Compressed Sensing in Multi-Signal Environments

by

Jae Young Park

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2013

Doctoral Committee:

Professor Anna C. Gilbert, Co-Chair
Assistant Professor Michael B. Wakin, Colorado School of Mines, Co-Chair
Professor Selim Esedoglu
Professor Jeffrey A. Fessler

To my youth

ACKNOWLEDGEMENTS

I have met so many great people in graduate school who have influenced me in the most positive way to be the person that I am today. I am grateful to each and every one of them and I would like to take this moment to say a few words of appreciation.

First and foremost, I must thank Professor Wakin for the endless guidance, support and encouragement he has given me over past five years. He was the perfect advisor I could ever wish for and it has absolutely been a privilege working with him. I would like to thank Professor Gilbert for her invaluable suggestions and contributions to this dissertation. I had so much fun working with her and I will definitely miss the enthusiastic conversations we had in our weekly meetings. I would like to thank Professor Fessler for the insightful comments he has given me to strengthen this dissertation. I am also grateful for his mentorship; he is by far the best graduate student mentor in the EECS department. I would also like to thank Professor Esedoglu for helpful suggestions on this dissertation.

I could not have survived grad school without the help and support of my friends I met in Ann Arbor. I must thank Hyun Jeong Cho who has taught me so much in so many ways since the day I started grad school. I will always be thankful for the encouragement and support she has given me to help me come this far. Thank you. I am also grateful to Yoojin Choi who was always there to listen and for being an incredible *hyoung* to me. I would also like to thank Jooseuk Kim, and Seung Hyun Oh for the joyful memories that I will always remember. And to all my other friends in Ann Arbor who have helped me get through graduate school; thank you all, you know who you are.

Also, big thanks to my HK friends across the world who made my yearly trips back home unforgettable; Alex Kim, Jung Ho Lee, Sam Pyo, Kiung Um, Kevin Kim, Eun Kee Choi, Jun Hyoung Hong, Mike Kim, Jong Gu Lee, Ki Cho Jung, Julia Ryu, Wendy Wang, Nayoung Kim, Jenny Choi, Maria Park.

Finally, I would like to thank my parents and my sister for always being there for me when I needed them most. Thank you *umma*, thank you *abba*, and thank you *nuna*.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF FIGURES vii

LIST OF APPENDICES xi

ABSTRACT xii

CHAPTER

1. Introduction 1

1.1 Multi-Signal Environments 1
1.2 Compressed Sensing in Multi-Signal Environment 2
1.3 Outline and Contributions 4

2. Background on Signal Models and Compressed Sensing 7

2.1 Concise Signal Models 7
2.2 Compressed Sensing 9

3. Compressed Sensing in Video Acquisition and Reconstruction 12

3.1 Introduction 12
3.1.1 Motivation and Overview 12
3.2 On the Temporal Bandwidth of Video Signals 15
3.2.1 Setup 15
3.2.2 Temporal Bandwidth Analysis 16
3.2.3 Supporting Experiments 19
3.2.4 Videos with Two Spatial Dimensions 23
3.3 Anchor Frames for Reducing Reconstruction Complexity 25
3.3.1 Sampling and Interpolation Principles 25
3.3.2 CS in Streaming Scenarios 26
3.4 Multiscale Reconstruction Algorithm 29
3.4.1 Problem Formulation 30
3.4.2 The Multiscale Approach 31
3.4.3 Lifting-based Invertible Motion Adaptive Transform (LIMAT) . . . 34
3.5 Discussion and Simulations 35
3.5.1 Related Work 35
3.5.2 Simulations 39

4. Compressed Sensing in Multiview Imaging and Reconstruction 46

4.1 Introduction 46
4.2 Problem Setup and Related Work 48
4.2.1 Multi-view Imaging using Image-by-Image Random Measurements 48
4.2.2 Current Approaches to DCS Multi-view Image Reconstruction . . . 49
4.3 Manifold Lifting Techniques for Multi-view Image Reconstruction 51
4.3.1 Far-field Multi-view Imaging 52
4.3.2 Near-field Multi-view Imaging 52
4.3.3 Dealing with Uncertainties in Camera Positions 55
4.4 Manifold Lifting Case Study 57
4.4.1 Problem Setup 57
4.4.2 Manifold Lifting Algorithm 58

5. Concentration of Measure for Block Diagonal Matrices with Applications to Compressive Signal Processing 64

5.1 Introduction 64
5.2 Background and Related Work 66
5.2.1 Subgaussian Random Variables 67
5.2.2 Concentration Inequalities 67
5.2.3 Applications of Concentration Inequalities 68
5.3 Non-uniform Concentration of Measure Inequalities 70
5.3.1 Distinct Block Diagonal (DBD) Matrices 70
5.3.2 Repeated Block Diagonal (RBD) Matrices 74
5.4 Applications 76
5.4.1 Stable Embeddings and the Johnson-Lindenstrauss Lemma 77
5.4.2 Signal Detection in the Compressed Domain 79
5.4.3 Classification in the Compressed Domain 82
5.5 Favorable Signal Classes 84
5.5.1 Delay Networks and Multiview Imaging 84
5.5.2 Frequency Sparse Signals 85
5.5.3 Difference Signals 87
5.5.4 Random Signals 88
5.6 Conclusion 90

6. Sketched SVD: Recovering Spectral Features from Compressive Measurements 92

6.1 Introduction 92
6.2 Problem Setup 95
6.3 Related Work 97
6.3.1 Absolute Bounds 97
6.3.2 Relative Bounds 99
6.3.3 Relation to Simultaneous Iteration 100
6.3.4 Randomized Algorithms for Linear Algebra 101
6.4 Main Result 102
6.4.1 Proposed Algorithm and Estimation Bounds 102
6.4.2 Proof of Theorem 6.2 104
6.5 Application to Spectral Analysis of Streaming Graphs 109
6.6 Conclusion 113

7. Compressed Sensing in Structural Health Monitoring Systems 114

7.1 Introduction . . . 114
7.1.1 Structural health monitoring systems . . . 114
7.1.2 The usefulness of wireless SHM systems . . . 115
7.2 Background . . . 118
7.2.1 Single-degree-of-freedom system . . . 119
7.2.2 Multiple-degree-of-freedom system . . . 120
7.3 Problem Formulation . . . 122
7.3.1 The analytic signal of $\{u(t)\}$ . . . 122
7.3.2 The relationship to singular value decomposition . . . 123
7.3.3 The implications of sampling . . . 124
7.4 Main Results . . . 125
7.4.1 Proposed method . . . 125
7.4.2 Uniform sampling . . . 126
7.4.3 Random sampling . . . 127
7.5 Proof of Main Results . . . 128
7.5.1 Perturbation analysis . . . 128
7.5.2 Proof for random sampling . . . 129
7.5.3 Proof for uniform sampling . . . 135
7.6 Experimental Results . . . 139

8. Conclusion . . . 141

8.1 Joint signal model and reconstruction algorithm . . . 141
8.1.1 Video application . . . 141
8.1.2 Multi-view application . . . 143
8.1.3 Structural health monitoring application . . . 144
8.2 Measurement matrices in multi-signal scenarios . . . 145

APPENDICES . . . 147

BIBLIOGRAPHY . . . 156

LIST OF FIGURES

Figure

2.1 *A manifold $\mathcal{M}$ can be viewed as a nonlinear surface in $\mathbb{R}^N$. When the mapping between θ and x_θ is well-behaved, as we trace out a path in the parameter space Θ, we trace out a similar path on $\mathcal{M}$. A random projection Φ from $\mathbb{R}^N$ to a lower dimensional space $\mathbb{R}^M$ can provide a stable embedding of $\mathcal{M}$, preserving all pairwise distances, and therefore preserving the structure within an ensemble of images. The goal of a manifold lifting algorithm is to recover an ensemble of images from their low-dimensional measurements.* . 8

2.2 *The multiscale structure of manifolds. The top manifold in this figure corresponds to the collection of images of a teapot that could be acquired from different camera positions θ. While manifolds like this containing images with sharp edges are not differentiable, manifolds of images containing smooth images* are *differentiable, and the more one smoothes the images, the smoother the manifold becomes.* 9

3.1 *(a) Butterfly structure in the two-dimensional Fourier transform of a simple translational 1D video. The slope of the lines is proportional to the maximum speed of the translation. (b) Bandlimiting the video in space (e.g., by spatial lowpass filtering) will also bandlimit the video in time. The resulting temporal bandwidth will be proportional to the spatial bandwidth times the maximum translational speed.* 18

3.2 *Top row: (a) Video $f(x,t) = g(x - h(t))$ with bandlimited profile $g(x)$ and bandlimited translation signal $h(t)$. (b) Estimated spectrum. Middle row: (c) 4096 sampled 1D rows from* Candle *video. (d) Estimated spectrum with windowing to alleviate border artifacts. Bottom row: (e) 4096 sampled 1D rows from* Pendulum + Cars *video. (f) Estimated spectrum with windowing to alleviate border artifacts.* 21

3.3 *Reconstructed anchor frames of* Candle *video. Out of 512 original frames, 32 anchor frames were reconstructed. From the top each row shows the 5th, 11th, 21st, and 28th anchor frame and its reconstruction via various reconstruction methods. From the left each column represents: (a) original frames, (b) 3D-DWT with rectangular interpolation kernel, PSNR = 33.67 dB, (c) C_n, $n = 32$, with rectangular interpolation kernel, PSNR = 33.68 dB, (d) LIMAT with rectangular interpolation kernel and zero-motion vectors, PSNR = 33.80 dB, (e) LIMAT with linear interpolation kernel, PSNR = 36.76 dB.* 40

3.4 *Reconstructed anchor frames of* Pendulum + Cars *video. Out of 512 original frames, 32 anchor frames were reconstructed. From the top each row shows the 4th, 13th, 22nd, and 30th anchor frame and its reconstruction via various reconstruction methods. From the left each column represents: (a) original frames, (b) 3D-DWT with rectangular interpolation kernel, PSNR = 28.22 dB, (c) C_n, $n = 32$, with rectangular interpolation kernel, PSNR = 29.58 dB, (d) LIMAT with rectangular interpolation kernel and zero-motion vectors, PSNR = 29.81 dB, (e) LIMAT with linear interpolation kernel, PSNR = 31.38 dB.* 41

3.5 *Reconstructed anchor frames of* Candle *video. Out of 512 original frames, 16 anchor frames were reconstructed. From the top each row shows the 3rd, 6th, 12th, and 15th anchor frame and its reconstruction via various reconstruction methods. From the left each column represents: (a) original frames, (b) 3D-DWT with rectangular interpolation kernel, PSNR = 32.79 dB, (c) C_n, $n = 32$, with rectangular interpolation kernel, PSNR = 32.70 dB, (d) LIMAT with rectangular interpolation kernel and zero-motion vectors, PSNR = 32.64 dB, (e) LIMAT with linear interpolation kernel, PSNR = 35.49 dB.* 43

