit (SOMP) [9, 69] which is formalized as Algorithm 4. For a large number of sensors  $J$ , close to  $K$  measurements per signal suffice for joint recovery (that is,  $c \rightarrow 1$  as  $J \rightarrow \infty$ ); see Figure 2.4 for an example of improving performance as  $J$  increases. On the contrary, with independent CS recovery, perfect recovery of all signals requires *increasing* each  $M_j$  in order to maintain the same probability of recovery of the signal ensemble. This surprise is due to the fact that each signal will experience an independent probability  $p \leq 1$  of successful recovery; therefore the overall probability of complete success is  $p^J$ . Consequently, each sensor must compensate by making additional measurements. We also note that when the supports of the innovations of the signals are small, signals that are well modeled by JSM-1 can also be modeled by JSM-2 by selecting a global support that contains all of the individual supports. Such approximation allows for a simpler recovery algorithm, while incurring a slight increase in the number of measurements required for recovery.

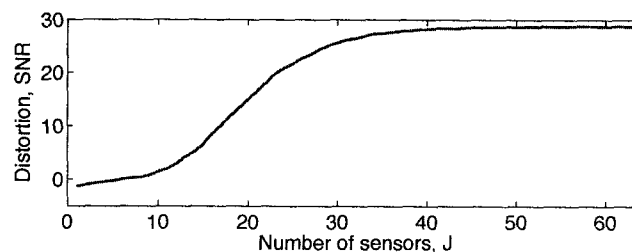


Figure 2.4 : *Joint recovery of synthetic JSM-2 signals having length  $N = 128$  and sparsity  $K = 10$  using  $M = 11$  random measurements per sensor. Each measurement is quantized to approximately 5 bits of precision. The recovery is robust to quantization and is progressive: as the number of sensors  $J$  increases we see improved recovery performance.*

For JSM-3, no individual signal  $\mathbf{x}_j$  is sparse, and so recovery of each signal separately would require a full  $N$  measurements per signal. To approach the recovery problem, we note that the common component  $\mathbf{z}_C$  is observed by all sensors. This is the main concept behind the Alternating Common and Innovation Estimation (ACIE) recovery algorithm [9], which alternates between two steps: (1) Estimate the common component  $\mathbf{z}_C$  by combining all measurements and treating the innovations  $\mathbf{z}_j$  as noise that can be averaged out; (2) Estimate the innovations  $\mathbf{z}_j$  from each sensor by subtracting the estimated common component  $\mathbf{z}_C$  and then applying standard CS recovery techniques. We have proved that, asymptotically, each sensor need only measure at the rate dictated by the sparsity  $K_j$  [9]. Thus, for a large number of sensors  $J$ , the impact of the common nonsparse component  $\mathbf{z}_C$  is eliminated.

---

**Algorithm 4** DCS-SOMP
 

---

Inputs: Measurement matrices  $\Phi_j$ , measurement vectors  $\mathbf{y}_j$ ,  $j = 1, \dots, J$ .

Outputs: Sparse signals  $\mathbf{x}_j$ ,  $j = 1, \dots, J$ .

Initialize:  $\Omega = \emptyset$ ,  $i = 0$

**for**  $j = 1, \dots, J$  **do**

$\hat{\mathbf{x}}_{j,0} = \mathbf{0}$ ,  $\mathbf{r}_j = \mathbf{y}_j$  {initialize}

**end for**

**while** halting criterion false **do**

1.  $i \leftarrow i + 1$

2.  $\mathbf{b}_j \leftarrow \Phi_j^T \mathbf{r}_j$ ,  $j = 1, \dots, J$  {form residual signal estimates}

3.  $\mathbf{b} = \sum_{j=1}^J |\mathbf{b}_j|$  {merge signal residual estimates  
in absolute value}

4.  $\Omega \leftarrow \Omega \cup \text{supp}(\mathfrak{T}(\mathbf{b}, 1))$  {add index of residual's largest  
magnitude entry to signal support}

**for**  $j = 1, \dots, J$  **do**

5a.  $\hat{\mathbf{x}}_{j,i}|\Omega \leftarrow \Psi_{j,\Omega}^\dagger \mathbf{y}_j$ ,  $\hat{\mathbf{x}}_{j,i}|\Omega^c \leftarrow 0$  {form signal estimates}

5b.  $\mathbf{r}_j \leftarrow \mathbf{y}_j - \Phi_j \hat{\mathbf{x}}_{j,i}$  {update measurement residuals}

**end for**

**end while**

return  $\hat{\mathbf{x}}_j \leftarrow \hat{\mathbf{x}}_{j,i}$ ,  $j = 1, \dots, J$

---

## Chapter 3

### Theoretical Measurement Bounds for Jointly Sparse Signals via Graphical Models

In Section 2.4, we summarized a framework for *distributed compressive sensing* (DCS) that enables new distributed coding algorithms to exploit both intra- and inter-signal correlation structures. In a typical DCS scenario, multiple sensors measure signals that are each individually sparse in some basis and also correlated among sensors. Each sensor *independently* encodes its signal by projecting it onto a small set of randomized vectors and then transmits the resulting coefficients to a single collection point. Under the right conditions, a decoder at the collection point can recover each of the signals precisely.

The DCS theory relies on the *joint sparsity* of a signal ensemble. Unlike the single-signal definition of sparsity, however, there are numerous plausible ways in which joint sparsity could be defined. In this chapter,<sup>1</sup> we provide a general framework for joint sparsity using graphical models. Using this framework, we derive upper and lower bounds for the number of noiseless measurements required for recovery. Our results are also applicable to cases where the signal ensembles are measured jointly, as well as to the single signal case.

---

<sup>1</sup>This work is in collaboration with Shriram Sarvotham, Dror Baron, Michael B. Wakin, and Richard G. Baraniuk [9, 70, 71]

### 3.1 Algebraic Framework

Our framework enables analysis of a given ensemble  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_J$  in a “jointly sparse” sense, as well as a metric for the complexities of different signal ensembles. It is based on a factored representation of the signal ensemble that decouples location and value information. We begin by illustrating the single signal case. For clarity and without loss of generality, we will assume in this chapter that the signals  $\mathbf{x}_j$ ,  $1 \leq j \leq J$ , are sparse in the canonical basis  $\Psi = \mathbf{I}$ .

#### 3.1.1 Single Signal Case

Consider a sparse vector  $\mathbf{x} \in \mathbb{R}^N$  with  $K < N$  nonzero entries. Alternatively, we can write  $\mathbf{x} = \mathbf{P}\theta$ , where  $\theta \in \mathbb{R}^K$  contains the nonzero values of  $\mathbf{x}$ , and  $\mathbf{P}$  is an *identity submatrix*, i.e.,  $\mathbf{P}$  contains  $K$  columns of the  $N \times N$  identity matrix  $\mathbf{I}$ . To model the set of all possible sparse signals, let  $\mathcal{P}$  be the set of all identity submatrices of all possible sizes  $N \times K'$ , with  $1 \leq K' \leq N$ . We refer to  $\mathcal{P}$  as a *sparsity model*. Given a signal  $\mathbf{x}$ , one may consider all possible factorizations  $\mathbf{x} = \mathbf{P}\theta$ , with  $\mathbf{P} \in \mathcal{P}$ . Whether a signal is sufficiently sparse is defined *in the context of this model*: given a signal  $\mathbf{x}$ , one can consider all possible factorizations  $\mathbf{x} = \mathbf{P}\theta$  with  $\mathbf{P} \in \mathcal{P}$ . Among these factorizations, the unique representation with smallest dimensionality for  $\theta$  equals the *sparsity level* of the signal  $\mathbf{x}$  under the model  $\mathcal{P}$ .

### 3.1.2 Multiple Signal Case

For multiple signals, consider factorizations of the form  $\mathbf{X} = \mathbf{P}\Theta$  where

$$\mathbf{X} = [\mathbf{x}_1^T \ \dots \ \mathbf{x}_J^T]^T, \ \mathbf{X} \in \mathbb{R}^{JN}$$

is the concatenation of the signals in the ensemble,  $\mathbf{P} \in \mathbb{R}^{JN \times D}$ , and  $\Theta \in \mathbb{R}^D$ . We refer to  $\mathbf{P}$  and  $\Theta$  as the *location matrix* and *value vector*, respectively. A JSM is defined in terms of a set  $\mathcal{P}$  of admissible location matrices  $\mathbf{P}$  with varying numbers of columns; we specify below additional conditions that the matrices  $P$  must satisfy for each model. For a given ensemble  $\mathbf{X}$ , we let  $\mathcal{P}_F(\mathbf{X}) \subseteq \mathcal{P}$  denote the set of feasible location matrices  $\mathbf{P} \in \mathcal{P}$  for which a factorization  $\mathbf{X} = \mathbf{P}\Theta$  exists. We define the *joint sparsity level* of the signal ensemble as follows.

**Definition 3.1** *The joint sparsity level  $D$  of the signal ensemble  $\mathbf{X}$  is the number of columns of the smallest matrix  $P \in \mathcal{P}_F(X)$ .*

In contrast to the single-signal case, there are several natural choices for what matrices  $\mathbf{P}$  should be members of a joint sparsity model  $\mathcal{P}$ . We restrict our attention in the sequel to what we call *common/innovation component JSMs*. In these models each signal  $\mathbf{x}_j$  is generated as a combination of two components: (i) a common component  $\mathbf{z}_C$ , which is present in all signals, and (ii) an innovation component  $\mathbf{z}_j$ , which is unique to each signal. These combine additively, giving

$$\mathbf{x}_j = \mathbf{z}_C + \mathbf{z}_j, \quad j \in \Lambda.$$

Note, however, that the individual components might be zero-valued in specific scenarios. We can express the component signals as

$$\mathbf{z}_C = \mathbf{P}_C \theta_C, \quad \mathbf{z}_j = \mathbf{P}_j \theta_j, \quad j \in \Lambda,$$

where  $\theta_C \in \mathbb{R}^{K_C}$  and each  $\theta_j \in \mathbb{R}^{K_j}$  have nonzero entries. Each matrix  $\mathbf{P} \in \mathcal{P}$  that can express such signals  $\{\mathbf{x}_j\}$  has the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_C & \mathbf{P}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{P}_C & \mathbf{0} & \mathbf{P}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_C & \mathbf{0} & \mathbf{0} & \dots & \mathbf{P}_J \end{bmatrix}, \quad (3.1)$$

where  $\mathbf{P}_C, \{\mathbf{P}_j\}_{j \in \Lambda}$  are identity submatrices. We define the value vector as  $\Theta = [\theta_C^T \theta_1^T \theta_2^T \dots \theta_J^T]^T$ , where  $\theta_C \in \mathbb{R}^{K_C}$  and each  $\theta_j \in \mathbb{R}^{K_j}$ , to obtain  $\mathbf{X} = \mathbf{P}\Theta$ . Although the values of  $K_C$  and  $K_j$  are dependent on the matrix  $\mathbf{P}$ , we omit this dependency in the sequel for brevity, except when necessary for clarity.

If a signal ensemble  $\mathbf{X} = \mathbf{P}\Theta$ ,  $\Theta \in \mathbb{R}^\delta$  were to be generated by a selection of  $\mathbf{P}_C$  and  $\{\mathbf{P}_j\}_{j \in \Lambda}$ , where all  $J+1$  identity submatrices share a common column vector, then  $\mathbf{P}$  would not be full rank. In other cases, we may observe a vector  $\Theta$  that has zero-valued entries; i.e., we may have  $\theta_j(k) = 0$  for some  $1 \leq k \leq K_j$  and some  $j \in \Lambda$ , or  $\theta_C(k) = 0$  for some  $1 \leq k \leq K_C$ . In both of these cases, by removing one instance of this column from any of the identity submatrices, one can obtain a matrix  $\mathbf{Q}$  with fewer columns for which there exists  $\Theta' \in \mathbb{R}^{\delta-1}$  that gives  $\mathbf{X} = \mathbf{Q}\Theta'$ . If  $\mathbf{Q} \in \mathcal{P}$ , then we term this phenomenon *sparsity reduction*. Sparsity reduction, when present, reduces the effective joint sparsity of a signal ensemble. As an example of



sparsity reduction, consider  $J = 2$  signals of length  $N = 2$ . Consider the coefficient  $\mathbf{z}_C(1) \neq 0$  of the common component  $\mathbf{z}_C$  and the corresponding innovation coefficients  $\mathbf{z}_1(1), \mathbf{z}_2(1) \neq 0$ . Suppose that all other coefficients are zero. The location matrix  $\mathbf{P}$  that arises is

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

The span of this location matrix (i.e., the set of signal ensembles  $\mathbf{X}$  that it can generate) remains unchanged if we remove any one of the columns, i.e., if we drop any entry of the value vector  $\Theta$ . This provides us with a lower-dimensional representation  $\Theta'$  of the same signal ensemble  $X$  under the JSM  $\mathcal{P}$ ; the joint sparsity of  $\mathbf{X}$  is  $D = 2$ .

### 3.2 Bound on Measurement Rates

In this section, we seek conditions on  $\mathcal{M} = (M_1, M_2, \dots, M_J)$ , the tuple of number of measurements from each sensor, such that we can guarantee perfect recovery of  $\mathbf{X}$  given  $\mathbf{Y}$ . To this end, we provide a graphical model for the general framework provided in Section 3.1. This graphical model is fundamental in the derivation of the number of measurements needed for each sensor, as well as in the formulation of a combinatorial recovery procedure.

Based on the models presented in Section 2.4.1, recovering  $\mathbf{X}$  requires determining a value vector  $\Theta$  and location matrix  $\mathbf{P}$  such that  $\mathbf{X} = \mathbf{P}\Theta$ . Two challenges immediately present themselves. First, a given measurement depends only on some

of the components of  $\Theta$ , and the measurement budget should be adjusted between the sensors according to the information that can be gathered on the components of  $\Theta$ . For example, if a component  $\Theta(d)$  does not affect any signal coefficient  $\mathbf{x}_j(\cdot)$  in sensor  $j$ , then the corresponding measurements  $\mathbf{y}_j$  provide no information about  $\Theta(d)$ . Second, the decoder must identify a location matrix  $\mathbf{P} \in \mathcal{P}_F(\mathbf{X})$  from the set  $\mathcal{P}$  and the measurements  $\mathbf{Y} = [\mathbf{y}_1^T \ \mathbf{y}_2^T \ \dots \ \mathbf{y}_J^T]^T$ .

### 3.2.1 Graphical Model Framework

We introduce a graphical representation that captures the dependencies between the measurements in  $\mathbf{Y}$  and the value vector  $\Theta$ , represented by  $\Phi$  and  $\mathbf{P}$ . Consider a feasible decomposition of  $\mathbf{X}$  into a full-rank matrix  $\mathbf{P} \in \mathcal{P}_F(\mathbf{X})$  and the corresponding  $\Theta$ ; the matrix  $\mathbf{P}$  defines the sparsities of the common and innovation components  $K_C$  and  $K_j$ ,  $1 \leq j \leq J$ , as well as the joint sparsity  $D = K_C + \sum_{j=1}^J K_j$ . Define the following sets of vertices, illustrated in Figure 3.1: (i) the set of *value vertices*  $V_V$  has elements with indices  $d \in \{1, \dots, D\}$  representing the entries of the value vector  $\Theta(d)$ ; (ii) the set of *signal vertices*  $V_S$  has elements with indices  $(j, n)$  representing the signal entries  $\mathbf{x}_j(n)$ , with  $j \in \Lambda$  and  $n \in \{1, \dots, N\}$ ; and (iii) the set of *measurement vertices*  $V_M$  has elements with indices  $(j, m)$  representing the measurements  $y_j(m)$ , with  $j \in \Lambda$  and  $m \in \{1, \dots, M_j\}$ . The cardinalities for these sets are  $|V_V| = D$ ,  $|V_S| = JN$ , and  $|V_M| = \sum_{j=1}^J M_j$ , respectively.

Let  $\mathbf{P}$  be partitioned into *location submatrices*  $\mathbf{P}^j$ ,  $j \in \Lambda$ , so that  $\mathbf{x}_j = \mathbf{P}^j \Theta$ ; here  $\mathbf{P}^j$  is the restriction of  $\mathbf{P}$  to the rows that generate the signal  $\mathbf{x}_j$ . We then define the bipartite graph  $G = (V_S, V_V, E)$ , determined by  $\mathbf{P}$  and shown in Figure 3.1(a), where

there exists an edge connecting  $(j, n)$  and  $d$  if and only if  $\mathbf{P}^j(n, d) \neq 0$ .

A similar bipartite graph  $G' = (V_M, V_S, E')$ , illustrated in Figure 3.1(a), connects the measurement vertices  $\{(j, m)\}$  to the signal vertices  $\{(j, n)\}$ ; there exists an edge in  $G'$  connecting  $(j, n) \in V_S$  and  $(j, m) \in V_M$  if  $\Phi_j(m, n) \neq 0$ . When the measurements matrices  $\Phi_j$  are dense, which occurs with probability one for i.i.d. Gaussian random matrices, the vertices corresponding to entries of a given signal  $\mathbf{x}_j$  in  $V_S$  are all connected to all vertices corresponding to the measurements  $\mathbf{y}_j$  in  $V_V$ . Figure 3.1(a) shows an example for dense measurement matrices: each measurement vertex  $(j, \cdot)$  is connected to each signal vertex  $(j, \cdot)$ .

The graphs  $G$  and  $G'$  can be merged into  $\hat{G} = (V_M, V_V, \hat{E})$  that relates entries of the value vector to measurements. Figure 3.1(b) shows the example composition of the previous two bipartite graphs.  $\hat{G}$  is used to recover  $\Theta$  from the measurement ensemble  $\mathbf{Y}$  when  $\mathbf{P}$  is known.

### 3.2.2 Quantifying Redundancies

In order to obtain sharp bounds on the number of measurements needed, our analysis of the measurement process must account for redundancies between the locations of the nonzero coefficients in the common and innovation components. To that end, we consider the overlaps between common and innovation components in each signal. When we have  $\mathbf{z}_C(n) \neq 0$  and  $\mathbf{z}_j(n) \neq 0$  for a certain signal  $j$  and some index  $1 \leq n \leq N$ , we cannot recover the values of both coefficients from the measurements of this signal alone; therefore, we will need to recover  $\mathbf{z}_C(n)$  using measurements of other signals that do not feature the same overlap. We thus quantify the size of the

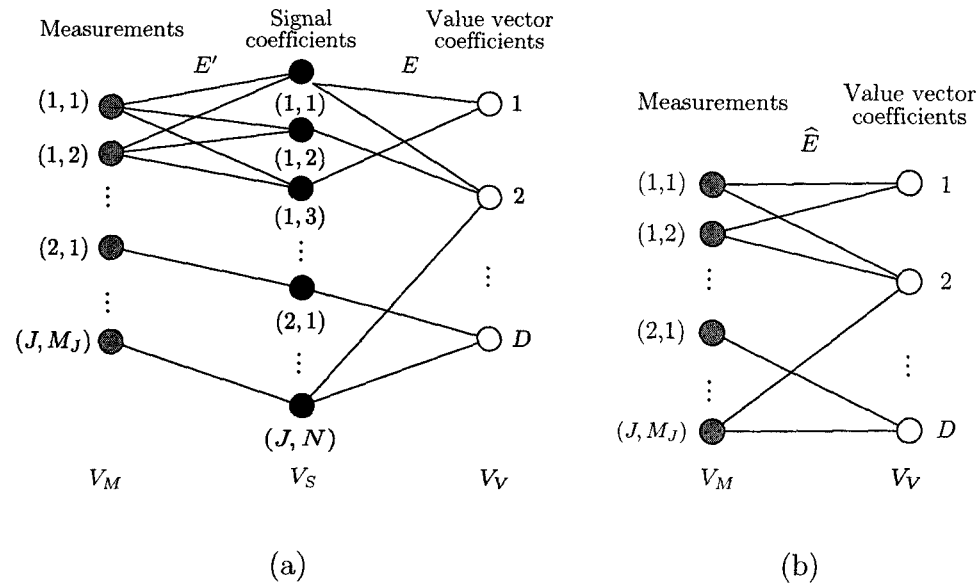


Figure 3.1 : Bipartite graphs for distributed compressive sensing. (a)  $G = (V_S, V_V, E)$  connects the entries of each signal with the value vector coefficients they depend on;  $G' = (V_M, V_S, E')$  connects the measurements at each sensor with observed signal entries. The matrix  $\Phi$  is a dense Gaussian random matrix, as shown in the graph. (b)  $\hat{G} = (V_M, V_V, \hat{E})$  is the composition of  $G$  and  $G'$ , and relates between value vector coefficients and measurements.

overlap for all subsets of signals  $\Gamma \subset \Lambda$  under a feasible representation given by  $\mathbf{P}$  and  $\Theta$ , as described in Section 3.1.

**Definition 3.2** The overlap size for the set of signals  $\Gamma \subset \Lambda$ , denoted  $K_C(\Gamma, \mathbf{P})$ , is the number of indices in which there is overlap between the common and the innovation component supports at all signals  $j \notin \Gamma$ :

$$K_C(\Gamma, \mathbf{P}) = |\{n \in \{1, \dots, N\} : \mathbf{z}_C(n) \neq 0 \text{ and } \forall j \notin \Gamma, \mathbf{z}_j(n) \neq 0\}|. \quad (3.2)$$

We also define  $K_C(\Lambda, \mathbf{P}) = K_C(\mathbf{P})$  and  $K_C(\emptyset, \mathbf{P}) = 0$ .

For  $\Gamma \subset \Lambda$ ,  $K_C(\Gamma, \mathbf{P})$  provides a penalty term due to the need for recovery of common component coefficients that are overlapped by innovations in all other signals  $j \notin \Gamma$ . Intuitively, for each entry counted in  $K_C(\Gamma, \mathbf{P})$ , some sensor in  $\Gamma$  must take one measurement to account for that entry of the common component — it is impossible to recover such entries from measurements made by sensors outside of  $\Gamma$ . When all signals  $j \in \Lambda$  are considered, it is clear that all of the common component coefficients must be recovered from the obtained measurements.

### 3.2.3 Measurement Bounds

Converse and achievable bounds for the number of measurements necessary for DCS recovery are given below. Our bounds consider each subset of sensors  $\Gamma \subseteq \Lambda$ , since the cost of sensing the common component can be amortized across sensors: it may be possible to reduce the rate at one sensor  $j_1 \in \Gamma$  (up to a point), as long as other sensors in  $\Gamma$  offset the rate reduction. We quantify the reduction possible through the following definition.

**Definition 3.3** *The conditional sparsity of the set of signals  $\Gamma$  is the number of entries of the vector  $\Theta$  that must be recovered by measurements  $\mathbf{y}_j$ ,  $j \in \Gamma$ :*

$$K_{\text{cond}}(\Gamma, \mathbf{P}) = \left( \sum_{j \in \Gamma} K_j(\mathbf{P}) \right) + K_C(\Gamma, \mathbf{P}).$$

The joint sparsity gives the number of degrees of freedom for the signals in  $\Lambda$ , while the conditional sparsity gives the number of degrees of freedom for signals in  $\Gamma$  when the signals in  $\Lambda \setminus \Gamma$  are available as side information.

The bipartite graph introduced in Section 3.2.1 is the cornerstone of Theorems 3.1, 3.2, and 3.3, which consider whether a perfect matching can be found in the graph; see the proofs in Appendices A, C, and D, respectively, for detail.

**Theorem 3.1** (Achievable, known  $\mathbf{P}$ ) *Assume that a signal ensemble  $\mathbf{X}$  is obtained from a common/innovation component JSM  $\mathcal{P}$ . Let  $\mathcal{M} = (M_1, M_2, \dots, M_J)$  be a measurement tuple, let  $\{\Phi_j\}_{j \in \Lambda}$  be random matrices having  $M_j$  rows of i.i.d. Gaussian entries for each  $j \in \Lambda$ , and write  $\mathbf{Y} = \Phi\mathbf{X}$ . Suppose there exists a full rank location matrix  $\mathbf{P} \in \mathcal{P}_F(\mathbf{X})$  such that*

$$\sum_{j \in \Gamma} M_j \geq K_{\text{cond}}(\Gamma, \mathbf{P}) \quad (3.3)$$

*for all  $\Gamma \subseteq \Lambda$ . Then with probability one over  $\{\Phi_j\}_{j \in \Gamma}$ , there exists a unique solution  $\hat{\Theta}$  to the system of equations  $\mathbf{Y} = \Phi\mathbf{P}\hat{\Theta}$ ; hence, the signal ensemble  $\mathbf{X}$  can be uniquely recovered as  $\mathbf{X} = \mathbf{P}\hat{\Theta}$ .*

**Theorem 3.2** (Achievable, unknown  $\mathbf{P}$ ) *Assume that a signal ensemble  $\mathbf{X}$  and measurement matrices  $\{\Phi_j\}_{j \in \Lambda}$  follow the assumptions of Theorem 3.1. Suppose there exists a full rank location matrix  $\mathbf{P}^* \in \mathcal{P}_F(\mathbf{X})$  such that*

$$\sum_{j \in \Gamma} M_j \geq K_{\text{cond}}(\Gamma, \mathbf{P}^*) + |\Gamma| \quad (3.4)$$

*for all  $\Gamma \subseteq \Lambda$ . Then  $\mathbf{X}$  can be uniquely recovered from  $\mathbf{Y}$  with probability one over  $\{\Phi_j\}_{j \in \Gamma}$ .*

**Theorem 3.3** (Converse) *Assume that a signal ensemble  $\mathbf{X}$  and measurement matrices  $\{\Phi_j\}_{j \in \Lambda}$  follow the assumptions of Theorem 3.1. Suppose there exists a full rank*

location matrix  $P \in \mathcal{P}_F(\mathbf{X})$  such that

$$\sum_{j \in \Gamma} M_j < K_{\text{cond}}(\Gamma, \mathbf{P}) \quad (3.5)$$

for some  $\Gamma \subseteq \Lambda$ . Then there exists a solution  $\hat{\Theta}$  such that  $\mathbf{Y} = \Phi \mathbf{P} \hat{\Theta}$  but  $\hat{\mathbf{X}} := \mathbf{P} \hat{\Theta} \neq \mathbf{X}$ .

The identification of a feasible location matrix  $\mathbf{P}$  causes the one measurement per sensor gap that prevents (3.4)–(3.5) from being a tight converse and achievable bound pair. We note in passing that the signal recovery procedure used in Theorem 3.2 is akin to  $\ell_0$ -norm minimization on  $\mathbf{X}$ ; see Appendix C for details.

#### 3.2.4 Discussion

The bounds in Theorems 3.1–3.3 are dependent on the dimensionality of the subspaces in which the signals reside. The number of noiseless measurements required for ensemble recovery is determined by the dimensionality  $\dim(\mathcal{S})$  of the subspace  $\mathcal{S}$  in the relevant signal model, because dimensionality and sparsity play a volumetric role akin to the entropy  $H$  used to characterize rates in source coding. Whereas in source coding each bit resolves between two options, and  $2^{NH}$  typical inputs are described using  $NH$  bits [72], in CS we have  $M = \dim(\mathcal{S}) + O(1)$ . Similar to Slepian-Wolf coding [73], the number of measurements required for each sensor must account for the minimal features unique to that sensor, while at the same time features that appear among multiple sensors must be amortized over the group.

Theorems 3.1–3.3 can also be applied to the single sensor and joint measurement settings. In the single-signal setting, we will have  $\mathbf{x} = \mathbf{P}\theta$  with  $\theta \in \mathbb{R}^K$ , and  $\Lambda = \{1\}$ ;

Theorem 3.2 provides the requirement  $M \geq K + 1$ . It is easy to show that the joint measurement is equivalent to the single-signal setting: we concatenate all the individual signals into a single signal vector, and in both cases all measurements are dependent on all the entries of the signal vector. However, the distribution of the measurements among the available sensors is irrelevant in a joint measurement setting. Therefore, we only obtain a necessary condition  $\sum_j M_j \geq D + 1$  on the total number of measurements required.



## Chapter 4

### Distributed Compressive Sensing for Sensor Networks

In this chapter,<sup>1</sup> we demonstrate the potential of DCS for universal distributed sensing in sensor networks. We develop and validate algorithms for several network-specific signal processing and compression tasks using random measurements on real sensor network data. The properties of DCS directly address the sensor network challenges outlined in Chapter 1. In particular, DCS algorithms: offer a universal encoding appropriate for any jointly sparse signal ensemble; are completely non-collaborative and involve no communication overhead; can be implemented on the simplest computing hardware on the sensor nodes since they shift nearly all computational complexity to the decoder at the collection point; are inherently fault tolerant, robust to measurement and quantization noise, and secure; are robust to lossy communication links; offer progressively better, tunable recovery as measurements stream in; and are applicable to a range of sensor network signal processing tasks, from signal compression to estimation and detection/classification. To coin a term, DCS sensors are “omniscient”: they omnisciently capture the relevant signal information despite being nescient (ignorant) of the actual structure.

---

<sup>1</sup>This work is in collaboration with Michael B. Wakin, Dror Baron, and Richard G. Baraniuk [74]

## 4.1 Related Work

Several approaches have been proposed for data collection in sensor networks, most of which exploit the correlation among the signals being recorded. DIMENSIONS [75] enables distributed information storage and multiresolution data retrieval; it achieves compression by assuming that the signal at each sensor node features temporal correlation and clustering sensors that observe correlated signals in a hierarchical fashion. The compression of signal ensembles thus requires high computation during clustering, and so the cluster heads must be capable of performing such tasks within their power and computational budgets. Fractional cascading [76] allows queries to be injected at any point in the network. Information is redundantly stored at several sensors, requiring again collaboration and computation to integrate measurements from local groups of sensors.

Other algorithms that exploit correlations in sensor networks include signal compression [77], routing [78], and signal processing tasks [79–81]. The general approach consists of clustering nodes that observe correlated signals and then performing local processing, routing, or compression at a node chosen as a cluster head; the process continues iteratively until a single cluster is obtained. Unfortunately, clustering techniques require collaboration amongst sensors, which increases power consumption for the nodes due to message passing inside clusters. Furthermore, not all sensor network architectures can support the computational complexity of the signal processing algorithms.

In contrast to these approaches, our proposed framework involves no collaboration

among the sensors, has low computational complexity, and facilitates easy measurement aggregation. Section 4.3 elaborates on these and other benefits.

## 4.2 Distributed Sensing Using Random Projections

In this section we describe the mechanics of implementing DCS in a sensor network environment. In the next section, we highlight the unique benefits afforded by such an approach.

### 4.2.1 Incoherent Measurements

We consider a collection of  $J$  synchronized sensor nodes that observe signals obeying one of the JSMs or their extensions (as described in Section 2.4.1). Each sensor *independently* collects a set of incoherent measurements and transmits them to a data sink. The signals are then recovered *jointly* using algorithms discussed in Section 2.4.2. We emphasize that, thanks to the *universal* nature of random measurements, the sensors need not be informed of the sparsity-inducing basis for the signals; this information is only required to perform recovery at the decoder.

We assume that sensor  $j$  acquires the  $N$ -sample signal  $\mathbf{x}_j$  observed during a time interval  $[t_0, t_0 + T]$  and computes a given number of measurements  $M_j$ . The period  $[t_0, t_0 + T]$  could be the complete duration of the signal of interest or could correspond to a length- $N$  block of a longer signal; the above process can be repeated periodically. We denote the measurement vector by  $\mathbf{y}_j = \Phi_j \mathbf{x}_j$ , where  $\Phi_j$  is the *measurement matrix* for sensor  $j$ ;  $\Phi_j$  is  $M_j \times N$  and, in general, the entries of  $\Phi_j$  are different for each  $j$ .

Since all measurements have the same relevance for signal recovery, their values are quantized using the same scheme for each index  $m$ ; the distortion in the recovery due to quantization is bounded [82].

The CS and DCS frameworks require knowledge during recovery of the measurement matrix  $\Phi_j$  for the different sensors  $j = 1, \dots, J$ . This can be accomplished by constructing each measurement matrix using a pseudorandom number generator, whose seed could be provided by the data sink or computed as a function of the node ID. While most of the existing theory for CS encoding applies specifically to random Gaussian or Bernoulli measurements, there is active research into developing lower-complexity alternatives [26, 83]. We have strong experimental evidence that structured measurement matrices  $\Phi_j$  (involving, for example, an FIR filter with pseudorandom taps [83]) can provide suitable mutual coherence with the sparse basis  $\Psi$ .

#### 4.2.2 Communication to the Data Sink

Each quantized measurement  $\hat{y}_j(m)$  is transmitted to the sink together with its timestamp  $t_0$ , index  $m$ , and node ID  $j$ . This is the only information necessary from the sensors to recover the signals. Since the measurements can arrive out of order, they can be sent individually over the network or grouped into packets if desired. Many different options exist for routing the measurements, including TreeCast [84] and DIMENSIONS [75].

### 4.2.3 Joint Recovery

As the measurements are received by the data sink, the measurement matrices  $\Phi_j$  for the different sensors are built accordingly through the same procedure as in the sensors. Once the data sink receives all  $M_j$  measurements from each sensor — or alternatively, once it starts receiving measurements for the next measurement period (beginning at  $t_0 + T$ ) — the data sink can begin recovering the signal ensemble as detailed in Section 2.4.2.

## 4.3 Advantages of Distributed Compressive Sensing for Sensor Networks

Our DCS implementation for sensor networks is robust and widely applicable in sensor network scenarios. This section describes in more detail several of the desirable features.

### 4.3.1 Simple, Universal Encoding

DCS coding is particularly appealing when we employ random projections at the sensors. Random projections are *universal* in the sense that they are incoherent with any fixed sparsity basis  $\Psi$  [49]. In fact, using the same set of random measurements the decoder can attempt to recover the signals using any supposed sparse basis  $\Psi$  or JSM. In addition to being universally incoherent, the CS/DCS random measurements are also *future-proof*: if a better sparsity-inducing basis is found (or a better JSM is proposed), then the same random measurements can be used to recover an even more

accurate view of the environment without requiring any changes in the deployed sensing hardware. Additionally, DCS can be applied to any number of sensors  $J \geq 2$ , and the sensors need not know their physical locations (other than to network their data).

The CS/DCS frameworks, in which measurements can be obtained with low complexity and without collaboration, also shifts the computational load of recovery from the sensor network to the data sink or cluster head. Each sensor only needs to compute its incoherent projections of the signal it observes, while the data sink or cluster head recovers all of the signals. This *computational asymmetry* is desirable in many sensor networks since data sinks and cluster heads have typically more computational power than sensor nodes.

#### 4.3.2 Robustness, Progressivity, and Resiliency

DCS enjoys remarkable *robustness* properties thanks to the robustness of the CS framework. CS measurements have been shown to be robust to quantization and noise [62, 82], making the framework applicable to real world settings. Additionally, the incoherent measurements coming from each sensor have equal priority, unlike transform coefficients in current coders. Thus, the CS measurements can be transmitted and received in any order. Signal recovery can be attempted using any number of the received measurements — as more measurements are received they allow a *progressively better recovery* of the data [49].

In this sense, DCS is automatically *robust to packet loss* in wireless sensor networks; any loss of measurements leads to a graceful degradation in the recovery

quality. This *loss resiliency* is particularly useful, as errors in wireless sensor network transmissions often cause as many as 10 – 30% of the packets to be dropped [85]. This effect is exacerbated in multi-hop networks.

One existing approach that is robust to packet drops is multiple description coding [86, 87]. These techniques enable data recovery at varying levels of quality depending on the number of packets that arrive. Unfortunately, multiple description coding techniques for distributed source coding have not been fully developed [88]. Another approach uses layered coding for unequal bit error protection, where the first layer is highly protected with strong channel coding and is also used as side information when decoding the second layer [89]. This layered approach also increases demands on the system resources because the stronger channel code requires substantial redundancy in terms of channel resources and power consumption.

#### 4.3.3 Security

Using a pseudorandom basis (with a random seed) effectively implements *encryption*: the randomized measurements will themselves resemble noise and be meaningless to an observer who does not know the seed.

#### 4.3.4 Fault Tolerance and Anomaly Detection

DCS recovery techniques can be extended to be *fault tolerant*. In the case where a small number of signals may not obey the overall JSM (due to a faulty sensor, for example), the joint recovery techniques can be tailored to detect such anomalies. In the case of JSM-2, for example, after running SOMP to determine the common

support set  $\Omega$ , the data sink could examine each sensor's measurements to check for agreement with  $\Omega$ . Those signals that appear to disagree can then be recovered separately from the remaining (JSM-faithful) nodes.

#### 4.3.5 Adaptivity to Channel Capacity

The DCS measurement and transmission rates can be scaled to adapt to the conditions of the wireless communication channel and the nuances of the observed signals. If, for example, the communication channel capacity is below the required rate to send  $M_j$  measurements, then the sensors can perform rate limitation in a similar manner to congestion control algorithms for communication networks. When the data sink detects congestion in the communication channel, it can send a congestion notification (using a trickle of feedback) to the nodes so that the bit rate of the information sent is reduced in one of two ways. First, the sensors could increase the quantization stepsize of the measurements, since the CS/DCS recovery is robust to quantization. Second, the sensors could reduce the number of measurements taken for each signal: due to the resiliency of CS measurements, the effect of having few measurements on the recovery distortion is gradual. Thus, the CS/DCS measurement process can easily *scale* to match the transmission capacity of the communication channel, which is reminiscent of joint source-channel coding.

#### 4.3.6 Information Scalability

Incoherent measurements obtained via DCS can be used to recover *different levels of information* about the sensed signals. It has been shown [90] that the CS framework



is *information scalable* beyond signal recovery to a much wider range of statistical inference tasks, including estimation, detection, and classification. Depending on the situation, the lower levels of information about the signals can often be extracted using lower computational complexity or fewer incoherent measurements than would be required to recover the signals. For example, statistical detection and classification do not require recovery of the signal, but only require an estimate of the relevant *sufficient statistics*. Consequently, it is possible to directly extract such statistics from a small number of random projections without ever recovering the signal. As a result, significantly fewer measurements are required for signal detection than for signal recovery [90]. Furthermore, as in recovery, random measurements are again *universal*, in the sense that with high probability the sufficient statistics can be extracted from them regardless of the signal structure.

As a first example, we consider sensor networks for surveillance applications [79]. Typically, a detection algorithm is executed continuously on the sensed data; when the algorithm returns an event detection, other algorithms such as classification, localization, and tracking are executed. These algorithms require a larger amount of information from the signals than that of detection. In our DCS scheme, we can adapt the measurement rate of the sensor nodes according to the tasks being performed. We apply a low measurement rate for detection; once the detection returns an event, the measurement rate is increased to that required by the other tasks.

As another example, one may be interested in estimating linear functions of the

sensed signals

$$\mathbf{v} = \sum_j \omega_j \mathbf{x}_j;$$

examples include averages and linear interpolations. Thanks to the linearity of the CS/DCS measurement process, we can extract such information from the incoherent measurements without first recovering the signals  $\mathbf{x}_j$ . More specifically, assuming we use the same measurement process  $\Phi_j = \Phi$  at each sensor, we can write

$$\Phi \mathbf{v} = \sum_j \omega_j \Phi \mathbf{x}_j = \sum_j \omega_j \mathbf{y}_j.$$

Assuming that  $\mathbf{v}$  is sparse, it can be recovered from  $\Phi \mathbf{v}$  using standard CS techniques. Thus, by aggregating the measurements  $\mathbf{y}_j$  using the desired linear function we can *directly* obtain incoherent measurements of  $\mathbf{v}$  without recovering the  $\mathbf{x}_j$ . We also note that the measurement vectors can be aggregated using matched source-channel communication [24, 25], in which the wireless nodes collaborate to coherently send their measurements so that a receiver directly obtains the weighted sum. This could enable a significant reduction in power. Such aggregation can also be implemented hierarchically in frameworks such as TreeCast [84] or DIMENSIONS [75].

## 4.4 Experiments

In this section, we consider four different sensor network datasets. Although the signals we consider are not strictly sparse, we see that the JSM models provide a good approximation for the joint sparsity structure and that DCS offers a promising approach for such sensing environments.

#### 4.4.1 Environmental Sensing

The first three datasets [91] contain temperature, humidity, and light readings from a group of 48 nodes deployed at the offices of Intel Research Labs in Berkeley, CA.<sup>2</sup> The signals in Figures 4.1(a), 4.2(a) and 4.3(a) were recorded in an office environment and therefore exhibit periodic behavior caused by the activity levels during day and night. Furthermore, there are small fluctuations at each one of these states; thus we expect the signals to be compressible both in the Fourier and wavelet domains. Since the signals are observations of physical processes, they are smoothly varying in time and space; this causes the sensor readings to be close in value to each other, a situation well captured by the JSM-1 and JSM-2 models.

We now confirm the joint sparsity of the signals under the JSM-2 model. The top panel in Figure 4.4 shows the distortion of the best  $K$ -term wavelet approximation for each signal in the light dataset as  $K$  increases. The figure shows that a modest value of  $K = 100$  gives low distortion for all signals. However, the union over all signals of the  $K$  best wavelet basis vectors per signal has size greater than  $K$ . The bottom panel in Figure 4.4 shows the size of this union (the “joint support” for the signals under JSM-2) as  $K$  increases. We see that approximately  $|\Omega| = 200$  vectors are required to include the  $K = 100$  most significant vectors for each signal, which makes the JSM-2 model feasible due to the shared compactness of the representation. Similar results are observed for the other datasets, which are compressible in the

---

<sup>2</sup>For the purposes of our experiments, we select signals of length  $N = 1024$  and interpolate small amounts of missing data.

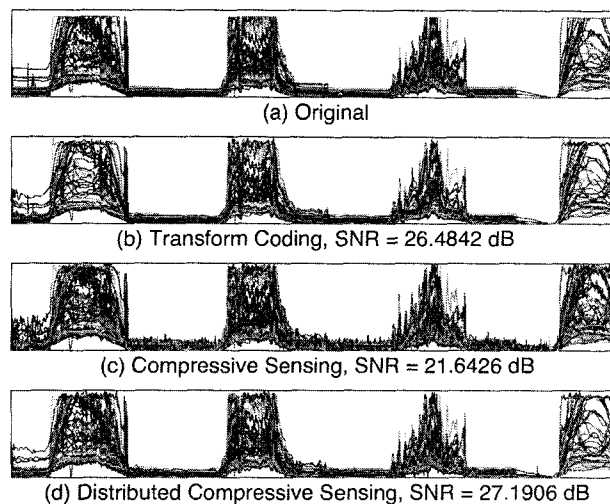


Figure 4.1 : *Recovery of light intensity signals from 48 sensors with length  $N = 1024$ . (a) Original signals; (b) wavelet thresholding using 100 coefficients per sensor, average SNR = 26.48dB; (c) separate recovery of each signal using CS from  $M = 400$  random projections per sensor, average SNR = 21.64dB; (d) joint recovery of the signal ensemble using DCS from  $M = 400$  random projections per sensor, average SNR = 27.19dB.*

wavelet domain as well. Thus, we expect that such datasets can be recovered from incoherent projections using DCS with the appropriate sparsity inducing bases.