3.6 *Reconstructed anchor frames of* Pendulum + Cars *video. Out of 512 original frames, 16 anchor frames were reconstructed. From the top each row shows the 2nd, 6th, 11th, and 15th anchor frame and its reconstruction via various reconstruction methods. From the left each column represents: (a) original frames, (b) 3D-DWT with rectangular interpolation kernel, PSNR = 27.81 dB, (c) C_n, $n = 32$, with rectangular interpolation kernel, PSNR = 28.30 dB, (d) LIMAT with rectangular interpolation kernel and zero-motion vectors, PSNR = 28.48 dB, (e) LIMAT with linear interpolation kernel, PSNR = 29.64 dB.* 44

4.1 *Multi-view compressive imaging setup. A common scene is observed by J cameras from different positions. Each camera j encodes a small number of random measurements y_j of its observed image x_j, and a single decoder jointly reconstructs all images $\{x_j\}$ from the ensemble of compressive measurements $\{y_j\}$.* 47

4.2 *Lumigraph geometry in compressive multi-view imaging. (a) Flatland-like illustration for collecting 1D images of an object in the 2D plane. At each camera position, all viewing directions may be considered. (b) Resulting 128×128 lumigraph for the ellipse-shaped object. Each row corresponds to a single "image". (In the real world each image is 2D and the full lumigraph is 4D.) The lumigraph can be repeated for viewing from all four sides of the object. (c) Wedgelets provide a simple parametric model for local patches of a 2D lumigraph; only two parameters are needed to describe the orientation and offset of the linear discontinuity. (d) Wedgelet-based lumigraph reconstruction from $M = 5$ compressive samples of each image (row of lumigraph). (e) Scene geometry estimated using local edge positions/orientations in the reconstructed lumigraph. Each blue line connects an estimated point on the object to a camera from which that point is visible. The true ellipse is shown in red.* 54

4.3 *(a) Setup for manifold lifting demonstration. The original image x (courtesy USGS) has size 192×192 and is observed by $J = 200$ satellites. The red boxes illustrate the limited field of view for a few such cameras. (b) Image-by-image reconstruction from random measurements, PSNR 14.4dB. (c) Joint reconstruction using our manifold lifting algorithm with unknown camera positions, PSNR 23.6dB.* 57

4.4 *(a) Initial ISOMAP embedding $v_1, v_2, \ldots, v_J$ of the measurement vectors. (b) Initial estimates $\{\widehat{\theta}_j\}$ of camera positions after rotating and scaling the $\{v_j\}$. (c) Final camera position estimates after running the manifold lifting algorithm. In (b) and (c), the colored points represent the estimated camera positions (color coded by the true θ_j^H value), while the blue vectors represent the error with respect to the true (but unknown) camera position.* 59

4.5 *(a) Scale-by-scale reconstruction of the underlying image proceeding from the coarsest scale s_1 on the left to the finest scale s_5 on the right. (b) The corresponding camera position estimates used in the reconstruction of the images in (a) proceeding from the coarsest scale s_1 on the left to the finest scale s_5 on the right.* 61

4.6 *(a) Reconstruction using fully dense Gaussian random matrix, PSNR 21.9dB. (b) Joint reconstruction using transform coding measurements with known camera positions, PSNR 22.8dB. (c) Joint reconstruction using random measurements with known camera positions, PSNR 24.7dB.* 62

5.1 *(a) Test signal for concentration in a DBD matrix. The main panel plots the energy distribution γ when the signal is partitioned into $J = 16$ blocks of length $N = 64$; the subpanel plots the length-1024 signal x itself. (b),(c) Test signals for concentration in a RBD matrix. The main panels plot the eigenvalue distributions λ for Sig. 1 and Sig. 2, respectively, when partitioned into $J = 16$ blocks of length $N = 64$; the subpanels plot the length-1024 signals themselves.* 73

5.2 *Comparisons of concentration behavior for various matrix and signal types; each figure shows the percentage of trials for which $(1-\epsilon) \leq \|\Phi x\|_2/\|x\|_2 \leq (1+\epsilon)$ as a function of ϵ. (a) Performance of the dense and DBD matrices on the signal shown in Figure 5.1(a). (b) Performance of the dense and RBD matrices on the signals shown in Figure 5.1(b),(c).* 73

5.3 *(a) Histogram of P_D for 10000 signals with uniform energy across blocks (signal class $\mathcal{S}_1$) and for 10000 signals with decaying energy across blocks (signal class $\mathcal{S}_2$) when measured with a DBD matrix. The compressive NP detector has the constraint $P_F \leq \alpha = 0.1$. (b) Plots of the probability of misclassification P_E over a range of values of $M = 1, \cdots, 20$. The first class of signals $\mathcal{S}_1$ are sparse in the frequency domain. The second class of signals $\mathcal{S}_2$ are nonzero only on a single block in the time domain. While P_E decreases with increasing M for both classes of signals, classification performs better for the signals in $\mathcal{S}_1$, which are more amenable to a stable embedding with a DBD matrix.* 81

5.4 *Histograms of the normalized quantity Γ_2 for frequency sparse signals. (a) The distribution of $\frac{\Gamma_2}{M}$ for randomly generated frequency sparse signals of length $N' = N \times J = 64 \times 64$ for sparsity levels $S \in \{5, 30, 64\}$. Note that $\frac{\Gamma_2}{M}$ accumulates near its upper bound of $J = 64$ for all three sparsity levels. (b) The distribution of $\frac{\Gamma_2}{MJ}$ for randomly generated frequency sparse signals with $S = 5$ and the number of signal blocks $J \in \{64, 200, 400\}$. Note that $\frac{\Gamma_2}{MJ}$ accumulates near its upper bound of 1.* 86

5.5 *Plots of the energy distributions of individual videos and of their differences for the best video pair and the worst video pair among all possible $\binom{8}{2}$ possible video pairs. (a) The difference of the video pair, "Bridge close" and "Bridge far", giving the best value of $\Gamma_2(x-y, \mathbf{M})/M = 149.9988$. (b) The difference of the video pair, "Coastguard" and "Miss America", giving the worst value of $\Gamma_2(x-y, \mathbf{M})/M = 148.7550$.* 89

7.1 A plot of $g(T_s) = \sin(|\omega_l - \omega_n|\frac{T_s}{2})$ in black and $f(T_s) = \frac{|\omega_l - \omega_n|T_s}{\pi}$ in red. This plot illustrates that $g(T_s) \geq f(T_s)$ when $T_s \leq \frac{\pi}{|\omega_l - \omega_n|}$. 137

7.2 *Comparison of FDD with CS+FDD and our proposed method SVD(Y). Each mode shape returned by CS+FDD and SVD(Y) is compared against the corresponding mode shape returned by FDD. The result is evaluated by computing the correlation (COR) between a pair of mode shapes. (a) CS+FDD: $COR = 0.9382$, SVD(Y): $COR = 0.9867$ (b) CS+FDD: $COR = 0.8726$, SVD(Y): $COR = 0.9701$, and (c) CS+FDD: $COR = 0.5209$, SVD(Y): $COR = 0.9882$.* 140

8.1 *Frame-by-frame PSNR values of high-rate* Pendulum + Cars *video estimate. This video was obtained by interpolating the 32 anchor frames that were reconstructed with our algorithm. Four of the reconstructed anchor frames are illustrated in Figure 3.4(e). The blue line shows the PSNR values of each frame, and the red squares mark the PSNR values of each anchor frame. The PSNR of the full reconstructed video sequence is* 31.31 *dB, which is only slightly lower than the PSNR of the reconstructed anchor frames.* 142

LIST OF APPENDICES

Appendix

A. Appendix A: Proof of Theorem 5.3 148

B. Appendix B: Proof of Theorem 5.5 150

C. Appendix C: Proof of Theorem 5.9 152

D. Appendix D: Proof of Lemma 5.10 154

ABSTRACT

Compressed Sensing in Multi-Signal Environments

by

Jae Young Park

Technological advances and the ability to build cheap high performance sensors make it possible to deploy tens or even hundreds of sensors to acquire information about a common phenomenon of interest. The increasing number of sensors allows us to acquire ever more detailed information about the underlying scene that was not possible before. This, however, directly translates to increasing amounts of data that needs to be acquired, transmitted, and processed. Compressed sensing (CS) is a novel acquisition and reconstruction scheme that is particularly useful in scenarios when high resolution signals are difficult or expensive to encode. When applying CS in a multi-signal scenario, there are several aspects that need to be considered such as the sensing matrix, the joint signal model, and the reconstruction algorithm. The purpose of this dissertation is to provide a complete treatment of these aspects in various multi-signal environments.

In the application of CS in video, based on the observation that many videos should have limited temporal bandwidth, we propose an algorithm that only involves reconstructing a low rate stream of anchor frames. Together with motion compensated temporal transform we propose a multiscale, iterative algorithm to successfully reconstruct video signals from streaming compressive measurements.

In the application of CS in multi-view imaging, we propose a manifold lifting algorithm for recovering an ensemble of images that describe a common scene. To model the joint signal correlations, we employ a geometric modeling framework in which the image ensemble is treated as a sampling of points from a low-dimensional manifold in the ambient signal space. We propose a multiscale manifold lifting algorithm to simultaneously recover the signal ensemble as well as the unknown camera positions.

In the application of CS in Structural Health Monitoring (SHM) systems, our proposed method ex-

ploits the joint signal structure of vibration data that can be observed through equations describing a simplified Multiple-Degree-Of-Freedom (MDOF) system. Our method is as simple as computing the SVD of the signal matrix, and we evaluate the performance of this method for both uniform and random sampling methods. For each sampling method, we give sufficient conditions on the required sampling rate, the total sampling time span, and the total number of measurements for the accurate recovery of mode shape vectors.

Finally, we study the characteristics of measurement matrices that appear frequently in multi-signal environments. We prove Concentration of Measure (CoM) results for block diagonal matrices and discuss applications of our results to various signal processing tasks. We also show that when each signal in a signal ensemble is measured with the same random matrix, the singular values and the right singular vectors of the underlying signal matrix can be well-preserved in the relative sense.

CHAPTER 1

Introduction

1.1 Multi-Signal Environments

Technological advances and the ability to build cheap high performance sensors make it possible to deploy tens or even hundreds of sensors to acquire information about a common phenomenon of interest. The increasing number of sensors allows us to acquire ever more detailed information about the underlying scene that was not possible before. It is incontrovertible that such multi-sensor environments have already become a ubiquitous part of our lives, and this trend will continue for the years to come.

There are countless applications where multiple sensors or multiple signals of the underlying phenomenon are beneficial or desired. For example, in televised sporting events multiple video cameras filming the game at various viewpoints and angles gives the viewer a dynamic experience. The videos acquired at different viewpoints can also be jointly processed to interpolate between neighboring viewpoints, which allows the viewer to follow the game in more detail. Or, in structural health monitoring systems [121, 49, 92, 98, 148] where the goal is to detect damage in a structure, increasing the number of sensors allows one to extract more detailed and accurate information about the structure. The list goes on with seismic imaging, medical imaging, radar, cognitive radio, and many more applications.

The increasing number of sensors, however, directly translates to increasing amounts of data that needs to be acquired, transmitted, and processed. The amount of data can be overwhelming, especially in applications that involve high-resolution signals such as images or videos. Fortunately, as the signals all describe a common phenomenon or scene there likely exists vast amount of redundancies and/or high correlation among the signals. In order to make the acquisition and processing as efficient and accurate as possible it will be crucial to both remove the redundancies and to exploit

the joint signal correlation in the signal ensemble.

In a typical multi-signal environment, the sensor first acquires data about the underlying phenomenon and then processes, e.g., compresses, the signal locally to reduce the number of measurements that are subsequently transmitted to the central node. For the purpose of encoding the data, there are two types of encoding strategies that one can think of in a multi-signal environment: joint encoding or disjoint encoding strategy. For example, for the sake of maximizing the data compression rate, it may be best to jointly collaborate with other sensors in the network and remove all intra- as well as inter-signal correlations. We refer to such an encoding strategy as a *joint encoding scheme*. However, at the expense of minimizing the number of bits of the data stream, this scheme will require significant resources for the additional communication and processing overhead. On the other hand, encoding techniques that encode their information in a completely disjoint fashion, thus referred to as *disjoint encoding schemes*, will require relatively much less resources but will generally result in a lower compression rate.

After the data have been compressed or processed, it is frequently the case that they are sent to a central node for further, more intense, processing. The user at the central node is then tasked with jointly processing the received signals to further infer any desired information through post-processing, such as estimation, detection, reconstruction, and so on. The main challenge at the decoder is to fully exploit the joint structure among the signals and to maximize the performance of the desired task with the given amount of information. The appropriate model for capturing the joint structure will vary drastically from one application to another and one must choose the model with care.

1.2 Compressed Sensing in Multi-Signal Environment

Compressed sensing (CS) [52, 26] is a novel acquisition and reconstruction scheme that has been a prevalent research topic in the signal processing and computational harmonic analysis communities over the past decade. The theory proposes to directly acquire a compressed version of the underlying signal, bypassing the need for any additional compression schemes. The measurement process is very simple and can be represented by a simple matrix-vector product such as $y = \Phi x$, where $x \in \mathbb{R}^N$ is the underlying signal, Φ an $M \times N$ measurement matrix with $M < N$, and $y \in \mathbb{R}^M$ the "compressed" measurements that we obtain. The compression comes from the fact that Φ has fewer rows than columns.

CS is particularly useful in two scenarios. The first is when a high-resolution signal is difficult to

measure directly. For example, conventional infrared cameras require expensive sensors, and with increasing resolution such cameras can become extremely costly. A compressive imaging camera has been proposed [55] that can acquire a digital image using far fewer (random) measurements than the number of pixels in the image. Such a camera is simple and inexpensive and can be used not only for imaging at visible wavelengths, but also for imaging at nonvisible wavelengths.

A second scenario where CS is useful is when one or more high-resolution signals are difficult or expensive to encode. Such scenarios arise, for example, in sensor networks and multi-view imaging, where it may be feasible to measure the raw data at each sensor, but joint, collaborative compression of that data among the sensors would require costly communication. As an alternative to conventional Distributed Source Coding (DSC) methods [120], a disjoint encoding scheme known as Distributed CS (DCS) [18] has been proposed that extends the idea of single-signal CS. In DCS, each sensor encodes only a random set of linear projections of its own observed signal. These projections could be obtained either by using CS hardware as described above (or below), or by using a random, compressive encoding of the data collected from a conventional sensor.

There are various aspects that need to be considered when applying CS in a multi-signal scenario. The first is the sensing aspect where we need to think about how we can physically acquire the CS measurements or random measurements. Over the past decade numerous CS acquisition devices have been proposed in a wide spectrum of applications. The hardware designs were proposed in applications that include image processing [55, 66, 11, 62, 78], medical imaging [90, 110], seismic imaging [95], cognitive radios [147], and biology [82], to name a few. The main difference between a CS acquisition device and a conventional one is that the former typically needs to blend in a source of randomness in the acquisition process. Additionally, for successful recovery, one also needs to consider whether or not the corresponding measurement matrix is favorable for CS reconstruction, e.g., whether or not the matrix satisfies the Restricted Isometry Property (see Definition 2.3).

Once we know what kind of measurement device to use to obtain CS measurements, the second aspect that we need consider is the choice of a joint signal model. The model of sparsity has been by far the prevalent model for signals in the CS community. To extend this notion of sparsity to multiple signals, researchers have proposed various models such as the joint sparsity model [56], the block-sparse model [58], or the union of subspaces model [59]. Depending on the application these models may be a natural fit but for other applications one needs to design an accurate model for the application at hand. Although CS has initially been proposed with the model of sparsity in mind, we would like to point out that CS can also be applied to non-sparsity based models. As long as the model conveys a notion of conciseness one may be able to adapt CS to the model for reconstruction.

For example, another notion of conciseness of signal ensembles is the broad class of low-dimensional manifolds. A successful application of manifold models has been considered in [135, 136, 105, 33], where [105] is discussed in Chapter 4 of this dissertation.

Once we choose the appropriate model for the joint signal structure, we need to tailor the algorithm that successfully incorporates this model in the reconstruction. There are numerous algorithms that have been proposed in the literature [13, 58, 56, 59, 136, 105] that adapt to the proposed model of inter- and/or intra-signal structure. Again, depending on the application it could be that one can modify an off-the-shelf reconstruction algorithm [26, 52, 102, 129, 21], or it may be that one needs to start from scratch and develop an application specific algorithm.

1.3 Outline and Contributions

In this dissertation, we consider various multi-signal applications. In each application we are interested in the case where there are limited resources available at the encoder and thus we require an efficient, energy saving acquisition scheme. As a suitable encoding strategy we advocate the use of a disjoint encoding scheme via a CS encoding protocol. More specifically, in all applications, representing the signals from J sensors as $x_1, x_2, \ldots, x_J \in \mathbb{R}^N$, we propose that each sensor obtains $y_i = \Phi_i x_i \in \mathbb{R}^M$. The following gives an outline of the dissertation and briefly highlights our contributions in more detail. We begin with a general background of concise signals models and more detailed coverage of the theory of CS in Chapter 2. Then, starting from Chapter 3 we look into specific multi-signal applications where we apply CS.

In Chapter 3, we consider the application of CS to video signals, where each frame x_i is independently encoded, i.e., $y_i = \Phi_i x_i$. Our work could apply to the single-pixel camera [55] or any type of device that is able to acquire streaming CS measurements. We propose a multiscale, iterative algorithm for reconstructing video signals from streaming compressive measurements. Our algorithm is based on the observation that, at the imaging sensor, many videos should have limited temporal bandwidth due to the spatial lowpass filtering that is inherent in typical imaging systems. Under modest assumptions about the motion of objects in the scene, this spatial filtering prevents the temporal complexity of the video from being arbitrarily high. Thus, even though streaming measurement systems may measure a video thousands of times per second, we propose an algorithm that only involves reconstructing a much lower rate stream of anchor frames. Our analysis of the temporal complexity of videos reveals an interesting tradeoff between the spatial resolution of the camera, the speed of any moving objects, and the temporal bandwidth of the video. We lever-

age this tradeoff in proposing a multiscale reconstruction algorithm that alternates between video reconstruction and motion estimation as it produces finer resolution estimates of the video.

In Chapter 4, we consider general multi-view imaging problems in which an ensemble of cameras collect images describing a common scene. Depending on the application, the CS measurements may be acquired via one of the CS cameras discussed above or any other camera that is able to take CS measurements. To capture the joint signal correlations, we propose a geometric modeling framework in which the image ensemble is treated as a sampling of points from a low-dimensional manifold in the ambient signal space. Building on results that guarantee stable embeddings of manifolds under random measurements, we propose a manifold lifting algorithm for recovering the ensemble that can operate even without knowledge of the camera positions. We divide our discussion into two scenarios, the near-field and far- field cases, and describe how the manifold lifting algorithm could be applied to these scenarios. At the end of this chapter, we present an in-depth case study of a far-field imaging scenario, where the aim is to reconstruct an ensemble of satellite images taken from different positions with limited but overlapping fields of view. In this case study, we demonstrate the impressive power of random measurements to capture single- and multi-image structure without explicitly searching for it, as the randomized measurement encoding in conjunction with the proposed manifold lifting algorithm can even outperform image-by-image transform coding.

The main contribution of Chapter 5 is the proof of the concentration of measure (CoM) [87] property of block diagonal (BD) measurement matrices. In the existing CS literature, for certain classes of measurement matrices Φ, the CoM property is an important result that is used to prove stable embedding results such as the Johnson-Lindenstrauss Lemma [81] and the RIP of random matrices. Block diagonal matrices arise frequently in multi-signal scenarios, where the overall system of equations can be written in terms of a BD measurement matrix. In particular, by concatenating all the measurements in one vector $[y_1^T, \ldots, y_J^T]^T$, the overall system of equations can be written as

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_J \end{pmatrix} = \begin{pmatrix} \Phi_1 & & & \\ & \Phi_2 & & \\ & & \ddots & \\ & & & \Phi_J \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_J \end{pmatrix}.$$

We show that the CoM behavior of BD matrices depends on the characteristics of the signals in the ensemble $x_1, \ldots, x_J$. This is in contrast to conventional results of fully populated random matrices, e.g., subgaussian random matrices, that exhibit uniform concentration behavior, independent of the

underlying signal. Our CoM results show that BD matrices with distinct subgaussian blocks, i.e., $\Phi_i \neq \Phi_j$, behave exactly the same as a fully populated subgaussian matrix when $\|x_1\|_2^2 = \cdots = \|x_J\|_2^2$. We also show analogous results for BD matrices that have blocks that are all equal to each other, i.e., $\Phi_i = \Phi_j$. The rest of the chapter discusses applications of our CoM results in ensuring stable embeddings for various signal families and in establishing performance guarantees for solving various signal processing tasks (such as detection and classification) directly in the compressed domain.

In Chapter 6, we consider a streaming data model in which n sensors observe individual streams of data, presented in a turnstile model. Our goal is to analyze the Singular Value Decomposition (SVD) of the matrix of data defined implicitly by the stream of updates. Each column i of the data matrix is given by the stream of updates seen at sensor i. Our approach is to sketch each column of the matrix, forming a "sketch matrix" Y, and then to compute the SVD of the sketch matrix. We show that the singular values and right singular vectors of Y are close to those of X, with small *relative* error.