We now consider a hypothetical implementation of DCS for these signals. For the light intensity signal we take  $M = 400$  random Gaussian measurements per sensor and compare DCS recovery (via DCS-SOMP using wavelets as the sparsity basis) with separable OMP recovery. For comparison, we also compare to wavelet thresholding at each signal using 100 terms. Figure 4.1 shows the recovery of the light intensity signal ensemble. We see average SNRs of 26.48dB, 21.64dB, and 27.19dB for wavelet thresholding, separate CS, and DCS recovery, respectively. The DCS recovery algorithm identifies the common structure emphasized by JSM-2, recovering salient

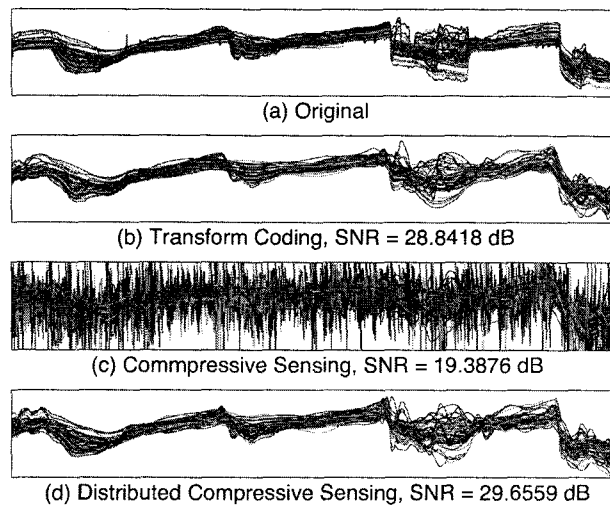


Figure 4.2 : *Recovery of humidity signals from 48 sensors with length  $N = 1024$ . (a) Original signals; (b) wavelet thresholding using 20 coefficients per sensor, average SNR = 28.84dB; (c) separate recovery if each signal using CS from  $M = 80$  random projections per sensor, average SNR = 19.39dB; (d) joint recovery of the signal ensemble using DCS from  $M = 80$  random projections per sensor, average SNR = 29.66dB.*

common features for all signals in the ensemble in addition to many of the distinct features in each signal. Similar results are seen for the humidity and temperature datasets in Figures 4.2, 4.3, and 4.5.

To illustrate progressivity, Figure 4.6 also plots the CS (OMP) and DCS (DCS-SOMP) recovery errors for the temperature signal ensemble at a variety of measurement rates  $M$ . SOMP recovery is superior at low and moderate rates, yet it is surpassed by OMP at high rates. This illustrates the applicability of the JSM-2 model, which becomes less valid as the very fine features of each signal (which vary between sensors) are incorporated. A joint recovery algorithm tailored to this fact would likely outperform both approaches.

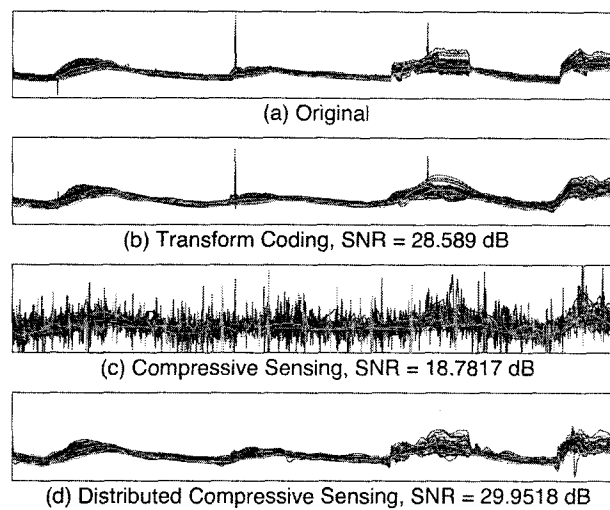


Figure 4.3 : *Recovery of temperature signals from 48 sensors with length  $N = 1024$ . (a) Original signals; (b) wavelet thresholding using 20 coefficients per sensor, average SNR = 28.59dB; (c) separate recovery of each signal using CS from  $M = 80$  random projections per sensor, average SNR = 18.78dB; (d) joint recovery of the signal ensemble using DCS from  $M = 80$  random projections per sensor, average SNR = 29.95dB.*

#### 4.4.2 Acoustic Sensing

Our fourth dataset [80] contains audio recordings of military vehicles from a 16-microphone sensor network array from the SITEX02 experiment of the DARPA SensIT program. The audio signals are compressible in the Fourier domain and follow the JSM-2 model (see Figure 4.7). Figure 4.8 shows an example DCS recovery (using SOMP with the Fourier sparse basis); the results are similar to those seen in the previous datasets.

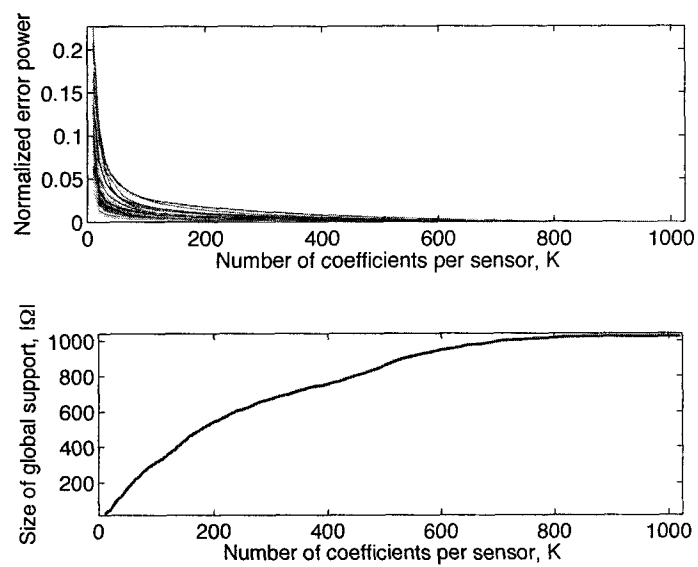


Figure 4.4 : Top: Quality of approximation of light intensity signals as a function of the number  $K$  of wavelet coefficients used per sensor. When  $K \geq 100$ , the approximations yield low distortion; thus the signals are compressible. Bottom: Number of wavelet vectors required to include the  $K$  largest wavelet coefficients for each signal. The slope of the curve is much smaller than  $J = 48$ , meaning that the supports of the compressible signals overlap, and that the ensemble is well represented by the JSM-2 model.

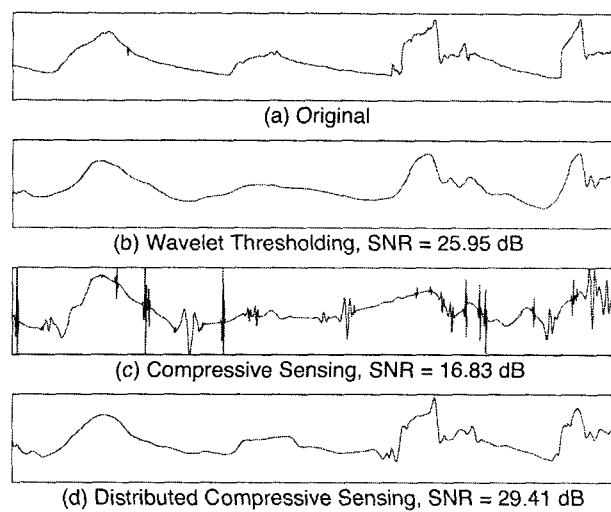


Figure 4.5 : Recovery of temperature signal #41 (extracted from Figure 4.3). (a) Original signal; (b) separate recovery of each signal using wavelet thresholding,  $\text{SNR} = 25.95\text{dB}$ ; (c) recovery using CS,  $\text{SNR} = 16.83\text{dB}$ ; (d) joint recovery of the signal ensemble using DCS,  $\text{SNR} = 29.41\text{dB}$ .

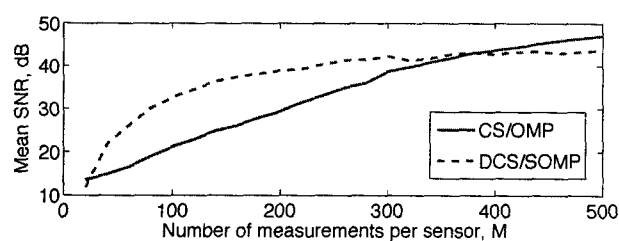


Figure 4.6 : Average SNR of temperature signals recovered from  $M$  measurements per sensor using CS (OMP) and DCS (DCS-SOMP).



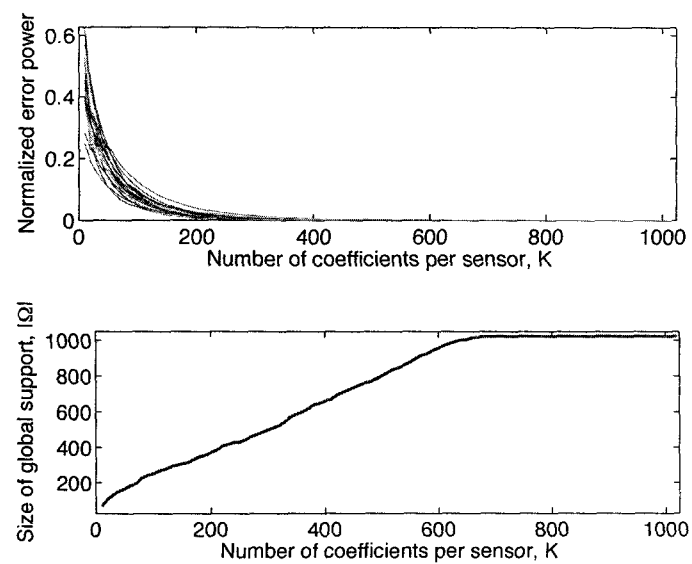


Figure 4.7 : Top: Quality of approximation of vehicle audio signals as a function of the number  $K$  of Fourier coefficients used per sensor. Bottom: Number of Fourier vectors required to include the  $K$  largest Fourier coefficients for each signal.

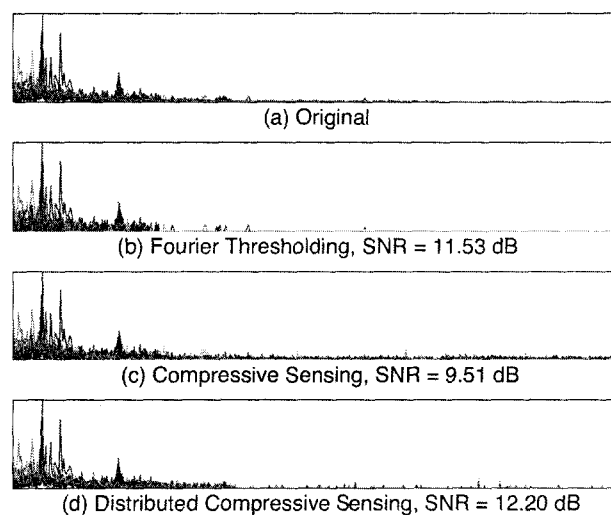


Figure 4.8 : Fourier coefficients for recovery of vehicle audio signals from 18 sensors with length  $N = 1024$ . (a) Original signals; (b) Fourier thresholding using 100 coefficients per sensor, average SNR = 11.53dB; (c) separate recovery using CS from  $M = 400$  random projections per sensor, average SNR = 9.51dB; (d) joint recovery using DCS from  $M = 400$  random projections per sensor, average SNR = 12.20dB.

## Chapter 5

### Compressive Sensing for Wavelet-Sparse Signals

The signal recovery algorithms for CS listed in Section 2.3.4 are generic, in the sense that they do not exploit any structure (aside from sparsity) that may exist in the sensed signals. An important subclass of sparse signals, however, is the class of *piecewise smooth* signals — many punctuated real-world phenomena give rise to such signals [2]. The wavelet transform of a piecewise smooth signal yields a sparse, *structured* representation of signals in this class: the largest coefficients tend to form a connected subtree of the wavelet coefficient tree. While other methods have been proposed for fast recovery of wavelet-sparse signals [46, 63], these methods do not fully exploit this connectedness property. In this chapter,<sup>1</sup> we propose algorithms for CS signal recovery that are specially tailored for the structure of sparse wavelet representations.

#### 5.1 The Structure of Multiscale Wavelet Transforms

Without loss of generality, we focus on 1-D signals, although similar arguments apply for 2-D and multidimensional data. Consider a signal  $\mathbf{x}$  of length  $N = 2^I$ , for an

---

<sup>1</sup>This work is in collaboration with Michael B. Wakin and Richard G. Baraniuk [92–94]

integer value of  $I$ . The wavelet representation of  $\mathbf{x}$  is given by

$$\mathbf{x} = v_0\nu + \sum_{i=0}^{I-1} \sum_{j=0}^{2^i-1} w_{i,j} \psi_{i,j}, \quad (5.1)$$

where  $\nu$  is the scaling function and  $\psi_{i,j}$  is the wavelet function at scale  $i$  and offset  $j$ . The wavelet transform consists of the scaling coefficient  $v_0$  and wavelet coefficients  $w_{i,j}$  at scale  $i$ ,  $0 \leq i \leq I-1$ , and position  $j$ ,  $0 \leq j \leq 2^i-1$ . In terms of our earlier matrix notation,  $\mathbf{x}$  has the representation  $\mathbf{x} = \Psi\theta$ , where  $\Psi$  is a matrix containing the scaling and wavelet functions as columns, and  $\theta = [v_0 \ w_{0,0} \ w_{1,0} \ w_{1,1} \ w_{2,0} \dots]^T$  is the vector of scaling and wavelet coefficients. We are, of course, interested in sparse and compressible  $\theta$ .

In a typical 1-D wavelet transform, each coefficient at scale  $j \in \{1, \dots, J := \log_2(N)\}$  describes a portion of the signal of size  $O(2^{-j})$ . With  $2^{j-1}$  such coefficients at each scale, a binary tree provides a natural organization for the coefficients. Each coefficient at scale  $j < \log_2(N)$  has 2 *children* at scale  $j+1$ , and each coefficient at scale  $j > 1$  has one *parent* at scale  $j-1$ .

### 5.1.1 Deterministic Signal Models

Due to the analysis properties of wavelets, coefficient values tend to persist through scale. A large wavelet coefficient (in magnitude) generally indicates the presence of a singularity inside its support; a small wavelet coefficient generally indicates a smooth region. Thanks to the nesting of child wavelets inside their parents, edges in general manifest themselves in the wavelet domain as chains of large coefficients propagating across scales in the wavelet tree; we call this phenomenon the *persistence property*.

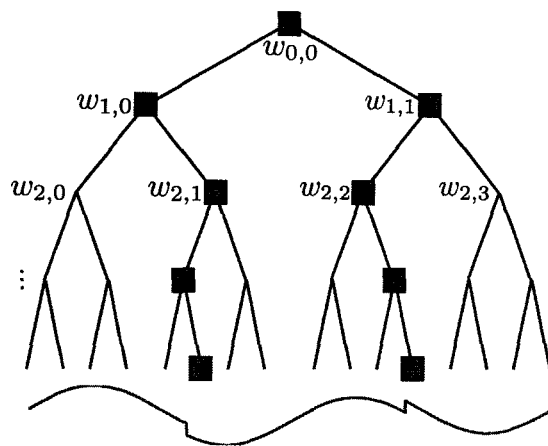


Figure 5.1 : *Binary wavelet tree for a 1-D signal. The squares denote the large wavelet coefficients that arise from the discontinuities in the piecewise smooth signal drawn below; the support of the large coefficients forms a rooted, connected tree.*

Additionally, wavelet coefficients also have *exponentially decaying magnitudes* at finer scales [2]. This causes the significant wavelet coefficients of piecewise smooth signals to concentrate within a connected subtree of the wavelet binary tree. This deterministic structure is illustrated in Figure 5.1.

In specific cases, we might observe signals that are piecewise smooth but that do not exhibit the connected subtree structure. The reasons for this are twofold. First, since wavelets are bandpass functions, wavelet coefficients oscillate between positive and negative values around singularities. Second, due to the linearity of the wavelet transform, two or more singularities in the signal may cause destructive interference among coarse scale wavelet coefficients; that is, the persistence of the wavelets across scale is *weaker at coarser scales*. Either of these factors may cause the wavelet coefficient corresponding to a discontinuity to be small yet have large

children, yielding a non-connected set of meaningful wavelet coefficients.

In summary, we have identified several properties of wavelet expansions:

- large/small values of wavelet coefficients generally persist across the scales of the wavelet tree;
- persistence becomes stronger as we move to finer scales; and
- the magnitude of the wavelet coefficients decreases exponentially as we move to finer scales.

### 5.1.2 Probabilistic Signal Models

The properties identified in Section 5.1.1 induce a joint structure among the wavelet coefficients that is far stronger than simple sparsity. We also note that the sparsity of the wavelet transform causes the coefficients to have a peaky, non-Gaussian distribution. The Hidden Markov Tree model (HMT) [95, 96] offers one modeling framework that succinctly and accurately captures this joint structure. HMT modeling has been used successfully to improve performance of denoising, classification, and segmentation algorithms for wavelet-sparse signals.

The HMT model sets the probability density function of each wavelet to be a Gaussian mixture density with a hidden binary state that determines whether the coefficient is large or small. The persistence across scale is captured by a tree-based Markov model that correlates the states of parent and children coefficients. The following properties are captured by the HMT.

### Non-Gaussianity

Sparse coefficients can be modeled probabilistically using a mixture of Gaussians: one component features a large variance that models large nonzero coefficients and receives a small weight (to encourage few such coefficients), while a second component features a small variance that models small and zero-valued coefficients and receives a large weight. We distinguish these two components by associating to each wavelet coefficient  $\theta(n)$  an unobserved hidden state  $\mathbf{s}(n) \in \{S, L\}$ ; the value of  $\mathbf{s}(n)$  determines which of the two components of the mixture model is used to generate  $\theta(n)$ . Thus we have

$$f(\theta(n)|\mathbf{s}(n) = S) = \mathcal{N}(0, \sigma_{S,n}^2),$$

$$f(\theta(n)|\mathbf{s}(n) = L) = \mathcal{N}(0, \sigma_{L,n}^2),$$

with  $\sigma_{L,n}^2 > \sigma_{S,n}^2$ . To generate the mixture, we apply a probability distribution to the available states:  $P(\mathbf{s}(n) = S) = p_n^S$  and  $P(\mathbf{s}(n) = L) = p_n^L$ , with  $p_n^S + p_n^L = 1$ .

### Persistence

The perpetuation of large and small coefficients from parent to child is well-modeled by a Markov model that links coefficient states. This induces a Markov tree where the state  $\mathbf{s}(n)$  of a coefficient  $\theta(n)$  is affected only by the state  $\mathbf{s}(\mathcal{P}(n))$  of its parent  $\mathcal{P}(n)$ . The Markov model is then completely determined by the set of state transition matrices for the different coefficients  $\theta(n)$  at wavelet scales  $1 < j \leq J$ :

$$\mathbf{A}_n = \begin{bmatrix} p_n^{S \rightarrow S} & p_n^{S \rightarrow L} \\ p_n^{L \rightarrow S} & p_n^{L \rightarrow L} \end{bmatrix}.$$

The persistence property implies that the values of  $p_n^{L \rightarrow L}$  and  $p_n^{S \rightarrow S}$  are significantly larger than their complements. If we are provided the hidden state probabilities for the wavelet coefficient in the coarsest scale  $p_1^S$  and  $p_1^L$ , then the probability distribution for any hidden state can be obtained recursively:

$$P(\mathbf{s}(n) = L) = p_{\mathcal{P}(n)}^S p_n^{S \rightarrow L} + p_{\mathcal{P}(n)}^L p_n^{L \rightarrow L}.$$

As posed, the HMT parameters include the probabilities for the hidden state  $\{p_1^S, p_1^L\}$ , the state transition matrices  $\mathbf{A}_n$ , and Gaussian distribution variances  $\{\sigma_{L,n}^2, \sigma_{S,n}^2\}$  for each of the wavelet coefficients  $\theta(n)$ . To simplify the model, the coefficient-dependent parameters are made equal for all coefficients within a scale; that is, the new model has parameters  $\mathbf{A}_j$  for  $1 < j \leq J$  and  $\{\sigma_{L,j}^2, \sigma_{S,j}^2\}$  for  $1 \leq j \leq J$ .

### Magnitude Decay

To enforce the decay of the coefficient magnitudes across scale, the variances  $\sigma_{L,j}^2$  and  $\sigma_{S,j}^2$  are modeled so that they decay exponentially as the scale becomes finer [97]:

$$\begin{aligned}\sigma_{L,j}^2 &= C_{\sigma_L} 2^{-j\alpha_L}, \\ \sigma_{S,j}^2 &= C_{\sigma_S} 2^{-j\alpha_S}.\end{aligned}$$

Since the wavelet coefficients that correspond to signal discontinuities decay slower than those representing smooth regions, the model sets  $\alpha_S \geq \alpha_L$ .

### Scale-Dependent Persistence

To capture the weaker persistence present in the coarsest scales, the values of the state transition matrices  $\mathbf{A}_j$  follow a model that strengthens the persistence at finer

scales [97]. Additionally, the model must reflect that in general, any large parent generally implies only one large child (that which is aligned with the discontinuity). This implies that the probability that  $s(n) = L$ , given that  $s(\mathcal{P}(n)) = L$ , should be roughly 1/2. HMT accounts for both factors by setting

$$p_j^{L \rightarrow L} = \frac{1}{2} + C_{LL}2^{-\gamma_L j}, \quad p_j^{L \rightarrow S} = \frac{1}{2} - C_{LL}2^{-\gamma_L j}$$

$$p_j^{S \rightarrow S} = 1 - C_{SS}2^{-\gamma_S j}, \quad \text{and } p_j^{S \rightarrow L} = C_{SS}2^{-\gamma_S j}.$$

### Estimation

We can obtain estimates of all parameters

$$\Pi = \{p_1^S, p_1^L, \alpha_S, \alpha_L, C_{\sigma_L}, C_{\sigma_S}, \gamma_L, \gamma_S, C_{LL}, C_{SS}\}$$

for a set of coefficients  $\theta$  using maximum likelihood estimation:

$$\Pi_{ML} = \arg \max_{\Pi} f(\theta|\Pi). \quad (5.2)$$

The expectation-maximization (EM) algorithm in [95] efficiently performs this estimation. Similarly, one can obtain the state probabilities  $P(s(n) = S|\theta, \Pi)$  using the Viterbi algorithm; the state probabilities for a given coefficient will be dependent on the states and coefficient values of all of its predecessors in the wavelet tree.

We aim to exploit this structure to improve the computational complexity and reduce the number of measurements required for recovery of piecewise smooth signals. In the next two sections, we will present two different algorithms that model the sparse wavelet structure during the signal recovery.



## 5.2 Iterative Greedy Algorithms for Signal Recovery

Not surprisingly, we observe for piecewise smooth signals that greedy recovery algorithms tend to select wavelet coefficients located near the top of the tree first and then continues selecting down the tree, effectively building a connected tree that contains the most significant coefficients from the top down. This suggests that it may not be necessary for the recovery algorithm to check *all* possible coefficients at each stage. Rather, the next most important coefficient at each stage is likely to be among the children of the currently selected coefficients.

We must refine this heuristic, however, to obtain an effective algorithm. In particular, for real world piecewise smooth signals, the nonzero coefficients generally do not form a perfect connected subtree. The reasons for this are twofold. First, since wavelets are bandpass functions, wavelet coefficients oscillate positive and negative around singularities [98]. Second, due to the linearity of the wavelet transform, two or more singularities in the signal may cause destructive interference among large wavelet coefficients. Either of these factors may cause the wavelet coefficient corresponding to a discontinuity to be small yet have large children, yielding a non-connected set of meaningful wavelet coefficients. We can still define a connected subtree that contains all of the nonzero valued coefficients, however, which will contain some *gaps* consisting of sequences of small or zero values. Our proposed algorithm features a parameter designed to address this complication.

### 5.2.1 Tree Matching Pursuit and Tree Orthogonal Matching Pursuit

The *Tree Matching Pursuit* (TMP) and *Tree Orthogonal Matching Pursuit* (TOMP) algorithms consider only a subset of the basis vectors at each iteration, and then expand that set as significant coefficients are found. For each iteration  $i$ , we define two sets of coefficients  $\mathcal{S}_i$  and  $\mathcal{C}_i$ , which contain the set of *selected* vectors (those vectors that correspond to nonzero coefficients in the estimate  $\hat{\alpha}$ ) and the *candidate* vectors (vectors with zero coefficients in  $\hat{\alpha}$  but whose projections will be evaluated at the next iteration). These sets are initialized as  $\mathcal{S}_0 = \emptyset$  and  $\mathcal{C}_0 = \{1\} \cup \mathcal{D}_b(1)$ , where the  $b$ -depth set of descendants  $\mathcal{D}_b(n)$  is the set of coefficients within  $b$  levels below coefficient  $n$  in the wavelet tree.<sup>2</sup>

At each iteration  $i$ , we search for the dictionary vector index  $n$  in  $\mathcal{S}_i \cup \mathcal{C}_i$  that yields the maximum inner product with the current residual; if the selected index comes from  $\mathcal{C}_i$ , then that index  $n$  and its ancestors, denoted  $\mathcal{A}(n)$ , are moved to the set of selected coefficients  $\mathcal{S}_i$  and removed from  $\mathcal{C}_i$ , and the descendant set  $\mathcal{D}_b(n)$  is added to  $\mathcal{C}_i$ . For TMP and TOMP, we adapt Step 2 of the MP and OMP algorithms (Algorithm 1 and 2, respectively), as shown in Algorithms 5 and 6, respectively.

While the existence of gaps in the wavelet subtree containing the set of meaningful coefficients will hamper the ability to reach some nonzero coefficients, the parameter  $b$  enables us to define a “lookahead” band of candidate coefficients wide enough that

---

<sup>2</sup>An independently obtained algorithm that is also called TOMP is proposed in [99, 100]; this algorithm evaluates the sums of the projections along each wavelet tree branch, rather than the projection of each single coefficient.

Table 5.1 : Computational complexity of CS algorithms.  $N$  = signal length;  $K$  = signal sparsity;  $I$  = convergence factor,  $C$  = oversampling factor;  $B$  = TMP band width.

Algorithm	BP	MP	OMP	$b$ -TMP	$b$ -TOMP
Complexity	$O(N^3 \log(N))$	$O(CKNI)$	$O(CK^2N)$	$O(2^b CK^2 I)$	$O(2^b CK^3)$

each possible gap is contained in the band. This modification has advantages and disadvantages; it is clear from the results shown in Figure 5.2 that the recovery will be the same or better as we add more descendants into  $\mathcal{D}_b(n)$ . However, the computational complexities of  $b$ -TMP and  $b$ -TOMP, given by  $O(2^b CK^2 I)$  and  $O(2^b CK^3)$ , respectively, will increase with  $b$ . For moderate  $b$  both still represent a significant improvement over their generic counterparts, and  $b$ -TOMP improves upon BP by a factor of  $O((N/K)^3)$ ; Table 5.2.1 summarizes the computational complexity of the various algorithms.

### 5.2.2 Experiments

We perform experiments to test the performance of standard and tree-based greedy algorithms on prototypical piecewise smooth signals. We use the standard piecewise constant signal *Blocks* of length  $N = 512$ , and obtain  $M = 200$  measurements using a random Gaussian matrix. Figure 5.2 shows that when the lookahead parameter  $b$  is set to be large enough to bypass discontinuities in the connected subtree, TMP achieves quality similar to that of standard MP while reducing the computational complexity of the recovery by about 50%. Additional experimental results that verify

the robustness of TMP and TOMP to noise can be found in [92].

### 5.2.3 Extensions

#### Complex Wavelet Transform

By using a *complex wavelet transform* (CWT) [98], we can avoid some of the pitfalls of the standard real wavelet transform. The CWT shares the same binary tree structure as the real wavelet transform, but the wavelet functions are complex-valued

$$\psi_c = \psi_r + j\psi_i.$$

The component  $\psi_r$  is real and even, while  $j\psi_i(t)$  is imaginary and odd; they form an approximate Hilbert transform pair. The CWT transform can be easily implemented using a dual-tree structure, where we simply compute two *real* wavelet transforms ( $\psi_r$  and  $\psi_i$ ) in parallel, obtaining the sequences of coefficients  $\alpha_r$  and  $\alpha_i$ . The complex wavelet coefficients are then defined as  $\alpha_c = \alpha_r + j\alpha_i$ .

Note that either the real or the imaginary part of the wavelet coefficients would suffice to recover a real signal; however, the dual representation establishes a strong coherency among the complex magnitudes. Due to the Hilbert transform relationship between the real and imaginary wavelets, when a discontinuity is present and the real (or imaginary) wavelet coefficient is small, the imaginary (or real) wavelet coefficient is large [98]. Thus, the shift-sensitivity of the standard real-wavelet transform is alleviated. As such, when the *b*-TMP algorithm is implemented using the CWT, a much smaller band will be necessary for efficient recovery. Figure 5.2 (bottom right) shows the approximate recovery of *Blocks* using a band of width 1. Unfortunately,

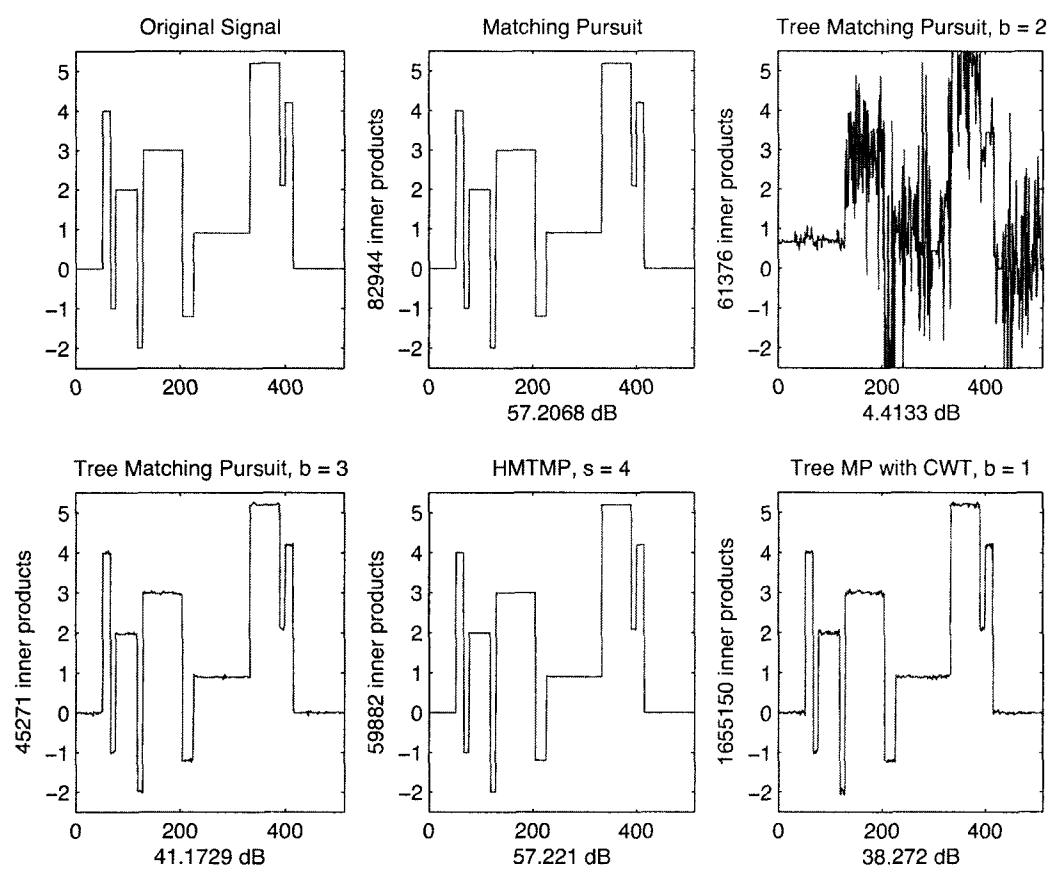


Figure 5.2 : CS recoveries of Blocks signal using several different algorithms. We set  $N = 512$  and obtain  $M = 200$  measurements using a random Gaussian matrix. Axis labels indicate recovery quality and computational complexity, measured by the number of inner products performed by the greedy algorithm. Top left: original signal. Top middle: MP. Top right:  $b$ -TMP with  $b = 2$ ; the band is too small to cover the gaps in the wavelet coefficients and recovery fails. Bottom left:  $b$ -TMP with  $b = 3$ ; the band is large enough to bypass the gap, leading to correct recovery. Bottom middle:  $s$ -MTMP,  $s = 4$ . Bottom right:  $b$ -TMP with the CWT,  $b = 1$ . Both of these modifications yield approximate recovery.

complex coefficients can still interfere destructively, suggesting  $b$  slightly greater than 1 as a conservative choice.

### Random Lookahead

We propose a second modification to TMP that can be applied to both the real and CWT variants. The modification involves a probabilistic definition of the candidate set  $C_t$  at each iteration, based on the HMT. We label the coefficients selected at each iteration as large, i.e.,  $P(\mathbf{s}(n_t) = L) = 1$ , and calculate the conditional probability that each of its descendants is in the  $L$  state. During the candidate set selection, for each leaf  $n_i$  in the subtree containing the set of selected coefficients, we select a random sample of descendants  $D_{\text{HMT}}(n_i)$  according to the probability that each descendant is in the large state, where for a coefficient  $n_j$  that is  $d$  levels below coefficient  $n_i$ ,  $P(\mathbf{s}(n_j) = L) = (p_{n_j}^{S \rightarrow S})^d$ . Thus, coefficients with higher estimates of  $P(\mathbf{s}(n_j) = L)$  are more likely to be selected in the candidate set.

We amend this formulation slightly for easier computation by choosing a constant  $s$  and then constructing  $D_{\text{HMT}}(i)$  by randomly selecting  $s$  descendant coefficients from each scale below  $i$ . We denote by Hidden Markov Tree Matching Pursuit (HMTMP) the TMP algorithm that uses this descendant set in the updates. It is worth noting that by setting  $s = 2^b$ , the descendants selected by the  $s$ -MTMP algorithm contain the set of descendants selected by the original TMP algorithm. The algorithm can enable recovery of signals having large gaps inside the set of meaningful coefficients, while keeping the number of coefficients in the candidate sets relatively small. In Figure 5.2 (bottom middle), we see that by using the random lookahead with  $s = 4$ ,

the significant coefficients below the gap are recovered.

### Regularization and Denoising

When the signal is sparse in the wavelet basis, we can effectively perform denoising by thresholding (see Section 2.2.4 and [32]) by varying the convergence criterion  $\epsilon$  as a function of the signal-to-noise ratio. We then identify only the most significant coefficients using the MP or TMP algorithm and effectively threshold their values at the recovery. CS recovery using the standard algorithms also typically suffers from artifacts since the energy of the signal is not discriminated by band; in this case, a small amount of the energy from the coefficients in the coarsest scales “leaks” to the finer scales and causes low-amplitude, high-frequency artifacts that resemble small-scale noise. By giving preference to the coarsest coefficients over the finest, the TMP algorithms help mitigate this effect during recovery.

## 5.3 Optimization-Based Signal Recovery

The connected subtree structure has been exploited in modifications to greedy algorithms; see the previous section and [99, 100]. While the TMP and TOMP algorithms enable faster recovery and lower recovery distortion by exploiting the connected tree structure of wavelet-sparse signals, for many real-world piecewise smooth signals the nonzero wavelet coefficients generally do not form a perfectly connected subtree. TMP and TOMP used heuristic rules to ameliorate the effect of this phenomenon. However, this considerably increases the computational complexity, and the success of

such heuristics varies markedly between different signals in the proposed class.

### 5.3.1 Iterative Reweighted $\ell_1$ -norm Minimization

When the complexity of the signal is measured using the  $\ell_1$ -norm, individual signal coefficients are penalized according to their magnitude; in contrast, when the  $\ell_0$ -norm is used to measure the signal complexity, the penalty for a nonzero coefficient is independent of its magnitude. The effect of this disparity is reflected in the increase of the overmeasuring factor  $M/K$  between the two algorithms.

A small variation to the  $\ell_1$ -norm penalty function has been suggested to rectify the imbalance between the  $\ell_0$ -norm and  $\ell_1$ -norm penalty functions [101]. The basic goal is to minimize a weighted  $\ell_1$ -norm penalty function  $\|\mathbf{W}\theta\|_1$ , where  $\mathbf{W}$  is a diagonal “weighting” matrix with entries  $\mathbf{W}_{n,n}$  approximately proportional to  $1/|\theta(n)|$ . This creates a penalty function that achieves higher magnitude independence. Since the true values of  $\theta$  are unknown (indeed they are sought), however, an iterative reweighted  $\ell_1$ -norm minimization ( $\text{IR}\ell_1$ ) algorithm is suggested.

The algorithm starts with the solution to the unweighted  $\ell_1$ -norm minimization algorithm (2.2), which we name  $\hat{\theta}^{(0)}$ . The algorithm then proceeds iteratively: on iteration  $i > 0$ , it solves the optimization problem

$$\hat{\theta}^{(i)} = \arg \min_{\theta} \|\mathbf{W}^{(i)}\theta\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{\Phi}\mathbf{\Psi}\theta, \quad (5.3)$$

where  $\mathbf{W}^{(i)}$  is a diagonal reweighting matrix with entries

$$\mathbf{W}^{(i)}(n, n) = \left( \left| \hat{\theta}^{(i-1)}(n) \right| + \epsilon \right)^{-1},$$



$1 \leq n \leq N$ , and  $\epsilon$  is a small regularization constant, and all other entries of  $\mathbf{W}$  are zero. The algorithm can be terminated when the change between consecutive solutions is smaller than an established threshold or after a fixed number of iterations. Each iteration of this algorithm can be posed as a linear program.

### 5.3.2 HMT-Based Weights for $\text{IR}\ell_1$

The  $\text{IR}\ell_1$  algorithm described in Sec. 5.3.1 provides an opportunity to implement flexible signal penalizations while retaining the favorable computational complexity of  $\ell_1$ -norm minimizations.

We now pose a new weight rule for the  $\text{IR}\ell_1$  algorithm that integrates the HMT model to enforce the wavelet coefficient structure during CS recovery. Our weighting scheme, dubbed HMT+ $\text{IR}\ell_1$ , employs the following weighting scheme:

$$\mathbf{W}^{(i)}(n, n) = \left( P\left(\mathbf{s}(n) = L|\hat{\theta}^{(i-1)}, \Pi\right) + \epsilon \right)^{-q}.$$

In words, for each wavelet coefficient in the current estimate we obtain the probability that the coefficient's hidden state is large; in the next iteration, we apply to that coefficient a weight that is inversely proportional to that probability. The parameter  $\epsilon$  is a regularization parameter for cases where  $P(\mathbf{s}(n) = L|\hat{\theta}^{(i-1)})$  is very small, and the exponent  $q$  is a parameter that regulates the strength of the penalization for small coefficients. The goal of this weighting scheme is to penalize coefficients with large magnitudes that have low likelihood of being generated by a wavelet sparse signal; these coefficients are often the largest contributors to the recovery error.

The first step of HMT+ $\text{IR}\ell_1$  consists of an initial training stage in which an EM

algorithm solves (5.2) to estimate the values of the parameters for a representative signal; additionally, the solution  $\hat{\theta}^{(0)}$  for the standard formulation (2.2) is obtained. Subsequently, we proceed iteratively with two alternating steps: a weight update step in which the Viterbi algorithm for state probability calculations is executed for the previous solution  $\hat{\theta}^{(i-1)}$ , and a recovery step in which the obtained weights are used in (5.3) to obtain an updated solution  $\hat{\theta}^{(i)}$ . The convergence criterion for this algorithm is the same as for the  $\text{IR}\ell_1$  algorithm.

Other probabilistic models for wavelet-sparse signals can also be used in combination with the  $\text{IR}\ell_1$  algorithm, including generalized Gaussian densities [102], Gaussian scales mixtures [103], and hierarchical Dirichlet processes [104].

### 5.3.3 Experiments

We now compare the  $\text{IR}\ell_1$  and HMT+ $\text{IR}\ell_1$  algorithms. We use piecewise-smooth signals of length  $N = 1024$ , with 5 randomly placed discontinuities and cubic polynomial pieces with random coefficients. Daubechies-4 wavelets are used to sparsify the signals. Measurements are obtained using a matrix with i.i.d. Gaussian entries. For values of  $M$  ranging from 102 to 512, we test the  $\ell_1$ -norm minimization and the  $\text{IR}\ell_1$ , TMP [93] and HMT+ $\text{IR}\ell_1$  algorithms. We fix the number of iterations for  $\text{IR}\ell_1$  and HMT+ $\text{IR}\ell_1$  to 10. The parameters are set for best performance to  $\epsilon = 0.2$ ,  $q = 0.1$ , and  $\epsilon = 10^{-10}$ . For each  $M$  we perform 100 simulations using different randomly generated signals and measurement matrices.

Figure 5.3 shows the magnitude of the recovery error for each of the algorithms, normalized by the error of the unweighted  $\ell_1$ -norm minimization recovery, as a func-

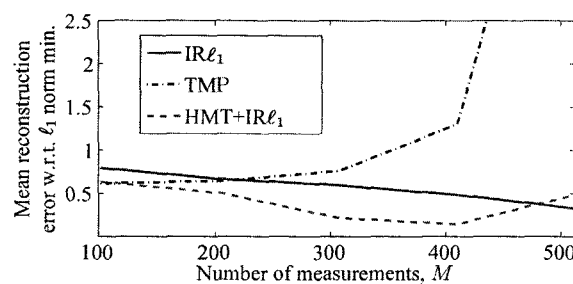


Figure 5.3 : Performance of  $IR\ell_1$  algorithm, normalized by the performance of  $\ell_1$ -norm minimization. Since all values are less than 1,  $IR\ell_1$  and HMT+ $IR\ell_1$  consistently outperforms  $\ell_1$ -norm minimization.

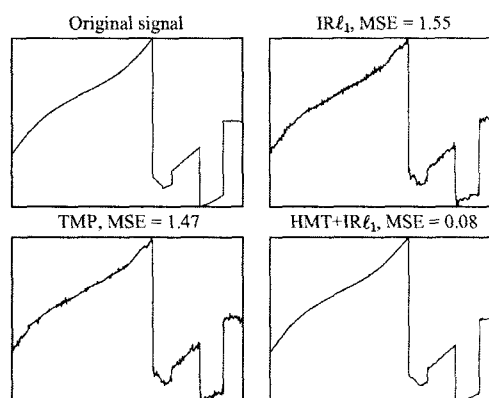


Figure 5.4 : Example outputs for the recovery algorithms.

tion of the iteration count. Figure 5.4 shows a recovery example. TMP performs well for smaller numbers of measurements  $M$ .  $IR\ell_1$  consistently outperforms  $\ell_1$  minimization. Our proposed HMT+ $IR\ell_1$  algorithm outperforms  $IR\ell_1$  for most values of  $M$ . For large  $M$  near  $N/2$ , HMT+ $IR\ell_1$  becomes less efficient than  $IR\ell_1$ ; we speculate that at this stage the recovered signal has roughly equal numbers of large and small wavelet coefficients, which begins to violate the HMT model. Figure 5.4 plots the various recovered signals for one realization of the experiment, with  $M = 300$ .

---

**Algorithm 5** Tree Matching Pursuit

---

Inputs: CS Matrix  $\Upsilon = \Phi\Psi$ , measurements  $\mathbf{y}$

Outputs:  $K$ -sparse approximation  $\hat{\mathbf{x}}$

Initialize:  $\hat{\mathbf{x}}_0 = 0$ ,  $\mathbf{r} = \mathbf{y}$ ,  $i = 0$ ,  $\mathcal{S}_0 = \emptyset$ ,  $\mathcal{C}_0 = \{1\} \cup \mathcal{D}_b(1)$ .

**while** halting criterion false **do**

1.  $i \leftarrow i + 1$

2.  $\mathbf{b} \leftarrow \Upsilon^T \mathbf{r}$  {form residual signal estimate}

3.  $\mathbf{b}|_{(\mathcal{S}_{i-1} \cup \mathcal{C}_{i-1})^c} = 0$  {restrict search to selected  
and candidate coefficients}

4.  $\hat{\mathbf{x}}_i \leftarrow \hat{\mathbf{x}}_{i-1} + \mathfrak{T}(\mathbf{b}, 1)$  {update largest magnitude coefficient  
in signal estimate}

5.  $\mathbf{r} \leftarrow \mathbf{y} - \Upsilon \mathfrak{T}(\mathbf{b}, 1)$  {update measurement residual}

6.  $\omega \leftarrow \text{supp}(\mathfrak{T}(\mathbf{b}, 1))$

**if**  $\omega \subseteq \mathcal{C}_{i-1}$  **then**

7a.  $\mathcal{S}_i = \mathcal{S}_{i-1} \cup \omega \cup \mathcal{A}(\omega)$  {update selected coefficient set}

7b.  $\mathcal{C}_i = (\mathcal{C}_{i-1} \setminus (\omega \cup \mathcal{A}(\omega))) \cup \mathcal{D}_b(\omega)$  {update candidate coefficient set}

**else**

7c.  $\mathcal{S}_i = \mathcal{S}_{i-1}$ ,  $\mathcal{C}_i = \mathcal{C}_{i-1}$  {preserve selected and candidate sets}

**end if**

**end while**

return  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}}_i$

---

---

**Algorithm 6** Tree Orthogonal Matching Pursuit

---

Inputs: CS Matrix  $\Upsilon = \Phi\Psi$ , measurements  $\mathbf{y}$

Outputs:  $K$ -sparse approximation  $\hat{\mathbf{x}}$

initialize:  $\hat{\mathbf{x}}_0 = 0$ ,  $\mathbf{r} = \mathbf{y}$ ,  $\Omega = \emptyset$ ,  $i = 0$ ,  $\mathcal{S}_0 = \emptyset$ ,  $\mathcal{C}_0 = \{1\} \cup \mathcal{D}_b(1)$ .

**while** halting criterion false **do**

1.  $i \leftarrow i + 1$

2.  $\mathbf{b} \leftarrow \Upsilon^T \mathbf{r}$  {form residual signal estimate}

3.  $\mathbf{b}|_{(\mathcal{S}_{i-1} \cup \mathcal{C}_{i-1})^c} = 0$  {restrict search to selected  
and candidate coefficients}

4.  $\Omega \leftarrow \Omega \cup \text{supp}(\mathfrak{T}(\mathbf{b}, 1))$  {add index of largest magnitude  
coefficient to signal support}

5.  $\hat{\mathbf{x}}_i|_{\Omega} \leftarrow \Upsilon_{\Omega}^{\dagger} \mathbf{y}$ ,  $\hat{\mathbf{x}}_i|_{\Omega^c} \leftarrow 0$  {form signal estimate}

6.  $\mathbf{r} \leftarrow \mathbf{y} - \Upsilon \hat{\mathbf{x}}_i$  {update measurement residual}

7.  $\omega \leftarrow \text{supp}(\mathfrak{T}(\mathbf{b}, 1))$

**if**  $\omega \subseteq \mathcal{C}_{i-1}$  **then**

8a.  $\mathcal{S}_i = \mathcal{S}_{i-1} \cup \omega \cup \mathcal{A}(\omega)$  {update selected coefficient set}

8b.  $\mathcal{C}_i = (\mathcal{C}_{i-1} \setminus (\omega \cup \mathcal{A}(\omega))) \cup \mathcal{D}_b(\omega)$  {update candidate coefficient set}

**else**

8c.  $\mathcal{S}_i = \mathcal{S}_{i-1}$ ,  $\mathcal{C}_i = \mathcal{C}_{i-1}$  {preserve selected and candidate sets}

**end if**

**end while**

return  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}}_i$

---

## Chapter 6

### Model-Based Compressive Sensing

Most research in CS has focused primarily on reducing both the number of measurements  $M$  (as a function of  $N$  and  $K$ ) and on increasing the robustness and reducing the computational complexity of the recovery algorithm. Today's state-of-the-art CS systems can robustly recover  $K$ -sparse and compressible signals from just  $M = \mathcal{O}(K \log(N/K))$  noisy measurements using polynomial-time optimization solvers or greedy algorithms (see Section 2.3.4).

While this represents significant progress from Nyquist-rate sampling, our contention in this chapter<sup>1</sup> is that it is possible to do even better by more fully leveraging concepts from state-of-the-art signal compression and processing algorithms. In many such algorithms, the key ingredient is a more realistic *structured sparsity signal model* that goes beyond simple sparsity by codifying the inter-dependency *structure* among the signal coefficients  $\theta$ .<sup>2</sup> For instance, modern wavelet image coders exploit

---

<sup>1</sup>This work is in collaboration with Richard G. Baraniuk, Volkan Cevher, and Chinmay Hegde [18, 105].

<sup>2</sup>Obviously, sparsity and compressibility correspond to simple signal models where each coefficient is treated independently; for example in a sparse model, the fact that the coefficient  $\theta(i)$  is large has no bearing on the size of any  $\theta(j)$ ,  $j \neq i$ . We will reserve the use of the term “model” for situations where we are enforcing dependencies between the values and the locations of the coefficients  $\theta(i)$ .

not only the fact that most of the wavelet coefficients of a natural image are small but also the fact that the values and locations of the large coefficients have a particular structure. Coding the coefficients according to a structured sparsity model enables these algorithms to compress images close to the maximum amount possible – significantly better than a naïve coder that just processes each large coefficient independently. In addition to the work described in Chapter 5, a previously developed CS recovery algorithm promotes structure in the sparse representation by tailoring the recovered signal according to a sparsity-promoting probabilistic model, such as an Ising model [106]. Such probabilistic models favor certain configurations for the magnitudes and indices of the significant coefficients of the signal.

We expand on this prior work by introducing a model-based CS theory that parallels the conventional theory and provides concrete guidelines on how to create model-based recovery algorithms with provable performance guarantees. By reducing the degrees of freedom of a sparse/compressible signal by permitting only certain configurations of the large and zero/small coefficients, structured sparsity models provide two immediate benefits to CS. First, they enable us to reduce, in some cases significantly, the number of measurements  $M$  required to stably recover a signal. Second, during signal recovery, they enable us to better differentiate true signal information from recovery artifacts, which leads to a more robust recovery.

To precisely quantify the benefits of model-based CS, we introduce and study several new theoretical concepts that could be of more general interest. We begin with structured sparsity models for  $K$ -sparse signals and make precise how the struc-

ture reduces the number of potential sparse signal supports in  $\alpha$ . Then using the *model-based restricted isometry property* (RIP) from [107, 108], we prove that such *structured sparse signals* can be robustly recovered from noisy compressive measurements. Moreover, we quantify the required number of measurements  $M$  and show that for some structured sparsity models  $M$  is independent of  $N$ . These results unify and generalize the limited related work to date on structured sparsity models for strictly sparse signals [9, 92, 93, 99, 100, 107–111]. We then introduce the notion of a *structured compressible signal*, whose coefficients  $\theta$  are no longer strictly sparse but have a structured power-law decay. To establish that structured compressible signals can be robustly recovered from compressive measurements, we generalize the CS RIP to a new *restricted amplification property* (RAmP). For some structured sparsity models, the required number of measurements  $M$  for recovery of compressible signals is independent of  $N$ .

To take practical advantage of this new theory, we demonstrate how to integrate structured sparsity models into two state-of-the-art CS recovery algorithms, CoSaMP and iterative hard thresholding (IHT) (see Section 2.3.4). The key modification is surprisingly simple: we merely replace the nonlinear sparse approximation step in these greedy algorithms with a structured sparse approximation. Thanks to our new theory, both new model-based recovery algorithms have provable robustness guarantees for both structured sparse and structured compressible signals.

To validate our theory and algorithms and demonstrate its general applicability and utility, we present two specific instances of model-based CS and conduct a range



of simulation experiments. The first structured sparsity model accounts for the fact that the large wavelet coefficients of piecewise smooth signals and images tend to live on a rooted, connected *tree structure*, as described in Section 5.1. Using the fact that the number of such trees is much smaller than  $\binom{N}{K}$ , the number of  $K$ -sparse signal supports in  $N$  dimensions, we prove that a tree-based CoSaMP algorithm needs only  $M = \mathcal{O}(K)$  measurements to robustly recover tree-sparse and tree-compressible signals.

The second structured sparsity model accounts for the fact that the large coefficients of many sparse signals cluster together [19, 110]. Such a so-called *block sparse* model is equivalent to a *joint sparsity* model for an ensemble of  $J$ , length- $N$  signals [9, 19] (see Section 2.4.1), where the supports of the signals' large coefficients are shared across the ensemble. Using the fact that the number of clustered supports is much smaller than  $\binom{JN}{JK}$ , we prove that a block-based CoSaMP algorithm needs only  $M = \mathcal{O}(JK + K \log(\frac{N}{K}))$  measurements to robustly recover block-sparse and block-compressible signals.

Our new theory and methods relate to a small body of previous work aimed at integrating structured sparsity models with CS. Several groups have developed structured sparse signal recovery algorithms [9, 92–94, 99, 100, 107–111] however, their approaches have either been ad hoc or focused on a single structured sparsity model. Most previous work on unions of subspaces [107, 108, 112] has focused exclusively on strictly sparse signals and has not considered feasible recovery algorithms.

A related CS modeling framework for structured sparse signals [19] collects the

$N$  samples of a signal into  $D$  groups,  $D \leq N$ , and allows signals where  $K$  out of  $D$  groups have nonzero coefficients. This framework is immediately applicable to block-sparse signals and signal ensembles with common sparse supports. While [19] provides recovery algorithms, measurement bounds, and recovery guarantees similar to those provided in Section 6.4, our proposed framework has the ability to focus on arbitrary subsets of the  $\binom{D}{K}$  groups that yield more elaborate structures, such as connected subtrees for wavelet coefficients. To the best of our knowledge, our general framework for model-based recovery, the concept of a structured compressible signal, and the associated RAmP are new to the literature.

## 6.1 Structured Sparsity and Compressibility

While many natural and manmade signals and images can be described to first-order as sparse or compressible, the support of their large coefficients often has an underlying inter-dependency structure. This phenomenon has received only limited attention by the CS community to date [9, 19, 92–94, 99, 100, 107–111]. In this section, we introduce a model-based theory of CS that captures such structure. A model reduces the degrees of freedom of a sparse/compressible signal by permitting only certain configurations of supports for the large coefficient. As we will show, this allows us to reduce, in some cases significantly, the number of compressive measurements  $M$  required to stably recover a signal.

### 6.1.1 Structured Sparse Signals

Recall from Section 2.2.6 that a  $K$ -sparse coefficient vector  $\theta$  lives in  $\Sigma_K \subset \mathbb{R}^N$ , which is a union of  $\binom{N}{K}$  subspaces of dimension  $K$ . Other than its  $K$ -sparsity, there are no further constraints on the support or values of its coefficients. A *structured sparsity model* endows the  $K$ -sparse coefficient vector  $\theta$  with additional structure that allows certain  $K$ -dimensional subspaces in  $\Sigma_K$  and disallows others [107, 108].

**Definition 6.1** A structured sparsity model  $\mathcal{M}_K$  is defined as the union of  $m_K$  canonical  $K$ -dimensional subspaces

$$\mathcal{M}_K = \bigcup_{m=1}^{m_K} \mathcal{X}_m, \text{ such that } \mathcal{X}_m := \{\theta \in \mathbb{R}^N : \theta|_{\Omega_m^c} = 0\},$$

where  $\{\Omega_1, \dots, \Omega_{m_K}\}$  is the set containing all allowed supports, each support having cardinality  $K$ , and each subspace  $\mathcal{X}_m$  contains all vectors  $\theta$  with  $\text{supp}(\theta) \in \Omega_m$ .

Signals from  $\mathcal{M}_K$  are called  *$K$ -model sparse*. Clearly,  $\mathcal{M}_K \subseteq \Sigma_K$  and contains  $m_K \leq \binom{N}{K}$  subspaces.

In Sections 6.3 and 6.4 below we consider two concrete structured sparsity models. The first model accounts for the fact that the large wavelet coefficients of piecewise smooth signals and images tend to live on a rooted, connected *tree structure*, as described in Section 5.1. The second model accounts for the fact that the large coefficients of sparse signals often *cluster* together [9, 19, 110].

### 6.1.2 Model-Based RIP

If we know that the coefficient vector  $\theta$  being acquired is  $K$ -model sparse, then we can relax the RIP constraint on the CS measurement matrix  $\Phi$  and still achieve stable recovery from the compressive measurements  $\mathbf{y} = \Phi\mathbf{x} = \Upsilon\theta$  [107, 108].

**Definition 6.2** [107, 108] *An  $M \times N$  matrix  $\Upsilon$  has the  $\mathcal{M}_K$ -restricted isometry property ( $\mathcal{M}_K$ -RIP) with constant  $\delta_{\mathcal{M}_K}$  if, for all  $\theta \in \mathcal{M}_K$ , we have*

$$(1 - \delta_{\mathcal{M}_K})\|\theta\|_2^2 \leq \|\Upsilon\theta\|_2^2 \leq (1 + \delta_{\mathcal{M}_K})\|\theta\|_2^2. \quad (6.1)$$

Blumensath and Davies [107] have quantified the number of measurements  $M$  necessary for a random CS matrix to have the  $\mathcal{M}_K$ -RIP with a given probability.

**Theorem 6.1** [107] *Let  $\mathcal{M}_K$  be the union of  $m_K$  subspaces of  $K$ -dimensions in  $\mathbb{R}^N$ . Then, for any  $t > 0$  and any*

$$M \geq \frac{2}{c\delta_{\mathcal{M}_K}^2} \left( \ln(2m_K) + K \ln \frac{12}{\delta_{\mathcal{M}_K}} + t \right),$$

*where  $c$  is a positive constant, an  $M \times N$  i.i.d. subgaussian random matrix has the  $\mathcal{M}_K$ -RIP with constant  $\delta_{\mathcal{M}_K}$  with probability at least  $1 - e^{-t}$ .*

This bound can be used to recover the conventional CS result by substituting  $m_K = \binom{N}{K} \approx (Ne/K)^K$ . Similarly, as the number of subspaces  $m_K$  that arise from the structure imposed can be significantly smaller than the standard  $\binom{N}{K}$ , the number of rows needed for a random matrix to have the  $\mathcal{M}_K$ -RIP can be significantly lower than the number of rows needed for the standard RIP. The  $\mathcal{M}_K$ -RIP property is sufficient for robust recovery of structured sparse signals, as we show below in Section 6.2.2.

### 6.1.3 Structured Compressible Signals

Just as compressible signals are “nearly  $K$ -sparse” and thus live close to the union of subspaces  $\Sigma_K$  in  $\mathbb{R}^N$ , structured compressible signals are “nearly  $K$ -model sparse” and live close to the restricted union of subspaces  $\mathcal{M}_K$ . In this section, we make this new concept rigorous. Recall from (2.13) that we defined compressible signals in terms of the decay of their  $K$ -term approximation error.

The  $\ell_2$  error incurred by approximating  $x \in \mathbb{R}^N$  by the best structured sparse approximation in  $\mathcal{M}_K$  is given by

$$\sigma_{\mathcal{M}_K}(\mathbf{x}) := \inf_{\bar{\mathbf{x}} \in \mathcal{M}_K} \|\mathbf{x} - \bar{\mathbf{x}}\|_2.$$

We define  $\mathbb{M}_B(\mathbf{x}, K)$  as the algorithm that obtains the best  $K$ -term structured sparse approximation of  $\mathbf{x}$  in the union of subspaces  $\mathcal{M}_K$ :

$$\mathbb{M}(\mathbf{x}, K) = \arg \min_{\bar{\mathbf{x}} \in \mathcal{M}_K} \|\mathbf{x} - \bar{\mathbf{x}}\|_2.$$

This implies that  $\|\mathbf{x} - \mathbb{M}(\mathbf{x}, K)\|_2 = \sigma_{\mathcal{M}_K}(\mathbf{x})$ . The decay of this approximation error defines the structured compressibility of a signal.

**Definition 6.3** *The set of  $s$ -structured compressible signals is defined as*

$$\mathfrak{M}_s = \{\theta \in \mathbb{R}^N : \sigma_{\mathcal{M}_K}(\theta) \leq GK^{-1/s}, 1 \leq K \leq N, G < \infty\}.$$

Define  $|\theta|_{\mathfrak{M}_s}$  as the smallest value of  $G$  for which this condition holds for  $\theta$  and  $s$ .

We say that  $\theta \in \mathfrak{M}_s$  is an  $s$ -structured compressible signal in the structured sparsity model  $\mathcal{M}_K$ . These approximation classes have been characterized for certain structured sparsity models; see Section 6.3 for an example.

#### 6.1.4 Nested Model Approximations and Residual Subspaces

In conventional CS, the same requirement (RIP) is a sufficient condition for the stable recovery of both sparse and compressible signals. In model-based recovery, however, the class of structured compressible signals is much larger than that of structured sparse signals, since the set of subspaces containing structured sparse signals does not span all  $K$ -dimensional subspaces.

To address this difference, we need to introduce some additional tools to develop a *sufficient* condition for the stable recovery of structured compressible signals. We will pay particular attention to structured sparsity models  $\mathcal{M}_K$  that generate *nested approximations*, since they are more amenable to analysis.

**Definition 6.4** *A structured sparsity model  $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \dots\}$  has the nested approximation property (NAP) if  $\text{supp}(\mathbb{M}(\theta, K)) \subset \text{supp}(\mathbb{M}(\theta, K'))$  for all  $K < K'$  and for all  $\theta \in \mathbb{R}^N$ .*

In words, a structured sparsity model generates nested approximations if the support of the best  $K'$ -term structured sparse approximation contains the support of the best  $K$ -term structured sparse approximation for all  $K < K'$ . An important example of a NAP-generating structured sparse model is the standard compressible signal model of (2.13).

When a structured sparsity model obeys the NAP, the support of the difference between the best  $jK$ -term structured sparse approximation and the best  $(j+1)K$ -term structured sparse approximation of a signal can be shown to lie in a small union of subspaces, thanks to the structure enforced by the model. This structure is

captured by the set of subspaces that are included in each subsequent approximation, as defined below.

**Definition 6.5** *The  $j^{\text{th}}$  set of residual subspaces of size  $K$  is defined as*

$$\mathcal{R}_{j,K}(\mathcal{M}) = \{\mathbf{u} \in \mathbb{R}^N \text{ such that } \mathbf{u} = \mathbb{M}(\theta, jK) - \mathbb{M}(\theta, (j-1)K) \text{ for some } \theta \in \mathbb{R}^N\},$$

for  $j = 1, \dots, \lceil N/K \rceil$ .

Under the NAP, each structured compressible coefficient vector  $\theta$  can be partitioned into its best  $K$ -term structured sparse approximation  $\theta_{T_1}$ , the additional components present in the best  $2K$ -term structured sparse approximation  $\theta_{T_2}$ , and so on, with  $\theta = \sum_{j=1}^{\lceil N/K \rceil} \theta_{T_j}$  and  $\theta_{T_j} \in \mathcal{R}_{j,K}(\mathcal{M})$  for each  $j$ . Each signal partition  $\theta_{T_j}$  is a  $K$ -sparse signal, and thus  $\mathcal{R}_{j,K}(\mathcal{M})$  is a union of subspaces of dimension  $K$ . We will denote by  $R_j$  the number of subspaces that compose  $\mathcal{R}_{j,K}(\mathcal{M})$  and omit the dependence on  $\mathcal{M}$  in the sequel for brevity.

Intuitively, the norms of the partitions  $\|\theta_{T_j}\|_2$  decay as  $j$  increases for signals that are structured compressible. As the next subsection shows, this observation is instrumental in relaxing the isometry restrictions on the measurement matrix  $\Phi$  and bounding the recovery error for  $s$ -structured compressible signals when the model obeys the NAP.

### 6.1.5 The Restricted Amplification Property (RAMP)

For exactly  $K$ -structured sparse signals, we discussed in Section 6.1.2 that the number of compressive measurements  $M$  required for a random matrix to have the  $\mathcal{M}_K$ -RIP

is determined by the number of canonical subspaces  $m_K$  via (6.2). Unfortunately, such structured sparse concepts and results do not immediately extend to structured compressible signals. Thus, we develop a generalization of the  $\mathcal{M}_K$ -RIP that we will use to quantify the stability of recovery for structured compressible signals.

One way to analyze the robustness of compressible signal recovery in conventional CS is to consider the tail of the signal outside its  $K$ -term approximation as contributing additional “noise” to the measurements of size  $\|\Upsilon(\theta - \mathfrak{T}(\theta, K))\|_2$  [65, 66, 113]. Consequently, the conventional  $K$ -sparse recovery performance result can be applied with the augmented noise  $\mathbf{n} + \Upsilon(\theta - \mathfrak{T}(\theta, K))$ .

This technique can also be used to quantify the robustness of structured compressible signal recovery. The key quantity we must control is the amplification of the structured sparse approximation residual through  $\Phi$ . The following property is a new generalization of the RIP and model-based RIP.

**Definition 6.6** *A matrix  $\Phi$  has the  $(\epsilon_K, r)$ -restricted amplification property (RAmP) for the residual subspaces  $\mathcal{R}_{j,K}$  of model  $\mathcal{M}$  if*

$$\|\Phi \mathbf{u}\|_2^2 \leq (1 + \epsilon_K) j^{2r} \|\mathbf{u}\|_2^2 \quad (6.2)$$

for any  $\mathbf{u} \in \mathcal{R}_{j,K}$  for each  $1 \leq j \leq \lceil N/K \rceil$ .

The regularity parameter  $r > 0$  caps the growth rate of the amplification of  $\mathbf{u} \in \mathcal{R}_{j,K}$  as a function of  $j$ . Its value can be chosen so that the growth in amplification with  $j$  balances the decay of the norm in each residual subspace  $\mathcal{R}_{j,K}$  with  $j$ .

We can quantify the number of compressive measurements  $M$  required for a ran-



dom measurement matrix  $\Phi$  to have the RAmP with high probability; we prove the following in Appendix E.

**Theorem 6.2** *Let  $\Phi$  be an  $M \times N$  matrix with i.i.d. subgaussian entries and let the set of residual subspaces  $\mathcal{R}_{j,K}$  of the structured sparsity model  $\mathcal{M}$  contain  $R_j$  subspaces of dimension  $K$  for each  $1 \leq j \leq \lceil N/K \rceil$ . If*

$$M \geq \max_{1 \leq j \leq \lceil N/K \rceil} \frac{1}{(j^r \sqrt{1 + \epsilon_K} - 1)^2} \left( 2K + 4 \ln \frac{R_j N}{K} + 2t \right), \quad (6.3)$$

*then the matrix  $\Phi$  has the  $(\epsilon_K, r)$ -RAmP with probability  $1 - e^{-t}$ .*

The order of the bound of Theorem 6.2 is lower than  $\mathcal{O}(K \log(N/K))$  as long as the number of subspaces  $R_j$  grows slower than  $N^K$ .

Armed with the RAmP, we can state the following result, which will provide robustness for the recovery of structured compressible signals; see Appendix F for the proof.

**Theorem 6.3** *Let  $\theta \in \mathfrak{M}_s$  be an  $s$ -model compressible signal in a structured sparsity model  $\mathcal{M}$  that obeys the NAP. If  $\Upsilon$  has the  $(\epsilon_K, r)$ -RAmP and  $r = s - 1$ , then we have*

$$\|\Upsilon(\theta - \mathbb{M}(\theta, K))\|_2 \leq \sqrt{1 + \epsilon_K} K^{-s} \ln \left\lceil \frac{N}{K} \right\rceil |\theta|_{\mathfrak{M}_s}.$$

## 6.2 Model-Based Signal Recovery Algorithms

To take practical advantage of our new theory for model-based CS, we demonstrate how to integrate structured sparsity models into two state-of-the-art CS recovery

algorithms, CoSaMP [66] (in this section) and iterative hard thresholding (IHT) [43, 44, 63–65] (in Appendix G). The key modification is simple: we merely replace the best  $K$ -term sparse approximation step in these greedy algorithms with a best  $K$ -term structured sparse approximation. Since at each iteration we need only search over the  $m_K$  subspaces of  $\mathcal{M}_K$  rather than  $\binom{N}{K}$  subspaces of  $\Sigma_K$ , fewer measurements will be required for the same degree of robust signal recovery. Or, alternatively, more accurate recovery can be achieved using the same number of measurements.

After presenting the modified CoSaMP algorithm, we prove robustness guarantees for both structured sparse and structured compressible signals. To this end, we must define an enlarged union of subspaces that includes sums of elements in the structured sparsity model.

**Definition 6.7** *The  $B$ -Minkowski sum for the set  $\mathcal{M}_K$ , with  $B > 1$  an integer, is defined as*

$$\mathcal{M}_K^B = \left\{ \theta = \sum_{r=1}^B \theta^{(r)}, \text{ with } \theta^{(r)} \in \mathcal{M}_K \right\}.$$

We also define  $\mathbb{M}_B(\theta, K)$  as the algorithm that obtains the best approximation of  $\theta$  in the enlarged union of subspaces  $\mathcal{M}_K^B$ :

$$\mathbb{M}_B(\theta, K) = \arg \min_{\bar{\theta} \in \mathcal{M}_K^B} \|\theta - \bar{\theta}\|_2.$$

We note that  $\mathbb{M}(\theta, K) = \mathbb{M}_1(\theta, K)$ . Note also that for many structured sparsity models, we will have  $\mathcal{M}_K^B \subset \mathcal{M}_{BK}$ , and so the algorithm  $\mathbb{M}(\theta, BK)$  will provide a strictly better approximation than  $\mathbb{M}_B(\theta, K)$ .

### 6.2.1 Model-based CoSaMP

We choose to modify the CoSaMP algorithm [66] for two reasons. First, it has robust recovery guarantees that are on par with the best convex optimization-based approaches. Second, it has a simple iterative, greedy structure based on a best  $BK$ -term approximation (with  $B$  a small integer) that is easily modified to incorporate a best  $B$ -Minkowski sum of  $K$ -term structured sparse approximation  $\mathbb{M}_B(\theta, K)$ . Pseudocode for the modified algorithm is given in Algorithm 7.

### 6.2.2 Performance of Structured Sparse Signal Recovery

A robustness guarantee for noisy measurements of structured sparse signals can be obtained using the model-based RIP (6.1). Our performance guarantee for structured sparse signal recovery will require that the measurement matrix  $\Phi$  be a near-isometry for all subspaces in  $\mathcal{M}_K^B$  for some  $B > 1$ . This requirement is a direct generalization of the  $2K$ -RIP,  $3K$ -RIP, and higher-order RIPs from the conventional CS theory. The following theorem is proven in Appendix H.

**Theorem 6.4** *Let  $\theta \in \mathcal{M}_K$  and let  $\mathbf{y} = \Upsilon\theta + \mathbf{n}$  be a set of noisy CS measurements. If  $\Upsilon$  has an  $\mathcal{M}_K^4$ -RIP constant of  $\delta_{\mathcal{M}_K^4} \leq 0.1$ , then the signal estimate  $\hat{\theta}_i$  obtained from iteration  $i$  of the model-based CoSaMP algorithm satisfies*

$$\|\theta - \hat{\theta}_i\|_2 \leq 2^{-i}\|\theta\|_2 + 15\|\mathbf{n}\|_2. \quad (6.4)$$

---

**Algorithm 7** Model-based CoSaMP
 

---

Inputs: CS matrix  $\Upsilon$ , measurements  $\mathbf{y}$ , structured sparse approx. algorithm  $\mathbb{M}_K$

Output:  $K$ -sparse approximation  $\hat{\theta}$  to true signal representation  $\theta$

Initialize:  $\hat{\theta}_0 = 0$ ,  $\mathbf{r} = \mathbf{y}$ ;  $i = 0$

**while** halting criterion false **do**

1.  $i \leftarrow i + 1$
2.  $\mathbf{e} \leftarrow \Upsilon^T \mathbf{r}$  {form signal residual estimate}
3.  $\Omega \leftarrow \text{supp}(\mathbb{M}_2(\mathbf{e}, K))$  {prune residual estimate according to structure}
4.  $T \leftarrow \Omega \cup \text{supp}(\hat{\theta}_{i-1})$  {merge supports}
5.  $\mathbf{b}|_T \leftarrow \Upsilon_T^\dagger \mathbf{y}$ ,  $\mathbf{b}|_{T^c}$  {form signal estimate}
6.  $\hat{\theta}_i \leftarrow \mathbb{M}(\mathbf{b}, K)$  {prune signal estimate according to structure}
7.  $\mathbf{r} \leftarrow \mathbf{y} - \Upsilon \hat{\theta}_i$  {update measurement residual}

**end while**

return  $\hat{\theta} \leftarrow \hat{\theta}_i$

---

This guarantee matches that of the CoSaMP algorithm [66, Theorem 4.1]; however, our guarantee is only for signals that exhibit the structured sparsity.

### 6.2.3 Performance of Structured Compressible Signal Recovery

Using the new tools introduced in Section 6.1, we can provide a robustness guarantee for noisy measurements of structured compressible signals, using the RAmP as a condition on the matrix  $\Upsilon$ .

**Theorem 6.5** *Let  $\theta \in \mathfrak{M}_s$  be an  $s$ -structured compressible signal from a structured sparsity model  $\mathcal{M}$  that obeys the NAP, and let  $\mathbf{y} = \Upsilon\theta + \mathbf{n}$  be a set of noisy CS measurements. If  $\Upsilon$  has the  $\mathcal{M}_K^4$ -RIP with  $\delta_{\mathcal{M}_K^4} \leq 0.1$  and the  $(\epsilon_K, r)$ -RAmP with  $\epsilon_K \leq 0.1$  and  $r = s - 1$ , then the signal estimate  $\hat{\theta}_i$  obtained from iteration  $i$  of the model-based CoSaMP algorithm satisfies*

$$\|\theta - \hat{\theta}_i\|_2 \leq 2^{-i}\|\theta\|_2 + 35 (\|\mathbf{n}\|_2 + |\theta|_{\mathfrak{M}_s} K^{-s}(1 + \ln[N/K])). \quad (6.5)$$

*Proof sketch.* To prove the theorem, we first bound the optimal structured sparse recovery error for an  $s$ -structured compressible signal  $\theta \in \mathfrak{M}_s$  when the matrix  $\Upsilon$  has the  $(\epsilon_K, r)$ -RAmP with  $r \leq s - 1$  (see Theorem 6.3). Then, using Theorem 6.4, we can easily prove the result by following the analogous proof in [66].  $\square$

The standard CoSaMP algorithm also features a similar guarantee for structured compressible signals, with the constant changing from 35 to 20.

#### 6.2.4 Robustness to Model Mismatch

We now analyze the robustness of model-based CS recovery to *model mismatch*, which occurs when the signal being recovered from compressive measurements does not conform exactly to the structured sparsity model used in the recovery algorithm.

We begin with optimistic results for signals that are “close” to matching the recovery structured sparsity model. First, consider a signal  $\theta$  that is not  $K$ -model sparse as the recovery algorithm assumes but rather  $(K + \kappa)$ -model sparse for some small integer  $\kappa$ . This signal can be decomposed into  $\theta_K = \mathbb{M}(\theta, K)$ , the signal’s  $K$ -

term structured sparse approximation, and  $\theta - \theta_K$ , the error of this approximation. For  $\kappa \leq K$ , we have that  $\theta - \theta_K \in \mathcal{R}_{2,K}$ . If the matrix  $\Upsilon$  has the  $(\epsilon_K, r)$ -RAmP, then it follows that

$$\|\Upsilon(\theta - \theta_K)\|_2 \leq 2^r \sqrt{1 + \epsilon_K} \|\theta - \theta_K\|_2. \quad (6.6)$$

Using equations (6.4) and (6.6), we obtain the following guarantee for the  $i^{\text{th}}$  iteration of model-based CoSaMP:

$$\|\theta - \hat{\theta}_i\|_2 \leq 2^{-i} \|\theta\|_2 + 16 \cdot 2^r \sqrt{1 + \epsilon_K} \|\theta - \theta_K\|_2 + 15 \|\mathbf{n}\|_2.$$

By noting that  $\|\theta - \theta_K\|_2$  is small, we obtain a guarantee that is close to (6.4).

Second, consider a signal  $\theta$  that is not  $s$ -model compressible as the recovery algorithm assumes but rather  $(s - \epsilon)$ -model compressible. The following bound can be obtained under the conditions of Theorem 6.5 by modifying the argument in Appendix F:

$$\|\theta - \hat{\theta}_i\|_2 \leq 2^{-i} \|\theta\|_2 + 35 \left( \|\mathbf{n}\|_2 + |\theta|_{\mathfrak{M}_s} K^{-s} \left( 1 + \frac{\lceil N/K \rceil^\epsilon - 1}{\epsilon} \right) \right).$$

As  $\epsilon$  becomes smaller, the factor  $\frac{\lceil N/K \rceil^\epsilon - 1}{\epsilon}$  approaches  $\log \lceil N/K \rceil$ , matching (6.5).

In summary, as long as the deviations from the structured sparse and structured compressible classes are small, our model-based recovery guarantees still apply within a small bounded constant factor.

We end with an intuitive worst-case result for signals that are arbitrarily far away from structured sparse or structured compressible. Consider such an arbitrary  $\theta \in \mathbb{R}^N$  and compute its nested model-based approximations  $\theta_{jK} = \mathbb{M}(\theta, jK)$ ,  $j = 1, \dots, \lceil N/K \rceil$ . If  $\theta$  is not structured compressible, then the model-based approximation error  $\sigma_{jK}(\theta)$  is not guaranteed to decay as  $j$  decreases. Additionally, the

number of residual subspaces  $\mathcal{R}_{j,K}$  could be as large as  $\binom{N}{K}$ ; that is, the  $j^{\text{th}}$  difference between subsequent model-based approximations  $\theta_{T_j} = \theta_{jK} - \theta_{(j-1)K}$  might lie in any arbitrary  $K$ -dimensional subspace. This worst case is equivalent to setting  $r = 0$  and  $R_j = \binom{N}{K}$  in Theorem 6.2. It is easy to see that this condition on the number of measurements  $M$  is nothing but the standard RIP for CS. Hence, if we inflate the number of measurements to  $M = \mathcal{O}(K \log(N/K))$  (the usual number for conventional CS), the performance of model-based CoSaMP recovery on an arbitrary signal  $\theta$  follows the distortion of the best  $K$ -term *model-based* approximation of  $\theta$  within a bounded constant factor.

### 6.2.5 Computational Complexity of Model-Based Recovery

The computational complexity of a model-based signal recovery algorithm differs from that of a standard algorithm by two factors. The first factor is the reduction in the number of measurements  $M$  necessary for recovery: since most current recovery algorithms have a computational complexity that is linear in the number of measurements, any reduction in  $M$  reduces the total complexity. The second factor is the cost of the model-based approximation. The  $K$ -term approximation used in most current recovery algorithms can be implemented with a simple sorting operation ( $\mathcal{O}(N \log N)$  complexity, in general). Ideally, the structured sparsity model should support a similarly efficient approximation algorithm.

To validate our theory and algorithms and demonstrate their general applicability and utility, we now present two specific instances of model-based CS and conduct a

range of simulation experiments.

### 6.3 Example: Wavelet Tree Model

In Chapter 5, we showed that wavelet coefficients can be naturally organized into a tree structure; for many kinds of natural and manmade signals, the largest coefficients cluster along the branches of this tree. This motivates a connected tree model for the wavelet coefficients [114–116].

While CS recovery for wavelet-sparse signals has been considered previously (see Chapter 5 and [99, 100, 109, 111]), the resulting algorithms integrated the tree constraint in an ad-hoc fashion. Furthermore, the algorithms provide no recovery guarantees or bounds on the necessary number of compressive measurements.

#### 6.3.1 Tree-Sparse Signals

We first describe tree sparsity in the context of sparse wavelet decompositions. We focus on one-dimensional signals and binary wavelet trees, but all of our results extend directly to  $d$ -dimensional signals and  $2^d$ -ary wavelet trees.

We remember from Section 5.1 that wavelet functions act as local discontinuity detectors, and using the nested support property of wavelets at different scales, it is straightforward to see that a signal discontinuity will give rise to a chain of large wavelet coefficients along a branch of the wavelet tree from a leaf to the root. Moreover, smooth signal regions will give rise to regions of small wavelet coefficients. This “connected tree” property has been well-exploited in a number of wavelet-based pro-



cessing [95, 97, 117] and compression [118, 119] algorithms. In this section, we will specialize the theory developed in Sections 6.1 and 6.2 to a connected tree model  $\mathcal{T}$ .

A set of wavelet coefficients  $\Omega$  is a *connected subtree* if, whenever a coefficient  $w_{i,j} \in \Omega$ , then its parent  $w_{i-1, \lceil j/2 \rceil} \in \Omega$  as well. Each such set  $\Omega$  defines a subspace of signals whose support is contained in  $\Omega$ ; that is, all wavelet coefficients outside  $\Omega$  are zero. In this way, we define the structured sparsity model  $\mathcal{T}_K$  as the union of all  $K$ -dimensional subspaces corresponding to supports  $\Omega$  that form connected subtrees.

**Definition 6.8** Define the set of  $K$ -tree sparse signals as

$$\mathcal{T}_K = \left\{ \mathbf{x} = v_0 \nu + \sum_{i=0}^{I-1} \sum_{j=1}^{2^i} w_{i,j} \psi_{i,j} : w|_{\Omega^c} = 0, |\Omega| = K, \Omega \text{ is a connected subtree} \right\}.$$

To quantify the number of subspaces in  $\mathcal{T}_K$ , it suffices to count the number of distinct connected subtrees of size  $K$  in a binary tree of size  $N$ . We prove the following result in Appendix I.

**Proposition 6.1** The number of subspaces in  $\mathcal{T}_K$  obeys  $T_K \leq \frac{4^{K+4}}{K e^2}$  for  $K \geq \log_2 N$  and  $T_K \leq \frac{(2e)^K}{K+1}$  for  $K < \log_2 N$ .

To simplify the presentation in the sequel, we will simply use the weaker bound  $T_K \leq \frac{(2e)^K}{K+1}$  for all values of  $K$  and  $N$ .

### 6.3.2 Tree-Based Approximation

To implement tree-based signal recovery, we seek an efficient algorithm  $\mathbb{T}(\mathbf{x}, K)$  to solve the optimal approximation

$$\mathbf{x}_K^{\mathcal{T}} = \arg \min_{\bar{\mathbf{x}} \in \mathcal{T}_K} \|\mathbf{x} - \bar{\mathbf{x}}\|_2. \quad (6.7)$$

Fortuitously, an efficient solver exists, called the *condensing sort and select algorithm* (CSSA) [114–116]. Recall that subtree approximation coincides with standard  $K$ -term approximation (and hence can be solved by simply sorting the wavelet coefficients) when the wavelet coefficients are monotonically nonincreasing along the tree branches out from the root. The CSSA solves (6.7) in the case of general wavelet coefficient values by *condensing* the nonmonotonic segments of the tree branches using an iterative sort-and-average routine during a greedy search through the nodes. For each node in the tree, the algorithm calculates the average wavelet coefficient magnitude for each subtree rooted at that node, and records the largest average among all the subtrees as the energy for that node. The CSSA then searches for the unselected node with the largest energy and adds the subtree corresponding to the node’s energy to the estimated support as a *supernode*: a single node that provides a condensed representation of the corresponding subtree [116]. Condensing a large coefficient far down the tree accounts for the potentially large cost (in terms of the total budget of tree nodes  $K$ ) of growing the tree to that point.

Since the first step of the CSSA involves sorting all of the wavelet coefficients, overall it requires  $\mathcal{O}(N \log N)$  computations. However, once the CSSA grows the optimal tree of size  $K$ , it is trivial to determine the optimal trees of size  $< K$  and computationally efficient to grow the optimal trees of size  $> K$  [114].

The constrained optimization (6.7) can also be rewritten as an unconstrained problem by introducing the Lagrange multiplier  $\lambda$  [120]:

$$\min_{\bar{\theta} \in \bar{\mathcal{T}}} \|\theta - \bar{\theta}\|_2^2 + \lambda(\|\bar{\theta}\|_0 - K),$$

where  $\bar{\mathcal{T}} = \cup_{n=1}^N \mathcal{T}_n$  and  $\bar{\theta}$  are the wavelet coefficients of  $\bar{\mathbf{x}}$ . Except for the inconsequential  $\lambda K$  term, this optimization coincides with Donoho's *complexity penalized sum of squares* [120], which can be solved in only  $\mathcal{O}(N)$  computations using coarse-to-fine dynamic programming on the tree. Its primary shortcoming is the nonobvious relationship between the tuning parameter  $\lambda$  and the resulting size  $K$  of the optimal connected subtree.

### 6.3.3 Tree-Compressible Signals

Specializing Definition 6.1 from Section 6.1.3 to  $\mathcal{T}$ , we make the following definition.

**Definition 6.9** *Define the set of  $s$ -tree compressible signals as*

$$\mathfrak{T}_s = \{\theta \in \mathbb{R}^N : \|\theta - \mathbb{T}(\theta, K)\|_2 \leq GK^{-s}, 1 \leq K \leq N, G < \infty\}.$$

Furthermore, define  $|\theta|_{\mathfrak{T}_s}$  as the smallest value of  $G$  for which this condition holds for  $\theta$  and  $s$ .

Tree approximation classes contain signals whose wavelet coefficients have a loose (and possibly interrupted) decay from coarse to fine scales. These classes have been well-characterized for wavelet-sparse signals [115, 116, 119] and are intrinsically linked with the Besov spaces  $B_q^s(L_p([0, 1]))$ . Besov spaces contain functions of one or more continuous variables that have (roughly speaking)  $s$  derivatives in  $L_p([0, 1])$ ; the parameter  $q$  provides finer distinctions of smoothness. When a Besov space signal  $x_a \in B_p^s(L_p([0, 1]))$  with  $s > 1/p - 1/2$  is sampled uniformly and converted to a length- $N$  vector  $\mathbf{x}$ , its wavelet coefficients  $\theta$  belong to the tree approximation space

$\mathfrak{T}_s$ , with

$$|\theta|_{\mathfrak{T}_s} \asymp \|x_a\|_{L_p([0,1])} + \|x_a\|_{B_q^s(L_p([0,1]))},$$

where “ $\asymp$ ” denotes an equivalent norm. The same result holds if  $s = 1/p - 1/2$  and  $q \leq p$ . A more thorough description of Besov spaces is provided in Section 7.4.