In Chapter 7, we consider the application of CS in Structural Health Monitoring (SHM) systems. The goal of SHM systems is to detect damage in structures that commonly arises due to natural disasters or continuous use. In SHM, wireless sensors are deployed on structures to measure real-time acceleration data when the structures vibrate. Our goal at the central repository is to infer modal parameters such as modal frequencies, mode shapes, and modal damping ratios from the acquired measurements; these can be used in damage detection algorithms. Our proposed method exploits the joint signal structure of the vibration data that can be observed through equations describing a simplified Multiple-Degree-Of-Freedom (MDOF) system. Our method is as simple as computing the SVD of the signal matrix obtained by stacking each vibration signal into a matrix. We evaluate the performance of this method for both uniform and random sampling methods. For each sampling method, we give sufficient conditions on the required sampling rate, the total sampling time span, and the total number of measurements for the accurate recovery of mode shape vectors.

CHAPTER 2

Background on Signal Models and Compressed Sensing

2.1 Concise Signal Models

Real-world signals typically contain some degree of structure that can be exploited to simplify their processing and recovery. *Sparsity* is one model of conciseness in which the signal of interest can be represented as a linear combination of only a few basis vectors from some dictionary. To provide a more formal statement, let us consider a signal $x \in \mathbb{R}^N$. (If the signal is a 2D image, we reshape it into a length-N vector.) We let $\Psi \in \mathbb{R}^{N \times N}$ denote an orthonormal basis[1] for $\mathbb{R}^N$, with its columns acting as basis vectors, and we write $x = \Psi\alpha$, where $\alpha := \Psi^T x \in \mathbb{R}^N$ denotes the expansion coefficients of x in the basis Ψ. We say that x is K-sparse in the basis Ψ if α contains only K nonzero entries. Sparse representations with $K \ll N$ provide exact or approximate models for wide varieties of signal classes, as long as the basis Ψ is chosen to match the structure in x. In the case of images, the 2D Discrete Wavelet Transform (DWT) and 2D Discrete Cosine Transform (DCT) are reasonable candidates for Ψ [94].

As an alternative to sparsity, *manifolds* have also been used to capture the concise structure of multi-signal ensembles [124, 53, 138, 15, 137]. Simply put, we can view a manifold as a low-dimensional nonlinear surface within $\mathbb{R}^N$. Manifold models arise, for example, in settings where a low-dimensional parameter controls the generation of the signal (see Figure 2.1). Assume, for instance, that $x = x_\theta \in \mathbb{R}^N$ depends on some parameter θ, which belongs to a p-dimensional parameter space that we call Θ.[2] One might imagine photographing some static scene and letting θ correspond to the position of the camera: for every value of θ, there is some N-pixel image x_θ that the camera will see. Supposing that the mapping $\theta \to x_\theta$ is well-behaved, then if we consider all possible

[1] It is also possible to consider other more general non-orthonormal dictionaries.

[2] Depending on the scenario, the parameter space Θ could be a subset of $\mathbb{R}^p$, or it could be some more general topological manifold such as SO(3), e.g., if θ corresponds to the orientation of some object in 3D space.

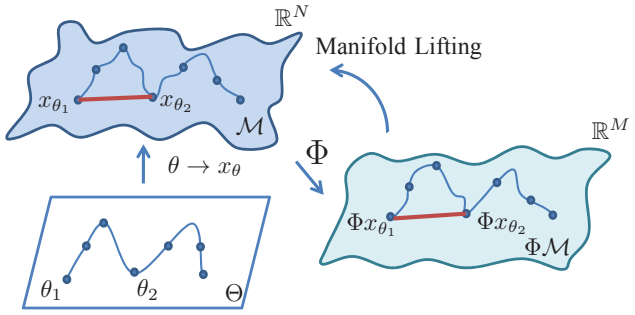


Figure 2.1: *A manifold $\mathcal{M}$ can be viewed as a nonlinear surface in $\mathbb{R}^N$. When the mapping between θ and x_θ is well-behaved, as we trace out a path in the parameter space Θ, we trace out a similar path on $\mathcal{M}$. A random projection Φ from $\mathbb{R}^N$ to a lower dimensional space $\mathbb{R}^M$ can provide a stable embedding of $\mathcal{M}$, preserving all pairwise distances, and therefore preserving the structure within an ensemble of images. The goal of a manifold lifting algorithm is to recover an ensemble of images from their low-dimensional measurements.*

signals that can be generated by all possible values of θ, the resulting set $\mathcal{M} := \{x_\theta : \theta \in \Theta\} \subset \mathbb{R}^N$ will in general correspond to a nonlinear p-dimensional surface within $\mathbb{R}^N$.

When the underlying signal x is an image, the resulting manifold $\mathcal{M}$ is called an *Image Appearance Manifold* (IAM). Recently, several important properties of IAMs have been revealed. For example, if the images x_θ contain sharp edges that move as a function of θ, the IAM is *nowhere differentiable* with respect to θ [53]. This poses difficulties for gradient-based parameter estimation techniques such as Newton's method because the tangent planes on the manifold (onto which one may wish to project) do not exist. However, it has also been shown that IAMs have a multiscale tangent structure [53, 138] that is accessible through a sequence of regularizations of the image, as shown in Figure 2.2. In particular, suppose we define a spatial blurring kernel (such as a lowpass filter) denoted by h_s, where $s > 0$ indicates the *scale* (e.g., the bandwidth or the cutoff frequency) of the filter. Then, although $\mathcal{M} = \{x_\theta : \theta \in \Theta\}$ will not be differentiable, the manifold $\mathcal{M}_s = \{h_s * x_\theta : \theta \in \Theta\}$ of regularized images will be differentiable, where $*$ denotes 2D convolution. Tangent planes do exist on these regularized manifolds $\mathcal{M}_s$, and as $s \to 0$, the orientation of these tangent planes along a given $\mathcal{M}_s$ changes more slowly as a function of θ. In the past, we have used this multiscale tangent structure to implement a coarse-to-fine Newton method for parameter estimation on IAMs [138].

The rich geometrical information that rests within an IAM makes it an excellent candidate for modeling in multi-view imaging. Letting θ represent camera position, all of the images in a multi-view ensemble will live along a common IAM, and as we will later discuss, image reconstruction in the IAM framework can ensure global consistency of the reconstructed images.

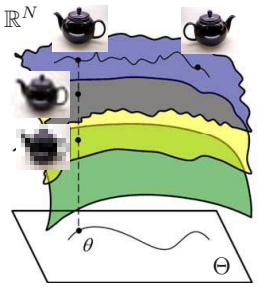


Figure 2.2: *The multiscale structure of manifolds. The top manifold in this figure corresponds to the collection of images of a teapot that could be acquired from different camera positions θ. While manifolds like this containing images with sharp edges are not differentiable, manifolds of images containing smooth images* are *differentiable, and the more one smoothes the images, the smoother the manifold becomes.*

2.2 Compressed Sensing

In conventional signal acquisition devices such as digital cameras and camcorders, we first acquire a full N-dimensional signal x and then apply a compression technique such as JPEG or MPEG [94]. These and other *transform coding* techniques essentially involve computing the expansion coefficients α describing the signal in some basis Ψ, keeping only the K-largest entries of α, and setting the rest to zero. While this can be a very effective way of consolidating the signal information, one could argue that this procedure of "first sample, then compress" is somewhat wasteful because we must measure N pieces of information only to retain $K < N$ coefficients. For certain sensing modalities (such as infrared), it may be difficult or expensive to acquire so many high-resolution samples of the signal.

The recently emerged theory of CS suggests an alternative acquisition scheme. CS utilizes an efficient encoding framework in which we directly acquire a compressed representation of the underlying signal by computing simple linear inner products with a small set of randomly generated test functions. Let us denote the full-resolution discrete signal as $x \in \mathbb{R}^N$ and suppose that we generate a collection of M random vectors, $\phi_i \in \mathbb{R}^N$, $i = 1, 2, \ldots, M$. We stack these vectors into an $M \times N$ matrix $\Phi = [\phi_1 \ \phi_2 \ \cdots \ \phi_M]^T$, which we refer to as a measurement matrix. A CS encoder or sensor produces the measurements $y = \Phi x \in \mathbb{R}^M$, possibly without ever sampling or storing x itself.

At the decoder, given the random measurements y and the measurement matrix Φ, one must attempt to recover x. The canonical approach in CS is to assume that x is sparse in a known basis Ψ and solve an optimization problem of the form [52, 26]

$$\min_{\alpha} \|\alpha\|_1 \text{ s.t. } y = \Phi\Psi\alpha, \tag{2.1}$$

which can be recast as a linear program. When there is bounded noise or uncertainty in the measurements, i.e., $y = \Phi x + n$ with $\|n\|_2 \leq \epsilon$, it is common to solve a similar problem [27]:

$$\min_{\alpha} \|\alpha\|_1 \text{ s.t. } \|y - \Phi\Psi\alpha\|_2 \leq \epsilon, \tag{2.2}$$

which is again convex and can be solved efficiently.

Depending on the measurement matrix Φ, recovery of sparse signals can be provably accurate, even in noise. One condition on Φ that has been used to establish recovery bounds is known as the Restricted Isometry Property (RIP) [30], which requires that pairwise distances between sparse signals be approximately preserved in the measurement space. The following gives a formal definition of the RIP.

Definition 2.1. An $m \times N$ matrix Φ satisfies the restricted isometry property of order $2K$ when there exists a constant $\delta_{2K} > 0$ such that the following holds for all K-sparse signals x_1 and x_2:

$$(1 - \delta_{2K})\|x_1 - x_2\|_2^2 \leq \|\Phi(x_1 - x_2)\|_2^2 \leq (1 + \delta_{2K})\|x_1 - x_2\|_2^2. \tag{2.3}$$

If Φ satisfies the RIP of order $2K$ with δ_{2K} sufficiently small, it is known that (2.1) will perfectly recover any K-sparse signal in the basis Ψ, and that (2.2) will incur a recovery error at worst proportional to ϵ [27]. The performance of both recovery techniques also degrades gracefully if x is not exactly K-sparse but rather is well approximated by a K-sparse signal.

It has been shown that we can obtain an RIP matrix Φ with high probability simply by taking $M = \mathcal{O}(K \log(N/K))$ and populating the matrix with i.i.d. Gaussian, Bernoulli, or more general subgaussian entries [14]. Thus, one of the hallmarks of CS is that this requisite number of measurements M is essentially proportional to the sparsity level K of the signal to be recovered.

In addition to families of K-sparse signals, random matrices can also provide stable embeddings for manifolds (see Figure 2.1). Letting $\mathcal{M}$ denote a smooth[3] p-dimensional manifold, if we take $M = \mathcal{O}(p \log(N))$ and generate Φ randomly from one of the distributions above, we will obtain an embedding $\Phi\mathcal{M} := \{\Phi x : x \in \mathcal{M}\} \in \mathbb{R}^M$ such that all pairwise distances between points on the manifold are approximately preserved [15], i.e., such that (2.3) holds for all $x_{\theta_1}, x_{\theta_2} \in \mathcal{M}$. Geodesic distances are also approximately preserved. Again, the requisite number of measurements is merely proportional to the information level of the signal, which in this case equals p (the dimension of the manifold), rather than the sparsity level of the signal in any particular dictionary. All of this suggests

[3]Although an IAM $\mathcal{M}$ may not itself be smooth, a regularized manifold $\mathcal{M}_s$ will be smooth, and later in this chapter we discuss image reconstruction strategies based on random projections of $\mathcal{M}_s$ at a sequence of scales s.

that manifolds may be viable models to use in CS recovery; see [136] for additional discussion on the topic of using manifold models to recover individual signals.