#### 6.3.4 Stable Tree-Based Recovery From Compressive Measurements

For tree-sparse signals, by applying Theorem 6.1 and Proposition 6.1, we find that a subgaussian random matrix has the  $\mathcal{T}_K$ -RIP property with constant  $\delta_{\mathcal{T}_K}$  and probability  $1 - e^{-t}$  if the number of measurements obeys

$$M \geq \frac{2}{c\delta_{\mathcal{T}_K}^2} \left( K \ln \frac{48}{\delta_{\mathcal{T}_K}} + \ln \frac{512}{Ke^2} + t \right).$$

Thus, the number of measurements necessary for stable recovery of tree-sparse signals is linear in  $K$ , without the dependence on  $N$  present in conventional non-model-based CS recovery.

For tree-compressible signals, we must quantify the number of subspaces  $R_j$  in each residual set  $\mathcal{R}_{j,K}$  for the approximation class. We can then apply the theory of Section 6.2.3 with Proposition 6.1 to calculate smallest allowable  $M$  via Theorem 6.5.

**Proposition 6.2** *The number of  $K$ -dimensional subspaces that comprise  $\mathcal{R}_{j,K}$  obeys*

$$R_j \leq \frac{(2e)^{K(2j+1)}}{(Kj + K + 1)(Kj + 1)}. \quad (6.8)$$

Using Proposition 6.2 and Theorem 6.5, we obtain the following condition for the matrix  $\Phi$  to have the RAmP, which is proved in Appendix J.

**Proposition 6.3** *Let  $\Phi$  be an  $M \times N$  matrix with i.i.d. subgaussian entries. If*

$$M \geq \frac{2}{(\sqrt{1 + \epsilon_K} - 1)^2} \left( 10K + 2 \ln \frac{N}{K(K+1)(2K+1)} + t \right),$$

*then the matrix  $\Phi$  has the  $(\epsilon_K, s)$ -RAmP for the structured sparsity model  $\mathcal{T}$  and all  $s > 0.5$  with probability  $1 - e^{-t}$ .*

Both cases give a simplified bound on the number of measurements required as  $M = \mathcal{O}(K)$ , which is a substantial improvement over the  $M = \mathcal{O}(K \log(N/K))$  required by conventional CS recovery methods. Thus, when  $\Phi$  satisfies Proposition 6.3, we have the guarantee (6.5) for sampled Besov space signals from  $B_q^s(L_p([0, 1]))$ .

### 6.3.5 Experiments

We now present the results of a number of numerical experiments that illustrate the effectiveness of a tree-based recovery algorithm. Our consistent observation is that explicit incorporation of the structured sparsity model in the recovery process significantly improves the quality of recovery for a given number of measurements. In addition, model-based recovery remains stable when the inputs are no longer tree-sparse, but rather are tree-compressible and/or corrupted with differing levels of noise. We employ the model-based CoSaMP recovery of Algorithm 7 with a CSSA-based structured sparse approximation step in all experiments.

We first study one-dimensional signals that match the connected wavelet-tree model described above. Among such signals is the class of piecewise smooth functions, which are commonly encountered in analysis and practice.

Figure 6.1 illustrates the results of recovering the tree-compressible *HeaviSine* signal of length  $N = 1024$  from  $M = 80$  noise-free random Gaussian measurements using CoSaMP,  $\ell_1$ -norm minimization using the `l1_eq` solver from the  $\ell_1$ -Magic toolbox,<sup>3</sup> and our tree-based recovery algorithm. It is clear that the number of measurements ( $M = 80$ ) is far fewer than the minimum number required by CoSaMP and  $\ell_1$ -norm minimization to accurately recover the signal. In contrast, tree-based recovery using  $K = 26$  is accurate and uses fewer iterations to converge than conventional CoSaMP. Moreover, the normalized magnitude of the squared error for tree-based recovery is equal to 0.037, which is remarkably close to the error between the noise-free signal and its *best*  $K$ -term tree-structured sparse approximation (0.036).

Figure 6.2(a) illustrates the results of a Monte Carlo simulation study on the impact of the number of measurements  $M$  on the performance of model-based and conventional recovery for a class of tree-sparse piecewise polynomial signals. Each data point was obtained by measuring the normalized recovery error of 500 sample trials. Each sample trial was conducted by generating a new piecewise polynomial signal of length  $N = 1024$  with five polynomial pieces of cubic degree and randomly placed discontinuities, computing its best  $K$ -term tree-approximation using the CSSA, and then measuring the resulting signal using a matrix with i.i.d. Gaussian entries. Model-based recovery attains near-perfect recovery at  $M = 3K$  measurements, while CoSaMP only matches this performance at  $M = 5K$ .

For the same class of signals, we empirically compared the recovery times of our

---

<sup>3</sup><http://www.acm.caltech.edu/l1magic>.

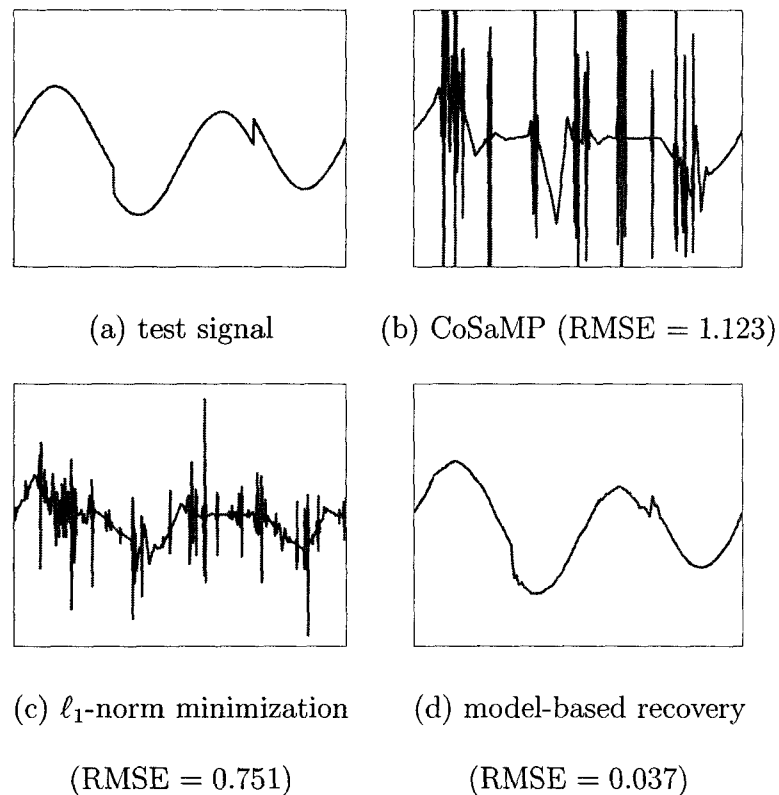


Figure 6.1 : *Example performance of model-based signal recovery for a piecewise smooth signal. (a) HeaviSine test signal of length  $N = 1024$ . This signal is compressible in a connected wavelet tree model. Signal recovered from  $M = 80$  random Gaussian measurements using (b) the iterative recovery algorithm CoSaMP, (c) standard  $\ell_1$ -norm minimization, and (d) the wavelet tree-based CoSaMP algorithm from Section 6.3. In all figures, root mean-squared error (RMSE) values are normalized with respect to the  $\ell_2$  norm of the signal.*

proposed algorithm with those of the standard approach (CoSaMP). Experiments were conducted on a Sun workstation with a 1.8GHz AMD Opteron dual-core processor and 2GB memory running UNIX, using non-optimized Matlab code and a function-handle based implementation of the random projection operator  $\Phi$ . As is evident from Figure 6.2(b), wavelet tree-based recovery is in general slower than CoSaMP. This is due to the fact that the CSSA step in the iterative procedure is

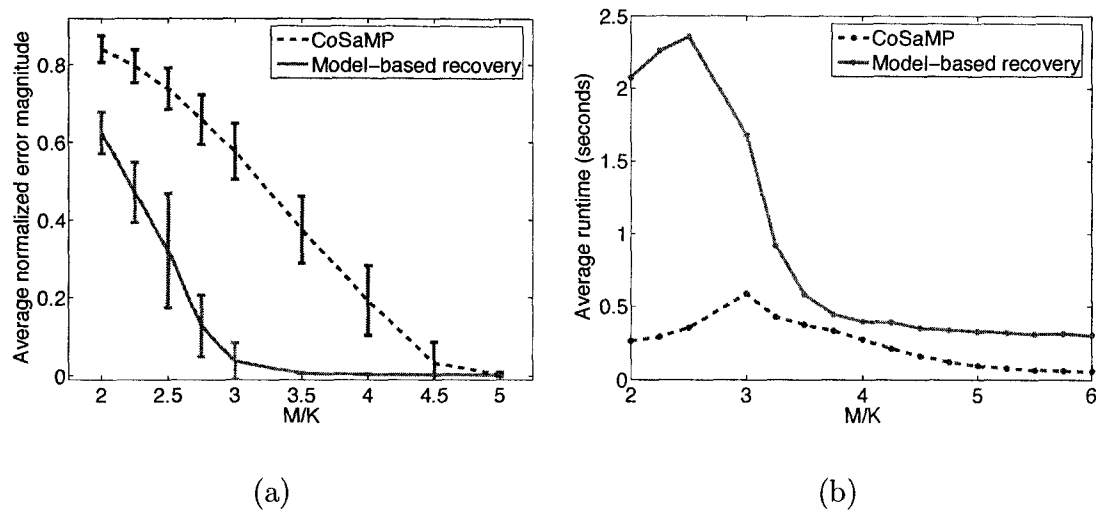


Figure 6.2 : Performance of CoSaMP vs. wavelet tree-based recovery on a class of piecewise cubic signals. (a) Average normalized recovery error and (b) average runtime for each recovery algorithm as a function of the overmeasuring factor  $M/K$ . The number of measurements  $M$  for which the wavelet tree-based algorithm obtains near-perfect recovery is much smaller than that required by CoSaMP. The penalty paid for this improvement is a modest increase in the runtime.

more computationally demanding than simple  $K$ -term approximation. Nevertheless, the highest benefits of model-based CS recovery are obtained around  $M = 3K$ ; in this regime, the runtimes of the two approaches are comparable, with tree-based recovery displaying faster convergence and yielding much smaller recovery error.

Figure 6.3 shows the growth of the overmeasuring factor  $M/K$  with the signal length  $N$  for conventional CS and model-based recovery. We generated 50 sample piecewise cubic signals and numerically computed the minimum number of measurements  $M$  required for the recovery error  $\|\theta - \hat{\theta}\|_2 \leq 2.5\sigma_{T_K}(\theta)$ , the *best* tree-approximation error, for every sample signal. The figure shows that while doubling the signal length increases the number of measurements required by standard recov-



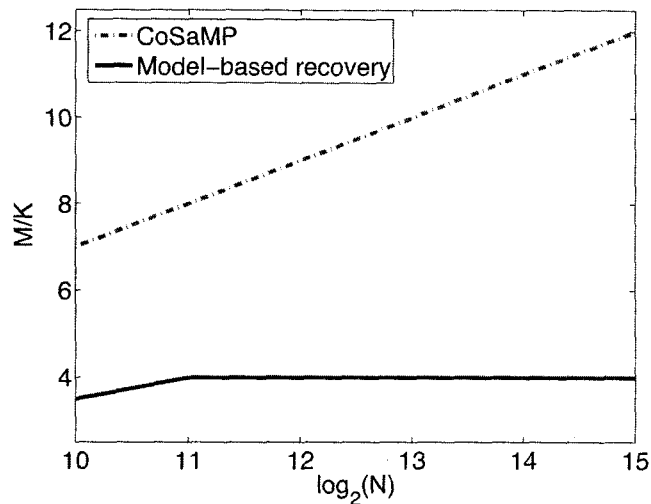


Figure 6.3 : Required overmeasuring factor  $M/K$  to achieve a target recovery error  $\|\mathbf{x} - \hat{\mathbf{x}}\|_2 \leq 2.5\sigma_{\mathcal{T}_K}(x)$  as a function of the signal length  $N$  for standard and model-based recovery of piecewise smooth signals. While standard recovery requires  $M$  to increase logarithmically with  $N$ , the required  $M$  is essentially constant for model-based recovery.

ery by  $K$ , the number of measurements required by model-based recovery is constant for all  $N$ . These experimental results verify the theoretical performance described in Proposition 6.3.

Furthermore, we demonstrate that model-based recovery performs stably in the presence of measurement noise. We generated sample piecewise polynomial signals as above, computed their best  $K$ -term tree-approximations, computed  $M$  measurements of each approximation, and finally added Gaussian noise of expected norm  $\|\mathbf{n}\|_2$  to each measurement. We emphasize that this noise model implies that the energy of the noise added will be larger as  $M$  increases. Then, we recovered the signal using CoSaMP and model-based recovery and measured the recovery error in each

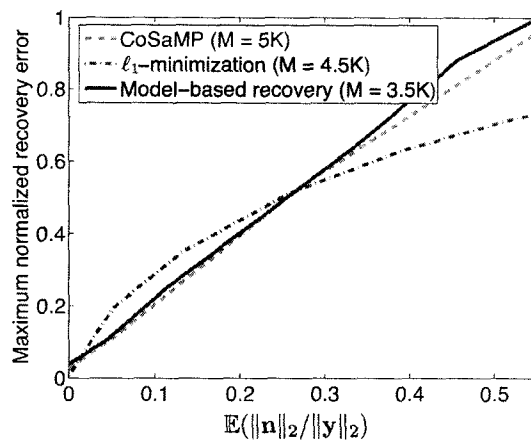


Figure 6.4 : Robustness to measurement noise for standard and wavelet tree-based CS recovery algorithms. We plot the maximum normalized recovery error over 200 sample trials as a function of the expected signal-to-noise ratio. The linear growth demonstrates that model-based recovery possesses the same robustness to noise as CoSaMP and  $\ell_1$ -norm minimization.

case. For comparison purposes, we also tested the recovery performance of the BPIC algorithm (see Section 2.2.4), which has been implemented as the `l1_qc` solver in the  $\ell_1$ -Magic toolbox. First, we determined the lowest value of  $M$  for which the respective algorithms provided near-perfect recovery in the absence of noise in the measurements. This corresponds to  $M = 3.5K$  for model-based recovery,  $M = 5K$  for CoSaMP, and  $M = 4.5K$  for  $\ell_1$  minimization. Next, we generated 200 sample tree-modeled signals, computed  $M$  *noisy* measurements, recovered the signal using the given algorithm and recorded the recovery error. Figure 6.4 illustrates the growth in maximum normalized recovery error (over the 200 sample trials) as a function of the expected measurement signal-to-noise ratio for the tree algorithms. We observe similar stability curves for all three algorithms, while noting that model-based recovery offers this kind of stability using significantly fewer measurements.

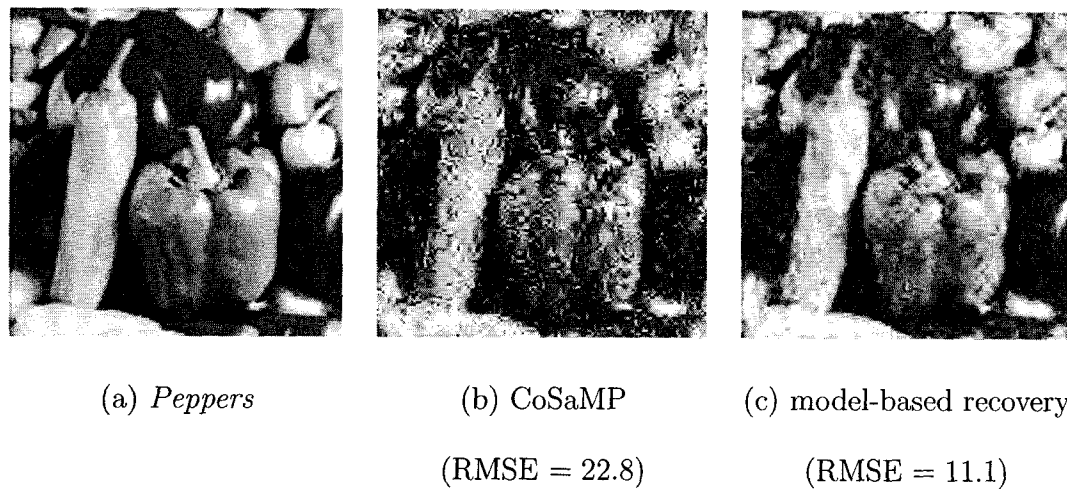


Figure 6.5 : *Example performance of standard and model-based recovery on images. (a)  $N = 128 \times 128 = 16384$ -pixel Peppers test image. Image recovery from  $M = 5000$  compressive measurements using (b) conventional CoSaMP and (c) our wavelet tree-based algorithm.*

Finally, we turn to two-dimensional images and a wavelet quadtree model. The connected wavelet-quadtree model has proven useful for compressing natural images [115]; thus, our algorithm provides a simple and provably efficient method for recovering a wide variety of natural images from compressive measurements. An example of recovery performance is given in Figure 6.5. The test image (*Peppers*) is of size  $N = 128 \times 128 = 16384$  pixels, and we computed  $M = 5000$  random Gaussian measurements. Model-based recovery again offers higher performance than standard signal recovery algorithms like CoSaMP, both in terms of recovery mean-squared error and visual quality.

## 6.4 Example: Block-Sparse Signals and Signal Ensembles

In a *block-sparse* signal, the locations of the significant coefficients cluster in blocks under a specific sorting order. Block-sparse signals have been previously studied in CS applications, including DNA microarrays and magnetoencephalography [19, 110]. An equivalent problem arises in CS for signal ensembles, such as sensor networks and MIMO communication [9, 19, 121], when signals have common sparse supports; see Section 2.4.1 for a description. Such signal ensembles can be re-shaped as a single vector by concatenation, and then the coefficients can be rearranged so that the concatenated vector exhibits block sparsity.

It has been shown that the block-sparse structure enables signal recovery from a reduced number of CS measurements, both for the single signal case [19, 110, 122] and the signal ensemble case [9], through the use of specially tailored recovery algorithms. However, the robustness guarantees for the algorithms [19, 110, 122] either are restricted to exactly sparse signals and noiseless measurements, do not have explicit bounds on the number of necessary measurements, or are asymptotic in nature. An optimization-based algorithm introduced in [19] provides similar recovery guarantees to those obtained by the algorithm we present in this chapter; thus, our method can be interpreted as a greedy-based counterpart to that provided in [19].

In this section, we formulate the block sparsity model as a union of subspaces and pose an approximation algorithm on this union of subspaces. The approximation algorithm is used to implement block-based signal recovery. We also define the corresponding class of block-compressible signals and quantify the number of mea-

surements necessary for robust recovery. For simplicity, and without loss of generality, we will assume in this section that the sparsity basis is  $\Psi = \mathbf{I}$ , so that  $\mathbf{x} = \theta$  is itself sparse.

#### 6.4.1 Block-Sparse Signals

Consider a class  $\mathcal{S}$  of signal vectors  $\mathbf{x} \in \mathbb{R}^{NJ}$ , with  $N$  and  $J$  integers. This signal can be reshaped into a  $N \times J$  matrix  $\mathbf{X}$ , and we use both notations interchangeably in this chapter. We will restrict entire rows of  $\mathbf{X}$  to be part of the support of the signal as a group. That is, signals  $\mathbf{X}$  in a block-sparse model have entire rows as zeros or nonzeros. The measure of sparsity for  $\mathbf{X}$  is its number of nonzero rows. More formally, we make the following definition.

**Definition 6.10** [19, 110] *Define the set of  $K$ -block sparse signals as*

$$\mathcal{S}_K = \{\mathbf{X} = [\mathbf{x}_1^T \dots \mathbf{x}_N^T]^T \in \mathbb{R}^{N \times J} \text{ s.t. } \mathbf{x}_n = 0 \text{ for } n \notin \Omega, \Omega \subseteq \{1, \dots, N\}, |\Omega| = K\}.$$

It is important to note that a  $K$ -block sparse signal has sparsity  $KJ$ , which is dependent on the size of the block  $J$ . We can extend this formulation to ensembles of  $J$ , length- $N$  signals with common sparse supports. Denote this signal ensemble by  $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_J\}$ , with  $\tilde{\mathbf{x}}_j \in \mathbb{R}^N$ ,  $1 \leq j \leq J$ . We formulate a matrix representation  $\tilde{\mathbf{X}}$  of the ensemble that features the signal  $\tilde{\mathbf{x}}_j$  in its  $j^{\text{th}}$  column:  $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1 \dots \tilde{\mathbf{x}}_J]$ . The matrix  $\tilde{\mathbf{X}}$  features the same structure as the matrix  $\mathbf{X}$  obtained from a block-sparse signal; thus, the matrix  $\tilde{\mathbf{X}}$  can be converted into a block-sparse vector  $\tilde{\mathbf{x}}$  that represents the signal ensemble.

### 6.4.2 Block-Based Approximation

To pose the block-based approximation algorithm, we need to define the mixed norm of a matrix.

**Definition 6.11** *The  $(p, q)$  mixed norm of the matrix  $\mathbf{X} = [\mathbf{x}_1^T \ \mathbf{x}_2^T \ \dots \ \mathbf{x}_N^T]^T$  is defined as*

$$\|\mathbf{X}\|_{(p,q)} = \left( \sum_{n=1}^N \|\mathbf{x}_n\|_p^q \right)^{1/q}.$$

When  $q = 0$ ,  $\|\mathbf{X}\|_{(p,0)}$  simply counts the number of nonzero rows in  $\mathbf{X}$ .

We immediately find that  $\|\mathbf{X}\|_{(p,p)} = \|\mathbf{x}\|_p$ , with  $\mathbf{x}$  the vectorization of  $\mathbf{X}$ . Intuitively, we pose the algorithm  $\mathbb{S}(\mathbf{X}, K)$  to obtain the best block-based approximation of the signal  $\mathbf{X}$  as follows:

$$\mathbf{X}_K^{\mathbb{S}} = \arg \min_{\tilde{\mathbf{X}} \in \mathbb{R}^{N \times J}} \|\mathbf{X} - \tilde{\mathbf{X}}\|_{(2,2)} \text{ subject to } \|\tilde{\mathbf{X}}\|_{(2,0)} \leq K. \quad (6.9)$$

It is easy to show that to obtain the approximation, it suffices to perform row-wise hard thresholding: let  $\rho$  be the  $K^{\text{th}}$  largest  $\ell_2$ -norm among the rows of  $\mathbf{X}$ . Then our approximation algorithm is  $\mathbb{S}(\mathbf{X}, K) = \mathbf{X}_K^{\mathbb{S}} = [(\mathbf{x}_{K,1}^{\mathbb{S}})^T \ \dots \ (\mathbf{x}_{K,N}^{\mathbb{S}})^T]^T$ , where

$$\mathbf{x}_{K,n}^{\mathbb{S}} = \begin{cases} \mathbf{x}_n & \|\mathbf{x}_n\|_2 \geq \rho, \\ \mathbf{0}_s & \|\mathbf{x}_n\|_2 < \rho, \end{cases}$$

for each  $1 \leq n \leq N$ . Alternatively, a recursive approximation algorithm can be obtained by sorting the rows of  $\mathbf{X}$  by their  $\ell_2$  norms, and then selecting the rows with largest norms. The complexity of this sorting process is  $\mathcal{O}(NJ + N \log N)$ .

### 6.4.3 Block-Compressible Signals

The approximation class in the block-compressible model corresponds to signals with blocks whose  $\ell_2$  norm has a power-law decay rate.

**Definition 6.12** *We define the set of  $s$ -block compressible signals as*

$$\mathfrak{S}_s = \{\mathbf{X} = [\mathbf{x}_1 \ \dots \ \mathbf{x}_N] \in \mathbb{R}^{J \times N} \text{ s.t. } \|\mathbf{x}_{\mathcal{I}(i)}\|_2 \leq Gi^{-s-1/2}, 1 \leq i \leq N, S < \infty\},$$

where  $\mathcal{I}$  indexes the sorted row norms.

We say that  $\mathbf{X}$  is an  $s$ -block compressible signal if  $\mathbf{X} \in \mathfrak{S}_s$ . For such signals, we have  $\|\mathbf{X} - \mathbf{X}_K\|_{(2,2)} = \sigma_{S_K}(\mathbf{X}) \leq G_1 K^{-s}$ , and  $\|\mathbf{X} - \mathbf{X}_K\|_{(2,1)} \leq G_2 K^{1/2-s}$ . Note that the block-compressible model does not impart a structure to the decay of the signal coefficients, so that the sets  $\mathcal{R}_{j,K}$  are equal for all values of  $j$ ; due to this property, the  $(\delta_{S_K}, s)$ -RAmP is implied by the  $S_K$ -RIP. Taking this into account, we can derive the following result from [66], which is proven similarly to Theorem 6.4.

**Theorem 6.6** *Let  $\mathbf{x}$  be a signal from the structured sparsity model  $\mathcal{S}$ , and let  $\mathbf{y} = \Phi\mathbf{x} + \mathbf{n}$  be a set of noisy CS measurements. If  $\Phi$  has the  $S_K^4$ -RIP with  $\delta_{S_K^4} \leq 0.1$ , then the estimate obtained from iteration  $i$  of block-based CoSaMP, using the approximation algorithm (6.9), satisfies*

$$\|\mathbf{x} - \hat{\mathbf{x}}_i\|_2 \leq 2^{-i} \|\mathbf{x}\|_2 + 20 \left( \|\mathbf{X} - \mathbf{X}_K^S\|_{(2,2)} + \frac{1}{\sqrt{K}} \|\mathbf{X} - \mathbf{X}_K^S\|_{(2,1)} + \|\mathbf{n}\|_2 \right).$$

Thus, the algorithm provides a recovered signal of similar quality to approximations of  $\mathbf{X}$  by a small number of nonzero rows. When the signal  $\mathbf{x}$  is  $K$ -block sparse,

we have that  $\|\mathbf{X} - \mathbf{X}_K^S\|_{(2,2)} = \|\mathbf{X} - \mathbf{X}_K^S\|_{(2,1)} = 0$ , obtaining the same result as Theorem 6.4, save for a constant factor.

#### 6.4.4 Stable Block-Based Recovery From Compressive Measurements

Since Theorem 6.6 poses the same requirement on the measurement matrix  $\Upsilon$  for sparse and compressible signals, the same number of measurements  $M$  is required to provide performance guarantees for block-sparse and block-compressible signals. The class  $\mathcal{S}_K$  contains  $S = \binom{N}{K}$  subspaces of dimension  $JK$ . Thus, a subgaussian random matrix has the  $\mathcal{S}_K$ -RIP property with constant  $\delta_{\mathcal{S}_K}$  and probability  $1 - e^{-t}$  if the number of measurements obeys

$$M \geq \frac{2}{c\delta_{\mathcal{S}_K}^2} \left( K \left( \ln \frac{2N}{K} + J \ln \frac{12}{\delta_{\mathcal{S}_K}} \right) + t \right). \quad (6.10)$$

To compare with the standard CS measurement bound, the number of measurements required for robust recovery scales as  $M = \mathcal{O}(JK + K \log(N/K))$ , which is a substantial improvement over the  $M = \mathcal{O}(JK \log(N/K))$  that would be required by conventional CS recovery methods. When the size of the block  $J$  is larger than  $\log(N/K)$ , then this term becomes  $\mathcal{O}(KJ)$ ; that is, it is linear on the total sparsity of the block-sparse signal.

We note in passing that the bound on the number of measurements (6.10) assumes a dense subgaussian measurement matrix, while the measurement matrices used in [9] have a block-diagonal structure. To obtain measurements from an  $M \times JN$  dense matrix in a distributed setting, it suffices to partition the matrix into  $J$  pieces of size  $M \times N$  and calculate the CS measurements at each sensor with the cor-



responding matrix; these individual measurements are then summed to obtain the complete measurement vector. For large  $J$ , (6.10) implies that the total number of measurements required for recovery of the signal ensemble is lower than the bound for the case where each signal recovery is performed independently for each signal ( $M = \mathcal{O}(JK \log(N/K))$ ).

#### 6.4.5 Experiments

We conducted several numerical experiments comparing model-based recovery to CoSaMP in the context of block-sparse signals. We employ the model-based CoSaMP recovery of Algorithm 7 with the block-based approximation algorithm (6.9) in all cases. For brevity, we exclude a thorough comparison of our model-based algorithm with  $\ell_1$ -norm minimization and defer it to future work. In practice, we observed that our algorithm performs several times faster than convex optimization-based procedures.

##### Block-sparse signals

Figure 6.6 illustrates an  $N = 4096$  signal that exhibits block sparsity, and its recovered version from  $M = 960$  measurements using CoSaMP and model-based recovery. The block size  $J = 64$  and there were  $K = 6$  active blocks in the signal. We observe the clear advantage of using the block-sparsity model in signal recovery.

We now consider block-compressible signals. An example recovery is illustrated in Figure 6.7. In this case, the  $\ell_2$ -norms of the blocks decay according to a power law, as described above. Again, the number of measurements is far below the mini-

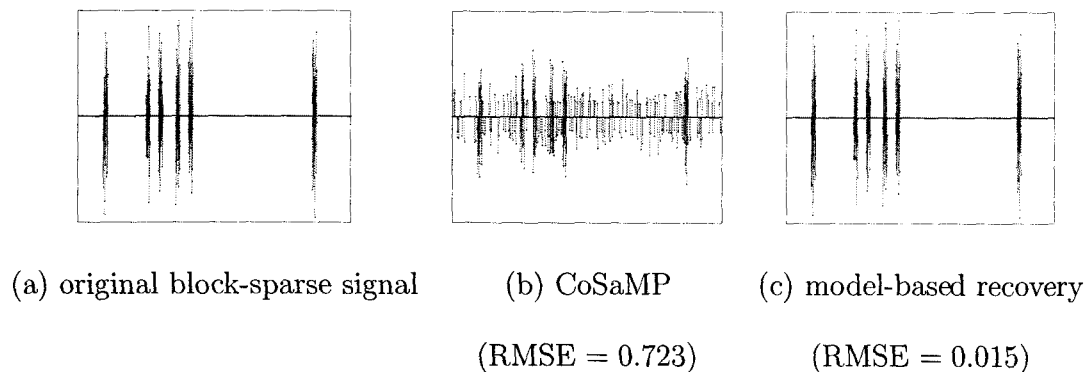


Figure 6.6 : Example performance of model-based signal recovery for a block-sparse signal. (a) Example block-sparse signal of length  $N = 4096$  with  $K = 6$  nonzero blocks of size  $J = 64$ . Recovered signal from  $M = 960$  measurements using (b) conventional CoSaMP recovery and (c) block-based recovery.

mum number required to guarantee stable recovery through conventional CS recovery. However, enforcing the structured sparsity model in the approximation process results in a solution that is very close to the best 5-block approximation of the signal.

Figure 6.8(a) indicates the decay in recovery error as a function of the numbers of measurements for CoSaMP and model-based recovery. We generated sample block-sparse signals as follows: we randomly selected a set of  $K$  blocks, each of size  $J$ , and endow them with coefficients that follow an i.i.d. Gaussian distribution. Each sample point in the curves is generated by performing 200 trials of the corresponding algorithm. As in the connected wavelet-tree case, we observe clear gains using model-based recovery, particularly for low-measurement regimes; CoSaMP matches model-based recovery only for  $M \geq 5K$ .

Figure 6.8(b) compares the recovery times of the two approaches. For this particular model, we observe that our proposed approach is in general much faster than

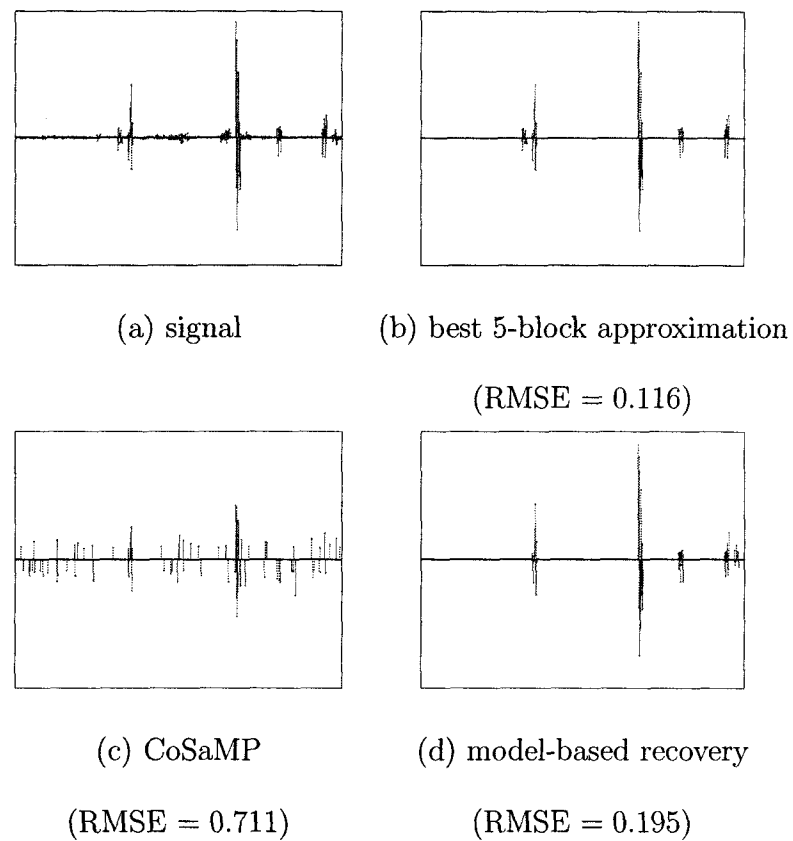


Figure 6.7 : *Example performance of model-based signal recovery for block-compressible signals. (a) Example block-compressible signal, length  $N = 1024$ . (b) Best block-based approximation with  $K = 5$  blocks. Recovered signal from  $M = 200$  measurements using both (c) conventional CoSaMP recovery and (d) block-based recovery.*

CoSaMP. This is because of two reasons: a) the block-based approximation step involves sorting fewer coefficients, and thus is faster than  $K$ -term approximation; b) block-based recovery requires fewer iterations to converge to the true solution.

### Signal ensembles with common sparse supports

We now consider the same environmental sensing dataset that was used in Section 4.4. The signals were recorded in an office environment and therefore exhibit periodic

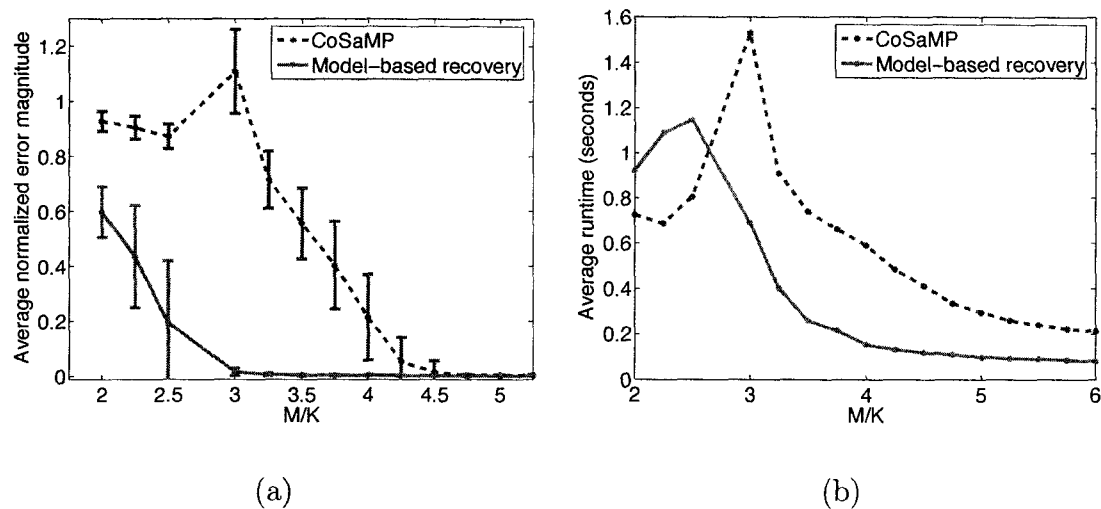


Figure 6.8 : Performance of CoSaMP vs. block-based recovery on a class of block-sparse signals. (a) Average normalized recovery error and (b) average runtime for each recovery algorithm as a function of the overmeasuring factor  $M/K$ . CoSaMP does not match the performance of the block-based algorithm until  $M = 5K$ . Furthermore, the block-based algorithm has faster convergence time than CoSaMP.

behavior caused by the activity levels during day and night. Therefore, we expect the signals to be compressible in the wavelet domain. Since the signals are observations of physical processes, they are smoothly varying in time and space; this causes the sensor readings to be close in value to each other, a situation well captured by the common sparse supports model.

We consider the recovery from CS measurements for these signals. We obtain  $M$  CS measurements for each signal using a matrix with random Gaussian distributed entries. We modify the model-based recovery algorithm due to the special structure observed by the distributed measurements performed. The resulting algorithm, which we call model-based distributed CoSaMP, is formalized as Algorithm 8. We then compare model-based recovery with standard CoSaMP recovery, where the parameter

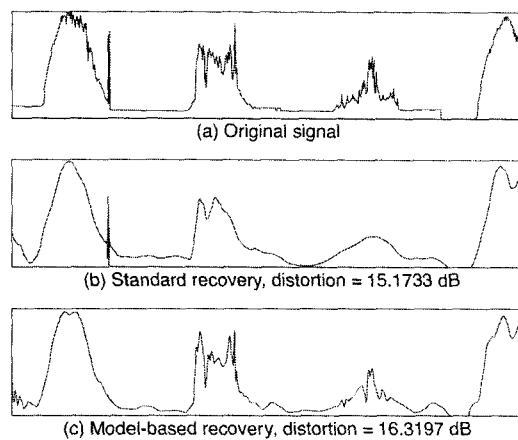


Figure 6.9 : *Recovery of light intensity signal 35 from the Intel Berkeley sensor network using the standard and model-based distributed CoSaMP (Algorithm 8).  $N = 1024$ ,  $M = 400$ . When the common sparse supports model is used in model-based distributed CoSaMP, the features that are salient in all signals are preserved, while those that are observed only in one signal (such as the spike on the left side of the signal) are removed.*

$K$  is chosen to achieve best performance.

Figure 6.9 shows the recovery for a representative example: the light intensity signal from sensor 35. The model-based recovery algorithm exploits the common sparse supports structure, recovering salient common features for all signals in the ensemble, and thus obtaining better recovery performance than standard CoSaMP from the same sets of measurements. Table 6.4.5 summarizes similar results for the different datasets.

We also study the performance of these algorithms for different numbers of measurements. Figures 6.10 – 6.12 plot the probability of exact recovery for the standard and model-based distributed CoSaMP recovery algorithms for the three environmental sensing datasets; we also show the performance of DCS-SOMP as a baseline. Model-based recovery is superior at low and moderate rates, yet it is surpassed by

Table 6.1 : Performance comparison for standard and model-based distributed CoSaMP recovery on 49 environmental sensing signals from the Intel Berkeley dataset.

Dataset	$M$	Standard	Model-based
Light	200	14.07dB	17.87dB
Humidity	80	20.45dB	26.68dB
Temperature	400	19.10dB	26.40dB

standard CoSaMP at high rates. This illustrates the applicability of the common sparse supports model, which becomes less valid as the very fine features of each signal (which vary between sensors) are incorporated. While the performance of model-based recovery is similar to that of DCS-SOMP, model-based recovery has the added benefit of the aforementioned recovery guarantees.

We now study the dependence of model-based recovery performance on the number of signals in the ensemble. Figure 6.13 compares the performance of the standard and model-based distributed CoSaMP algorithms on synthetically generated exactly sparse signals with common sparse supports. Over 100 repetitions, we select the signal supports at random and assign coefficients from a standard Gaussian distribution. We then obtain CS measurements for each signal using matrices with entries following a standard Gaussian distribution. The figure shows that while standard CoSaMP recovery requires more measurements to achieve high probability of successful recovery — as each sensor must succeed independently — the model-based recovery algorithm requires fewer measurements as the number of signals increases, as it is simpler to

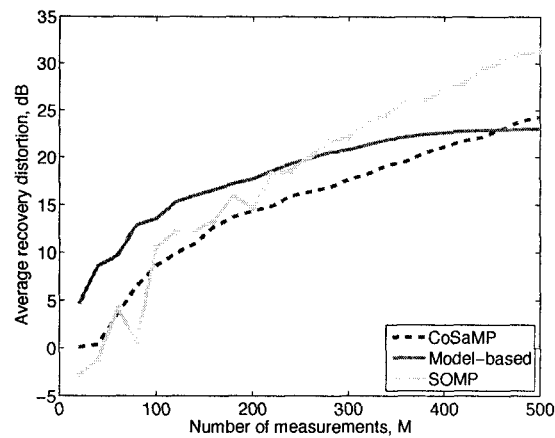


Figure 6.10 : *Performance of CoSaMP, DCS-SOMP and block-based recovery on a group of light signals from the Intel Berkeley sensor network as a function of the number of measurements  $M$ .*

establish the common sparse support structure. We also see that the number of measurements necessary for recovery appears to converge to  $M = 2K$  as the number of sensors becomes larger; in comparison, for the DCS-SOMP algorithm this number of measurements converged to  $M = K$  [74]. We believe that this increase in the bound is due to the enlarged support estimate obtained in model-based distributed CoSaMP.

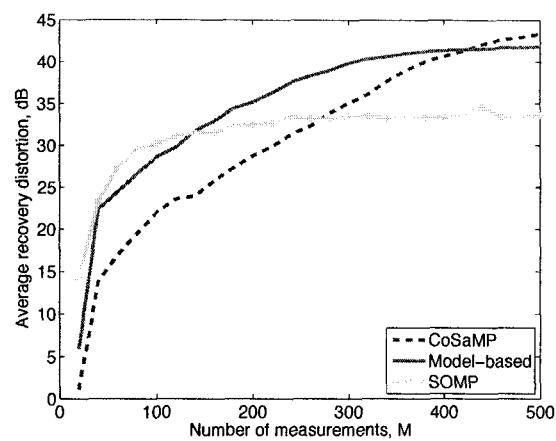


Figure 6.11 : Performance of CoSaMP, DCS-SOMP and block-based recovery on a group of humidity signals from the Intel Berkeley sensor network as a function of the number of measurements  $M$ .

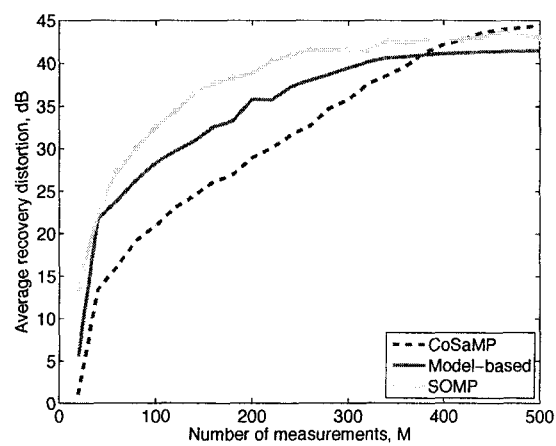


Figure 6.12 : Performance of CoSaMP, DCS-SOMP and block-based recovery on a group of temperature signals from the Intel Berkeley sensor network as a function of the number of measurements  $M$ .