We see from the above that random measurements have a remarkable "universal" ability to capture the key information in a signal, and this occurs with a number of measurements just proportional to the number of degrees of freedom in the signal. Only the decoder attempts to exploit the signal structure, and it can do so by positing any number of possible signal models.

In summary, in settings where a high-resolution signal x is difficult or expensive to measure directly, CS allows us to replace the "first sample, then compress" paradigm with a technique for directly acquiring compressive measurements of x. To do this in practice, we might resort to CS hardware which directly acquires the linear measurements y without ever sampling or storing x directly. Several forms of compressive imaging architectures have been proposed, ranging from existing data collection schemes in Magnetic Resonance Imaging (MRI) [91] to more exotic CS-based techniques. One architecture [55], for example, replaces the conventional CCD/CMOS sensor in a digital camera with a digital micromirror device (DMD), which modulates the incoming light and reflects it onto a single photodiode for measurement. Some intriguing uses of this inexpensive "single pixel camera" could include infrared or hyperspectral imaging, where conventional high-resolution sensors can cost hundreds of thousands of dollars.

Before proceeding, however, we note that CS can also be useful in settings where it is possible to *acquire* high-resolution signals, but is difficult or expensive to subsequently *encode* them. For example, x might represent a video signal, for which direct measurement is possible, but for which subsequent compression typically requires exploiting complicated spatio-temporal correlations [96, 104]. A more straightforward encoder might simply compute $y = \Phi x$ for some random, compressive Φ. Other scenarios where data is difficult to encode efficiently might be in sensor networks or in multi-view imaging, which is the topic of Chapter 4 and is discussed further in that chapter.

CHAPTER 3

Compressed Sensing in Video Acquisition and Reconstruction

3.1 Introduction

3.1.1 Motivation and Overview

The emerging theory of Compressed Sensing (CS) has inspired a number of efficient new designs for signal acquisition in general and imaging in particular. Architectures such as the "single-pixel camera" [139, 55] provide a promising proof-of-concept that still images can be acquired using small numbers of randomized measurements. Despite the apparently incomplete data collected by such devices, reconstruction of the signal can be accomplished by employing a sparse model specifying, for example, that a high-dimensional image may have only a small number of significant coefficients when expanded in the two-dimensional (2D) wavelet domain [94].

There are numerous applications where it could be helpful to extend the CS imaging framework beyond still images to incorporate video. Standard video capture systems require a complete set of samples to be obtained for each frame, at which point a compression algorithm may be applied to exploit spatial and temporal redundancy. In some applications, such as imaging at non-visible (e.g., infrared) wavelengths, it may be difficult or expensive to obtain these raw samples. In other applications, it could be computationally challenging to implement a state-of-the-art video compression algorithm at the sensor. We argue that these burdens may be reduced by using compressive imaging hardware where random measurements are collected independently from each frame in the video and no additional compression protocol is needed.

This work is in collaboration with Michael B. Wakin [106], and builds upon earlier work that appeared in [104] in collaboration with Michael B. Wakin, and also upon a technical report in [134] by Michael B. Wakin. The theoretical discussions on the bandwidth of video signals presented in this work have initially been discussed in [134].

From a hardware perspective, it is not difficult to envision extending standard compressive imaging architectures to acquire compressive measurements of a video. For example, the single-pixel camera takes random measurements serially in time. Each measurement corresponds to a random linear function of the image on the focal plane at the instant that measurement is collected. When the single-pixel camera is photographing a fixed scene, each measurement corresponds to the same image. When the single-pixel camera is photographing a scene containing motion, each measurement will correspond to a different "frame" of the video. The single-pixel camera is capable of taking many thousands of measurements per second. While we will continue to use the single-pixel camera as a specific example in this chapter, other compressive imaging architectures (such as a CMOS-based transform imager [112] and a coded-aperture imager [97]) could be similarly used to acquire streaming measurements of a video.

From a data processing perspective, though, there are two major reasons why implementing a CS video system may be significantly more difficult than implementing a CS imaging system:

- **Challenge 1:** The complexity of a CS reconstruction algorithm is dependent on the number of unknowns that must be recovered from the compressive samples. The sheer volume of data in a raw video signal makes CS reconstruction a formidable task. As we explain in Section 3.3, this problem is only exacerbated in compressive video systems that collect streaming measurements, where the number of measured frames can be in the thousands per second. Naively reconstructing all of the pixels in all of these frames could literally involve solving for billions of unknowns every second.

- **Challenge 2:** Reconstructing a signal from compressive measurements requires an efficient sparsifying transform and a corresponding algorithm that can promote this sparsity in the reconstructed signal. In the long literature of standard video compression [143] (not video CS), a variety of methods have been proposed to exploit spatial and temporal redundancies. One common approach combines motion compensation and estimation algorithms [79] with image compression techniques; for example, given a set of vectors describing the motion in the video, the LIMAT framework [118] yields a motion-compensated wavelet transform (across the temporal and spatial dimensions) intended to provide a sparse representation of the video. While some of these central ideas can be absorbed into the CS framework, there is an important challenge that we must address. Unlike the standard video compression problem where the frames of the video are explicitly available to perform motion estimation, in CS only random measurements of the underlying video are available. We are faced with a chicken-or-egg prob-

lem [104]: Given the video frames, we could estimate the motion; but given the motion we could better estimate the frames themselves.

In this chapter, we offer suggestions for confronting both of these challenges. We begin in Section 3.2 with a short discussion concerning the temporal bandwidth of video signals. We argue analytically that, at the imaging sensor, many videos should have limited temporal bandwidth due to the spatial lowpass filtering that is inherent in typical imaging systems. Under modest assumptions about the motion of objects in the scene, this spatial filtering prevents the temporal complexity of the video from being arbitrarily high.

We then explain in Section 3.3 how this limited temporal complexity can be exploited in addressing Challenge 1 above. Following standard arguments in sampling theory, we note that under various interpolation kernels, a stream of high-rate video frames (such as those measured by a single-pixel camera) can be represented as a linear combination of a low-rate (e.g., Nyquist-rate) "anchor" set of sampled video frames. We then explain how the CS video problem can be reformulated by setting up a system of linear equations that relate the compressive measurements to the underlying degrees of freedom of the video (specifically, the anchor frames). This significantly reduces the number of unknowns that must be solved for. As we demonstrate, our use of interpolation kernels for reducing the burden of processing streaming measurements can also be much more effective than the traditional technique of partitioning the measurements into short groups and assuming that all measurements within a group come from the same frame. Such raw aggregation of measurements—which actually corresponds to using a rectangular interpolation kernel in our formulation—can introduce significant interpolation error and degrade the reconstruction quality.

Our analysis of the temporal complexity of videos reveals an interesting tradeoff between the spatial resolution of the camera, the speed of any moving objects, and the temporal bandwidth of the video. In Section 3.4, we explain how this tradeoff can be leveraged to address Challenge 2 above. We propose a novel, multiscale algorithm for reconstructing video signals from compressive measurements. Our algorithm begins by reconstructing a coarse-scale approximation to the video, having low spatial and temporal resolution. From this, we obtain a crude estimate of the motion vectors in the video, and we then use these motion vectors to define a sparsifying transform that enables reconstruction of the next-finer scale approximation to the video. Our representation framework is built around the LIMAT [118] method for standard video compression, in which motion compensation is used to improve sparsity in the three-dimensional (3D) wavelet domain. We solve the chicken-or-egg problem by alternating between motion estimation and video reconstruction, pro-

ceeding to higher and higher spatial and temporal resolutions. At each scale, we employ the anchor frames described above to manage the complexity of the reconstruction process.

We conclude in Section 3.5 by describing the differences between our work and several other important ones in the literature. We also present simulations that demonstrate the performance of our algorithm. We stress that to some degree, the algorithm we present in this chapter is a proof-of-concept inspired by our temporal bandwidth analysis. We see our work as an addition to—not a replacement for—the nascent CS video literature, and we believe that the ideas we expound (such as using anchor frames to reduce the complexity of reconstruction) could be combined with other existing ideas in the literature.

3.2 On the Temporal Bandwidth of Video Signals

3.2.1 Setup

3.2.1.1 Signal model and objectives

For the sake of simplicity, we begin by considering "videos" that have just one spatial dimension; we extend this to videos with two spatial dimensions in Section 3.2.4. We also begin by considering continuous-time, continuous-space videos; our analysis will reveal the implications of sampling these videos. We use the variable $t \in \mathbb{R}$ to index time (which we measure in seconds), and we use the variable $x \in \mathbb{R}$ to index spatial position on the focal plane. For convenience, we measure x in an arbitrary real-valued unit we call "pix"; this unit is intended to symbolize what might be the typical pixel size in a subsequent discretization of this video. One could easily replace pix with micrometers or any other arbitrary unit of distance.

We consider videos belonging to a simple but representative translational model. Let $g(x)$ denote a 1D function of space (think of this as a continuous-space "still image"), and consider a continuous-space, continuous-time video $f(x,t)$ in which each "frame" of the video merely consists of a shifted version of this prototype frame. More formally, suppose that

$$f(x,t) = g(x - h(t)),$$

where $h(t)$ is some function that controls how much (in pix) the prototype frame is shifted in the focal plane at each time step. Because we have an interest in video imaging systems with high temporal sampling rates, our purpose is to characterize the temporal bandwidth of the video $f(x,t)$.

In particular, *we suggest that $f(x,t)$ could have limited temporal bandwidth in plausible scenarios*

where the prototype frame $g(x)$ and translation signal $h(t)$ have limited complexity. For example, in a physical imaging system, one may envision $f(x,t)$ as the video that exists at the focal plane prior to being sampled by the imaging sensor. It is reasonable to expect that, due to optical blurring and due to the implicit filtering that occurs from the spatial extent of each light integrator, the prototype frame $g(x)$ will have limited spatial bandwidth. Similarly, if the camera motion is constrained or due to the physics governing the movement of objects in the scene, one might expect that the translation signal $h(t)$ will have limited slope and/or limited temporal bandwidth. In the sections that follow, we explain how such scenarios can allow us to bound the approximate temporal bandwidth of $f(x,t)$.

3.2.1.2 Fourier setup

Let $F(\omega_x, \omega_t)$ denote the 2D Fourier transform of $f(x,t)$, and let $G(\omega_x)$ denote the 1D Fourier transform of $g(x)$; in terms of units, ω_x is measured in rad/pix, and ω_t is measured in rad/s. From the separability property of the 2D Fourier transform and the shift property of the 1D Fourier transform, it follows that

$$F(\omega_x, \omega_t) = G(\omega_x) \cdot L(\omega_x, \omega_t),$$

where

$$L(\omega_x, \omega_t) := \mathcal{F}_t\{e^{-j\omega_x h(t)}\}(\omega_x, \omega_t) \tag{3.1}$$

and $\mathcal{F}_t\{\cdot\}$ denotes an operator that performs a 1D Fourier transform in the temporal direction. That is, for fixed ω_x, $L(\omega_x, \omega_t)$ equals the 1D Fourier transform of $e^{-j\omega_x h(t)}$ with respect to time, evaluated at the frequency ω_t.