---

**Algorithm 8** Model-based distributed CoSaMP

---

Inputs: CS matrices  $\{\Upsilon\}_{j=1}^J$ , measurements  $\{\mathbf{y}_j\}_{j=1}^J$

Output:  $K$ -sparse approximations  $\{\hat{\theta}_j\}_{j=1}^J$  to true signal representations  $\{\theta_j\}_{j=1}^J$

Initialize:  $i = 0$

**for**  $j = 1, \dots, J$  **do**

$\hat{\theta}_{j,0} = 0$ ,  $\mathbf{r}_j = \mathbf{y}_j$ ; {initialize}

**end for**

**while** halting criterion false **do**

1.  $i \leftarrow i + 1$

2.  $\mathbf{e}_j \leftarrow \Upsilon_j^T \mathbf{r}_j$ ,  $j = 1, \dots, J$  {form signal residual estimates}

3.  $\mathbf{e} = \sum_{j=1}^J (\mathbf{e}_j \cdot \mathbf{e}_j)$  {merge signal residual estimates  
in squared  $\ell_2$  norm}

4.  $\Omega \leftarrow \text{supp}(\mathfrak{T}(\mathbf{e}, 2K))$  {prune merged residual estimate  
according to structured sparsity}

5.  $T \leftarrow \Omega \cup \text{supp}(\hat{\theta}_{j,i-1})$  {merge supports}

6.  $\mathbf{b}_j|_T \leftarrow \Upsilon_{j,T}^\dagger \mathbf{y}_j$ ,  $\mathbf{b}_j|_{T^c} \leftarrow 0$  {form signal estimates}

7.  $\mathbf{b} = \sum_{j=1}^J (\mathbf{b}_j \cdot \mathbf{b}_j)$  {merge signal estimates  
in squared  $\ell_2$  norm}

8.  $\Lambda \leftarrow \text{supp}(\mathfrak{T}(\mathbf{b}, K))$  {prune signal estimate support}

9.  $\hat{\theta}_{j,i}|_\Lambda \leftarrow \mathbf{b}_j|_\Lambda$ ,  $\hat{\theta}_{j,i}|\_{\Lambda^c} \leftarrow 0$ ,  $j = 1, \dots, J$  {prune signal estimates}

10.  $\mathbf{r}_j \leftarrow \mathbf{y}_j - \Upsilon \hat{\theta}_{j,i}$ ,  $j = 1, \dots, J$  {update measurement residuals}

**end while**

return  $\hat{\theta} \leftarrow \hat{\theta}_i$

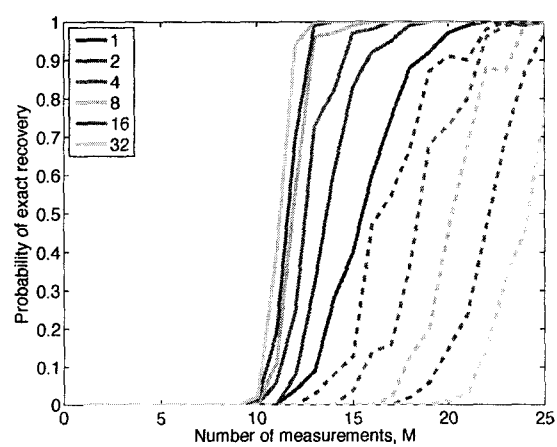


Figure 6.13 : Performance of CoSaMP (dashed lines) and model-based distributed CoSaMP (solid lines) on a class of signals with common sparse supports ( $K = 5$ ) as a function of  $M$  for several numbers of sensors  $J$ . While more measurements are required with CoSaMP as  $J$  increases, model-based CoSaMP requires a decreasing number of measurements, appearing to converge to  $M = 2K$  as  $J \rightarrow \infty$ .

## Chapter 7

### Kronecker Product Compressive Sensing

In this chapter,<sup>1</sup> we show that Kronecker product matrices are a natural way to generate sparsifying and measurement matrices for CS of multidimensional signals. Kronecker product sparsity bases *combine* the structures encoded by the sparsity bases for each signal dimension into a *single matrix*. Similarly, Kronecker product measurement matrices for multidimensional signals can be implemented by performing a *sequence of separate multiplexing operations on each dimension*. The Kronecker product formulation for sparsity bases and measurement matrices enables the derivation of analytical bounds for recovery of compressible multidimensional signals from randomized or incoherent measurements.

We can use Kronecker product matrices as sparsifying bases for multidimensional signals to jointly model the signal structure along each one of its dimensions when such structures can be expressed using sparsity or compressibility. In some cases, such as wavelet bases, it is possible to obtain bounds for the magnitude rate of decay for the coefficients of a signal when a Kronecker product basis is used. The Kronecker product basis rate of decay is dependent on the rates of decay for the coefficients of slices of the signals across the different dimensions using the individual bases.

---

<sup>1</sup>This work is in collaboration with Richard G. Baraniuk [123]. Thanks to Kevin Kelly, Ting Sun, and Dharmpal Takhar for providing experimental data for the single-pixel hyperspectral imager.

When the rates of decay using the corresponding bases for each of the dimensions are different, the Kronecker product basis rate will fall between the maximum and minimum rates among the different dimensions; when the rates of decay are all the same, they are matched by that of the Kronecker product basis.

Additionally, many of the CS measurements schemes proposed for multidimensional signals can be easily expressed as Kronecker product matrices. In particular, when partitioned measurements are used and the same measurement matrix is applied to each piece of the signal, the resulting measurement matrix can be expressed as the Kronecker product of an identity matrix and the measurement matrix used. We can also build new Kronecker measurement matrices that are performed in two stages: a first stage uses the same measurement vectors on each piece of a partitioned signal, and a second stage combines those measurements together using fixed linear combinations on measurements with matching indices.

When Kronecker matrices are used in CS, we can provide metrics to evaluate partitioned measurement schemes against Kronecker measurement matrices, as well as guidance on the improvements that may be afforded by the use of such multidimensional structures. We provide some initial results by studying the special case of signals that are compressible in a Kronecker products of wavelet bases, comparing the rates of decay for the CS recovery error when Kronecker products are used to that of standard CS recovery along a single dimension. We also verify our theoretical findings using experimental results with synthetic and real-world multidimensional signals.

## 7.1 Stylized Applications

### 7.1.1 Hyperspectral Imaging

Our Kronecker Compressive Sensing (KCS) concept is immediately applicable to several CS applications that use partitioned measurements. As an example, consider the hyperspectral single-pixel camera [7], which computes inner products of the pixels in each band of a hyperspectral datacube against a measurement vector with 0/1 entries by employing a digital micromirror device (DMD) as a spatial light modulator. Each spectral band's image is multiplexed by the same binary functions, as the DMD reflects all of the imaged spectra. This results in the same measurement matrix being applied to each spectral image, which results in a Kronecker product measurement matrix. Additionally, there are known compressibility bases for each spectral band as well as each pixel's spectral signature, which can be integrated into a single Kronecker product compressibility basis. An example datacube captured with a single-pixel hyperspectral camera is shown in Figure 7.1 [27].

### 7.1.2 Video Acquisition

Similarly, consider the example of compressive video acquisition, where a single-pixel camera applies the same set of measurements to each frame of the video sequence, resulting once again in a Kronecker product measurement matrix. It is possible to sparsify or compress the video sequence observed at each pixel using a Fourier or wavelet transform, depending on the video characteristics. Furthermore, as each frame of the video is an image, it is possible to sparsify each frame using standard

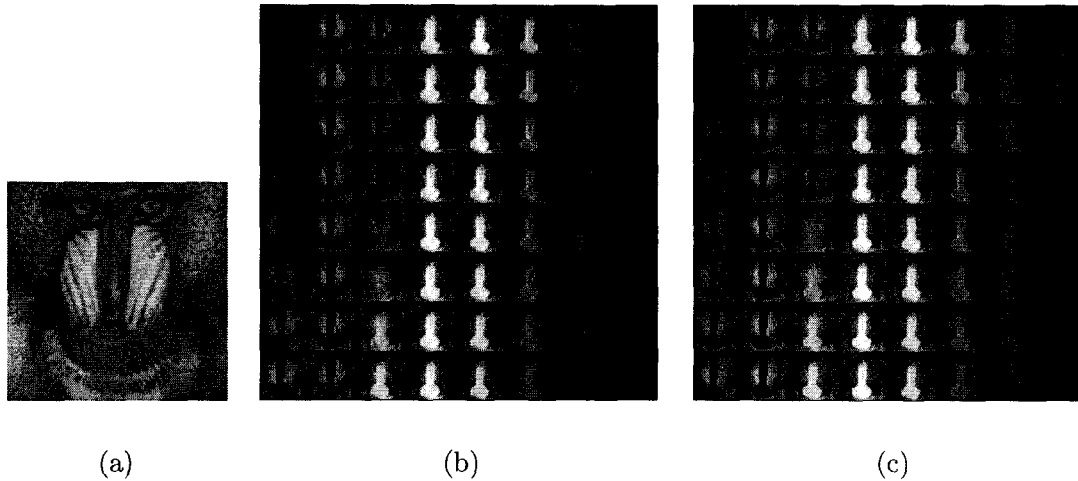


Figure 7.1 : Example capture from a single-pixel hyperspectral camera at resolution  $128 \times 128$  pixels by 64 spectral bands ( $2^{20}$  voxels) from  $M = 5000$  CS measurements ( $210\times$  sub-Nyquist) [27]. (a) Mandrill test image. (b) Hyperspectral datacube obtained via independent CS recovery of each spectral band. (c) Datacube obtained via KCS; marked improvement is seen in bands with low signal-to-noise ratios. Measurement data provided by Kevin Kelly, Ting Sun, and Dharmpal Takhar.

cosine or wavelet transforms. We can then use a Kronecker product of these two bases to sparsify or compress the video sequence.

### 7.1.3 Source Localization

Finally, consider the sparsity-based distributed localization problem [124], where a sparse vector encodes the locations of the sources in a localization grid, and the CS matrix encodes the propagation physics and known source signal. We can instead assume that the signal propagated by the target is not exactly known, but that it is sparse in a known basis. It is therefore possible to employ a Kronecker product matrix that encodes both the propagation physics and the sparse or compressible structure of the source signal. Such a structure has applications not only in sensors

and surveillance networks, but also in the localization of action potentials in multineuron recordings, where the neural activity or spikes are recorded by several electrodes with varying amplitudes (due to decay) and large electrical noise [125].

## 7.2 Background

### 7.2.1 Tensor and Kronecker Products

Let  $V$  and  $W$  represent Hilbert spaces. The *tensor product* of  $V$  and  $W$  is a new vector space  $V \otimes W$  together with a bilinear map  $\mathbb{T} : V \times W \rightarrow V \otimes W$  that is universal in the following sense: for every vector space  $X$  and every bilinear map  $\mathbb{S} : V \times W \rightarrow X$  there is a unique linear map  $\mathbb{S}' : V \otimes W \rightarrow X$  such that for all  $v \in V$  and  $w \in W$ ,  $\mathbb{S}(v, w) = \mathbb{S}'(\mathbb{T}(v, w))$ .

The *Kronecker product* of two matrices  $A$  and  $B$  of sizes  $P \times Q$  and  $R \times S$ , respectively, is defined as

$$A \otimes B := \begin{bmatrix} A(1,1)B & A(1,2)B & \dots & A(1,Q)B \\ A(2,1)B & A(2,2)B & \dots & A(2,Q)B \\ \vdots & \vdots & \ddots & \vdots \\ A(P,1)B & A(P,2)B & \dots & A(P,Q)B \end{bmatrix}, \quad (7.1)$$

Thus,  $A \otimes B$  is a matrix of size  $PR \times QS$ . The definition has a straightforward extension to the Kronecker product of vectors  $a \otimes b$ . In the case where  $V = \mathbb{R}^v$  and  $W = \mathbb{R}^w$ , it can be shown that  $V \otimes W \cong \mathbb{R}^{vw}$ , and a suitable map  $\mathbb{T} : \mathbb{R}^v \times \mathbb{R}^w \rightarrow \mathbb{R}^v \otimes \mathbb{R}^w$  is defined by the Kronecker product as  $\mathbb{T}(a, b) := a \otimes b$ .

Let  $\Psi_V = \{\psi_{V,1}, \psi_{V,2}, \dots\}$  and  $\Psi_W = \{\psi_{W,1}, \psi_{W,2}, \dots\}$  be bases for the spaces  $V$

and  $W$ , respectively. Then one can pose a basis for  $V \otimes W$  as  $\Psi_{V \otimes W} = \{\mathbb{T}(\psi_v, \psi_w) : \psi_v \in \Psi_V, \psi_w \in \Psi_W\}$ . Once again, when  $V = \mathbb{R}^v$  and  $W = \mathbb{R}^w$ , we will have  $\Psi_{V \otimes W} = \Psi_V \otimes \Psi_W$ .

### 7.2.2 Signal Ensembles

In distributed sensing problems, we aim to acquire an ensemble of signals  $\mathbf{x}_1, \dots, \mathbf{x}_J \in \mathbb{R}^N$  that vary in time, space, etc. We assume that each signal's structure can be encoded using sparsity with an appropriate basis  $\Psi'$ . This ensemble of signals can be expressed as a  $N \times J$  matrix  $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_J] = [\mathbf{x}^{1T} \ \mathbf{x}^{2T} \ \dots \ \mathbf{x}^{JT}]^T$ , where the individual signals  $\mathbf{x}_1, \dots, \mathbf{x}_J$  corresponding to columns of the matrix, and where the rows  $\mathbf{x}^1, \dots, \mathbf{x}^N$  of the matrix correspond to different *snapshots* of the signal ensembles at different values of time, space, etc. For brevity we refer to the rows of  $\mathbf{X}$  as signals and to its columns as snapshots. Under this construction, the structure of each signal is observable on each of the columns of the matrix, while the structure of each snapshot (spanning all the signals) is present on each of the rows of the matrix  $\mathbf{X}$ .

We expect that, in certain applications, the inter-signal correlations can also be modeled using sparsity; that is, that a basis or frame  $\Psi$  can be used to compress or sparsify  $\mathbf{x}^1, \dots, \mathbf{x}^N$ . For example, in sensor network applications, the structure of each snapshot is determined by the geometry of the sensing deployment, and can also be captured by a sparsity basis [124]. In such cases, it is desirable to obtain a single sparsity basis for the signal ensemble that encodes both intra- and inter-signal correlations; such representation would significantly simplify the analysis of



joint sparsity structures.

### 7.3 Kronecker Product Matrices for Multidimensional Compressive Sensing

We now describe our framework for the use of Kronecker product matrices in multidimensional CS. In this section, we will assume that the signal  $\mathbf{X}$  is 2-D and slightly abuse terminology by calling its rows and columns *snapshots* and *signals*, respectively; this allows us more easily to bridge the multidimensional signal and signal ensemble applications. While our exposition is based on 2-D signals for simplicity, the framework is extendable to multidimensional settings.

#### 7.3.1 Kronecker Product Sparsity Bases

It is possible to simultaneously exploit the sparsity properties of a multidimensional signal along each of its dimensions to provide a new representation for their structure. We obtain a single sparsity/compressibility basis for all signals and snapshots as the Kronecker product of the bases used for the individual signals and snapshots. For multidimensional signals, this encodes all of the available structure using a single transformation. For signal ensembles, we obtain a single coefficient vector to represent all the signals observed.

More formally, we denote the individual signals as  $\mathbf{x}_j \in \mathbb{R}^N$ ,  $1 \leq j \leq J$ , and the individual snapshots as  $\mathbf{x}^n \in \mathbb{R}^J$ ,  $1 \leq n \leq N$ . The multidimensional signal  $\mathbf{X}$  is then in  $\mathbb{R}^N \otimes \mathbb{R}^J \cong \mathbb{R}^{NJ}$ , where its columns corresponds to the individual

signals and the rows correspond to individual snapshots. We further assume that the snapshots  $\mathbf{x}^n$  are sparse or compressible in a basis  $\Psi$  and that the signals  $\mathbf{x}_j$  are sparse or compressible in a basis  $\Psi'$ . We then pose a sparsity/compressibility basis for  $\mathbf{X}$  obtained from Kronecker products as  $\bar{\Psi} = \Psi \otimes \Psi' = \{\psi \otimes \psi', \psi \in \Psi, \psi' \in \Psi'\}$ , and obtain a coefficient vector  $\Theta$  for the signal ensemble so that  $\tilde{\mathbf{X}} = \bar{\Psi}\Theta$ , where  $\tilde{\mathbf{X}} = [\mathbf{x}_1^T \ \mathbf{x}_2^T \ \dots \ \mathbf{x}_J^T]^T$  is a vector-reshaped representation of  $\mathbf{X}$ .

### 7.3.2 Kronecker Product Measurement Matrices

We can also design measurement matrices that are formulated as Kronecker products; such matrices correspond to measurement processes that operate first on each individual signal/snapshot, followed by operations on the measurements obtained for the different signals/snapshots, respectively. The resulting measurement matrix can be expressed as  $\tilde{\Phi} = \Phi \otimes \Phi'$ , with  $\Phi \in \mathbb{R}^{M_1 \times N}$ ,  $\Phi' \in \mathbb{R}^{M_2 \times J}$ , and  $\tilde{\Phi} \in \mathbb{R}^{M_1 M_2 \times NJ}$ . This results in a matrix that provides  $M = M_1 M_2$  measurements of the multidimensional signal  $\mathbf{X}$ .

Consider the example of distributed measurements, whose structure is succinctly captured by Kronecker products. We say that the measurements taken are distributed when for each signal  $\mathbf{x}_j$ ,  $1 \leq j \leq J$ , we obtain independent measurements  $\mathbf{y}_j = \Phi_j \mathbf{x}_j$  with an individual measurement matrix being applied to each signal. To compactly

represent the signal and measurement ensembles, we denote

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_J \end{bmatrix} \quad \text{and} \quad \tilde{\Phi} = \begin{bmatrix} \Phi_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Phi_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Phi_J \end{bmatrix}, \quad (7.2)$$

with  $\mathbf{0}$  denoting a matrix of appropriate size with all entries equal to 0. We then have  $\mathbf{Y} = \tilde{\Phi}\tilde{\mathbf{X}}$ . Equation (7.2) shows that the measurement matrix that arises from distributed sensing has a characteristic block-diagonal structure when the entries of the sparse vector are grouped by signal. If a matrix  $\Phi_j = \Phi'$  is used at each sensor to obtain its individual measurements, then we can express the joint measurement matrix the matrix as  $\tilde{\Phi} = \mathbf{I}_J \otimes \Phi'$ , where  $\mathbf{I}_J$  denotes the  $J \times J$  identity matrix.

### 7.3.3 Compressive Sensing Performance for Kronecker Product Matrices

We now derive results for metrics of Kronecker product sparsifying and sensing matrices that are relevant to obtain CS performance guarantees. The results obtained provide a link between the performance of the Kronecker product matrix and that of the individual matrices used in the product for CS recovery.

#### Mutual Coherence

Consider a Kronecker sparsity basis  $\tilde{\Psi} = \Psi \otimes \Psi'$  and a global measurement basis obtained through a Kronecker product of individual measurement bases:  $\tilde{\Phi} = \Phi \otimes \Phi'$ , with  $\Phi$  and  $\Psi$  and  $\Phi'$  and  $\Psi'$  being mutually incoherent. The following lemma provides a conservation of mutual coherence across Kronecker products.

**Lemma 7.1** *Let  $\Phi, \Psi$  and  $\Phi', \Psi'$  be bases or frames for  $\mathbb{R}^N$  and  $\mathbb{R}^J$ , respectively.*

*Then*

$$\mu(\Phi \otimes \Phi', \Psi \otimes \Psi') = \mu(\Phi, \Psi)\mu(\Phi', \Psi').$$

**Proof.** First we consider the inner product of columns from  $\tilde{\Phi} = \Phi \otimes \Phi'$  and  $\tilde{\Psi} = \Psi \otimes \Psi'$ .

$$\langle \tilde{\phi}_r, \tilde{\psi}_s \rangle = \sum_{n=1}^N \langle \phi_t(n) \phi'_u, \psi_v(n) \psi'_w \rangle$$

where  $r, s$  are aleatory indices for columns of  $\tilde{\Phi}$  and  $\tilde{\Psi}$ , respectively, and  $t, u, v, w$  are the indices for the columns of  $\Phi, \Phi', \Psi, \Psi'$ , respectively, involved in  $\tilde{\phi}_r$  and  $\tilde{\psi}_s$ .

We have

$$\langle \tilde{\phi}_r, \tilde{\psi}_s \rangle = \langle \phi'_u, \psi'_w \rangle \sum_{n=1}^N \phi_t(n) \psi_v^*(n) = \langle \phi'_u, \psi'_w \rangle \langle \phi_t, \psi_v \rangle. \quad (7.3)$$

Since the absolute values for the two inner products at the end of (7.3) attain the value of the mutual coherences for some values of  $t, u, v, w$ , then there exist  $r', s'$  for which

$$\left| \langle \tilde{\phi}_{r'}, \tilde{\psi}_{s'} \rangle \right| = \mu(\Phi', \Psi') \mu(\Phi, \Psi).$$

Furthermore, since for all other  $r, s$  the mutual coherence is an upper bound for the inner products at the end of (7.3),

$$\left| \langle \tilde{\phi}_r, \tilde{\psi}_s \rangle \right| = |\langle \phi'_u, \psi'_w \rangle| |\langle \phi_t, \psi_v \rangle| \leq \mu(\Phi', \Psi') \mu(\Phi, \Psi), \quad (r, s) \neq (r', s').$$

Thus, we have shown that  $\mu(\Phi \otimes \Phi', \Psi \otimes \Psi') = \mu(\Phi, \Psi)\mu(\Phi', \Psi')$ .  $\square$

Since the mutual incoherence of the snapshot sparsity and measurement basis is upper bounded by one, the number of Kronecker product-based measurements

necessary for successful recovery of the signal ensemble is strictly lower than the corresponding number of necessary partitioned measurements. This reduction is maximized when the snapshot measurement basis is  $\Phi$  maximally incoherent with the snapshot sparsity basis  $\Psi$ .

### Restricted Isometry Constants

The restricted isometry constants for  $\Phi$  are intrinsically tied to the singular values of all submatrices of  $\Phi$  of a certain size. The structure of Kronecker products enables simple bounds for their mutual coherence.

**Lemma 7.2** *Let  $\Phi$  and  $\Phi'$  be matrices with  $N$  and  $J$  columns, respectively, and with restricted isometry constants  $\delta_K(\Phi)$  and  $\delta_K(\Phi')$ , respectively. Then,*

$$\delta_K(\Phi \otimes \Phi') \leq \delta_K(\Phi) + \delta_K(\Phi') + \delta_K(\Phi)\delta_K(\Phi').$$

**Proof.** We denote by  $\tilde{\Phi}_\Omega$  the  $K$ -column submatrix of  $\tilde{\Phi}$  containing the columns  $\tilde{\phi}_t$ ,  $t \in \Omega$ ; its nonzero singular values obey

$$1 - \delta_K(\tilde{\Phi}) \leq \sigma_{\min}(\tilde{\Phi}_\Omega) \leq \sigma_{\max}(\tilde{\Phi}_\Omega) \leq 1 + \delta_K(\tilde{\Phi}).$$

Since each  $\tilde{\phi}_t = \phi_u \otimes \phi'_v$  for specific  $u, v$ , we can build sets  $\Omega_1, \Omega_2$  of cardinality up to  $K$  that contain the values of  $u, v$ , respectively, corresponding to  $t \in \Omega$ . Then, it is easy to see that  $\tilde{\Phi}_\Omega$  is a submatrix of  $\Phi_{\Omega_1} \otimes \Phi'_{\Omega_2}$ , which has up to  $K^2$  columns. Furthermore, it is well known that  $\sigma_{\min}(\Phi \otimes \Phi') = \sigma_{\min}(\Phi)\sigma_{\min}(\Phi')$  and  $\sigma_{\max}(\Phi \otimes \Phi') = \sigma_{\max}(\Phi)\sigma_{\max}(\Phi')$ . Additionally, the range of singular values of a submatrix are

interlaced inside those of the original matrix [126]. Thus,

$$\begin{aligned}\sigma_{\min}(\Phi_{\Omega_1} \otimes \Phi'_{\Omega_2}) &\leq \sigma_{\min}(\tilde{\Phi}_{\Omega}) \leq \sigma_{\max}(\tilde{\Phi}_{\Omega}) \leq \sigma_{\max}(\Phi_{\Omega_1} \otimes \Phi'_{\Omega_2}), \\ \sigma_{\min}(\Phi_{\Omega_1})\sigma_{\min}(\Phi'_{\Omega_2}) &\leq \sigma_{\min}(\tilde{\Phi}_{\Omega}) \leq \sigma_{\max}(\tilde{\Phi}_{\Omega}) \leq \sigma_{\max}(\Phi_{\Omega_1})\sigma_{\max}(\Phi'_{\Omega_2}).\end{aligned}$$

By using the  $K$ -restricted isometry constants for  $\Phi$  and  $\Phi'$ , we obtain the following bounds:

$$(1 - \delta_K(\Phi))(1 - \delta_K(\Phi')) \leq \sigma_{\min}(\tilde{\Phi}_{\Omega}) \leq \sigma_{\max}(\tilde{\Phi}_{\Omega}) \leq (1 + \delta_K(\Phi))(1 + \delta_K(\Phi')),$$

proving the lemma.  $\square$

When  $\Phi$  is an orthonormal basis, it has restricted isometry constant  $\delta_K(\Phi) = 0$  for all  $K \leq N$ . Therefore the restricted isometry constant of the Kronecker product of an orthonormal basis and a matrix is equal to that of the (second) matrix. We note, however, that the sparsity of the multidimensional signal in the basis  $\Phi \otimes \Phi'$  is larger than the sparsity of any of its pieces in each independent basis  $\Phi, \Phi'$ .

#### 7.3.4 Extensions to multidimensional settings

The framework provided in this section can be extended to multidimensional signals in two ways. One option is to use a sequence of multiple Kronecker products. For example, we can obtain a basis for 3-D volumes as the result of a dual Kronecker product  $\tilde{\Phi} = \Phi_1 \otimes \Phi_2 \otimes \Phi_3$ . Another option is to choose sparsity bases and measurement matrices for signal sections of 2-D or higher dimension. This entails reshaping the corresponding data sections into vectors, so that the basis can be expressed as a matrix and the Kronecker product can be performed. For example, in Section 7.5 we

obtain a basis for a 3-D hyperspectral datacube as the Kronecker product of a 2-D wavelet basis along the spatial dimensions with a 1-D wavelet basis along the spectral dimension.

## 7.4 Case Study: CS with Multidimensional Wavelet Bases

Kronecker products are prevalent in the extension of wavelet transforms to multi-dimensional settings. We describe several different wavelet basis constructions depending on the choice of basis vectors involved in the Kronecker products. For these constructions, our interest is in the relationship between the compressibility of each signal in the wavelet component basis and the compressibility of the signal ensemble in the wavelet Kronecker product basis. In the rest of this section, we assume that the  $N$ -length,  $D$ -D signal  $\mathbf{X}$  is a sampled representation of a continuous-indexed  $D$ -D signal  $f(t_1, \dots, t_D)$ , with  $t_d \in \Omega := [0, 1]$ ,  $1 \leq d \leq D$ , such that  $\mathbf{X}(n_1, \dots, n_D) = f(n_1/N_1, \dots, n_D/N_D)$ , with  $N = N_1 \times \dots \times N_D$ .

### 7.4.1 Isotropic, Anisotropic, and Hyperbolic Wavelets

Consider a 1-D signal  $g(t) : \Omega \rightarrow \mathbb{R}$  with  $\Omega = [0, 1]$ ; its wavelet representation is given by

$$g = v_0 \nu + \sum_{i \geq 0} \sum_{j=0}^{2^i-1} w_{i,j} \psi_{i,j},$$

where  $\nu$  is the scaling function and  $\psi_{i,j}$  is the wavelet function at scale  $i$  and offset  $j$ :

$$\psi_{i,j}(t) = \frac{1}{2^{i/2}} \psi \left( \frac{t}{2^i} - j \right).$$

The wavelet transform consists of the scaling coefficient  $v_0$  and wavelet coefficients  $w_{i,j}$  at scale  $i$ ,  $i \geq 0$ , and position  $j$ ,  $0 \leq j \leq 2^i - 1$ ; the support of the corresponding wavelet  $\psi_{i,j}$  is roughly  $[2^i j, 2^i(j+1)]$ . In terms of the sampled signal  $\mathbf{x}$  and our earlier matrix notation,  $\mathbf{x}$  has the representation  $\mathbf{x} = \Psi\theta$ , where  $\Psi$  is a matrix containing the scaling and wavelet functions for scales  $1, \dots, L = \log_2 N$  as columns, and  $\theta = [v_0 \ w_{0,0} \ w_{1,0} \ w_{1,1} \ w_{2,0} \dots]^T$  is the vector of corresponding scaling and wavelet coefficients. We are, of course, interested in sparse and compressible  $\theta$ .

Several different extensions exist for construction of  $D$ -D wavelet basis vectors as Kronecker product of 1-D wavelet functions [2, 127, 128]. In each case, a  $D$ -D wavelet basis vector is obtained from the Kronecker product of  $D$  1-D wavelet basis vectors:  $\psi_{i_1, j_1, \dots, i_D, j_D} = \psi_{i_1, j_1} \otimes \dots \otimes \psi_{i_D, j_D}$ . Different bases for the multidimensional space can then be obtained through the use of appropriate combinations of 1-D wavelet basis vectors in the Kronecker product. For example, *isotropic wavelets* arise when the same scale  $j = j_1 = \dots = j_D$  is selected for all wavelet functions involved, while *anisotropic wavelets* force a fixed factor between any two scales, i.e.  $a_{d,d'} = j_d/j_{d'}$ ,  $1 \leq d, d' \leq D$ . Additionally, *hyperbolic wavelets* result when no restriction is placed on the scales  $j_1, \dots, j_D$ . Therefore, the anisotropic wavelet basis can also be obtained as the Kronecker product of the individual wavelet basis matrices [127, 128]. In the sequel, we identify the isotropic, anisotropic, and hyperbolic wavelet bases as  $\Psi_I$ ,  $\Psi_A$ , and  $\Psi_H$ , respectively; example basis elements for each type of multidimensional wavelet basis are shown in Figure 7.2.



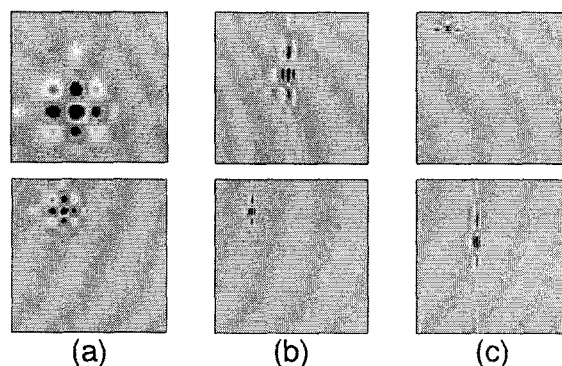


Figure 7.2 : *Example basis elements from 2-D wavelet bases. In each case, zeroes are represented by green pixels, while blue and red (dark-colored) pixels represent large values. (a) Isotropic wavelets have the same degree of smoothness on all dimensions, and are obtained from the Kronecker product of two 1-D wavelets of the same scale; (b) Anisotropic wavelets have different degrees of smoothness in each dimension, but with a constant ratio and are obtained from the Kronecker product of two 1-D wavelets at ratio-matching scales; (c) Hyperbolic wavelets have different degrees of smoothness in each dimension without restrictions and are obtained from the Kronecker product of two 1-D wavelets of any scale.*

#### 7.4.2 Isotropic Besov Spaces

The most popular type of multidimensional wavelet bases are isotropic wavelets, which have been found suitable for analysis of images and for specific video sequences [11]. Significant study has been devoted to identify the types of signals that are sparse or compressible in an isotropic wavelet basis. A fundamental result in this direction states that the discretizations of signals in isotropic Besov spaces are compressible in an appropriate wavelet transform. Such signals have the same degree of smoothness in all dimensions. We begin by providing a formal definition of Besov spaces.

We define the *derivative* of  $f$  in the direction  $h$  as  $(\Delta_h f)(t) := f(t+h) - f(t)$ , with higher-degree derivatives defined as  $(\Delta_h^m f)(t) := (\Delta_h(\Delta_h^{m-1} f))(t)$ ,  $m \geq 2$ . Here and later we define  $(\Delta_h f)(t) = 0$  if  $t+h \notin \Omega^D$ . For  $r \in \mathbb{R}^+$ ,  $m \in \mathbb{N}$  and  $0 < p < \infty$ ,

we define the *modulus of smoothness* as

$$\omega_m(f, r, \Omega^D)_p = \sup_{|h| \leq r} \|\Delta_h^m f\|_{p, \Omega^D}.$$

It is easy to see that  $\omega_m(f, r, \Omega^D)_p \rightarrow 0$  as  $r \rightarrow 0$ ; smoother functions have faster decay in this asymptotic behavior.

A signal can be classified according to its smoothness simply by posing conditions on the rate of decay of its moduli of smoothness. The resulting classes are known as *Besov spaces*. A Besov space  $B_{p,q}^s$  contains  $D$ -D or multidimensional functions that have (roughly speaking)  $s$  derivatives in  $L_p(\Omega^D)$ ; this smoothness is measured by the rate of decay of the modulus of smoothness as a function of the step size  $r$ . The Besov quasi-seminorm is then defined as

$$|f|_{B_{p,q}^s} = \left( \int_0^1 [r^{-s} \omega_m(f, r, \Omega^D)_p]^q \frac{dr}{r} \right)^{1/q}.$$

Here the parameter  $q$  provides finer distinctions of smoothness. Thus, we say that a signal  $f \in B_{p,q}^s$  if it has finite Besov norm, defined as  $\|f\|_{B_{p,q}^s} = \|f\|_p + |f|_{B_{p,q}^s}$ .

Similarly to the discrete signal case, we define the best  $K$ -term approximation error in the basis  $\Psi$  as

$$\sigma_K(f, \Psi)_p = \min \left\{ \|f - g\|_p, g = \sum_{k=1}^K c_j \psi_{i_k}, \psi_{i_k} \in \Psi \text{ for each } i = 1, \dots, K \right\}.$$

Such isotropic wavelet-based nonlinear approximations provide provable decay rates for the approximation error.

**Theorem 7.1** [129] *If the scaling function  $\nu \in B_{p,q}^s$ ,  $\nu$  has at least  $s$  vanishing moments, and  $f \in B_{p,q}^r$ , with  $r \geq D/p - D/2$  and  $0 < r < s$ , then  $\sigma_K(f, \Psi_I)_p < CK^{-r}$ .*

In words, Theorem 7.1 states that Besov-smooth signals are compressible in a sufficiently smooth wavelet transform.

### 7.4.3 Anisotropic Besov Spaces

In many applications the type of structure present is different in each of the signal's dimensions. For example, the smoothness of a video sequence is of different scales in the spatial and temporal dimensions, while the smoothness of a hyperspectral datacube can be different in the spatial and spectral dimensions. In these cases, anisotropic and hyperbolic wavelets can be used to achieve sparse and compressible representations for signals of this type. Similarly to isotropic Besov spaces, signals in anisotropic Besov spaces have discretizations that are compressible in an anisotropic wavelet basis. We first provide a formal definition of anisotropic Besov spaces, which closely mirrors that of standard Besov spaces, except that the smoothness in each dimension is specified separately.

We let  $f(t) := f((t_1, \dots, t_D)) : \Omega^D \rightarrow \mathbb{R}$  be a  $D$ -D function. We define the directional derivatives as  $(\Delta_{h,d}f)(t) := f(t + he_d) - f(t)$ ,  $1 \leq d \leq D$ , where  $e_d$  is the  $d^{th}$  canonical vector, i.e., its  $d^{th}$  entry is one and all others are zero. We also define higher-degree directional derivatives as  $(\Delta_{h,d}^m f)(t) := (\Delta_{h,d}(\Delta_{h,d}^{m-1}f))(t)$ ,  $m \geq 2$ . For  $r \in \mathbb{R}^+$ ,  $m \in \mathbb{N}$  and  $0 < p < \infty$ , we define the directional moduli of smoothness as

$$\omega_{m,d}(f, r, \Omega^D)_p = \sup_{|h| \leq r} \|\Delta_{h,d}^m f\|_{p, \Omega^D}.$$

By letting  $\vec{s} = (s_1, \dots, s_D)$ , we define the anisotropic Besov quasi-seminorm as

$$|f|_{B_{p,q}^{\vec{s}}} = \left( \int_0^1 \left[ \sum_{d=1}^D r^{-s_d} \omega_{m,d}(f, r, \Omega^D)_p \right]^q \frac{dr}{r} \right)^{1/q}.$$

An anisotropic Besov space  $B_{p,q}^{\bar{s}}$  contains functions of  $D$  continuous variables that have (roughly speaking)  $s_d$  derivatives in  $L_p(\Omega)$  for any slice of the  $D$ -D function along the  $d^{\text{th}}$  dimension; the parameter  $q$  provides finer distinctions of smoothness. An example is a multidimensional signal that is expressed as the Kronecker product of two signals that are compressible in wavelet bases.

We now study the conditions for compressibility of a signal in an anisotropic wavelets, as a function of the smoothness of the signal in its different dimensions. We will observe that the rate of decay for the wavelet coefficients will depend on the characteristics of the anisotropic Besov space in which the signal lives. Some conditions must be imposed on the wavelets used for compressibility. We denote by  $\nu_{i,j}(t) = 2^{j/2}\nu(2^j t - i)$  the scaling function dilated to scale  $j$  and translated to offset  $i$ , and  $\nu_{i_1,j_1,\dots,i_D,j_D} = \nu_{i_1,j_1} \otimes \dots \otimes \nu_{i_D,j_D}$ .

**Definition 7.1** *A scaling function  $\nu$  is  $B_{p,q}^{\bar{s}}$ -smooth,  $\bar{s} > 0$  (i.e.  $s_d > 0$ ,  $1 \leq d \leq D$ ), if for  $(m_1, \dots, m_D) > \bar{s}$  and  $j_1, \dots, j_D \in \mathbb{N}_0^D$  there are  $\bar{i}_1, \dots, \bar{i}_D$  such that for each  $k \in \mathbb{N}_0$ ,*

$$\omega_{m_d,d}(\nu_{i_1,j_1,\dots,i_D,j_D}, 2^{-k}, \Omega^D)_p < C \omega_{m_d,d}(\nu_{\bar{i}_1,j_1,\dots,\bar{i}_D,j_D}, 2^{-k}, \Omega^D)_p,$$

for  $1 \leq i_d \leq 2^{j_d}$ ,  $d = 1, \dots, D$ , and if for each  $(j_1, \dots, j_D) \in \mathbb{N}_0^D$  it holds that

$$|\nu_{\bar{i}_1,j_1,\dots,\bar{i}_D,j_D}|_{B_{p,q}^{\bar{s}}} < C 2^{(j_1+\dots+j_D)(1/2-1/p)} \sum_{d=1}^D 2^{j_d s_d}.$$

It can be shown that the scaling function formed from a Kronecker product of scaling functions has this smoothness property when the two individual scaling functions are smooth enough. This condition suffices to obtain results on approximation rates for

the different types of Kronecker product wavelet bases. The following theorem is an extension of a result from [128] to the  $D$ -D setting, and is proven in Appendix K.

**Theorem 7.2** *Assume the scaling function  $\nu$  that generates the anisotropic wavelet basis  $\Psi_A$  with anisotropy parameter  $\bar{s} = (s_1, \dots, s_D)$  is  $B_{p,q}^{\bar{s}}$ -smooth and that the function  $f \in B_{p,q}^{\bar{r}}$ , with  $\bar{r} = (r_1, \dots, r_D)$  and  $0 < \bar{r} < \bar{s}$ . Define  $\rho = \min_{1 \leq d \leq D} r_d$  and  $\lambda = \frac{D}{\sum_{d=1}^D 1/r_d}$ . If  $\rho > D/p + D/2$  then the approximation rate for the function  $f$  in an isotropic wavelet basis is  $\sigma_K(f, \Psi_I)_p < CK^{-\rho}$ . Similarly, if  $\lambda > D/p + D/2$ , then the approximation rate for the function  $f$  in both an anisotropic and a hyperbolic wavelet basis is  $\sigma_K(f, \Psi_A)_p < C_A K^{-\lambda}$  and  $\sigma_K(f, \Psi_H)_p < C_H K^{-\lambda}$ .*

To give some perspective for this theorem, we study two example cases: isotropy and extreme anisotropy. In the anisotropic case, all the individual rates  $r_d \approx r$ ,  $1 \leq d \leq D$ , and the approximation rate under anisotropic and hyperbolic wavelets matches that of isotropic wavelets:  $\rho \approx r$ . In the extreme anisotropic case, we have that one of the approximation rates is much smaller than all others:  $r_e \ll r_d$  for all  $e \neq d$ . In contrast, in this case we obtain a rate of approximation under anisotropic and hyperbolic wavelets of  $\rho \approx Dr_e$ , which is  $D$  times larger than the rate for isotropic wavelets. Thus, the approximation rate with anisotropic and hyperbolic wavelets is in the range  $\rho \in [1, D] \min_{1 \leq d \leq D} r_d$ .

The disadvantage of anisotropic wavelets, as compared with hyperbolic wavelets, is that they must have smoothness ratios between the dimensions that match that of the signal in order to achieve the optimal approximation rate. Additionally, the anisotropic wavelet basis is the only one out of the three basis types described that can

be expressed as the Kronecker product of lower dimensional wavelet *bases*. Therefore, we use hyperbolic wavelets in the sequel and in the experiments in Section 7.5.

#### 7.4.4 Performance of Kronecker Product CS Recovery with Multidimensional Wavelet Bases

When Kronecker product matrices are used for measurement and transform coding of compressible signals – a scheme we abbreviate as *Kronecker Compressive Sensing* (KCS) – it is possible to compare the rates of approximation that can be obtained by using independent measurements of each signal snapshot (or signal). The following Theorem is obtained by merging the results of Theorems 2.5, 2.7 and 7.2 and Lemma 7.1.

**Theorem 7.3** *Assume that a  $D$ - $D$  signal  $\mathbf{X} \in \mathbb{R}^{N_1 \times \dots \times N_D}$  is the sampled version of a continuous-time signal in  $B_{p,q}^{\bar{s}}$ , with  $\bar{s} = (s_1, \dots, s_D)$ , under the conditions of Theorem 7.2. That is,  $\mathbf{X}$  has  $s_d$ -compressible sections along its  $d^{\text{th}}$  dimension in a wavelet bases  $\Psi_d$ ,  $1 \leq d \leq D$ . Denote by  $\Phi_d$ ,  $1 \leq d \leq D$  a set of CS measurement bases that can be applied along each dimension of  $\mathbf{X}$ . If  $M$  measurements are obtained using a random subset of the columns of  $\Phi_1 \otimes \dots \otimes \Phi_D$ , then with high probability the recovery error from these measurements has the property*

$$\|\mathbf{X} - \hat{\mathbf{X}}\|_2 \leq CM^{-\beta} \prod_{d=1}^D \mu(\Phi_d, \Psi_d)^\beta, \quad (7.4)$$

where  $\beta = \frac{D}{2 \sum_{d=1}^D 1/s_d} - \frac{1}{4}$ , while the recovery from  $M$  measurements equally distributed among sections of the signal in the  $d^{\text{th}}$  dimension has the property

$$\|\mathbf{X} - \hat{\mathbf{X}}\|_2 \leq CM^{-\gamma_d} \mu(\Phi_d, \Psi_d)^{\gamma_d}, \quad (7.5)$$

for  $d = 1, \dots, D$ , where  $\gamma_d = \frac{s_d}{2} - \frac{1}{4}$ .

To summarize the theorem, the recovery error decay rate as the number of measurements increases matches that of the signal's compressibility approximation error; however, there is an additional factor dependent on the inverse of the mutual coherences that affects the decay with the same exponential rate of decay.

To put Theorem 7.3 in perspective, we consider the isotropic and extreme anisotropic cases. In the anisotropic case ( $s_d = s$ ,  $1 \leq d \leq D$ ), all approaches provide the same CS recovery approximation rate, i.e.,  $\beta = \gamma_d$ ,  $1 \leq d \leq D$ . In the extreme anisotropic case ( $s_e \ll s_d$ ,  $d \neq e$ ) the approximation rate of KCS recovery approaches  $\beta \approx Ds_e$ , while the approximation rate using standard CS on the sections of the signal along the  $d^{th}$  dimension is approximately  $\gamma_d \approx s_d$ . Thus, using KCS would only provide an advantage if the measurements are to be distributed along the  $e^{th}$  dimension.

It is desirable to find a meaningful comparison of these  $D + 1$  choices for CS recovery to determine the values of mutual coherences and compression error decay rates for which KCS is more advantageous. For example, the upper bound obtained from (7.4) is smaller than that of (7.5) for the  $e^{th}$  dimension when

$$M < \mu(\Phi_e, \Psi_e) \prod_{d \neq e} \mu(\Phi_d, \Psi_d)^{\frac{\beta}{\beta - \gamma_e}},$$

which provides the maximum number of measurements for which KCS outperforms standard CS with partitioned measurements.

## 7.5 Experimental Results

In this section, we perform experiments to verify the compressibility properties of multidimensional hyperspectral signals in a Kronecker product wavelet basis. We also perform experiments that showcase the advantage of using Kronecker product sparsity bases and measurement matrices when compared with schemes that operate on partitioned versions of the multidimensional signals.

### 7.5.1 Performance of Kronecker CS

Our first experiment considers synthetically generated  $8 \times 8 \times 8$  signals ( $N = 512$ ) that are  $K = 10$ -sparse in a Kronecker product basis, and compares three CS recovery schemes: the first one uses a single recovery from dense, *global* measurements; the second one uses a single KCS recovery from the set of measurements obtained independently from each  $8 \times 8$  slice; and the third one uses *independent* recovery of each  $8 \times 8$  slice from its individual measurements. We let the number of measurements  $M$  vary from 0 to  $N$ , with the measurements evenly split among the slices in the *independent* and KCS cases. For each value of  $M$ , we perform 100 iterations by generating  $K$ -sparse signals  $\mathbf{x}$  with independent and identically distributed (i.i.d.) Gaussian entries and with support following a uniform distribution among all supports of size  $K$ , and generating measurement matrices with i.i.d. Gaussian entries for each slice as well. We then measure the probability of successful recovery for each value of  $M$ , where a success is declared if the signal estimate  $\hat{\mathbf{x}}$  obeys  $\|\mathbf{x} - \hat{\mathbf{x}}\|_2 \leq 10^{-3}\|\mathbf{x}\|_2$ . The results are shown in Figure 7.3, which shows that KCS outperforms separate slice-by-



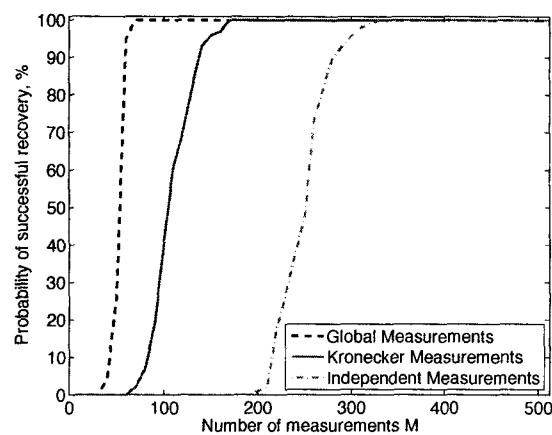


Figure 7.3 : *Performance of Kronecker product CS measurements.* We generate signals of size  $8 \times 8 \times 8$  that are 10-sparse in a Kronecker product of a 2D and 1D wavelet bases. We compare the performance of recovery from measurements taken using a Kronecker product of a 2D random dictionary and a 1D identity basis, measurements taken using a full 3D random matrix, and measurements taken using a 2D random dictionary separately for each 2D slice.

slice recovery, while achieving lower success probabilities than global measurements.

In fact, the overmeasuring factors  $M/K$  required for 95% success rate are 6, 15, and 30 for global measurements, KCS, and independent recovery, respectively.

### 7.5.2 Hyperspectral Data

Our second experiment performs an experimental evaluation of the compressibility of a real-world hyperspectral datacube using independent spatial and spectral sparsity bases and compares it with a Kronecker product basis. The datacube for this experiment is obtained from the AVIRIS database. A  $128 \times 128 \times 128$  voxel sample is taken, obtaining a signal of length  $N = 2^{21}$  samples. We then process the signal through three different transforms: the first two (*Space, Frequency*) perform wavelet

transforms along a subset of the dimensions of the data; the third one (*Kronecker*) transforms the entire datacube with a basis formed from the Kronecker product of a 2D isotropic wavelet basis in space and a 1D wavelet basis in frequency, providing a hyperbolic wavelet basis. For each one of these transforms, we measured the compression signal-to-noise ratio (SNR) when transform coding is used to preserve  $K$  coefficients of the data for varying values of  $K$ . The results are shown in Figures 7.4 and 7.5; the Kronecker transform provides the sparsest representation of the signal, outperforming the partial transforms in compression SNR. However, Figure 7.5(b) shows that the rate of decay for the normalized error of the Kronecker transform is only slightly higher than the minimum rate of decay among the individual transforms. Our analysis indicates that this result is due to the difference between the degrees of smoothness among the signal dimensions.

We also compare the performance of KCS to that of CS using standard bases to sparsify individual spectral frames or pixels. In our simulations we obtain CS measurements using the subsampled permuted Hadamard transform of [7] on each spectral frame. For KCS we use a single Kronecker product measurement matrix as shown in (7.2), while for standard CS we perform independent recovery of each spectral frame. We also obtain *global* CS measurements that depend on all the voxels of the datacube as a baseline; such measurements result in a fully dense measurement matrix  $\Phi$  and therefore are difficult to obtain in real-world applications. We perform the experiment using the datacube from the previous experiment; we also “flatten” it in the spectral dimension to 16, 32, and 64 bands through averaging of neighboring

bands.

Figure 7.6 shows the recovery error from several different setups: *Independent* recovery operates on each spectral band independently using a wavelet basis to sparsify each spectral band. KCS employs the Kronecker product formulations to perform joint recovery. We test two different Kronecker product bases: *KCS Wavelet* uses a Kronecker products of wavelet bases for both the spectral and spatial dimensions, and *KCS Fourier* uses a Kronecker products of a Fourier basis in the spectral dimension and a 2-D wavelet basis in the spatial dimensions. We also show results using the Kronecker product bases for sparsity together with *Global* measurements that depend on all voxels of the datacube.

For the smaller datacubes used in Figure 7.6(a–b), we see a strong advantage to the use of Kronecker product compressibility bases as compared to independent recovery. We also see an improvement for distributed measurements (used in KCS) over global measurements when the number of measurements  $M$  obtained for each band is small; as  $M$  increases, this advantage vanishes due to the availability of sufficient information. However, as the spectral resolution of the datacube increases (Figure 7.6(c–d)), the distributed (in the spectral dimension) measurement vectors become coherent with an increasing number of wavelets at fine scales, therefore deteriorating the performance of KCS. Furthermore, the datacube is likely to become less compressible in the bases chosen due to more sudden fluctuations in intensity for the wider spectral bands caused by the finer spectral resolution.

### 7.5.3 Video Data

Our third experiment performs an experimental evaluation of the compressibility in independent spatial (per frame) and temporal (per pixel) sparsity bases and compares it with a standard isotropic wavelet bases, as well as Kronecker product wavelet basis. We use the standard video sequences known as *Foreman*, *Mobile*, *Akiyo*, *Hall*, and *MotherDaughter*. We also test the *Dawn* sequence used in [11]. Each sequence is cropped to have frames of  $128 \times 128$  pixels and we preserve 128 frames for each sequence to obtain signals of length  $N = 2^{21}$  samples; the *Dawn* sequence, however, is originally of size  $64 \times 64 \times 64$ . We then process the signals through three different transforms: the first (*Space*) performs wavelet transforms along the spatial dimensions of the data; the second (*Isotropic*) uses standard isotropic 3D wavelets for the entire video sequence, and the third (*Kronecker*) transforms the entire sequence with a basis formed from the Kronecker product of a 2D isotropic wavelet basis in space and a 1D wavelet basis in time, providing a hyperbolic wavelet basis. For each one of these transforms, we measured the compression signal-to-noise ratio (SNR) when transform coding is used to preserve  $K$  coefficients of the data for varying values of  $K$ . The results are shown in Figure 7.7 for each sequence, and closely resemble those obtained for hyperspectral data. Additionally, the Kronecker product outperforms the isotropic wavelet transform, due to the difference in smoothness between the temporal and spatial dimensions. The *Dawn* sequence is the exception here.

We also compare the performance of KCS to that of CS using standard bases to sparsify individual frames. In our simulations we obtain CS measurements using

the subsampled permuted Hadamard transform of [7] on each video frame. For KCS we use a single Kronecker product measurement matrix as shown in (7.2), while for standard CS we perform independent recovery of each frame. We also obtain *global* CS measurements that depend on all the pixels of the video sequence as a baseline.

Figures 7.8 and 7.9 show the recovery error from several different setups: *Independent* recovery operates on each video frame independently, using a wavelet sparsifying basis. KCS employs the Kronecker product matrices to perform joint recovery of all frames. We also show results using the Kronecker product bases for sparsity together with *Global* measurements, as well as results using an *Isotropic* wavelet basis both with *Global* and *Distributed* measurements for comparison. The sequences used for the figures were *Foreman* and *Akiyo*, respectively. The *Foreman* and *Mobile* sequences feature camera movement, which is reflected in sharp changes in the value of each pixel across frames; in contrast, the *Akiyo*, *Hall* and *MotherDaughter* sequences have static camera placement, making the temporal changes of each pixel much smoother. The CS and KCS performance is very similar for the video sequences in each group, and so we omit the additional results.

Figures 7.8 and 7.9 show, once again, that the strong advantage of KCS with distributed sensing fades as the measurement vectors become coherent with more wavelet basis vectors, i.e., as the number of frames in the video increases. While the Kronecker product basis outperforms the isotropic wavelet basis when global measurements are used, the advantage is lost when we switch to distributed measurements, due to their high mutual coherence with the Kronecker product basis. In other words, using mea-

surements that are practically feasible incurs a penalty in the CS performance using Kronecker product bases.

#### 7.5.4 Single-Pixel Hyperspectral Camera

Our third experiment uses real-world data obtained from a imaging device that performs compressive sensing of multidimensional data using the distributed measurements of (7.2). It is possible to construct a powerful compressive hyperspectral imaging device simply by replacing the photosensor of the single-pixel camera of [7] by a spectrometer [27]. The resulting single-pixel hyperspectral camera effectively applies the same CS measurement matrix  $\Phi$  to each spectral band of the hyperspectral datacube, as all wavelengths are modulated in the same fashion by a digital micromirror device (DMD). The spectral band division can be performed dynamically since the spectral granularity of the CS measurements is determined by the spectrometer used.

Figure 7.1(a) shows an example capture from the single-pixel hyperspectral camera. The target is a printout of the *Mandrill* test image (illuminated by a desk lamp), for which 64 spectral bands spanning the 450–850 nm range at a resolution of  $128 \times 128$  pixels were obtained. In Figure 7.1(b), each spectral band was recovered independently. In Figure 7.1(c), the spectral bands were recovered jointly with KCS using the measurement structure of (7.2) and a hyperbolic wavelet basis. The results show considerable improvement in the quality of the recovery, particularly for those spectral frames with low signal power.

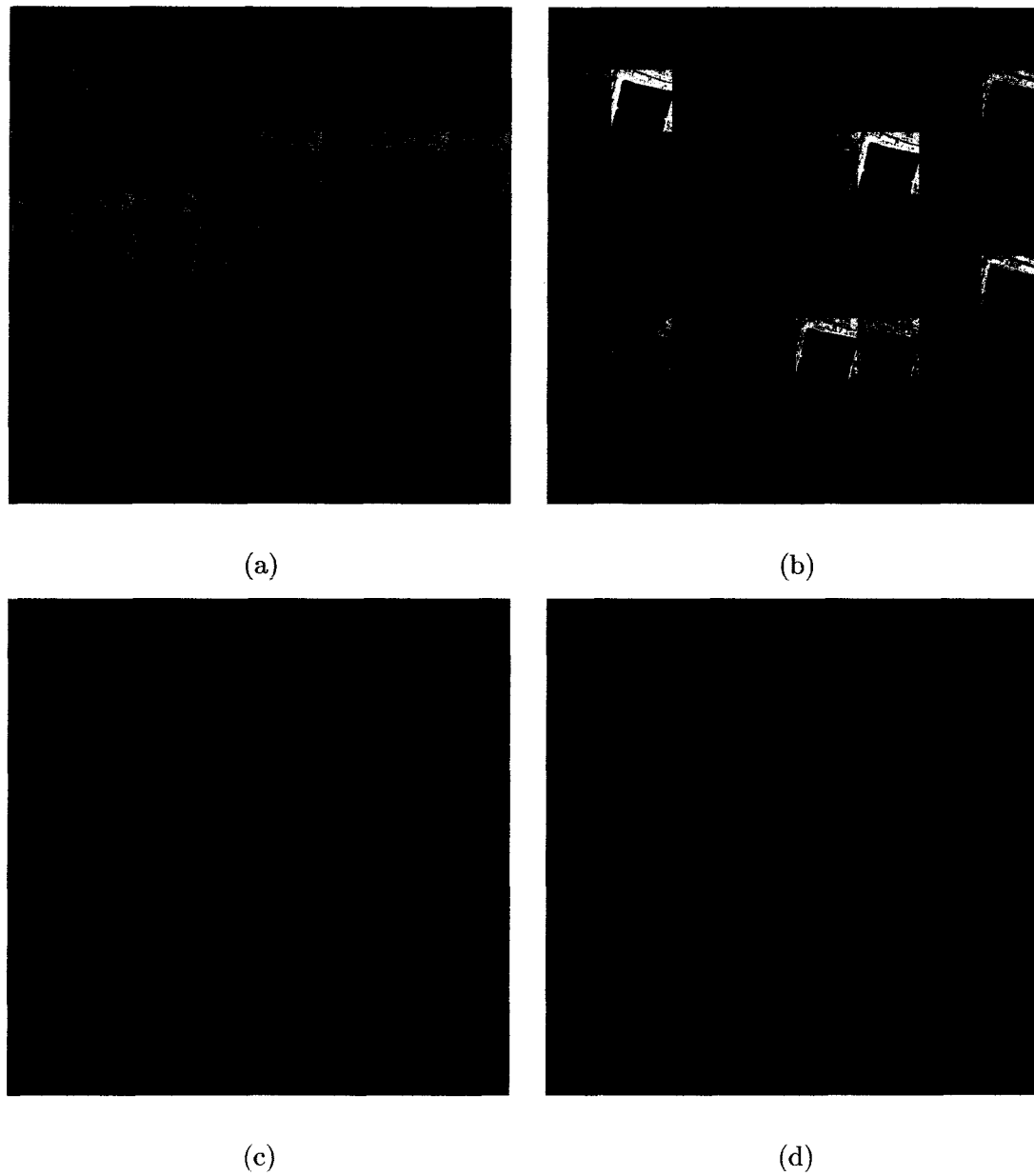


Figure 7.4 : Examples of transform coding of a hyperspectral datacube of size  $128 \times 128 \times 64$ . (a) Original data; (b) Coefficients of a wavelet transform applied at each pixel in the spectral domain; (c) Coefficients of a wavelet transform applied at each pixel in the spatial domain; (d) Coefficients of a Kronecker product hyperbolic wavelet transform. Each figure shows the datacube or coefficients flattened to 2D by concatenating each spectral band's image, left to right, top to bottom. In (b-d), dark blue pixels represent coefficients with small magnitudes.

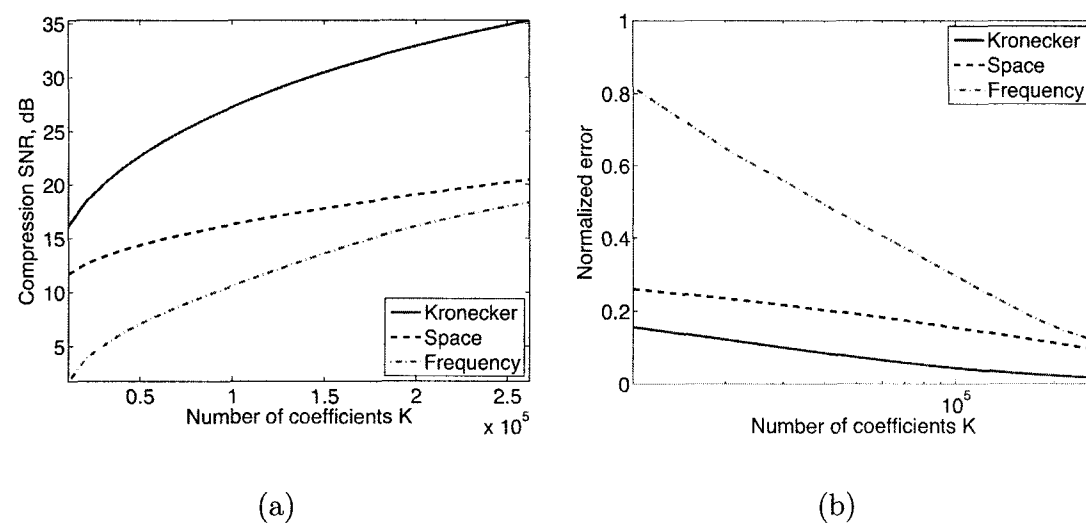


Figure 7.5 : *Performance of Kronecker product sparsity for hyperspectral imaging. A  $128 \times 128 \times 128$  voxel datacube is subject to transform coding using a 2D wavelet basis for each spectral slice, a 1D wavelet basis for each pixel, and a Kronecker product of these two bases for the entire datacube. (a) The Kronecker product performs better in distortion than either basis independently; however, (b) the rate of decay of the compression error using the Kronecker product basis is approximately the same as the lower rate obtained from the individual bases.*



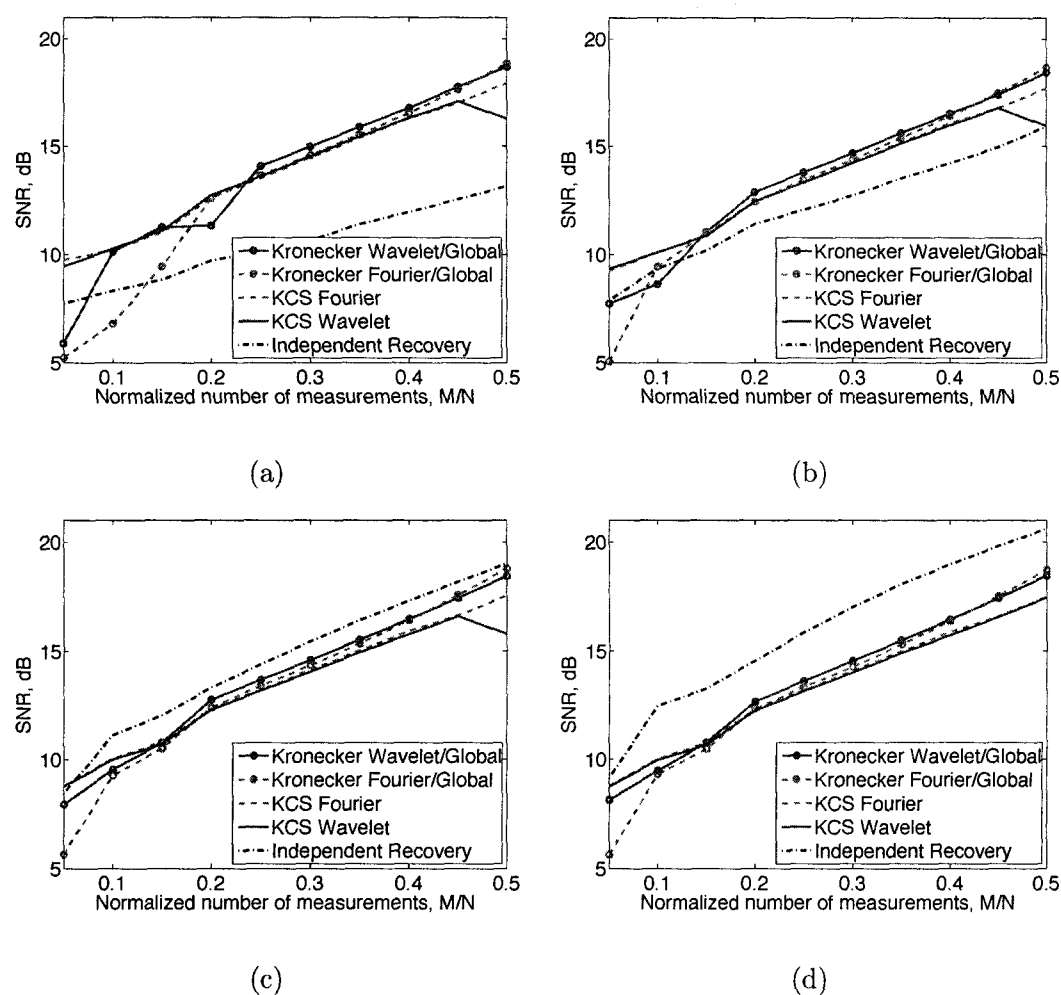


Figure 7.6 : Performance of Kronecker product sparsity and measurements matrices for hyperspectral imaging. Four versions of a datacube are subject to transform coding using a 2D wavelet basis for each spectral slice and a Kronecker product of a wavelet and a wavelet or Fourier basis for the entire datacube. The four versions used are of sizes (a)  $128 \times 128 \times 16$  voxels, (b)  $128 \times 128 \times 32$  voxels, (c)  $128 \times 128 \times 64$  voxels, and (d)  $128 \times 128 \times 128$  voxels. Recovery using the Kronecker product sparsifying basis outperforms separate recovery for the smaller datacubes. Additionally, there is an advantage to applying distributed rather than global measurements when the number of measurements  $M$  is low. However, when the resolution of the spectral dimension is increased, the distributed measurements in the spectral dimension become coherent with an increasing number of fine-scale wavelet basis vectors, therefore deteriorating the performance of KCS. Furthermore, the datacube is likely to become less compressible due to more sudden fluctuations in intensity.

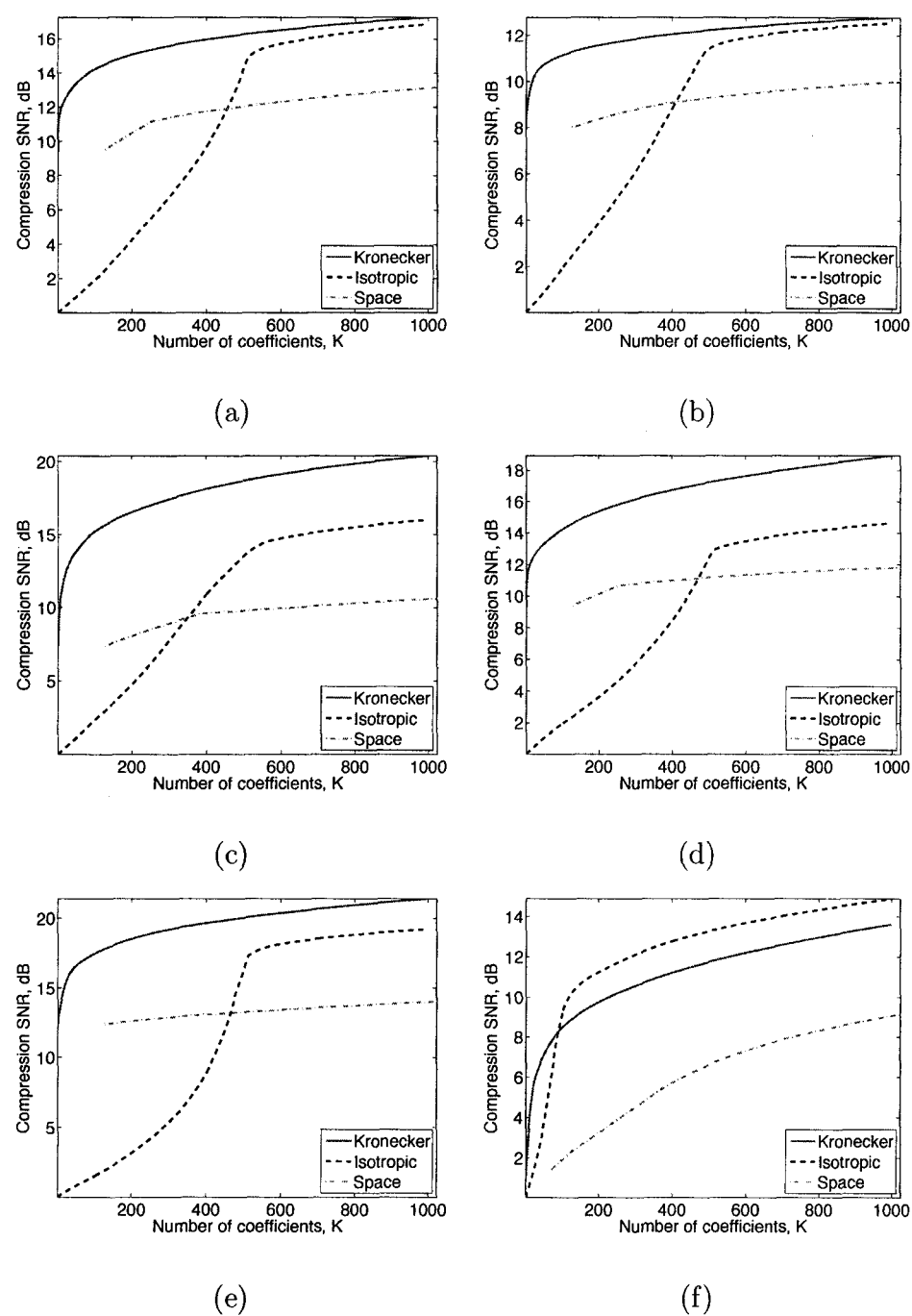


Figure 7.7 : Performance of Kronecker product sparsity basis for transform coding of video sequences. The sequences are of size  $128 \times 128 \times 128$  voxels. We subject to transform coding using a 2D wavelet basis for each frame, an isotropic wavelet bases for the sequence and a Kronecker product of a 2-D and a 1-D wavelet basis for the entire datacube. The sequences used are (a) Foreman, (b) Mobile, (c) Akiyo, (d) Hall, (e) MotherDaughter, and (f) Dawn (size  $64 \times 64 \times 64$ ). For (a-e), the Kronecker product performs better in distortion than the alternative bases.

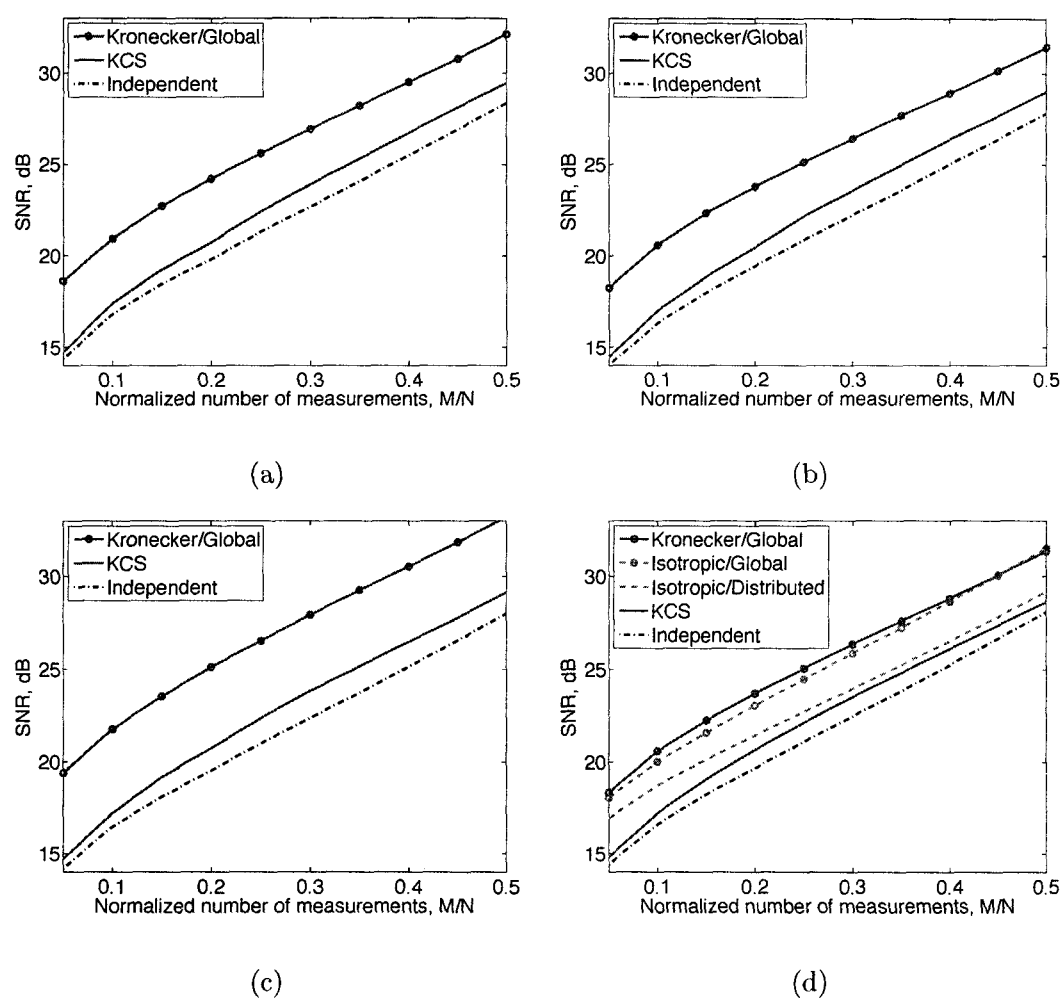


Figure 7.8 : Performance of Kronecker product sparsity and measurements matrices for the Foreman video sequence. Four versions of the video sequence are subject to transform coding using a 2D wavelet basis for each spectral slice and a Kronecker product of a wavelet and a wavelet or Fourier basis for the entire datacube. The four versions used are contain (a) 16, (b) 32, (c) 64 , and (d) 128 frames. Recovery using the Kronecker product sparsifying basis outperforms separate recovery. Additionally, the Kronecker basis outperforms isotropic wavelets when global measurements are used. However, when the measurements are distributed, the isotropic wavelet basis outperforms KCS due to the higher mutual coherence between distributed measurements and the hyperbolic wavelet basis.

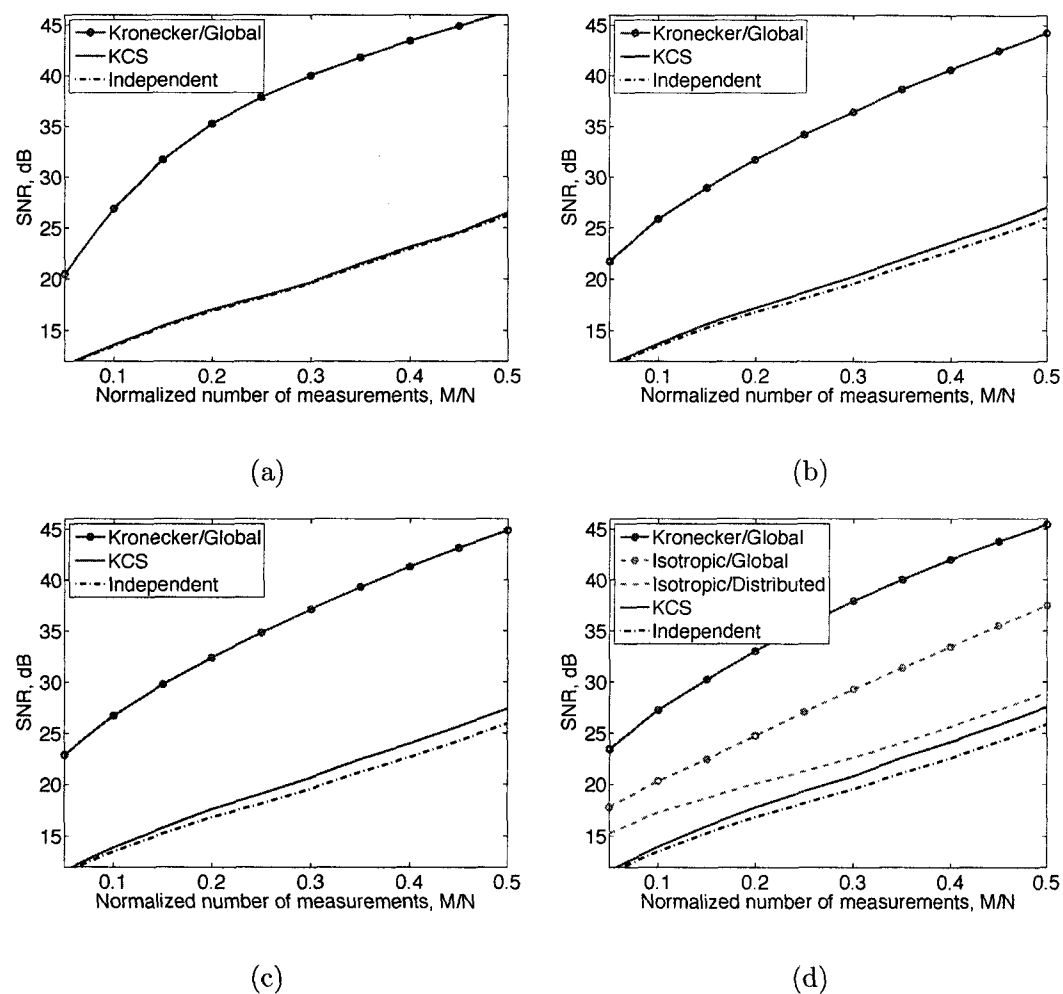


Figure 7.9 : Performance of Kronecker product sparsity and measurements matrices for the Akiyo video sequence. Four versions of the video sequence are subject to transform coding using a 2D wavelet basis for each spectral slice and a Kronecker product of a wavelet and a wavelet or Fourier basis for the entire datacube. The four versions used are contain (a) 16, (b) 32, (c) 64 , and (d) 128 frames. Recovery using the Kronecker product sparsifying basis matches separate recovery for the shortest video. However, the performance of KCS has a small improvement when the number of frames in the video increases. The Kronecker basis outperforms isotropic wavelets when global measurements are used. However, when the measurements are distributed, the isotropic wavelet basis outperforms KCS due to the higher mutual coherence between distributed measurements and the hyperbolic wavelet basis.

## Chapter 8

### Conclusions and Future Work

#### 8.1 Conclusions

In this thesis, we have proposed and studied a group of sparsity and compressibility models for signal ensembles and multidimensional signals. The models contributed here can be leveraged for signal compression through transform coding, as well as for compressive sensing (CS) and signal processing. Our focus was on prominent applications that are well suited to the properties of CS, such as sensor and camera networks [7, 9, 74, 130, 131], antenna and microphone arrays [132, 133], hyperspectral imaging [8, 10, 27], and video acquisition [11, 13–16]. The fact that the signals obtained in these applications span two or more physical dimensionalities that exhibit different types of structure allowed us to propose a variety of representations that exploit the structure present in each different dimension in a different fashion.

In Chapter 3, we provided fundamental lower bounds on the number of measurements required of each signal when a joint sparsity model is used. The bounds guarantee successful signal recovery using an algorithm with combinatorial computational complexity; while this recovery method is not feasible for routine applications, it does provide us with significant insight on the type of savings that joint sparsity models can provide. The results provided in Theorem 3.1, 3.2 and 3.3 are very reminiscent of

the Slepian-Wolf theorem for distributed source coding [72], with the obvious differences between sparsity metrics for finite-dimensional signals and the entropy metrics for source statistics. Similar to the Slepian-Wolf theorem, the bounds we obtained show that the number of measurements required for each group of signals must suffice to recover the nonzero coefficient information that is observed exclusively by that group. Similarly, for coefficients that are observed in a group of signals, we can partition the measurement burden between the corresponding sensors. Additionally, we verified the reduction in the number of measurements afforded by the use of joint sparsity models in real-world applications by applying our distributed CS framework on environmental sensor network data in Chapter 4.

In Chapters 5 and 6, we demonstrated that there are significant performance gains to be made by exploiting more realistic and richer signal models beyond the simplistic sparse and compressible models that dominate the CS literature. Building on unions of subspaces models, in Chapter 6 we provided an initial contribution towards what promises to be a general theory for model-based CS. We introduced the notion of a structured compressible signal and the associated restricted amplification property (RAmP) condition it imposes on the measurement matrix  $\Phi$ . For the volumes of natural and manmade signals and images that are wavelet-sparse or compressible, our tree-based CS recovery algorithms offer performance that significantly exceeds today's state-of-the-art while requiring only  $M = \mathcal{O}(K)$  rather than  $M = \mathcal{O}(K \log(N/K))$  random measurements. For block-sparse signals and signal ensembles with common sparse support, our block-based CS recovery algorithms offer not only excellent per-

formance but also require just  $M = \mathcal{O}(JK)$  measurements, where  $JK$  is the signal sparsity or ensemble joint sparsity, respectively. Therefore, block-based recovery can recover signal ensembles using fewer measurements than the number required when each signal is recovered independently. Additional structured sparsity models have been developed using our general framework in [134, 135].

In Chapter 7, we presented initial analytical results on the performance of CS using Kronecker product matrices. This theoretical framework is motivated by new sensing applications that acquire multidimensional signals in a progressive fashion, as well as by settings where the measurement process is distributed, such as sensor networks and arrays. We also provided analytical results for the recovery of signals that live in anisotropic Besov spaces, where there is a well-defined relationship between the degrees of compressibility obtained using lower-dimensional wavelet bases on subsets of the signal and multidimensional anisotropic wavelet bases on the entire signal. Furthermore, because the formulation follows the standard CS approach of single measurement and sparsifying matrices, standard recovery algorithms that provide provable recovery guarantees can be used; this obviates the need to develop ad-hoc algorithms to exploit additional signal structure.

## 8.2 Future Work

There are many avenues for future work on the topics of this thesis.

While joint sparsity models are based on the assumption that the signals are sparse or compressible in some basis, in some cases the event observed is governed by a small

number of parameters. Manifold models are often used for parameter estimation and signal processing tasks in this domain; fortunately, the structure of these models is also preserved by randomized linear projections [136].

Manifold models can also be extended to signal ensembles [137]. Consider the example of an array of antennas that sense a known signal that is emitted from an unknown location. In this case, determining the location of the emitter allows us to compute the recorded signals through the use of a physical model for the transmission medium and emitting and sensing devices. Therefore, when such a physical model is available, the received signal can be described succinctly by the location of the emitter, which provides a two-dimensional signal parameterization. By noting that the same parameter values underlie each of the recorded signals, it is straightforward to pose a single manifold model for a concatenation of the signals in the ensemble.

Similarly, for model-based CS, we have only considered the recovery of signals from models that can be geometrically described as a union of subspaces; possible extensions include other, more complex geometries (for example, high-dimensional polytopes, nonlinear manifolds). Furthermore, our framework will benefit from the formulation of new structured sparsity models that are endowed with efficient structured sparse approximation algorithms. We also expect that the core of our proposed algorithms — a structured sparse approximation step — can be integrated into other iterative algorithms, including relaxed  $\ell_1$ -norm minimization methods. Several such algorithms employ soft thresholding operations, in contrast to the hard thresholding operations performed by most greedy and greedy-inspired CS recov-



ery algorithms [64, 65]. Therefore, there is some promise in the modification of soft thresholding algorithms to enforce the structure present in a union-of-subspaces model [138, 139].

Further work for Kronecker CS remains in finding additional signal classes for which the use of multidimensional structures provide an advantage during compression. Some promising candidates include modulation spaces, which contain signals that can be compressed using Wilson and brushlet bases [140, 141]. This framework also motivates the formulation of novel structured representations using sparsifying bases in applications where transform coding compression schemes have not been developed, such as terahertz imaging.

Finally, the Kronecker CS can enable several different types of analysis for the performance of CS using partitioned measurements. For example, the size of the partitioning for a Kronecker product measurement matrix affects the performance; intuitively, the size directly controls the amount of randomization present in the matrix, with a minimum appearing when the size of each piece is close to the number of pieces in the partition. Similarly, consider the case of a Kronecker product basis that is known to provide good performance for transform coding of a multidimensional signal. We can design Kronecker product measurement matrices composed of measurement bases that are incoherent with the corresponding basis used in the Kronecker product basis for the corresponding dimension. As an example, distributed measurements along a given dimension — which correspond to a measurement matrix equal to a submatrix of the identity — will be optimal for signals that are sparse or compressible

in a Fourier basis along the same dimension, and will work well with signals that are sparse or compressible with bases whose vectors have dense or global supports.

## Appendix A

### Proof of Theorem 3.1

Let

$$D := K_C + \sum_{j \in \Lambda} K_j \quad (\text{A.1})$$

denote the number of columns in  $\mathbf{P}$ . Because  $\mathbf{P} \in \mathcal{P}_F(X)$ , there exists  $\Theta \in \mathbb{R}^D$  such that  $\mathbf{X} = \mathbf{P}\Theta$ . Because  $\mathbf{Y} = \Phi\mathbf{X}$ , then  $\Theta$  is a solution to  $\mathbf{Y} = \Phi\mathbf{P}\Theta$ . We will argue that, with probability one over  $\Phi$ ,

$$\Upsilon := \Phi\mathbf{P}$$

has rank  $D$ , and thus  $\Theta$  is the unique solution to the equation  $\mathbf{Y} = \Phi\mathbf{P}\Theta = \Upsilon\Theta$ .

We recall that, under our common/innovation model,  $P$  has the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_C & \mathbf{P}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{P}_C & \mathbf{0} & \mathbf{P}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_C & \mathbf{0} & \mathbf{0} & \dots & \mathbf{P}_J \end{bmatrix},$$

where  $\mathbf{P}_C$  is an  $N \times K_C$  submatrix of the  $N \times N$  identity, and each  $\mathbf{P}_j$ ,  $j \in \Lambda$ , is an  $N \times K_j$  submatrix of the  $N \times N$  identity.

To prove that  $\Upsilon$  has rank  $\Xi$ , we will require the following lemma, which we prove in Appendix B.