3.2.2 Temporal Bandwidth Analysis

The appearance of the $h(t)$ term within an exponent in (3.1) can complicate the task of characterizing the bandwidth of $f(x,t)$. However, by imposing certain assumptions on $h(t)$, this analysis can become tractable.

3.2.2.1 Constant velocity model for $h(t)$

As a starting example, we briefly discuss a "constant velocity" model for $h(t)$ that is commonly seen in textbook discussions of video bandwidth (see, e.g., [142]). We assume that $h(t) = \Gamma t$ for some constant Γ (having units of pix/s). In this case we have $L(\omega_x, \omega_t) = \delta(\omega_t + \omega_x \Gamma)$, and so

$F(\omega_x, \omega_t) = G(\omega_x) \cdot \delta(\omega_t + \omega_x \Gamma)$, which corresponds to a diagonal line in the 2D Fourier plane with slope ($\Delta\omega_t$ over $\Delta\omega_x$) that depends linearly on Γ.

To see the implications of this in terms of bandwidth, suppose that $G(\omega_x)$ is bandlimited (or essentially bandlimited) to some range of frequencies $\omega_x \in [-\Omega_x, \Omega_x]$ rad/pix. (Again, even if the moving object has sharp edges, $G(\omega_x)$ may be bandlimited due to blurring in the imaging system.) In this case, it follows that $F(\omega_x, \omega_t)$ must be bandlimited (or essentially bandlimited) to the range of frequencies $(\omega_x, \omega_t) \in [-\Omega_x, \Omega_x] \times [-\Gamma\Omega_x, \Gamma\Omega_x]$. In other words, the temporal bandwidth of the video is no greater than $\Gamma\Omega_x$ rad/s.

3.2.2.2 Bounded velocity model for $h(t)$

We now consider a more robust "bounded velocity" model for $h(t)$; we note that similar mathematics have appeared in the analysis of plenoptic [48] and plenacoustic [7, 8] functions, which arise from measuring light and sound, respectively, at various positions in a room. We assume that the position function $h(t)$ has bounded slope, i.e., that for some $\Gamma > 0$, $|dh(t)/dt| \leq \Gamma$ pix/s for all t. This corresponds to a bound on the speed at which the object can move in the video, without requiring that this speed be constant. We also assume that the translation signal $h(t)$ is bandlimited, with bandwidth given by Ω_h rad/s.

For any fixed ω_x, we can recognize $e^{-j\omega_x h(t)}$ as a frequency-modulated (FM) sinusoid [149] having carrier frequency 0 and instantaneous frequency (in rad/s) $\omega_i(t) = -\omega_x \cdot dh(t)/dt$. Let us also define the deviation term

$$D := \frac{|\omega_x|}{\Omega_h} \max \left| \frac{dh(t)}{dt} \right|.$$

From Carson's bandwidth rule for frequency modulation [149], we have that for fixed ω_x, at least 98% of the total power of $e^{-j\omega_x h(t)}$ must be concentrated in the frequency range $\omega_t \in [-2(D+1)\Omega_h, 2(D+1)\Omega_h]$ rad/s. Since $D \leq \frac{|\omega_x|\Gamma}{\Omega_h}$, we conclude that at least 98% of the total power of $e^{-j\omega_x h(t)}$ must be concentrated in the frequency range $\omega_t \in [-(2|\omega_x|\Gamma + 2\Omega_h), 2|\omega_x|\Gamma + 2\Omega_h]$ rad/s. We note that the dependence of this bandwidth on ω_x is essentially linear.

We conclude that $L(\omega_x, \omega_t)$ will have a characteristic "butterfly shape" with most of its total power concentrated between two diagonal lines that intercept the ω_t-axis at $\pm 2\Omega_h$ and have slope approximately $\pm 2\Gamma$. This shape is illustrated in Figure 3.1(a). Though not shown, the corresponding figure for the constant velocity model discussed in Section 3.2.2.1 would involve a single diagonal line intersecting the origin and having slope $-\Gamma$ (which is half as large as the slope that appears in our more general bounded velocity analysis).

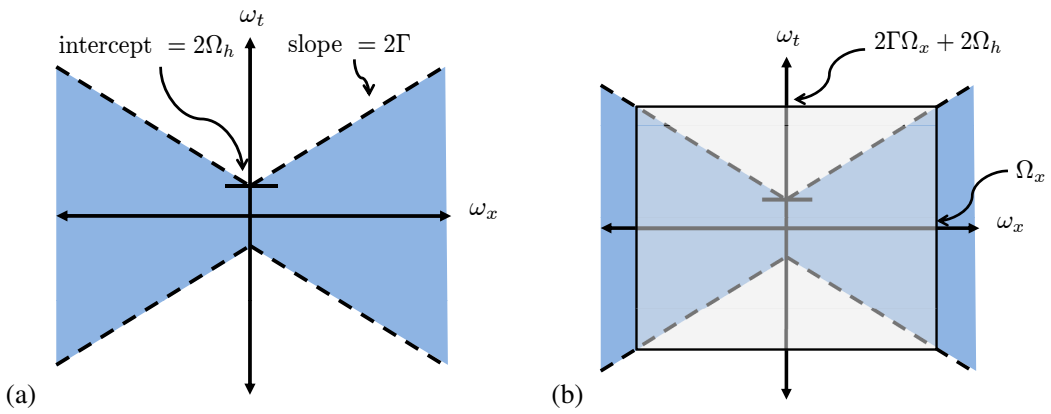


Figure 3.1: *(a) Butterfly structure in the two-dimensional Fourier transform of a simple translational 1D video. The slope of the lines is proportional to the maximum speed of the translation. (b) Bandlimiting the video in space (e.g., by spatial lowpass filtering) will also bandlimit the video in time. The resulting temporal bandwidth will be proportional to the spatial bandwidth times the maximum translational speed.*

To see the implications of this in terms of bandwidth, let us again suppose that $G(\omega_x)$ is bandlimited (or essentially bandlimited) to the range of frequencies $\omega_x \in [-\Omega_x, \Omega_x]$. We must then have that $F(\omega_x, \omega_t) = G(\omega_x) \cdot L(\omega_x, \omega_t)$ is also (essentially) bandlimited in the spatial direction to the range of frequencies $\omega_x \in [-\Omega_x, \Omega_x]$. Because of the butterfly structure in $L(\omega_x, \omega_t)$, however, this will also cause $F(\omega_x, \omega_t)$ to be (essentially) bandlimited in the temporal direction to the range of frequencies

$$\omega_t \in [-(2\Omega_x\Gamma + 2\Omega_h), 2\Omega_x\Gamma + 2\Omega_h] \text{ rad/s}. \tag{3.2}$$

This fact, which is illustrated in Figure 3.1(b), exemplifies a central theme of our work: *filtering a video in the spatial direction can cause it to be bandlimited both in space and in time.* Once again, we note that similar conclusions were reached in the analysis of plenoptic [48] and plenacoustic [7, 8] functions.

3.2.2.3 Sampling implications

From a classical (not compressive) sampling perspective, the Nyquist theorem and the temporal bandwidth predicted in (3.2) suggest that in order to avoid aliasing, the video should be sampled at a minimum rate of $\frac{2\Omega_x\Gamma+2\Omega_h}{\pi}$ samples/s. Let us plug in some plausible numbers to illustrate the implications of these bounds. First, consider the spatial bandwidth Ω_x of the prototype frame. In a reasonable imaging system, we might expect the pixel size to be balanced with the spatial bandwidth of the frame so that spatial aliasing is avoided. (This should occur naturally if we assume each pixel

integrates spatially over a window of size approximately 1 pix.) Thus, one might anticipate that Ω_x will be on the order of π rad/pix. (This corresponds to a spatial bandwidth of $\frac{\pi}{2\pi} = \frac{1}{2}$ cycles/pix, which suggests a spatial Nyquist sample rate of one sample per pix.)

Under the assumption that $\Omega_x = \pi$, (3.2) suggests the video will have temporal bandwidth limited to approximately $2\pi\Gamma + 2\Omega_h$ rad/s. We note that Ω_h, the temporal bandwidth of $h(t)$, does not depend on the amplitude or slope of $h(t)$, but only on its shape and smoothness. The term Γ, in contrast, increases with the amplitude or slope of $h(t)$, which in turn could increase for objects closer to the camera. We conjecture that in practice (aside from exceptionally non-smooth motions $h(t)$), the $2\pi\Gamma$ term will typically dominate the $2\Omega_h$ term, and therefore *in general a temporal Nyquist sampling rate of roughly* 2Γ *samples/s should suffice to avoid temporal aliasing.* Stated differently, to avoid temporal aliasing we should not allow a moving object to traverse more than $\approx \frac{1}{2}$ pix between adjacent sampling times. While this of course makes strong intuitive sense, we have arrived at this conclusion through a principled analysis—one that illustrates the direct relationship between the speed of object motion in the video and the video's overall temporal bandwidth.

Again, to be clear, the paragraphs above concern the implications of our analysis in classical (not compressive) sampling of a video. As we discuss in Section 3.3, the streaming compressive measurements produced by a single-pixel camera may actually need to be acquired much faster than the video's temporal Nyquist rate (because only one measurement is collected from each frame). The specific reason that the temporal bandwidth of the video will be relevant is because it will impact the spacing of the "anchor frames" that we use to reduce the complexity of the reconstruction problem.

3.2.3 Supporting Experiments

3.2.3.1 Within our model assumptions

As an illustration, we can analytically define a continuous-space, continuous-time video that allows us to test our predictions about spectral support. We let the prototype function $g(x) = \text{sinc}\left(\frac{\Omega_x x}{\pi}\right)$, where $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ and $\Omega_x = \pi$ rad/pix. This definition ensures that $g(x)$ is bandlimited and that its bandwidth equals precisely Ω_x rad/pix. We let the motion signal

$$h(t) = \sum_{i=1}^{5} a_i \text{sinc}\left(\frac{\Omega_h(t - d_i)}{\pi}\right), \tag{3.3}$$

where Ω_h controls the total bandwidth (we set $\Omega_h = 15$ rad/s), the delays d_i are chosen randomly, and the amplitudes a_i are chosen somewhat arbitrarily but ensure that the maximum value attained

by $|h(t)|$ equals some parameter Γ, which we set to 25 pix/s. (By changing Ω_h and Γ, we can independently articulate the bandwidth and the maximum slope of this signal.) Figure 3.2(a) shows the video $f(x,t)$.

We oversample this video compared to its predicted spatial and temporal bandwidths, and in Figure 3.2(b) we show the approximate spectrum using the FFT.[1] The blue lines in Figure 3.2(b) indicate the predicted "butterfly shape" which should bound the nonzero support of $F(\omega_x, \omega_t)$. We see that the empirical spectrum does largely concentrate within this region. For this video, the temporal bandwidth is predicted not to exceed $2\Omega_x\Gamma + 2\Omega_h \approx 187$ rad/s.

In other experiments, we have observed that as we vary the bandwidth and velocity parameters, the approximate support of the estimated spectrum stays within the "butterfly shape" predicted by our theory in Section 3.2.2.2. For the sake of space, these experiments are omitted from the current manuscript; however, they are available in a companion technical report [134]. For several videos, we have computed the empirical temporal bandwidth based on our estimated spectrum. To do this, we determine the value of Ω_t for which 99.99% of the energy in the FFT (or windowed FFT) falls within the range $|\omega_t| \leq \Omega_t$. In each case, the empirical bandwidth Ω_t equals roughly half of the bandwidth $2\Omega_x\Gamma + 2\Omega_h$ predicted by our theory. (There are occasional exceptions where the unwindowed FFT gives a higher estimate, but this is likely due to sampling artifacts.) This suggests that in some cases, the bandwidth prediction based on Carson's bandwidth rule may be pessimistic.

3.2.3.2 Beyond our model assumptions

Our formal analysis and the experiments described in Section 3.2.3.1 pertain specifically to translational videos in which $g(x)$ is bandlimited, $h(t)$ has bounded velocity and bandwidth, and the entire contents of the frame translate *en masse*. However, real world videos may contain objects whose appearance (neglecting translation) changes over time, objects that move in front of a stationary background, multiple moving objects, and so on. We suspect that as a general rule of thumb, the temporal bandwidth of real world videos will be dictated by the same tradeoffs of spatial resolution and object motion that our theory suggests. In particular, the prediction of $2\Omega_x\Gamma + 2\Omega_h$ given by our theory may be approximately correct, if we let Ω_x be the essential spatial bandwidth of the imaging system, Γ be the maximum speed of any object moving in the video, and Ω_h be the essential bandwidth of any object motion. This last parameter is perhaps the most difficult to predict for a given video, but we suspect that in many cases its value will be small and thus its role minor in

[1] All spectral plots in Figure 3.2 show the magnitude of the FFT on a $\log_{10}$ scale. In panels (d) and (f) we apply a smooth Blackman-Harris window to the samples before computing the FFT; this helps remove artifacts from the borders of the sampling region.

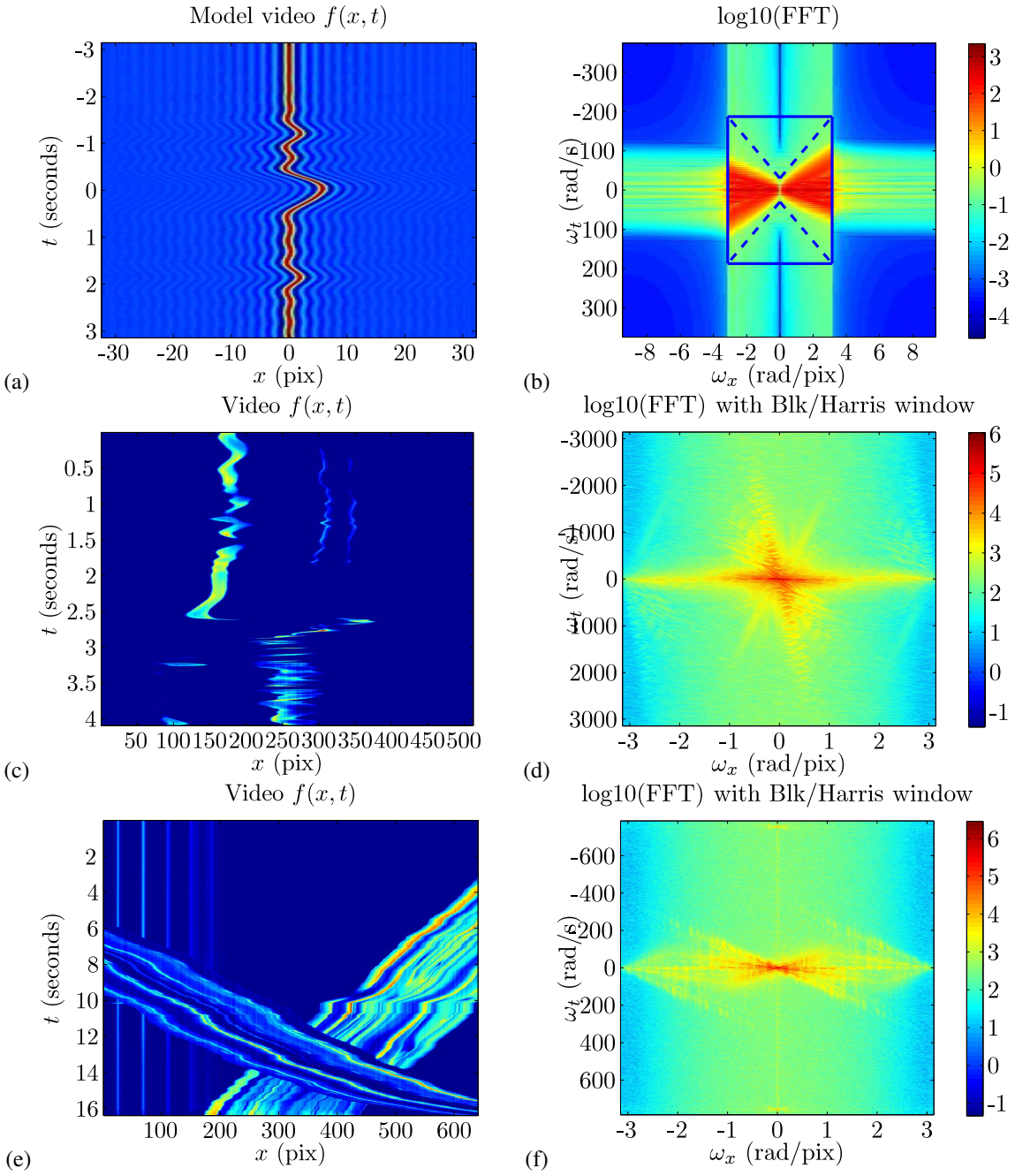


Figure 3.2: *Top row: (a) Video $f(x,t) = g(x - h(t))$ with bandlimited profile $g(x)$ and bandlimited translation signal $h(t)$. (b) Estimated spectrum. Middle row: (c) 4096 sampled 1D rows from* Candle *video. (d) Estimated spectrum with windowing to alleviate border artifacts. Bottom row: (e) 4096 sampled 1D rows from* Pendulum + Cars *video. (f) Estimated spectrum with windowing to alleviate border artifacts.*

determining the overall temporal bandwidth.

To support these conjectures, we have identified the characteristic "butterfly shape" in experiments where $g(x)$ is not bandlimited (e.g., a Gaussian bump convolved with the unit-step function), where $h(t)$ is not bandlimited (e.g., a triangle wave), where there are multiple moving edges, and where there are occluding objects. Again, these experiments are available in a companion technical report [134]. In general, the estimated spectra continue to follow the predicted butterfly shape. However, it remains an open problem to support this with theoretical analysis. We do note that we believe our results are largely consistent with the classical study by Dong and Atick concerning the statistics of real-world videos [50]. Although the videos in that study had relatively low temporal resolution, Dong and Atick did note a certain radial symmetry to the spectrum (with one term depending on ω_t/ω_x) and observe that at low spatial frequencies the power spectrum will have a strong decay as a function of the temporal frequency.

We conclude this examination with our own series of experiments on real-world videos. These videos (courtesy of MERL) were collected in a laboratory setting using a high-speed video camera, but the scenes being imaged contained natural (not particularly high-speed) motions. For each video, we select a 2D "slice" of the 3D video cube, extracting one spatial dimension and one temporal dimension.

We begin with the *Candle* video which features two candle flames in front of a dark background; the video was acquired at a rate of 1000 frames per second. We select 4096 consecutive time samples from the video, and we extract 512 pixels from a certain row in each frame. Figure 3.2(c) shows the video, and Figure 3.2(d) shows the estimated spectrum. Again, we recognize the approximate butterfly shape and an approximate limitation to the video bandwidth, both spatially and temporally. More specifically, we see a collection of lines having various slopes, with the lines passing roughly through the origin. However, there is also a bit of "thickness" near the origin due to a possible Ω_h term. The slopes of these lines match what might be expected based on an empirical examination of the video itself. For example, the fastest motion in the video appears to occur at roughly $t = 2.6$ s, where the candle flame translates to the right with a speed of roughly 1500 pix/s. Consequently, we see a portion of the estimated spectrum oriented along a line with slope of approximately 4000 pix/s. Overall, for this video the empirical temporal bandwidth (in this case, the value of Ω_t for which 99% of the energy in the windowed FFT falls within the range $|\omega_t| \leq \Omega_t$) equals 348 rad/s. This suggests that the video's temporal sampling rate (1000 frames/s) may have been higher than necessary.

Next, we consider the *Pendulum + Cars* video, featuring two translating cars and an occlusion

as one car passes in front of the other; the video was acquired at a rate of 250 frames per second. We select 4096 consecutive time samples from the video, and we extract 640 pixels from a certain row in each frame. Figure 3.2(e) shows the video, and Figure 3.2(f) shows the estimated spectrum. Once again, we recognize the approximate butterfly shape and an approximate limitation to the video bandwidth, both spatially and temporally, and once again, we see a collection of lines with various slopes, but with a bit of "thickness" near the origin due to a possible Ω_h term. The slopes of these lines match what might be expected based on an empirical examination of the video itself. For example, the maximum slope appears to be on the order of 140 pix/s, while the maximum translational speed of the cars appears to be on the order of 70 pix/s. The overall empirical temporal bandwidth is less than 35 rad/s, and consequently, this video's temporal sampling rate (250 frames/s) may also have been higher than necessary.

3.2.4 Videos with Two Spatial Dimensions

3.2.4.1 Signal model

Our analysis is easily generalized to the more conventional case of videos having two spatial dimensions. We again use the variable $t \in \mathbb{R}$ to index time in seconds, and we use the variables $x, y \in \mathbb{R}$ to index spatial position on the focal plane in pix. We again consider videos belonging to a simple but representative translational model. Let $g(x, y)$ denote a 2D function of space (one can think of this as a continuous-space "still image"), and consider a continuous-space, continuous-time video $f(x, y, t)$ in which each "frame" of the video consists of a shifted version of this prototype frame. More formally, suppose that $f(x, y, t) = g(x - h_x(t), y - h_y(t))$, where $h(t) = (h_x(t), h_y(t))$ is a function that controls how much (in pix) the prototype frame is shifted in the x- and y-directions at each time step.