**Lemma A.1** *If (3.3) holds, then there exists a mapping  $\mathfrak{M} : \{1, 2, \dots, K_C\} \rightarrow \Lambda$ , assigning each element of the common component to one of the sensors, such that for each  $\Gamma \subseteq \Lambda$ ,*

$$\sum_{j \in \Gamma} M_j \geq \sum_{j \in \Gamma} K_j + \sum_{k=1}^{K_C} 1_{\mathfrak{M}(k) \in \Gamma} \quad (\text{A.2})$$

*and such that for each  $k \in \{1, 2, \dots, K_C\}$ , the  $k^{\text{th}}$  column of  $\mathbf{P}_C$  does not also appear as a column of  $\mathbf{P}_{\mathfrak{M}(k)}$ .*

Intuitively, the existence of such a mapping suggests that (i) each sensor has taken enough measurements to cover its own innovation (requiring  $K_j$  measurements) and perhaps some of the common component, (ii) for any  $\Gamma \subseteq \Lambda$ , the sensors in  $\Gamma$  have collectively taken enough extra measurements to cover the requisite  $K_C(\Gamma, \mathbf{P})$  elements of the common component, and (iii) the extra measurements are taken at sensors where the common and innovation components do not overlap. Formally, we will use the existence of such a mapping to prove that  $\Upsilon$  has rank  $D$ .

We proceed by noting that  $\Upsilon$  has the form

$$\Upsilon = \begin{bmatrix} \Phi_1 \mathbf{P}_C & \Phi_1 \mathbf{P}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \Phi_2 \mathbf{P}_C & \mathbf{0} & \Phi_2 \mathbf{P}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi_J \mathbf{P}_C & \mathbf{0} & \mathbf{0} & \dots & \Phi_J \mathbf{P}_J \end{bmatrix},$$

where each  $\Phi_j \mathbf{P}_C$  (respectively,  $\Phi_j \mathbf{P}_j$ ) is an  $M_j \times K_C$  (respectively,  $M_j \times K_j$ ) submatrix of  $\Phi_j$  obtained by selecting columns from  $\Phi_j$  according to the nonzero entries of  $\mathbf{P}_C$  (respectively,  $\mathbf{P}_j$ ). In total,  $\Upsilon$  has  $D$  columns (A.1). To argue that  $\Upsilon$  has rank  $D$ , we will consider a sequence of three matrices  $\Upsilon_0$ ,  $\Upsilon_1$ , and  $\Upsilon_2$  constructed from small

modifications to  $\Upsilon$ .

We begin by letting  $\Upsilon_0$  denote the “partially zeroed” matrix obtained from  $\Upsilon$  using the following construction. We first let  $\Upsilon_0 = \Upsilon$  and then make the following adjustments:

1. Let  $k = 1$ .
2. For each  $j$  such that  $\mathbf{P}_j$  has a column that matches column  $k$  of  $\mathbf{P}_C$  (note that by Lemma A.1 this cannot happen if  $\mathfrak{M}(k) = j$ ), let  $k'$  represent the column index of the full matrix  $\mathbf{P}$  where this column of  $\mathbf{P}_j$  occurs. Subtract column  $k'$  of  $\Upsilon_0$  from column  $k$  of  $\Upsilon_0$ . This forces to zero all entries of  $\Upsilon_0$  formerly corresponding to column  $k$  of the block  $\Phi_j \mathbf{P}_C$ .
3. If  $k < K_C$ , add one to  $k$  and go to step 2.

The matrix  $\Upsilon_0$  is identical to  $\Upsilon$  everywhere except on the first  $K_C$  columns, where any portion of a column overlapping with a column of  $\Phi_j \mathbf{P}_j$  to its right has been set to zero. Thus,  $\Upsilon_0$  satisfies the following two properties, which will be inherited by matrices  $\Upsilon_1$  and  $\Upsilon_2$  that we subsequently define:

P1. Each entry of  $\Upsilon_0$  is either zero or a Gaussian random variable.

P2. All Gaussian random variables in  $\Upsilon_0$  are i.i.d.

Finally, because  $\Upsilon_0$  was constructed only by subtracting columns of  $\Upsilon$  from one another,

$$\text{rank}(\Upsilon_0) = \text{rank}(\Upsilon). \quad (\text{A.3})$$

We now let  $\Upsilon_1$  be the matrix obtained from  $\Upsilon_0$  using the following construction. For each  $j \in \Lambda$ , we select  $K_j + \sum_{k=1}^{K_C} 1_{\mathfrak{M}(k)=j}$  arbitrary rows from the portion of  $\Upsilon_0$  corresponding to sensor  $j$ . Using (A.1), the resulting matrix  $\Upsilon_1$  has

$$\sum_{j \in \Lambda} \left( K_j + \sum_{k=1}^{K_C} 1_{\mathfrak{M}(k)=j} \right) = \sum_{j \in \Lambda} K_j + K_C = D$$

rows. Also, because  $\Upsilon_1$  was obtained by selecting a subset of rows from  $\Upsilon_0$ , it has  $D$  columns (A.1) and satisfies

$$\text{rank}(\Upsilon_1) \leq \text{rank}(\Upsilon_0). \quad (\text{A.4})$$

We now let  $\Upsilon_2$  be the  $D \times D$  matrix obtained by permuting columns of  $\Upsilon_1$  using the following construction:

1. Let  $\Upsilon_2 = [ ]$ , and let  $j = 1$ .
2. For each  $k$  such that  $\mathfrak{M}(k) = j$ , let  $\Upsilon_1(k)$  denote the  $k^{\text{th}}$  column of  $\Upsilon_1$ , and concatenate  $\Upsilon_1(k)$  to  $\Upsilon_2$ , i.e., let  $\Upsilon_2 \leftarrow [\Upsilon_2 \ \Upsilon_1(k)]$ . There are  $\sum_{k=1}^{K_C} 1_{\mathfrak{M}(k)=j}$  such columns.
3. Let  $\Upsilon'_1$  denote the columns of  $\Upsilon_1$  corresponding to the entries of  $\Phi_j P_j$  (the innovation components of sensor  $j$ ), and concatenate  $\Upsilon'_1$  to  $\Upsilon_2$ , i.e., let  $\Upsilon_2 \leftarrow [\Upsilon_2 \ \Upsilon'_1]$ . There are  $K_j$  such columns.
4. If  $j < J$ , let  $j \leftarrow j + 1$  and go to Step 2.

Because  $\Upsilon_1$  and  $\Upsilon_2$  share the same columns up to reordering, it follows that

$$\text{rank}(\Upsilon_2) = \text{rank}(\Upsilon_1). \quad (\text{A.5})$$

Based on its dependence on  $\Upsilon_0$ , and following from Lemma A.1, the square matrix  $\Upsilon_2$  meets properties P1 and P2 defined above in addition to a third property:

P3. All diagonal entries of  $\Upsilon_2$  are Gaussian random variables.

This follows because for each  $j$ ,  $K_j + \sum_{k=1}^{K_C} 1_{\mathfrak{M}(k)=j}$  rows of  $\Upsilon_1$  are assigned in its construction, while  $K_j + \sum_{k=1}^{K_C} 1_{\mathfrak{M}(k)=j}$  columns of  $\Upsilon_2$  are assigned in its construction. Thus, each diagonal element of  $\Upsilon_2$  will either be an entry of some  $\Phi_j \mathbf{P}_j$ , which remains Gaussian throughout our constructions, or it will be an entry of some  $k^{\text{th}}$  column of some  $\Phi_j \mathbf{P}_C$  for which  $\mathfrak{M}(k) = j$ . In the latter case, we know by Lemma A.1 and the construction of  $\Upsilon_0$  that this entry remains Gaussian throughout our constructions.

Having identified these three properties satisfied by  $\Upsilon_2$ , we will prove by induction that, with probability one over  $\Phi$ , such a matrix has full rank.

**Lemma A.2** *Let  $\Upsilon^{(d-1)}$  be a  $(d-1) \times (d-1)$  matrix having full rank. Construct a  $d \times d$  matrix  $\Upsilon^{(d)}$  as follows:*

$$\Upsilon^{(d)} := \begin{bmatrix} \Upsilon^{(d-1)} & \mathbf{v}_1 \\ \mathbf{v}_2^t & \omega \end{bmatrix}$$

where  $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^{d-1}$  are vectors with each entry being either zero or a Gaussian random variable,  $\omega$  is a Gaussian random variable, and all random variables are i.i.d. and independent of  $\Upsilon^{(d-1)}$ . Then with probability one,  $\Upsilon^{(d)}$  has full rank.

Applying Lemma A.2 inductively  $D$  times, the success probability remains one. It follows that with probability one over  $\Phi$ ,  $\text{rank}(\Upsilon_2) = D$ . Combining this last result with (A.3-A.5), we obtain  $\text{rank}(\Upsilon) = D$  with probability one over  $\Phi$ . It remains to prove Lemma A.2.

**Proof of Lemma A.2:** When  $d = 1$ ,  $\Upsilon^{(d)} = [\omega]$ , which has full rank if and only if  $\omega \neq 0$ , which occurs with probability one.

When  $d > 1$ , using expansion by minors, the determinant of  $\Upsilon^{(d)}$  satisfies

$$\det(\Upsilon^{(d)}) = \omega \cdot \det(\Upsilon^{(d-1)}) + C,$$

where  $C = C(\Upsilon^{(d-1)}, \mathbf{v}_1, \mathbf{v}_2)$  is independent of  $\omega$ . The matrix  $\Upsilon^{(d)}$  has full rank if and only if  $\det(\Upsilon^{(d)}) \neq 0$ , which is satisfied if and only if

$$\omega \neq \frac{-C}{\det(\Upsilon^{(d-1)})}.$$

By assumption,  $\det(\Upsilon^{(d-1)}) \neq 0$  and  $\omega$  is a Gaussian random variable that is independent of  $C$  and  $\det(\Upsilon^{(d-1)})$ . Thus,  $\omega \neq \frac{-C}{\det(\Upsilon^{(d-1)})}$  with probability one.  $\square$



## Appendix B

### Proof of Lemma A.1

To prove this lemma, we apply tools from graph theory. We begin by specifying a bipartite graph  $\tilde{G} = (V_V, V_M, \tilde{E})$  that depends on the structure of the location matrix  $\mathbf{P} \in \mathcal{P}_F(X)$ . The graph  $\tilde{G}$  has two sets of vertices  $V_V$  and  $V_M$  and a collection of edges  $\tilde{E}$  joining elements of  $V_V$  to  $V_M$ . The set  $V_V$  has vertices with indices  $k \in \{1, 2, \dots, D\}$ , which are known as *value vertices* and represent entries of the value vector  $\Theta$  (equivalently, columns of the matrix  $\mathbf{P}$ ). The set  $V_M$  has vertices with indices  $(j, m)$ , with  $j \in \Lambda$  and  $m \in \{1, 2, \dots, M_j\}$ , which are known as *measurement vertices* and represent entries  $y_j(m)$  of the measurement vectors (equivalently, rows of the matrix  $\Phi$ ). The edges  $\tilde{E}$  are specified as follows:

- For every  $k \in \{1, 2, \dots, K_C\} \subseteq V_V$  and  $j \in \Lambda$  such that column  $k$  of  $\mathbf{P}_C$  does not also appear as a column of  $\mathbf{P}_j$ , we have an edge connecting  $k$  to each vertex  $(j, m) \in V_M$  for  $1 \leq m \leq M_j$ .
- For every  $k \in \{K_C+1, K_C+2, \dots, D\} \subseteq V_V$ , we consider the sensor  $j$  associated with column  $k$  of  $\mathbf{P}$ , and we have an edge connecting  $k$  to each vertex  $(j, m) \in V_M$  for  $1 \leq m \leq M_j$ .

This graph  $\tilde{G}$  is a subgraph of the graph  $\hat{G}$  shown in Figure 3.1(c), from which we remove the edges going from common component vertices in  $V_V$  to measurement

vertices in  $V_M$  that have incoming edges from innovation component vertices in  $V_V$ .

We seek a matching within this graph, i.e., a subgraph  $(V_V, V_M, \bar{E})$  with  $\bar{E} \subseteq \tilde{E}$  that pairs each element of  $V_V$  with a unique element of  $V_M$ . Such a matching will immediately give us the desired mapping  $\mathfrak{M}$  as follows: for each  $k \in \{1, 2, \dots, K_C\} \subseteq V_V$ , we let  $(j, m) \in V_M$  denote the single node matched to  $k$  by an edge in  $\tilde{E}$ , and we set  $\mathfrak{M}(k) = j$ .

To prove the existence of such a matching within the graph, we invoke a version of Hall's marriage theorem for bipartite graphs [142]. Hall's theorem states that within a bipartite graph  $(V_1, V_2, E)$ , there exists a matching that assigns each element of  $V_1$  to a unique element of  $V_2$  if for any collection of elements  $\Pi \subseteq V_1$ , the set  $E(\Pi)$  of neighbors of  $\Pi$  in  $V_2$  has cardinality  $|E(\Pi)| \geq |\Pi|$ .

In the context of our lemma, Hall's condition requires that for any set of entries in the value vector,  $\Pi \subseteq V_V$ , the set  $\tilde{E}(\Pi)$  of neighbors of  $\Pi$  in  $V_M$  has size  $|\tilde{E}(\Pi)| \geq |\Pi|$ . We will prove that if (3.3) is satisfied, then Hall's condition is satisfied, and thus a matching must exist.

Let us consider an arbitrary set  $\Pi \subseteq V_V$ . We let  $\tilde{E}(\Pi)$  denote the set of neighbors of  $\Pi$  in  $V_M$  joined by edges in  $\tilde{E}$ , and we let  $S_\Pi = \{j \in \Lambda : (j, m) \in \tilde{E}(\Pi) \text{ for some } m\}$ . Thus,  $S_\Pi \subseteq \Lambda$  denotes the set of signal indices whose measurement nodes have edges that connect to  $\Pi$ . It follows that  $|\tilde{E}(\Pi)| = \sum_{j \in S_\Pi} M_j$ . Thus, in order to satisfy Hall's condition for  $\Pi$ , we require

$$\sum_{j \in S_\Pi} M_j \geq |\Pi|. \quad (\text{B.1})$$

We would now like to show that  $\sum_{j \in S_\Pi} K_j + K_C(S_\Pi, \mathbf{P}) \geq |\Pi|$ , and thus if (3.3) is satisfied for all  $\Gamma \subseteq \Lambda$ , then (B.1) is satisfied in particular for  $S_\Pi \subseteq \Lambda$ .

In general, the set  $\Pi$  may contain vertices for both common components and innovation components. We write  $\Pi = \Pi_I \cup \Pi_C$  to denote the disjoint union of these two sets.

By construction,  $\sum_{j \in S_\Pi} K_j = |\Pi_I|$  because  $I(S_\Pi, \mathbf{P})$  counts all innovations with neighbors in  $S_\Pi$ , and because  $S_\Pi$  contains all neighbors for nodes in  $\Pi_I$ . We will also argue that  $K_C(S_\Pi, \mathbf{P}) \geq |\Pi_C|$  as follows. By definition, for a set  $\Gamma \subseteq \Lambda$ ,  $K_C(\Gamma, \mathbf{P})$  counts the number of columns in  $\mathbf{P}_C$  that also appear in  $\mathbf{P}_j$  for all  $j \notin \Gamma$ . By construction, for each  $k \in \Pi_C$ , node  $k$  has no connection to nodes  $(j, m)$  for  $j \notin S_\Pi$ ; thus it must follow that the  $k^{\text{th}}$  column of  $\mathbf{P}_C$  is present in  $\mathbf{P}_j$  for all  $j \notin S_\Pi$ , due to the construction of the graph  $G$ . Consequently,  $K_C(S_\Pi, \mathbf{P}) \geq |\Pi_C|$ .

Thus,  $\sum_{j \in S_\Pi} K_j + K_C(S_\Pi, \mathbf{P}) \geq |\Pi_I| + |\Pi_C| = |\Pi|$ , and so (3.3) implies (B.1) for any  $\Pi$ , and so Hall's condition is satisfied, and a matching exists. Because in such matching a set of vertices in  $V_M$  matches to a set in  $V_V$  of lower or equal cardinality, we have in particular that (A.2) holds for each  $\Gamma \subseteq \Lambda$ .  $\square$

## Appendix C

### Proof of Theorem 3.2

Given the measurements  $\mathbf{Y}$  and measurement matrix  $\Phi$ , we show that it is possible to recover some  $\mathbf{P} \in \mathcal{P}_F(\mathbf{X})$  and a corresponding vector  $\Theta$  such that  $\mathbf{X} = \mathbf{P}\Theta$  using the following algorithm:

- Take the last measurement of each sensor for verification, and sum these  $J$  measurements to obtain a single *global* test measurement  $\bar{y}$ . Similarly, add the corresponding rows of  $\Phi$  into a single row  $\bar{\phi}$ .
- Group all the remaining  $\sum_{j \in \Lambda} M_j - J$  measurements into a vector  $\bar{\mathbf{y}}$  and a matrix  $\bar{\Phi}$ .
- For each matrix  $\mathbf{P} \in \mathcal{P}$ 
  - choose a single solution  $\Theta_{\mathbf{P}}$  to  $\bar{\mathbf{y}} = \bar{\Phi}\mathbf{P}\Theta_{\mathbf{P}}$  independently of  $\bar{\phi}$  – if no solution exists, skip the next two steps;
  - define  $\mathbf{X}_{\mathbf{P}} = \mathbf{P}\Theta_{\mathbf{P}}$ ;
  - cross-validate: check if  $\bar{y} = \bar{\phi}\mathbf{X}_{\mathbf{P}}$ ; if so, return the estimate  $(\mathbf{P}, \Theta_{\mathbf{P}})$ ; if not, continue with the next matrix.

We begin by showing that, with probability one over  $\Phi$ , the algorithm only terminates when it gets a correct solution – in other words, that for each  $\mathbf{P} \in \mathcal{P}$  the cross-

validation measurement  $\bar{\mathbf{y}}$  can determine whether  $\mathbf{X}_{\mathbf{P}} = \mathbf{X}$ . We note that all entries of the vector  $\bar{\phi}$  are i.i.d. Gaussian, and independent from  $\bar{\Phi}$ . Assume for the sake of contradiction that there exists a matrix  $\mathbf{P} \in \mathcal{P}$  such that  $\bar{\mathbf{y}} = \bar{\phi}\mathbf{X}_{\mathbf{P}}$ , but  $\mathbf{X}_{\mathbf{P}} = \mathbf{P}\Theta_{\mathbf{P}} \neq \mathbf{X}$ ; this implies  $\bar{\phi}(\mathbf{X} - \mathbf{X}_{\mathbf{P}}) = 0$ , which occurs with probability zero over  $\Phi$ . Thus, if  $\mathbf{X}_{\mathbf{P}} \neq \mathbf{X}$ , then  $\bar{\phi}\mathbf{X}_{\mathbf{P}} \neq \bar{\mathbf{y}}$  with probability one over  $\Phi$ . Since we only need to search over a finite number of matrices  $\mathbf{P} \in \mathcal{P}$ , cross validation will determine whether each matrix  $\mathbf{P} \in \mathcal{P}$  gives the correct solution with probability one.

We now show that there is a matrix in  $\mathcal{P}$  for which the algorithm will terminate with the correct solution. We know that the matrix  $\mathbf{P}^* \in \mathcal{P}_F(\mathbf{X}) \subseteq \mathcal{P}$  will be part of our search, and that the unique solution  $\Theta_{\mathbf{P}^*}$  to  $\bar{\mathbf{Y}} = \bar{\Phi}\mathbf{P}^*\Theta_{\mathbf{P}^*}$  yields  $\mathbf{X} = \mathbf{P}^*\Theta_{\mathbf{P}^*}$  when (3.4) holds for  $\mathbf{P}^*$ , as shown in Theorem 3.1. Thus, the algorithm will find at least one matrix  $\mathbf{P}$  and vector  $\Theta_{\mathbf{P}}$  such that  $\mathbf{X} = \mathbf{P}\Theta_{\mathbf{P}}$ ; when such matrix is found the cross-validation step will return this solution and end the algorithm.  $\square$

*Remark.* Consider the algorithm used in the proof: if the matrices in  $\mathcal{P}$  are sorted by number of columns, the algorithm is akin to  $\ell_0$  minimization on  $\Theta$  with an additional cross-validation step.

## Appendix D

### Proof of Theorem 3.3

We let  $D$  denote the number of columns in  $\mathbf{P}$ . Because  $\mathbf{P} \in \mathcal{P}_F(\mathbf{X})$ , there exists  $\Theta \in \mathbb{R}^D$  such that  $\mathbf{X} = \mathbf{P}\Theta$ . Because  $\mathbf{Y} = \Phi\mathbf{X}$ , then  $\Theta$  is a solution to  $\mathbf{Y} = \Phi\mathbf{P}\Theta$ . We will argue for  $\Upsilon := \Phi\mathbf{P}$  that  $\text{rank}(\Upsilon) < D$ , and thus there exists  $\hat{\Theta} \neq \Theta$  such that  $\mathbf{Y} = \Upsilon\Theta = \Upsilon\hat{\Theta}$ . Moreover, since  $\mathbf{P}$  has full rank, it follows that  $\hat{\mathbf{X}} := \mathbf{P}\hat{\Theta} \neq \mathbf{P}\Theta = \mathbf{X}$ .

We let  $\Upsilon_0$  be the “partially zeroed” matrix obtained from  $\Upsilon$  using the identical procedure detailed in Appendix A. Again, because  $\Upsilon_0$  was constructed only by subtracting columns of  $\Upsilon$  from one another, it follows that  $\text{rank}(\Upsilon_0) = \text{rank}(\Upsilon)$ .

Suppose  $\Gamma \subseteq \Lambda$  is a set for which (3.5) holds. We let  $\Upsilon_1$  be the submatrix of  $\Upsilon_0$  obtained by selecting the following columns:

- For any  $k \in \{1, 2, \dots, K_C\}$  such that column  $k$  of  $\mathbf{P}_C$  also appears as a column in all  $\mathbf{P}_j$  for  $j \notin \Gamma$ , we include column  $k$  of  $\Upsilon_0$  as a column in  $\Upsilon_1$ . There are  $K_C(\Gamma, \mathbf{P})$  such columns  $k$ .
- For any  $k \in \{K_C + 1, K_C + 2, \dots, D\}$  such that column  $k$  of  $\mathbf{P}$  corresponds to an innovation for some sensor  $j \in \Gamma$ , we include column  $k$  of  $\Upsilon_0$  as a column in  $\Upsilon_1$ . There are  $\sum_{j \in \Gamma} K_j$  such columns  $k$ .

This submatrix has  $\sum_{j \in \Gamma} K_j + K_C(\Gamma, \mathbf{P})$  columns. Because  $\Upsilon_0$  has the same size as  $\Upsilon$ , and in particular has only  $D$  columns, then in order to have that  $\text{rank}(\Upsilon_0) = D$ ,

it is necessary that all  $\sum_{j \in \Gamma} K_j + K_C(\Gamma, \mathbf{P})$  columns of  $\Upsilon_1$  be linearly independent.

Based on the method described for constructing  $\Upsilon_0$ , it follows that  $\Upsilon_1$  is zero for all measurement rows not corresponding to the set  $\Gamma$ . Therefore, let us consider the submatrix  $\Upsilon_2$  of  $\Upsilon_1$  obtained by selecting only the measurement rows corresponding to the set  $\Gamma$ . Because of the zeros in  $\Upsilon_1$ , it follows that  $\text{rank}(\Upsilon_1) = \text{rank}(\Upsilon_2)$ . However, since  $\Upsilon_2$  has only  $\sum_{j \in \Gamma} M_j$  rows, we invoke (3.5) and have that  $\text{rank}(\Upsilon_1) = \text{rank}(\Upsilon_2) \leq \sum_{j \in \Gamma} M_j < \sum_{j \in \Gamma} K_j + K_C(\Gamma, \mathbf{P})$ . Thus, all  $\sum_{j \in \Gamma} K_j + K_C(\Gamma, \mathbf{P})$  columns of  $\Upsilon_1$  cannot be linearly independent, and so  $\Upsilon$  does not have full rank.  $\square$

## Appendix E

### Proof of Theorem 6.2

To prove this theorem, we will study the distribution of the maximum singular value of a submatrix  $\Phi_T$  of a matrix with i.i.d. Gaussian entries  $\Phi$  corresponding to the columns indexed by  $T$ . From this we obtain the probability that RAmP does not hold for a fixed support  $T$ . We will then evaluate the same probability for all supports  $T$  of elements of  $\mathcal{R}_{j,K}$ , where the desired bound on the amplification is dependent on the value of  $j$ . This gives us the probability that the RAmP does not hold for a given residual subspace set  $\mathcal{R}_{j,K}$ . We fix the probability of failure on each of these sets; we then obtain probability that the matrix  $\Phi$  does not have the RAmP using a union bound. We end by obtaining conditions on the number of rows  $M$  of  $\Phi$  to obtain a desired probability of failure.

We begin from the following concentration of measure for the largest singular value of a  $M \times K$  submatrix  $\Phi_T$ ,  $|T| = K$ , of an  $M \times N$  matrix  $\Phi$  with i.i.d. subgaussian entries that are properly normalized [50, 54, 143]:

$$P \left( \sigma_{\max}(\Phi_T) > 1 + \sqrt{\frac{K}{M}} + \tau + \beta \right) \leq e^{-M\tau^2/2}.$$

For large enough  $M$ ,  $\beta \ll 1$ ; thus we ignore this small constant in the sequel. By letting  $\tau = j^r \sqrt{1 + \epsilon_K} - 1 - \sqrt{\frac{K}{M}}$  (with the appropriate value of  $j$  for  $T$ ), we obtain

$$P \left( \sigma_{\max}(\Phi_T) > j^r \sqrt{1 + \epsilon_K} \right) \leq e^{-\frac{M}{2} \left( j^r \sqrt{1 + \epsilon_K} - 1 - \sqrt{\frac{K}{M}} \right)^2}.$$



We use a union bound over all possible  $R_j$  supports for  $\mathbf{u} \in \mathcal{R}_{j,K}$  to obtain the probability that  $\Phi$  does not amplify the norm of  $\mathbf{u}$  by more than  $j^r \sqrt{1 + \epsilon_K}$ :

$$P(\|\Phi \mathbf{u}\|_2 > (j^r \sqrt{1 + \epsilon_K}) \|\mathbf{u}\|_2 \mid \mathbf{u} \in \mathcal{R}_{j,K}) \leq R_j e^{-\frac{1}{2}(\sqrt{M}(j^r \sqrt{1 + \epsilon_K} - 1) - \sqrt{K})^2}.$$

Bound the right hand side by a constant  $\mu$ ; this requires

$$R_j \leq e^{\frac{1}{2}(\sqrt{M}(j^r \sqrt{1 + \epsilon_K} - 1) - \sqrt{K})^2} \mu \quad (\text{E.1})$$

for each  $j$ . We use another union bound among the residual subspaces  $\mathcal{R}_{j,K}$  to measure the probability that the RAmP does not hold:

$$P(\|\Phi \mathbf{u}\|_2 > (j^r \sqrt{1 + \epsilon_K}) \|\mathbf{u}\|_2 \mid \mathbf{u} \in \mathcal{R}_{j,K}, \forall j, 1 \leq j \leq \lceil N/K \rceil) \leq \left\lceil \frac{N}{K} \right\rceil \mu.$$

To bound this probability by  $e^{-t}$ , we need  $\mu = \frac{K}{N} e^{-t}$ ; plugging this into (E.1), we obtain

$$R_j \leq e^{\frac{1}{2}(\sqrt{M}(j^r \sqrt{1 + \epsilon_K} - 1) - \sqrt{K})^2} \frac{K}{N} e^{-t}$$

for each  $j$ . Simplifying, we obtain that for  $\Phi$  to possess the RAmP with probability  $1 - e^{-t}$ , the following must hold for all  $j$ :

$$M \geq \frac{1}{(j^r \sqrt{1 + \epsilon_K} - 1)^2} \left( \sqrt{2 \left( \ln \frac{R_j N}{K} + t \right)} + \sqrt{K} \right)^2. \quad (\text{E.2})$$

Since  $(\sqrt{a} + \sqrt{b})^2 \leq 2a + 2b$  for  $a, b > 0$ , then the hypothesis (6.3) implies (E.2), proving the theorem.  $\square$

## Appendix F

### Proof of Theorem 6.3

In this proof, we denote  $\mathbb{M}(\theta, K) = \theta_K$  for brevity. To bound  $\|\Upsilon(\theta - \theta_K)\|_2$ , we write  $\theta$  as

$$\theta = \theta_K + \sum_{j=2}^{\lceil N/K \rceil} \theta_{T_j},$$

where

$$\theta_{T_j} = \theta_{jK} - \theta_{(j-1)K}, j = 2, \dots, \lceil N/K \rceil$$

is the difference between the best  $jK$  structured sparse approximation and the best  $(j-1)K$  structured sparse approximation. Additionally, each piece  $\theta_{T_j} \in \mathcal{R}_{j,K}$ . Therefore, since  $\Upsilon$  satisfies the  $(\epsilon_K, s-1)$ -RAmP, we obtain

$$\|\Upsilon(\theta - \theta_K)\|_2 = \left\| \Upsilon \left( \sum_{j=2}^{\lceil N/K \rceil} \theta_{T_j} \right) \right\|_2 \leq \sum_{j=2}^{\lceil N/K \rceil} \|\Upsilon \theta_{T_j}\|_2 \leq \sum_{j=2}^{\lceil N/K \rceil} \sqrt{1 + \epsilon_K} j^{s-1} \|\theta_{T_j}\|_2. \quad (\text{F.1})$$

Since  $\theta \in \mathfrak{M}_s$ , the norm of each piece can be bounded as

$$\|\theta_{T_j}\|_2 = \|\theta_{jK} - \theta_{(j-1)K}\|_2 \leq \|\theta - \theta_{(j-1)K}\|_2 + \|\theta - \theta_{jK}\|_2 \leq |\theta|_{\mathfrak{M}_s} K^{-s} ((j-1)^{-s} + j^{-s}).$$

Applying this bound in (F.1), we obtain

$$\begin{aligned}
 \|\Upsilon(\theta - \theta_K)\|_2 &\leq \sqrt{1 + \epsilon_K} \sum_{j=2}^{\lceil N/K \rceil} j^{s-1} \|\theta_{T_j}\|_2, \\
 &\leq \sqrt{1 + \epsilon_K} |\theta|_{\mathfrak{M}_s} K^{-s} \sum_{j=2}^{\lceil N/K \rceil} j^{s-1} ((j-1)^{-s} + j^{-s}), \\
 &\leq \sqrt{1 + \epsilon_K} |\theta|_{\mathfrak{M}_s} K^{-s} \sum_{j=2}^{\lceil N/K \rceil} j^{-1}.
 \end{aligned}$$

It is easy to show, using Euler-Maclaurin summations, that  $\sum_{j=2}^{\lceil N/K \rceil} j^{-1} \leq \ln \lceil N/K \rceil$ ;

we then obtain

$$\|\Upsilon(\theta - \theta_K)\|_2 \leq \sqrt{1 + \epsilon_K} K^{-s} \ln \left\lceil \frac{N}{K} \right\rceil |\theta|_{\mathfrak{M}_s},$$

which proves the theorem.  $\square$

## Appendix G

### Model-based Iterative Hard Thresholding

Our proposed model-based iterative hard thresholding (IHT) is given in Algorithm 9. For this algorithm, Theorems 6.4, 6.5, and 6.6 can be proven with only a few modifications:  $\Upsilon$  must have the  $\mathcal{M}_K^3$ -RIP with  $\delta_{\mathcal{M}_K^3} \leq 0.1$ , and the constant factor in the bound changes from 15 to 4 in Theorem 6.4, from 35 to 10 in Theorem 6.5, and from 20 to 5 in Theorem 6.6.

To illustrate the performance of the algorithm, we repeat the *HeaviSine* experiment from Figure 6.1. Recall that  $N = 1024$ , and  $M = 80$  for this example. The advantages of using our tree-model-based approximation step (instead of mere hard thresholding) are evident from Figure G.1. In practice, we have observed that our model-based algorithm converges in fewer steps than IHT and yields much more accurate results in terms of recovery error.

---

**Algorithm 9** Model-based Iterative Hard Thresholding
 

---

Inputs: CS matrix  $\Upsilon$ , measurements  $\mathbf{y}$ , structured sparse approx. algorithm  $\mathbb{M}_K$

Output:  $K$ -sparse approximations  $\hat{\theta}$  to true signal representation  $\theta$

Initialize:  $\hat{\theta}_0 = 0$ ,  $\mathbf{r} = \mathbf{y}$ ;  $i = 0$

**while** halting criterion false **do**

1.  $i \leftarrow i + 1$

2.  $\mathbf{b} \leftarrow \hat{\theta}_{i-1} + \Upsilon^T d$       {form signal estimate}

3.  $\hat{\theta}_i \leftarrow \mathbb{M}(\mathbf{b}, K)$       {prune residual estimate according to structure}

4.  $\mathbf{r} \leftarrow \mathbf{y} - \Upsilon \hat{\theta}_i$       {update measurement residual}

**end while**

return  $\hat{\theta} \leftarrow \hat{\theta}_i$

---

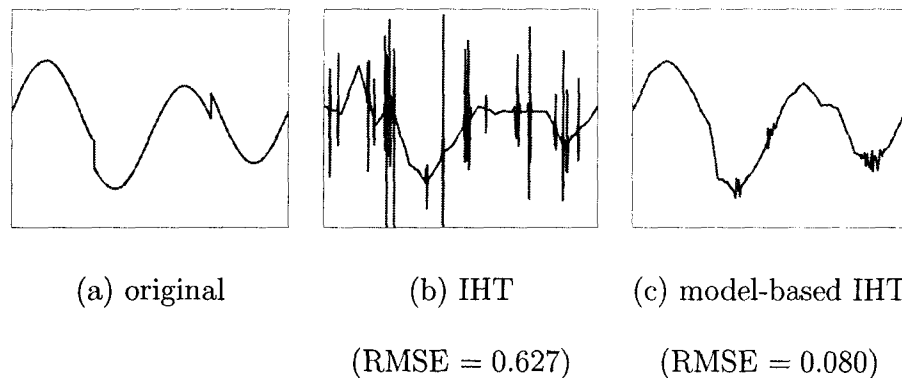


Figure G.1 : *Example performance of model-based IHT. (a) Piecewise smooth HeaviSine test signal, length  $N = 1024$ . Signal recovered from  $M = 80$  measurements using both (b) standard and (c) model-based IHT recovery. Root mean-squared error (RMSE) values are normalized with respect to the  $\ell_2$  norm of the signal.*

## Appendix H

### Proof of Theorem 6.4

The proof of this theorem is identical to that of the CoSaMP algorithm in [66, Section 4.6], and requires a set of six lemmas. The sequence of Lemmas H.1–H.6 below are modifications of the lemmas in [66] that are restricted to the structured sparsity model. Lemma H.4 does not need any changes from [66], so we state it without proof. The proof of Lemmas H.3–H.6 use the properties in Lemmas H.1 and H.2, which are simple to prove.

**Lemma H.1** *Suppose  $\Upsilon$  has  $\mathcal{M}$ -RIP with constant  $\delta_{\mathcal{M}}$ . Let  $\Omega$  be a support corresponding to a subspace in  $\mathcal{M}$ . Then we have the following handy bounds.*

$$\begin{aligned} \|\Upsilon_{\Omega}^T \mathbf{u}\|_2 &\leq \sqrt{1 + \delta_{\mathcal{M}}} \|\mathbf{u}\|_2, \\ \|\Upsilon_{\Omega}^{\dagger} \mathbf{u}\|_2 &\leq \frac{1}{\sqrt{1 - \delta_{\mathcal{M}}}} \|\mathbf{u}\|_2, \\ \|\Upsilon_{\Omega}^T \Upsilon_{\Omega} \mathbf{u}\|_2 &\leq (1 + \delta_{\mathcal{M}}) \|\mathbf{u}\|_2, \\ \|\Upsilon_{\Omega}^T \Upsilon_{\Omega} \mathbf{u}\|_2 &\geq (1 - \delta_{\mathcal{M}}) \|\mathbf{u}\|_2, \\ \|(\Upsilon_{\Omega}^T \Upsilon_{\Omega})^{-1} \mathbf{u}\|_2 &\leq \frac{1}{1 - \delta_{\mathcal{M}}} \|\mathbf{u}\|_2, \\ \|(\Upsilon_{\Omega}^T \Upsilon_{\Omega})^{-1} \mathbf{u}\|_2 &\geq \frac{1}{1 + \delta_{\mathcal{M}}} \|\mathbf{u}\|_2. \end{aligned}$$

**Lemma H.2** *Suppose  $\Upsilon$  has  $\mathcal{M}_K^2$ -RIP with constant  $\delta_{\mathcal{M}_K^2}$ . Let  $\Omega$  be a support corre-*

sponding to a subspace in  $\mathcal{M}_K$ , and let  $\theta \in \mathcal{M}_K$ . Then  $\|\Upsilon_\Omega^T \Upsilon \theta|_{\Omega^c}\|_2 \leq \delta_{\mathcal{M}_K^2} \|\theta|_{\Omega^c}\|_2$ .

We begin the proof of Theorem 6.4 by fixing an iteration  $i \geq 1$  of model-based CoSaMP. We write  $\hat{\theta} = \hat{\theta}_{i-1}$  for the signal estimate at the beginning of the  $i^{\text{th}}$  iteration. Define the signal residual  $\mathbf{s} = \theta - \hat{\theta}$ , which implies that  $\mathbf{s} \in \mathcal{M}_K^2$ . We note that we can write  $\mathbf{r} = \mathbf{y} - \Upsilon \hat{\theta} = \Upsilon(\theta - \hat{\theta}) + \mathbf{n} = \Upsilon \mathbf{s} + \mathbf{n}$ .

**Lemma H.3** (Identification) *The set  $\Omega = \text{supp}(\mathbb{M}_2(\mathbf{e}, K))$ , where  $\mathbf{e} = \Upsilon^T \mathbf{r}$ , identifies a subspace in  $\mathcal{M}_K^2$ , and obeys*

$$\|\mathbf{s}|_{\Omega^c}\|_2 \leq 0.2223\|\mathbf{s}\|_2 + 2.34\|\mathbf{n}\|_2.$$

*Proof of Lemma H.3:* Define the set  $\Pi = \text{supp}(\mathbf{s})$ . Let  $\mathbf{e}_\Omega = \mathbb{M}_2(\mathbf{e}, K)$  be the model-based approximation to  $\mathbf{e}$  with support  $\Omega$ , and similarly let  $\mathbf{e}_\Pi$  be the approximation to  $\mathbf{e}$  with support  $\Pi$ . Each approximation is equal to  $\mathbf{e}$  for the coefficients in the support, and zero elsewhere. Since  $\Omega$  is the support of the best approximation in

$\mathcal{M}_K^2$ , we must have:

$$\begin{aligned}
\|\mathbf{e} - \mathbf{e}_\Omega\|_2^2 &\leq \|\mathbf{e} - \mathbf{e}_\Pi\|_2^2, \\
\sum_{n=1}^N (\mathbf{e}[n] - \mathbf{e}_\Omega[n])^2 &\leq \sum_{n=1}^N (\mathbf{e}[n] - \mathbf{e}_\Pi[n])^2, \\
\sum_{n \notin \Omega} \mathbf{e}[n]^2 &\leq \sum_{n \notin \Pi} \mathbf{e}[n]^2, \\
\sum_{n=1}^N \mathbf{e}[n]^2 - \sum_{n \notin \Omega} \mathbf{e}[n]^2 &\geq \sum_{n=1}^N \mathbf{e}[n]^2 - \sum_{n \notin \Pi} \mathbf{e}[n]^2, \\
\sum_{n \in \Omega} \mathbf{e}[n]^2 &\geq \sum_{n \in \Pi} \mathbf{e}[n]^2, \\
\sum_{n \in \Omega \setminus \Pi} \mathbf{e}[n]^2 &\geq \sum_{n \in \Pi \setminus \Omega} \mathbf{e}[n]^2, \\
\|\mathbf{e}|_{\Omega \setminus \Pi}\|_2^2 &\geq \|\mathbf{e}|_{\Pi \setminus \Omega}\|_2^2,
\end{aligned}$$

where  $\Omega \setminus \Pi$  denotes the set difference of  $\Omega$  and  $\Pi$ . These signals are in  $\mathcal{M}_K^4$  (since they arise as the difference of two elements from  $\mathcal{M}_K^2$ ); therefore, we can apply the  $\mathcal{M}_K^4$ -RIP constants and Lemmas H.1 and H.2 to provide the following bounds on both sides (see [66] for details):

$$\|\mathbf{e}|_{\Omega \setminus \Pi}\|_2 \leq \delta_{\mathcal{M}_K^4} \|\mathbf{s}\|_2 + \sqrt{1 + \delta_{\mathcal{M}_K^2}} \|\mathbf{e}\|_2, \quad (\text{H.1})$$

$$\|\mathbf{e}|_{\Pi \setminus \Omega}\|_2 \geq (1 - \delta_{\mathcal{M}_K^2}) \|\mathbf{s}|_{\Omega^c}\|_2 - \delta_{\mathcal{M}_K^2} \|\mathbf{s}\|_2 - \sqrt{1 + \delta_{\mathcal{M}_K^2}} \|\mathbf{e}\|_2. \quad (\text{H.2})$$

Combining (H.1) and (H.2), we obtain

$$\|\mathbf{s}|_{\Omega^c}\|_2 \leq \frac{(\delta_{\mathcal{M}_K^2} + \delta_{\mathcal{M}_K^4}) \|\mathbf{s}\|_2 + 2\sqrt{1 + \delta_{\mathcal{M}_K^2}} \|\mathbf{e}\|_2}{1 - \delta_{\mathcal{M}_K^2}}.$$

The argument is completed by noting that  $\delta_{\mathcal{M}_K^2} \leq \delta_{\mathcal{M}_K^4} \leq 0.1$ .  $\square$

**Lemma H.4** (Support Merger) *Let  $\Omega$  be a set of at most  $2K$  indices. Then the set  $\Lambda = \Omega \cup \text{supp}(\hat{\mathbf{x}})$  contains at most  $3K$  indices, and  $\|\mathbf{x}|_{\Lambda^c}\|_2 \leq \|\mathbf{s}|_{\Omega^c}\|_2$ .*



**Lemma H.5** (Estimation) *Let  $\Lambda$  be a support corresponding to a subspace in  $\mathcal{M}_K^3$ , and define the least squares signal estimate  $\mathbf{b}$  by  $\mathbf{b}|_T = \Upsilon_T^\dagger y$ ,  $\mathbf{b}|_{T^c} = 0$ . Then*

$$\|\mathbf{x} - \mathbf{b}\|_2 \leq 1.112\|\mathbf{x}|_{\Lambda^c}\|_2 + 1.06\|\mathbf{n}\|_2.$$

*Proof of Lemma H.5:* It can be shown [66] that

$$\|\mathbf{x} - \mathbf{b}\|_2 \leq \|\mathbf{x}|_{\Lambda^c}\|_2 + \|(\Upsilon_\Lambda^T \Upsilon_\Lambda)^{-1} \Upsilon_\Lambda^T \Upsilon_{\Pi^c} \mathbf{x}\|_2 + \|\Upsilon_\Pi^\dagger \mathbf{n}\|_2.$$

Since  $\Lambda$  is a support corresponding to a subspace in  $\mathcal{M}_K^3$  and  $\mathbf{x} \in \mathcal{M}_K$ , we use Lemmas H.1 and H.2 to obtain

$$\begin{aligned} \|\mathbf{x} - \mathbf{b}\|_2 &\leq \|\mathbf{x}|_{\Lambda^c}\|_2 + \frac{1}{1 - \delta_{\mathcal{M}_K^3}} \|\Upsilon_\Lambda^T \Upsilon_{\Pi^c} \mathbf{x}\|_2 + \frac{1}{\sqrt{1 - \delta_{\mathcal{M}_K^3}}} \|\mathbf{n}\|_2, \\ &\leq \left(1 + \frac{\delta_{\mathcal{M}_K^4}}{1 - \delta_{\mathcal{M}_K^3}}\right) \|\mathbf{x}|_{\Pi^c}\|_2 + \frac{1}{\sqrt{1 - \delta_{\mathcal{M}_K^3}}} \|\mathbf{n}\|_2. \end{aligned}$$

Finally, note that  $\delta_{\mathcal{M}_K^3} \leq \delta_{\mathcal{M}_K^4} \leq 0.1$ . □

**Lemma H.6** (Pruning) *The pruned approximation  $\hat{\mathbf{x}}_i = \mathbb{M}(\mathbf{b}, K)$  is such that*

$$\|\mathbf{x} - \hat{\mathbf{x}}_i\|_2 \leq 2\|\mathbf{x} - \mathbf{b}\|_2.$$

*Proof of Lemma H.6:* Since  $\hat{\mathbf{x}}_i$  is the best approximation in  $\mathcal{M}_K$  to  $\mathbf{b}$ , and  $\mathbf{x} \in \mathcal{M}_K$ , we obtain

$$\|\mathbf{x} - \hat{\mathbf{x}}_i\|_2 \leq \|\mathbf{x} - \mathbf{b}\|_2 + \|\mathbf{b} - \hat{\mathbf{x}}_i\|_2 \leq 2\|\mathbf{x} - \mathbf{b}\|_2.$$

□

We use these lemmas in reverse sequence for the inequalities below:

$$\begin{aligned}
\|\mathbf{x} - \widehat{\mathbf{x}}_i\|_2 &\leq 2\|\mathbf{x} - \mathbf{b}\|_2, \\
&\leq 2(1.112\|\mathbf{x}|_{\Lambda^C}\|_2 + 1.06\|\mathbf{n}\|_2), \\
&\leq 2.224\|\mathbf{s}|_{\Omega^C}\|_2 + 2.12\|\mathbf{n}\|_2, \\
&\leq 2.224(0.2223\|\mathbf{s}\|_2 + 2.34\|\mathbf{n}\|_2) + 2.12\|\mathbf{n}\|_2, \\
&\leq 0.5\|\mathbf{s}\|_2 + 7.5\|\mathbf{n}\|_2, \\
&\leq 0.5\|\mathbf{x} - \widehat{\mathbf{x}}_{i-1}\|_2 + 7.5\|\mathbf{n}\|_2.
\end{aligned}$$

From the recursion on  $\widehat{\mathbf{x}}_i$ , we obtain  $\|\mathbf{x} - \widehat{\mathbf{x}}_i\|_2 \leq 2^{-i}\|\mathbf{x}\|_2 + 15\|\mathbf{n}\|_2$ . This completes the proof of Theorem 6.4.  $\square$

## Appendix I

### Proof of Proposition 6.1

When  $K < \log_2 N$ , the number of subtrees of size  $K$  of a binary tree of size  $N$  is the Catalan number [144]

$$T_{K,N} = \frac{1}{K+1} \binom{2K}{K} \leq \frac{(2e)^K}{K+1},$$

using Stirling's approximation. When  $K > \log_2 N$ , we partition this count of subtrees into the numbers of subtrees  $t_{K,h}$  of size  $K$  and height  $h$ , to obtain

$$T_{K,N} = \sum_{h=\lfloor \log_2 K \rfloor + 1}^{\log_2 N} t_{K,h}$$

We obtain the following asymptotic identity from [144, page 51]:

$$\begin{aligned} t_{K,h} &= \frac{4^{K+1.5}}{h^4} \sum_{m \geq 1} \left[ \frac{2K}{h^2} (2\pi m)^4 - 3(2\pi m)^2 \right] e^{-\frac{K(2\pi m)^2}{h^2}} + 4^K \mathcal{O}\left(e^{-\ln^2 h}\right) \\ &\quad + 4^K \mathcal{O}\left(\frac{\ln^8 h}{h^5}\right) + 4^K \mathcal{O}\left(\frac{\ln^8 h}{h^4}\right), \\ &\leq \frac{4^{K+2}}{h^4} \sum_{m \geq 1} \left[ \frac{2K}{h^2} (2\pi m)^4 - 3(2\pi m)^2 \right] e^{-\frac{K(2\pi m)^2}{h^2}}. \end{aligned} \quad (\text{I.1})$$

We now simplify the formula slightly: we seek a bound for the sum term (which we denote by  $\beta_h$  for brevity):

$$\beta_h = \sum_{m \geq 1} \left[ \frac{2K}{h^2} (2\pi m)^4 - 3(2\pi m)^2 \right] e^{-\frac{K(2\pi m)^2}{h^2}} \leq \sum_{m \geq 1} \frac{2K}{h^2} (2\pi m)^4 e^{-\frac{K(2\pi m)^2}{h^2}}. \quad (\text{I.2})$$

Let  $m_{\max} = \frac{h}{\pi\sqrt{2K}}$ , the value of  $m$  for which the term inside the sum (I.2) is maximum;

this is not necessarily an integer. Then,

$$\begin{aligned}
 \beta_h &\leq \sum_{m=1}^{\lfloor m_{\max} \rfloor - 1} \frac{2K}{h^2} (2\pi m)^4 e^{-\frac{K(2\pi m)^2}{h^2}} + \sum_{m=\lfloor m_{\max} \rfloor}^{\lceil m_{\max} \rceil} \frac{2K}{h^2} (2\pi m)^4 e^{-\frac{K(2\pi m)^2}{h^2}} \\
 &\quad + \sum_{m \geq \lceil m_{\max} \rceil + 1} \frac{2K}{h^2} (2\pi m)^4 e^{-\frac{K(2\pi m)^2}{h^2}}, \\
 &\leq \int_1^{\lfloor m_{\max} \rfloor} \frac{2K}{h^2} (2\pi x)^4 e^{-\frac{K(2\pi x)^2}{h^2}} dx + \sum_{m=\lfloor m_{\max} \rfloor}^{\lceil m_{\max} \rceil} \frac{2K}{h^2} (2\pi m)^4 e^{-\frac{K(2\pi m)^2}{h^2}} \\
 &\quad + \int_{\lceil m_{\max} \rceil}^{\infty} \frac{2K}{h^2} (2\pi x)^4 e^{-\frac{K(2\pi x)^2}{h^2}} dx,
 \end{aligned}$$

where the second inequality comes from the fact that the series in the sum is strictly increasing for  $m \leq \lfloor m_{\max} \rfloor$  and strictly decreasing for  $m > \lceil m_{\max} \rceil$ . One of the terms in the sum can be added to one of the integrals. If we have that

$$(2\pi \lfloor m_{\max} \rfloor)^4 e^{-\frac{K(2\pi \lfloor m_{\max} \rfloor)^2}{h^2}} < (2\pi \lceil m_{\max} \rceil)^4 e^{-\frac{K(2\pi \lceil m_{\max} \rceil)^2}{h^2}}, \quad (\text{I.3})$$

then we can obtain

$$\begin{aligned}
 \beta_h &\leq \int_1^{\lceil m_{\max} \rceil} \frac{2K}{h^2} (2\pi x)^4 e^{-\frac{K(2\pi x)^2}{h^2}} dx + \frac{2K}{h^2} (2\pi \lceil m_{\max} \rceil)^4 e^{-\frac{K(2\pi \lceil m_{\max} \rceil)^2}{h^2}} \\
 &\quad + \int_{\lceil m_{\max} \rceil}^{\infty} \frac{2K}{h^2} (2\pi x)^4 e^{-\frac{K(2\pi x)^2}{h^2}} dx.
 \end{aligned}$$

When the opposite of (I.3) is true, we have that

$$\begin{aligned}
 \beta_h &\leq \int_1^{\lfloor m_{\max} \rfloor} \frac{2K}{h^2} (2\pi x)^4 e^{-\frac{K(2\pi x)^2}{h^2}} dx + \frac{2K}{h^2} (2\pi \lfloor m_{\max} \rfloor)^4 e^{-\frac{K(2\pi \lfloor m_{\max} \rfloor)^2}{h^2}} \\
 &\quad + \int_{\lfloor m_{\max} \rfloor}^{\infty} \frac{2K}{h^2} (2\pi x)^4 e^{-\frac{K(2\pi x)^2}{h^2}} dx.
 \end{aligned}$$

Since the term in the sum reaches its maximum for  $m_{\max}$ , we will have in all three cases that

$$\beta_h \leq \int_1^{\infty} \frac{2K}{h^2} (2\pi x)^4 e^{-\frac{K(2\pi x)^2}{h^2}} dx + \frac{8h^2}{Ke^2}.$$

We perform a change of variables  $u = 2\pi x$  and define  $\sigma = h/\sqrt{2K}$  to obtain

$$\beta_h \leq \frac{1}{2\pi} \int_0^\infty \frac{1}{\sigma^2} u^4 e^{-u^2/2\sigma^2} dx + \frac{8h^2}{Ke^2} \leq \frac{1}{2\sigma\sqrt{2\pi}} \int_{-\infty}^\infty \frac{1}{\sqrt{2\pi}\sigma} u^4 e^{-u^2/2\sigma^2} dx + \frac{8h^2}{Ke^2}.$$

Using the formula for the fourth central moment of a Gaussian distribution:

$$\int_{-\infty}^\infty \frac{1}{\sqrt{2\pi}\sigma} u^4 e^{-u^2/2\sigma^2} dx = 3\sigma^4,$$

we obtain

$$\beta_h \leq \frac{3\sigma^3}{2\sqrt{2\pi}} + \frac{8h^2}{Ke^2} = \frac{3h^3}{8\sqrt{\pi K^3}} + \frac{8h^2}{Ke^2}.$$

Thus, (I.1) simplifies to

$$t_{K,h} \leq \frac{4^K}{K} \left( \frac{6}{h\sqrt{\pi K}} + \frac{128}{h^2 e^2} \right).$$

Correspondingly,  $T_{K,N}$  becomes

$$\begin{aligned} T_{K,N} &\leq \sum_{h=\lfloor \log_2 K \rfloor + 1}^{\log_2 N} \frac{4^K}{K} \left( \frac{6}{h\sqrt{\pi K}} + \frac{128}{h^2 e^2} \right), \\ &\leq \frac{4^K}{K} \left( \frac{6}{\sqrt{\pi K}} \sum_{h=\lfloor \log_2 K \rfloor + 1}^{\log_2 N} \frac{1}{h} + \frac{128}{e^2} \sum_{h=\lfloor \log_2 K \rfloor + 1}^{\log_2 N} \frac{1}{h^2 e^2} \right). \end{aligned}$$

It is easy to show, using Euler-Maclaurin summations, that

$$\sum_{j=a}^b j^{-1} \leq \ln \frac{b}{a-1} \text{ and } \sum_{j=a}^b j^{-1} \leq \frac{1}{a-1};$$

we then obtain

$$T_{K,N} \leq \frac{4^K}{K} \left( \frac{6}{\sqrt{\pi K}} \ln \frac{\log_2 N}{\lfloor \log_2 K \rfloor} + \frac{128}{e^2 \lfloor \log_2 K \rfloor} \right) \leq \frac{4^{K+4}}{Ke^2 \lfloor \log_2 K \rfloor} \leq \frac{4^{K+4}}{Ke^2}.$$

This proves the proposition.  $\square$

## Appendix J

### Proof of Proposition 6.3

We wish to find the value of the bound (6.3) for the subspace count given in (6.8).

We obtain  $M \geq \max_{1 \leq j \leq \lceil N/K \rceil} M_j$ , where  $M_j$  follows one of these three regimes:

$$M_j = \frac{1}{(j^r \sqrt{1 + \epsilon_K} - 1)^2} \left( 2K + 4 \ln \frac{(2e)^{K(2j+1)} N}{K(Kj+1)(Kj+K+1)} + 2t \right).$$

We separate the terms that are linear on  $K$  and  $j$ , and obtain

$$\begin{aligned} M_j &= \frac{\left( K(3 + 4 \ln 2) + 8Kj(1 + \ln 2) + 4 \ln \frac{N}{K(Kj+1)(Kj+K+1)} + 2t \right)}{(j^r \sqrt{1 + \epsilon_K} - 1)^2}, \\ &= \frac{\left( 8K(1 + \ln 2) + \frac{K(3+4 \ln 2)}{j} + \frac{4}{j} \ln \frac{N}{K(Kj+1)(Kj+K+1)} + \frac{2t}{j} \right)}{(j^{s-0.5} \sqrt{1 + \epsilon_K} - j^{-0.5})^2}. \end{aligned}$$

The sequence  $\{M_j\}_{j=1}^{\lceil \frac{N}{K} \rceil}$  is a decreasing sequence, since the numerator is a decreasing sequences and the denominator is an increasing sequence whenever  $s > 0.5$ . We then have

$$M \geq \frac{1}{(\sqrt{1 + \epsilon_K} - 1)^2} \left( K(11 + 12 \ln 2) + 4 \ln \frac{N}{K(K+1)(2K+1)} + 2t \right).$$

This completes the proof of Proposition 6.3.  $\square$

## Appendix K

### Proof of Theorem 7.2

Following [128], we define the anisotropy for the smoothness values  $s_1, \dots, s_D$  through the use of constants  $a_1, \dots, a_D$  such that

$$\frac{s_d}{s_e} = \frac{a_e}{a_d}, \quad 1 \leq d, e \leq D.$$

In order for this set of equations to have a unique solution, we pose the additional constraint  $\sum_{d=1}^D a_d = D$ . We search for the resulting approximation rate  $\lambda$  such that the following holds:

$$\lambda \left( \frac{1}{a_1}, \dots, \frac{1}{a_D} \right) = (s_1, \dots, s_D).$$

In other words, we require  $s_d = \frac{\lambda}{a_d}$ ,  $1 \leq d \leq D$ . Thus, we can write  $\lambda$  as the average of the products

$$\begin{aligned} \lambda &= \frac{s_1 a_1 + s_2 a_2 + \dots + s_D a_D}{D}, \\ &= \frac{\lambda + s_2(D - a_1 - a_3 - \dots - a_D) + \dots + s_D(D - a_1 - \dots - a_{D-1})}{D}, \\ &= \frac{D \sum_{d=2}^D s_d + \lambda - a_1 \sum_{d=2}^D s_d - a_2 \sum_{d=3}^D s_d - a_3 \sum_{d=2, d \neq 3}^D s_d - \dots - a_D \sum_{d=2}^{D-1} s_d}{D}, \\ &= \sum_{d=2}^D s_d + \frac{\lambda - \frac{\lambda}{s_1} \sum_{d=2}^D s_d - \frac{\lambda}{s_2} \sum_{d=3}^D s_d - \frac{\lambda}{s_3} \sum_{d=2, d \neq 3}^D s_d - \dots - \frac{\lambda}{s_D} \sum_{d=2}^{D-1} s_d}{D}, \\ &= \sum_{d=2}^D s_d + \frac{\lambda}{D} \left( 1 - \sum_{d=2}^D \frac{s_d}{s_1} - \sum_{d=3}^D \frac{s_d}{s_2} - \sum_{d=2, d \neq 3}^D \frac{s_d}{s_3} - \dots - \sum_{d=2}^{D-1} \frac{s_d}{s_D} \right), \\ &= \sum_{d=2}^D s_d + \frac{\lambda}{D} \left( 1 - \frac{1}{s_1} \sum_{d=2}^D s_d - \sum_{e=2}^D \frac{1}{s_e} \sum_{d=2, d \neq e}^D s_d \right). \end{aligned}$$

Grouping the terms with  $\lambda$ , we obtain

$$\frac{\lambda}{D} \left( D - 1 + \frac{1}{s_1} \sum_{d=2}^D s_d + \sum_{e=2}^D \frac{1}{s_e} \sum_{d=2, d \neq e}^D s_d \right) = \sum_{d=2}^D s_d.$$

Solving for  $\lambda$ , we obtain

$$\begin{aligned} \lambda &= \frac{D \sum_{d=2}^D s_d}{D - 1 + \frac{1}{s_1} \sum_{d=2}^D s_d + \sum_{e=2}^D \frac{1}{s_e} \sum_{d=2, d \neq e}^D s_d}, \\ &= \frac{D \sum_{d=2}^D s_d}{D - 1 + \frac{1}{s_1} \sum_{d=2}^D s_d + \sum_{e=2}^D \frac{1}{s_e} \left( \sum_{d=2}^D s_d - s_e \right)}, \\ &= \frac{D \sum_{d=2}^D s_d}{D - 1 + \frac{1}{s_1} \sum_{d=2}^D s_d + \sum_{e=2}^D \frac{1}{s_e} \sum_{d=2}^D s_d - (D - 1)}, \\ &= \frac{D}{\frac{1}{s_1} + \sum_{e=2}^D \frac{1}{s_e}}, \\ &= \frac{D}{\sum_{d=1}^D \frac{1}{s_d}}, \end{aligned}$$

proving the theorem. □



## Bibliography

- [1] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-time signal processing*. Upper Saddle River, N.J.: Prentice Hall, 2nd ed., 1999.
- [2] S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego: Academic Press, 1999.
- [3] W. Pennebaker and J. Mitchell, “JPEG: Still image data compression standard,” *Van Nostrand Reinhold*, 1993.
- [4] D. S. Taubman and M. W. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Kluwer, 2001.
- [5] E. J. Candès, “Compressive sampling,” in *Int. Congress of Mathematicians*, vol. 3, (Madrid, Spain), pp. 1433–1452, 2006.
- [6] D. Donoho, “Compressed sensing,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 1289–1306, Apr. 2006.
- [7] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, “Single pixel imaging via compressive sampling,” *IEEE Signal Proc. Mag.*, vol. 25, pp. 83–91, March 2008.
- [8] A. Wagadarikar, R. John, R. Willett, and D. Brady, “Single disperser design

for coded aperture snapshot spectral imaging,” *App. Optics*, vol. 47, no. 10, pp. B44–B51, 2008.

- [9] D. Baron, M. F. Duarte, S. Sarvotham, M. B. Wakin, and R. G. Baraniuk, “Distributed compressive sensing,” 2005. Preprint.
- [10] R. M. Willett, M. E. Gehm, and D. J. Brady, “Multiscale reconstruction for computational hyperspectral imaging,” in *Computational Imaging V*, vol. 6498 of *Proc. SPIE*, (San Jose, CA), Jan. 2007.
- [11] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, “Compressive imaging for video representation and coding,” in *Picture Coding Symp.*, (Beijing, China), Apr. 2006.
- [12] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, “Compressed sensing MRI,” *IEEE Signal Processing Mag.*, vol. 25, pp. 72–82, Mar. 2008.
- [13] L.-W. Kang and C.-S. Lu, “Distributed compressed video sensing,” in *IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, (Taipei, Taiwan), Apr. 2009.
- [14] V. Stankovic, L. Stankovic, and S. Cheng, “Compressive video sampling,” in *European Signal Processing Conf. (EUSIPCO)*, (Lausanne, Switzerland), Aug. 2008.
- [15] R. F. Marcia and R. M. Willett, “Compressive coded aperture video reconstruction,” in *European Signal Processing Conf. (EUSIPCO)*, (Lausanne, Switzer-

land), Aug. 2008.

- [16] J. Y. Park and M. B. Wakin, “A multiscale framework for compressive sensing of video,” in *Picture Coding Symposium*, (Chicago, IL), May 2009.
- [17] P. Ye, J. L. Paredes, G. R. Arce, Y. Wu, C. Chen, and D. W. Prather, “Compressive confocal microscopy: 3D reconstruction algorithms,” in *SPIE Photonics West*, (San Jose, CA), Jan. 2009.
- [18] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, “Model-based compressive sensing,” Aug. 2008. Preprint.
- [19] Y. Eldar and M. Mishali, “Robust recovery of signals from a union of subspaces,” July 2008. Preprint.
- [20] M. Rabbat, J. D. Haupt, A. Singh, and R. D. Nowak, “Decentralized compression and predistribution via randomized gossiping,” in *Int. Workshop on Inform. Processing in Sensor Networks (IPSN)*, (Nashville, TN), pp. 51–59, Apr. 2006.
- [21] W. Wang, M. Garofalakis, and K. Ramchandran, “Distributed sparse random projections for refinable approximation,” in *Int. Workshop on Inform. Processing in Sensor Networks (IPSN)*, (Cambridge, MA), pp. 331–339, Apr. 2007.
- [22] S. Aeron, M. Zhao, and V. Saligrama, “On sensing capacity of sensor networks for the class of linear observation, fixed SNR models,” in *IEEE Workshop on Statistical Signal Processing (SSP)*, (Madison, WI), pp. 388–392, Aug. 2007.

- [23] S. Aeron, M. Zhao, and V. Saligrama, “Fundamental limits on sensing capacity for sensor networks and compressed sensing,” 2008. Preprint.
- [24] W. Bajwa, A. Sayeed, and R. Nowak, “Matched source-channel communication for field estimation in wireless sensor networks,” in *Int. Workshop on Inform. Processing in Sensor Networks (IPSN)*, (Los Angeles, CA), pp. 332–339, Apr. 2005.
- [25] W. U. Bajwa, J. D. Haupt, A. M. Sayeed, and R. D. Nowak, “Joint source-channel communication for distributed estimation in sensor networks,” *IEEE Trans. Inform. Theory*, vol. 53, pp. 3629–3653, Oct. 2007.
- [26] D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, K. F. Kelly, and R. G. Baraniuk, “A compressed sensing camera: New theory and an implementation using digital micromirrors,” in *Computational Imaging IV*, vol. 6065 of *Proc. SPIE*, (San Jose, CA), Jan. 2006.
- [27] D. Takhar, *Compressed sensing for imaging applications*. PhD thesis, Rice University, Houston, TX, Feb. 2008.
- [28] P. L. Dragotti, G. Poggi, and A. Ragozini, “Compression of multispectral images by three-dimensional spiht algorithm,” *IEEE Trans. Geoscience and Remote Sensing*, vol. 38, pp. 416–428, Jan. 2000.
- [29] J. Fowler, “Compressive-projection principal component analysis,” *IEEE Trans. Image Processing*, 2008. Submitted.

- [30] Y. Rivenson and A. Stern, "Compressed imaging with a separable sensing operator," *IEEE Signal Processing Letters*, vol. 16, pp. 449–452, June 2009.
- [31] S. Jokar and V. Mehrmann, "Sparse representation of solutions of Kronecker product systems," 2008. Preprint.
- [32] D. Donoho, "Denoising by soft thresholding," *IEEE Trans. on Inform. Theory*, vol. 41, pp. 613–627, May 1995.
- [33] D. Donoho and J. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, Sept. 1994.
- [34] D. Donoho and I. Johnstone, "Minimax estimation via wavelet shrinkage," *Annals of Stat.*, vol. 3, pp. 879–921, June 1998.
- [35] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. on Sci. Comp.*, vol. 20, no. 1, pp. 33–61, 1998.
- [36] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, pp. 3397–3415, Dec. 1993.
- [37] R. DeVore and V. N. Temlyakov, "Some remarks on greedy algorithms," *Adv. Comput. Math.*, vol. 5, pp. 173–187, 1996.
- [38] Y. C. Pati, R. Rezaifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Asilomar Conf. Signals, Systems and Computation*, vol. 1, (Pacific Grove, CA), pp. 40–44, Nov. 1993.

- [39] D. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via  $\ell_1$  minimization," *Proc. Nat. Acad. Sci.*, vol. 100, pp. 2197–2202, Mar. 2003.
- [40] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inform. Theory*, vol. 50, pp. 2231–2242, Oct. 2004.
- [41] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Statist. Soc.*, vol. 58, no. 1, pp. 267–288, 1996.
- [42] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Stat.*, vol. 32, pp. 407–499, Apr. 2004.
- [43] M. Figueiredo and R. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Processing*, vol. 12, pp. 906–916, Dec. 2003.
- [44] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, pp. 1413–1457, 2004.
- [45] D. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inform. Theory*, vol. 52, pp. 6–18, Jan. 2006.
- [46] D. Donoho and Y. Tsaig, "Extensions of compressed sensing," *Signal Processing*, vol. 86, pp. 533–548, Mar. 2006.

- [47] E. Candès and J. Romberg, “Quantitative robust uncertainty principles and optimally sparse decompositions,” *Found. of Comp. Math.*, vol. 6, pp. 227–254, Apr. 2006.
- [48] E. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 489–509, Feb. 2006.
- [49] E. Candès and T. Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 5406–5425, Dec. 2006.
- [50] E. J. Candès and T. Tao, “Decoding by linear programming,” *IEEE Trans. Inform. Theory*, vol. 51, pp. 4203–4215, Dec. 2005.
- [51] E. Candès, M. Rudelson, T. Tao, and R. Vershynin, “Error correction via linear programming,” in *IEEE Symp. on Found. of Comp. Sci. (FOCS)*, (Pittsburgh, PA), pp. 295–308, Oct. 2005.
- [52] E. Candès and M. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, pp. 21–30, Mar. 2008.
- [53] R. G. Baraniuk, “Compressive sensing,” *IEEE Signal Processing Mag.*, vol. 24, no. 4, pp. 118–120, 124, July 2007.
- [54] S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann, “Uniform uncertainty principle for Bernoulli and subgaussian ensembles,” *Constructive Approx.*, Feb.

2008. To appear.

- [55] R. G. Baraniuk, M. A. Davenport, R. A. DeVore, and M. B. Wakin, “A simple proof of the restricted isometry property for random matrices,” *Constructive Approx.*, vol. 28, pp. 253–263, Dec. 2008.
- [56] E. J. Candès and J. Romberg, “Sparsity and incoherence in compressive sampling,” *Inverse Problems*, vol. 23, pp. 969–985, June 2007.
- [57] J. Tropp and A. Gilbert, “Signal recovery from partial information via orthogonal matching pursuit,” *IEEE Trans. Inform. Theory*, vol. 53, pp. 4655–4666, Dec. 2007.
- [58] E. T. Hale, W. Yin, and Y. Zhang, “Fixed-point continuation method for  $\ell_1$ -minimization with applications to compressed sensing,” *SIAM J. Optimization*, vol. 19, pp. 1107–1130, Oct. 2008.
- [59] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, “Gradient projections for sparse reconstruction: Application to compressed sensing and other inverse problems,” *IEEE J. Selected Topics in Signal Processing*, vol. 1, pp. 586–598, Apr. 2007.
- [60] E. van den Berg and M. P. Friedlander, “Probing the Pareto frontier for basis pursuit solutions,” *SIAM J. on Sci. Comp.*, vol. 31, pp. 890–912, Nov. 2008.
- [61] S. Wright, R. Nowak, and M. A. T. Figueiredo, “Sparse reconstruction by separable approximation,” *IEEE Trans. Signal Processing*, vol. 57, pp. 2479–2493,



July 2009.

- [62] J. Haupt and R. Nowak, “Signal reconstruction from noisy random projections,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 4036–4048, Sept. 2006.
- [63] E. J. Candès and J. K. Romberg, “Signal recovery from random projections,” in *Computational Imaging III*, vol. 5674 of *Proc. SPIE*, (San Jose, CA), pp. 76–86, Jan. 2005.
- [64] T. Blumensath and M. E. Davies, “Iterative thresholding for sparse approximations,” *J. Fourier Analysis and Applications*, vol. 14, pp. 629–654, Dec. 2008.
- [65] T. Blumensath and M. E. Davies, “Iterative hard thresholding for compressed sensing,” *Appl. Comput. Harmon. Anal.*, July 2008. To appear.
- [66] D. Needell and J. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” *Appl. Comput. Harmon. Anal.*, vol. 26, pp. 301–321, May 2008.
- [67] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing: Closing the gap between performance and complexity,” Mar. 2008. Preprint.
- [68] M. Rudelson and R. Vershynin, “On sparse reconstruction from Fourier and Gaussian measurements,” *Comm. Pure Appl. Math.*, vol. 61, pp. 1025–1171, Aug. 2008.
- [69] J. Tropp, A. C. Gilbert, and M. J. Strauss, “Simultaneous sparse approximation via greedy pursuit,” in *IEEE Int. Conf. Acoustics, Speech, Signal Process-*

ing (ICASSP), vol. V, (Philadelphia, PA), pp. 721–724, Mar. 2005.

- [70] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, “Performance limits for jointly sparse signals via graphical models,” in *Workshop on Sensor, Signal and Information Processing (SENSIP)*, (Sedona, AZ), May 2008.
- [71] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, “Theoretical performance limits for jointly sparse signals via graphical models,” Tech. Rep. TREE-0802, Rice University, Houston, TX, July 2008.
- [72] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
- [73] D. Slepian and J. K. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, July 1973.
- [74] M. F. Duarte, M. B. Wakin, D. Baron, and R. G. Baraniuk, “Universal distributed sensing via random projections,” in *Int. Workshop on Inform. Processing in Sensor Networks (IPSN)*, (Nashville, TN), pp. 177–185, Apr. 2006.
- [75] D. Ganesan, D. Estrin, and J. Heidemann, “DIMENSIONS: Why do we need a new data handling architecture for sensor networks?,” in *ACM Workshop on Hot Topics in Networks*, (Princeton, NJ), pp. 143–148, Oct. 2002.
- [76] J. Gao, L. J. Guibas, J. Hershberger, and L. Zhang, “Fractionally cascaded information in a sensor network,” in *Int. Workshop on Inform. Processing in*

*Sensor Networks (IPSN)*, (Berkeley, CA), pp. 311–319, Apr. 2004.

- [77] A. Kashyap, L. A. Lastras-Montano, C. Xia, and Z. Liu, “Distributed source coding in dense sensor networks,” in *Data Compression Conf. (DCC)*, (Snowbird, UT), Mar. 2005.
- [78] S. Patten, B. Krishnamachari, and R. Govindan, “The impact of spatial correlation on routing with compression in wireless sensor networks,” in *Int. Workshop on Inform. Processing in Sensor Networks (IPSN)*, (Berkeley, CA), pp. 28–35, Apr. 2004.
- [79] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, “Detection, classification, and tracking of targets,” *IEEE Signal Processing Mag.*, vol. 19, pp. 17–29, Feb. 2002.
- [80] M. F. Duarte and Y. H. Hu, “Vehicle classification in distributed sensor networks,” *J. Parallel and Distributed Computing*, vol. 64, pp. 826–838, July 2004.
- [81] A. Sayeed, “A statistical signal modeling framework for wireless sensor networks,” tech. rep., Univ. of Wisconsin - Madison, Feb 2004.
- [82] E. Candès, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Comm. Pure Appl. Math.*, vol. 59, pp. 1207–1223, Aug. 2006.
- [83] J. A. Tropp, M. B. Wakin, M. F. Duarte, D. Baron, and R. G. Baraniuk, “Random filters for compressive sampling and reconstruction,” in *IEEE Int. Conf.*

*Acoustics, Speech, Signal Processing (ICASSP)*, vol. III, (Toulouse, France), pp. 872–875, May 2006.

- [84] S. PalChaudhuri, S. Du, A. K. Saha, and D. B. Johnson, “TreeCast: A stateless addressing and routing architecture for sensor networks,” in *Int. Parallel and Distributed Processing Symp. (IPDPS)*, (Santa Fe, NM), pp. 221–228, Apr. 2004.
- [85] J. Zhao and R. Govindan, “Understanding packet delivery performance in dense wireless sensor networks,” in *Int. Conf. Embedded Networked Sensor Systems (SenSys)*, (Los Angeles, CA), pp. 1–13, Nov. 2003.
- [86] S. D. Servetto, K. Ramchandran, V. A. Vaishampayan, and K. Nahrstedt, “Multiple Description Wavelet Based Image Coding,” *IEEE Trans. Image Processing*, vol. 9, pp. 813–826, May 2000.
- [87] Y. Wang, M. T. Orchard, and A. Reibman, “Multiple description image coding for noisy channels by pairing transform coefficients,” in *IEEE Workshop on Multimedia Signal Processing (MMSP)*, (Princeton, NJ), pp. 419–424, June 1997.
- [88] S. N. Diggavi and V. A. Vaishampayan, “On multiple description source coding with decoder side information,” in *Inform. Theory Workshop (ITW)*, (San Antonio, TX), pp. 88–93, Oct. 2004.
- [89] S. D. Rane, A. Aaron, and B. Girod, “Systematic lossy forward error protection for error-resilient digital video broadcasting,” in *Security, Steganography, and*

*Watermarking of Multimedia Contents VI*, vol. 5306 of *Proc. SPIE*, (San Jose, CA), pp. 588–595, Jan. 2004.

- [90] M. F. Duarte, M. A. Davenport, M. B. Wakin, and R. G. Baraniuk, “Sparse signal detection from incoherent projections,” in *IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, vol. III, (Toulouse, France), pp. 305–308, Mar. 2006.
- [91] F. Koushanfar, N. Taft, and M. Potkonjak, “Sleeping coordination for comprehensive sensing using isotonic regression and domatic partitions,” in *Int. Conf. Comp. Communication (INFOCOM)*, (Barcelona, Spain), pp. 1–13, Apr. 2006.
- [92] M. F. Duarte and R. G. Baraniuk, “Fast reconstruction from random incoherent projections,” Tech. Rep. TREE-0507, Rice University ECE Department, Houston, TX, May 2005.
- [93] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk, “Fast reconstruction of piecewise smooth signals from random projections,” in *Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, (Rennes, France), Nov. 2005.
- [94] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk, “Wavelet-domain compressive signal reconstruction using a hidden Markov tree model,” in *IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, (Las Vegas, NV), pp. 5137–5140, April 2008.

- [95] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using Hidden Markov Models," *IEEE Trans. Signal Processing*, vol. 46, pp. 886–902, Apr. 1998.
- [96] H. Choi, J. Romberg, R. Baraniuk, and N. Kingsbury, "Hidden Markov tree modeling of complex wavelet transforms," in *IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, vol. 1, (Istanbul, Turkey), pp. 133–136, June 2000.
- [97] J. K. Romberg, H. Choi, and R. G. Baraniuk, "Bayesian tree-structured image modeling using wavelet-domain Hidden Markov Models," *IEEE Trans. Image Processing*, vol. 10, pp. 1056–1068, July 2001.
- [98] R. Baraniuk, N. Kingsbury, and I. Selesnick, "The dual-tree complex wavelet transform," *IEEE Signal Processing Mag.*, vol. 22, pp. 123–151, June 2005.
- [99] C. La and M. N. Do, "Tree-based orthogonal matching pursuit algorithm for signal reconstruction," in *IEEE Int. Conf. Image Processing (ICIP)*, (Atlanta, GA), pp. 1277–1280, Oct. 2006.
- [100] C. La and M. N. Do, "Signal reconstruction using sparse tree representation," in *Wavelets XI*, vol. 5914 of *Proc. SPIE*, (San Diego, CA), Aug. 2005.
- [101] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by weighted  $\ell_1$  minimization," tech. rep., California Institute of Technology, Pasadena, CA, Oct. 2007.
- [102] M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized

- Gaussian density and Kullback-Leibler distance,” *IEEE Trans. Image Processing*, vol. 11, pp. 146–158, Feb. 2002.
- [103] M. J. Wainwright and E. P. Simoncelli, “Scale mixtures of Gaussians and the statistics of natural images,” in *Workshop on Neural Inform. Processing Systems (NIPS)*, (Vancouver, Canada), Dec. 2000.
- [104] J. Kivinen, E. Sudderth, and M. Jordan, “Image denoising with nonparametric Hidden Markov Trees,” in *IEEE Int. Conf. Image Processing (ICIP)*, vol. III, (San Antonio, TX), pp. 121–124, Sept. 2007.
- [105] M. F. Duarte and V. Cevher, “Model-based compressive sensing for signal ensembles,” July 2009. Preprint.
- [106] V. Cevher, M. F. Duarte, C. Hegde, and R. G. Baraniuk, “Sparse signal recovery using Markov Random Fields,” in *Workshop on Neural Inform. Processing Systems (NIPS)*, (Vancouver, Canada), Dec. 2008.
- [107] T. Blumensath and M. E. Davies, “Sampling theorems for signals from the union of finite-dimensional linear subspaces,” *IEEE Trans. Inform. Theory*, Dec. 2008. To appear.
- [108] Y. M. Lu and M. N. Do, “Sampling signals from a union of subspaces,” *IEEE Signal Processing Mag.*, vol. 25, pp. 41–47, Mar. 2008.
- [109] M. N. Do and C. N. H. La, “Tree-based majorize-minimize algorithm for compressed sensing with sparse-tree prior,” in *Int. Workshop on Computational*

*Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, (Saint Thomas, US Virgin Islands), pp. 129–132, Dec. 2007.

- [110] M. Stojnic, F. Parvaresh, and B. Hassibi, “On the reconstruction of block-sparse signals with an optimal number of measurements,” *IEEE Trans. Signal Processing*, Mar. 2008. To appear.
- [111] L. He and L. Carin, “Exploiting structure in wavelet-based Bayesian compressive sensing,” Sept. 2008. Preprint.
- [112] K. Lee and Y. Bresler, “Selecting good Fourier measurements for compressed sensing.” SIAM Conf. Imaging Science, July 2008.
- [113] E. J. Candès, “The restricted isometry property and its implications for compressed sensing,” *Compte Rendus de l’Academie des Sciences, Series I*, vol. 346, pp. 589–592, May 2008.
- [114] R. G. Baraniuk and D. L. Jones, “A signal-dependent time-frequency representation: Fast algorithm for optimal kernel design,” *IEEE Trans. Signal Processing*, vol. 42, pp. 134–146, Jan. 1994.
- [115] R. G. Baraniuk, “Optimal tree approximation with wavelets,” in *Wavelet Applications in Signal and Image Processing VII*, vol. 3813 of *Proc. SPIE*, (Denver, CO), pp. 196–207, July 1999.
- [116] R. G. Baraniuk, R. A. DeVore, G. Kyriazis, and X. M. Yu, “Near best tree approximation,” *Advances in Comp. Math.*, vol. 16, pp. 357–373, May 2002.



- [117] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, “Image denoising using a scale mixture of Gaussians in the wavelet domain,” *IEEE Trans. Image Processing*, vol. 12, pp. 1338–1351, Nov. 2003.
- [118] J. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.
- [119] A. Cohen, W. Dahmen, I. Daubechies, and R. A. DeVore, “Tree approximation and optimal encoding,” *Appl. Comput. Harmon. Anal.*, vol. 11, pp. 192–226, Sept. 2001.
- [120] D. Donoho, “CART and best ortho-basis: A connection,” *Annals of Stat.*, vol. 25, pp. 1870–1911, Oct. 1997.
- [121] M. B. Wakin, S. Sarvotham, M. F. Duarte, D. Baron, and R. G. Baraniuk, “Recovery of jointly sparse signals from few random projections,” in *Workshop on Neural Inform. Processing Systems (NIPS)*, (Vancouver, Canada), Nov. 2005.
- [122] J. Tropp, A. C. Gilbert, and M. J. Strauss, “Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit,” *Signal Processing*, vol. 86, pp. 572–588, Apr. 2006.
- [123] M. F. Duarte and R. G. Baraniuk, “Kronecker products for compressive sensing,” July 2009. Preprint.
- [124] V. Cevher, M. F. Duarte, and R. Baraniuk, “Localization via spatial sparsity,” in *European Signal Processing Conf. (EUSIPCO)*, (Lausanne, Switzerland),

land), 2008.

- [125] M. Aghagolzadeh and K. Oweiss, “Compressed and distributed sensing of neuronal activity for real time spike train decoding,” *IEEE Trans. Neural Systems and Rehabilitation Eng.*, vol. 17, pp. 116–127, Apr. 2009.
- [126] R. C. Thompson, “Principal submatrices XI: Interlacing inequalities for singular values,” *Linear Algebra and Appl.*, vol. 5, pp. 1–12, 1972.
- [127] R. Hochmuth, “Wavelet characterizations for anisotropic Besov spaces,” *Appl. Comput. Harmon. Anal.*, vol. 12, pp. 179–208, Mar. 2002.
- [128] R. Hochmuth, “N-term approximation in anisotropic function spaces,” *Mathematische Nachrichten*, vol. 244, pp. 131–149, Oct. 2002.
- [129] R. A. DeVore, “Nonlinear approximation,” *Acta Numerica*, vol. 7, pp. 51–150, 1998.
- [130] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, “Compressive sensing for background subtraction,” in *European Conf. Comp. Vision (ECCV)*, (Marseille, France), pp. 155–168, Oct. 2008.
- [131] D. Reddy, A. C. Sankaranarayanan, V. Cevher, and R. Chellappa, “Compressed sensing for multi-view tracking and 3-D voxel reconstruction,” in *IEEE Int. Conf. Image Processing (ICIP)*, (San Diego, CA), pp. 221–224, Oct. 2008.
- [132] V. Cevher, A. C. Gurbuz, J. H. McClellan, and R. Chellappa, “Compressive wireless arrays for bearing estimation,” in *IEEE Int. Conf. Acoustics, Speech,*

*Signal Processing (ICASSP)*, (Las Vegas, NV), pp. 2497–2500, Apr. 2008.

- [133] A. C. Gurbuz, V. Cevher, and J. H. McClellan, “A compressive beamformer,” in *IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, (Las Vegas, NV), pp. 2617–2620, Apr. 2008.
- [134] C. Hegde, M. F. Duarte, and V. Cevher, “Compressive sensing recovery of spike trains using a structured sparsity model,” in *Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, (Saint Malo, France), Apr. 2009.
- [135] V. Cevher, P. Indyk, C. Hegde, and R. G. Baraniuk, “Recovery of clustered sparse signals from compressive measurements,” in *Int. Conf. on Sampling Theory and Applications (SAMPTA)*, (Marseille, France), May 2009.
- [136] R. G. Baraniuk and M. B. Wakin, “Random projections of smooth manifolds,” *Found. of Comp. Math.*, vol. 9, pp. 51–77, Feb. 2009.
- [137] M. A. Davenport, C. Hegde, M. F. Duarte, and R. G. Baraniuk, “A theoretical analysis of joint manifolds,” Tech. Rep. TREE-0901, Rice University ECE Department, Houston, TX, Jan. 2009.
- [138] I. Daubechies, M. Fornasier, and I. Loris, “Accelerated projected gradient method for linear inverse problems with sparsity constraints,” *J. Fourier Analysis and Applications*, vol. 14, pp. 764–792, 2008.
- [139] I. Loris, M. Bertero, C. De Mol, R. Zanella, and L. Zanni, “Accelerating gradient

projection methods for  $\ell_1$ - constrained signal recovery by steplength selection,”  
*Appl. Comput. Harmon. Anal.*, 2009. To appear.

- [140] H. G. Feichtinger, K. Gröchenig, and D. Walnut, “Wilson bases and modulation spaces,” *Mathematische Nachrichten*, vol. 155, pp. 7–17, Jan. 1992.
- [141] L. Borup and M. Nielsen, “Nonlinear approximation in  $\alpha$ -modulation spaces,” *Mathematische Nachrichten*, vol. 279, pp. 101–120, Jan. 2006.
- [142] D. B. West, *Introduction to Graph Theory*. Prentice Hall, 1996.
- [143] M. Ledoux, *The Concentration of Measure Phenomenon*. American Mathematical Society, 2001.
- [144] G. G. Brown and B. O. Shubert, “On random binary trees,” *Mathematics of Operations Research*, vol. 9, pp. 43–65, Feb. 1984.