3.2.4.2 Fourier setup

Let $F(\omega_x, \omega_y, \omega_t)$ denote the 3D Fourier transform of $f(x, y, t)$, and let $G(\omega_x, \omega_y)$ denote the 2D Fourier transform of $g(x, y)$. Using similar analysis to the above, we will have

$$F(\omega_x, \omega_y, \omega_t) = G(\omega_x, \omega_y) \cdot L(\omega_x, \omega_y, \omega_t),$$

where $L(\omega_x, \omega_y, \omega_t) := \mathcal{F}_t\{e^{-j\omega_x h_x(t) - j\omega_y h_y(t)}\}(\omega_x, \omega_y, \omega_t)$. For fixed ω_x, ω_y, $L(\omega_x, \omega_y, \omega_t)$ equals the 1D Fourier transform of $e^{-j\omega_x h_x(t) - j\omega_y h_y(t)}$ with respect to time, evaluated at the frequency ω_t.

3.2.4.3 Temporal bandwidth analysis

We assume that the position functions $h_x(t)$ and $h_y(t)$ have bounded slope, i.e., that for some $\Gamma_x, \Gamma_y > 0$, $|dh_x(t)/dt| \leq \Gamma_x$ pix/s and $|dh_y(t)/dt| \leq \Gamma_y$ pix/s for all t. This corresponds to a bound on the "speed" at which the object can move in each direction. We also assume that both translation signals $h_x(t)$ and $h_y(t)$ have bandwidths bounded by Ω_h rad/s.

Using arguments that parallel our 1D analysis (see [134]), we conclude that $L(\omega_x, \omega_y, \omega_t)$ will have a characteristic "polytope hourglass shape." Considering the first octant of the 3D frequency space (in which $\omega_x, \omega_y, \omega_t \geq 0$), most of the total power of $L(\omega_x, \omega_y, \omega_t)$ will fall below (in the temporal direction) a plane passing through the points $(0, 0, 2\Omega_h)$, $(1, 0, 2\Gamma_x + 2\Omega_h)$, and $(0, 1, 2\Gamma_y + 2\Omega_h)$. The other seven octants follow symmetrically.

To see the implications of this in terms of bandwidth, suppose that $G(\omega_x, \omega_y)$ is bandlimited (or essentially bandlimited) to the range of frequencies $(\omega_x, \omega_y) \in [-\Omega_x, \Omega_x] \times [-\Omega_y, \Omega_y]$. We must then have that $F(\omega_x, \omega_y, \omega_t) = G(\omega_x, \omega_y) \cdot L(\omega_x, \omega_y, \omega_t)$ is also essentially bandlimited in the spatial direction to the range of frequencies $(\omega_x, \omega_y) \in [-\Omega_x, \Omega_x] \times [-\Omega_y, \Omega_y]$. Because of the hourglass structure in $L(\omega_x, \omega_y, \omega_t)$, however, this will also cause $F(\omega_x, \omega_y, \omega_t)$ to be essentially bandlimited in the temporal direction to the range of frequencies

$$\omega_t \in [-(2\Omega_x\Gamma_x + 2\Omega_y\Gamma_y + 2\Omega_h), 2\Omega_x\Gamma_x + 2\Omega_y\Gamma_y + 2\Omega_h]. \tag{3.4}$$

Therefore, we see that filtering such a video in the spatial directions can cause it to be essentially bandlimited both in space and in time.

From a classical (not compressive) sampling perspective, if we expect that both Ω_x and Ω_y will be on the order of π rad/pix, and if we assume that the $2\Omega_x\Gamma_x + 2\Omega_y\Gamma_y$ term will typically dominate the $2\Omega_h$ term, then in typical scenarios, to avoid temporal aliasing we should not allow a moving object to traverse more than $\approx \frac{1}{2}$ pix in any direction between adjacent sampling times. Aside from exceptionally non-smooth motions $h(t)$, we do strongly suspect that the influence of the temporal bandwidth Ω_h will be minor in comparison to the influence of the $2\Omega_x\Gamma_x + 2\Omega_y\Gamma_y$ term, and so *in general a temporal Nyquist sampling rate of* $2(\Gamma_x + \Gamma_y)$ *samples/s will likely serve as a reasonable rule of thumb.* Again, this rule of thumb illustrates the direct relationship between the speed of object motion in the video and the video's overall temporal bandwidth. As we will see in Section 3.3, this bandwidth will impact the spacing of the anchor frames that we use to reduce the complexity of the CS reconstruction problem.

3.3 Anchor Frames for Reducing Reconstruction Complexity

The insight we have developed in Section 3.2 suggests that many videos of interest may indeed be exactly or approximately bandlimited in the temporal direction. For problems involving CS of such videos, this implies that there may be a limit to the "complexity" of the information collected by streaming compressive measurement devices. One way of exploiting this limited complexity draws from classical interpolation identities for bandlimited signals. We briefly review these identities in Section 3.3.1 before exploring their applications for CS reconstruction in Section 3.3.2.

3.3.1 Sampling and Interpolation Principles

Before considering 2D or 3D video signals, let us first review the basic principles involved in non-compressive sampling and interpolation of a bandlimited 1D signal. Suppose that $f(t)$ is a signal with temporal bandwidth bounded by Ω_t rad/s. The Nyquist theorem states that this signal can be reconstructed from a discrete set of samples $\{f(nT_s)\}_{n\in\mathbb{Z}}$, where the sampling interval $T_s \leq \frac{\pi}{\Omega_t}$ seconds. In particular, it holds that

$$f(t) = \sum_{n\in\mathbb{Z}} f(nT_s)\text{sinc}\left(\frac{t-nT_s}{T_s}\right). \tag{3.5}$$

Instead of actually *reconstructing* the continuous-time signal $f(t)$, a more important consequence of (3.5) for us will be the fact that, for any $t_0 \in \mathbb{R}$, $f(t_0)$ can be *represented* as a linear combination of the discrete samples $\{f(nT_s)\}_{n\in\mathbb{Z}}$.

With varying degrees of approximation, it is possible to replace the sinc interpolation kernel in (3.5) with other, more localized kernels. We will write

$$f(t) \approx \sum_{n\in\mathbb{Z}} f(nT_s)\gamma\left(\frac{t-nT_s}{T_s}\right), \tag{3.6}$$

where $\gamma(t)$ is a prototype interpolation kernel. In addition to the sinc kernel, for which $\gamma(t) = \text{sinc}\,(t)$, other possible choices include the zero-order hold (rectangular) kernel, for which $\gamma(t) = 1$ when $|t| \leq \frac{1}{2}$ and $\gamma(t) = 0$ otherwise, the first-order hold (triangular, or "linear interpolation") kernel, for which $\gamma(t) = 1 - |t|$ when $|t| \leq 1$ and $\gamma(t) = 0$ otherwise, and a variety of cubic interpolation kernels [100].

In general, a smoother choice for $\gamma(t)$ will better approximate the ideal sinc kernel. However, smoother kernels tend to have wider temporal supports, and it can be desirable in some applications (such as the CS recovery problem discussed below) to limit the temporal support of the kernel.

One way to improve the performance of the lower-order, more narrow interpolation kernels is to decrease the sampling interval T_s. However, for our CS recovery problem discussed below, this too will increase the complexity of the recovery algorithm by increasing the number of anchor frames.

For a 2D or 3D video with limited temporal bandwidth, the separability of the Fourier transform implies that the interpolation formulas presented above should hold for each spatial location (i.e., for each pixel). Using the videos discussed in Section 3.2.3, we have confirmed this in experiments evaluating the quality of interpolation as a function of the video properties, interpolation kernel, etc. Again, these experiments are available in a companion technical report [134].

3.3.2 CS in Streaming Scenarios

3.3.2.1 Measurement process

To set up the CS problem, consider a continuous-space, continuous-time video $f(x, y, t)$. Let $f_d : \{1, 2, \ldots, N\} \times \mathbb{R} \to \mathbb{R}$ denote a sampled discrete-space, continuous-time version of this video, where for $p = 1, 2, \ldots, N$,

$$f_d(p, t) = f(x_p, y_p, t). \tag{3.7}$$

In the expression above, N specifies the number of pixels in each sampled frame, and (x_p, y_p) represents the spatial location of pixel number p. Typically, $\{(x_p, y_p)\}_{p=1}^N$ will form a 2D grid of points in the focal plane. For notational convenience, we rasterize these pixel values and index spatial position in f_d using only the pixel number p.

We consider an imaging system that collects streaming measurements of the video $f(x, y, t)$ according to the following model. We let T denote a measurement interval (in seconds), and we suppose that one linear measurement is collected from f_d every T seconds. (As we will discuss below, typically T will be much smaller than the Nyquist sampling interval suggested by the video's bandwidth.) Letting $y(m)$ denote the measurement collected at time mT, we can write

$$y(m) = \sum_{p=1}^{N} \phi_m(p) f_d(p, mT) = \langle \phi_m, f_d(:, mT) \rangle, \tag{3.8}$$

where $\phi_m \in \mathbb{R}^N$ is a vector of random numbers (see Section 3.4.2 for details on the measurement vectors we prescribe), and we use "Matlab notation" to refer to a vector $f_d(:, t) \in \mathbb{R}^N$. In total, we

suppose that M measurements are collected. Stacking all of the measurements, we have

$$y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(M) \end{bmatrix} = \begin{bmatrix} \langle \phi_1, f_d(:,T) \rangle \\ \langle \phi_2, f_d(:,2T) \rangle \\ \vdots \\ \langle \phi_M, f_d(:,MT) \rangle \end{bmatrix} = \underbrace{\begin{bmatrix} \phi_1^T & & & \\ & \phi_2^T & & \\ & & \ddots & \\ & & & \phi_M^T \end{bmatrix}}_{\Phi:\ M \times MN} \underbrace{\begin{bmatrix} f_d(:,T) \\ f_d(:,2T) \\ \vdots \\ f_d(:,MT) \end{bmatrix}}_{x_d:\ MN \times 1}. \tag{3.9}$$

Stacking the raw frames into a vector $x_d = [f_d(:,T)^T \ f_d(:,2T)^T \ \cdots \ f_d(:,MT)^T]^T \in \mathbb{R}^{MN}$, and letting Φ denote the $M \times MN$ block diagonal measurement matrix appearing in (3.9), we can write $y = \Phi x_d$. This appears to be a standard CS problem, with measurements on the left, unknowns on the right, and a measurement matrix that relates the two. Unfortunately, this is a very difficult CS problem to solve directly: first, the recovery problem is very highly underdetermined, with the number of measurements representing only $\frac{1}{N}$ times the number of unknowns; and second, the size of the recovery problem, with MN unknowns, can be immense. We stress that the frames $f_d(:,T), f_d(:,2T), \ldots, f_d(:,MT)$ are assumed to be very close together in time (perhaps less than a thousandth of a second apart), and so x_d gets very large very quickly.

3.3.2.2 Simplifying the linear equations

Fortunately, if we assume that the video $f(x,y,t)$ has limited temporal bandwidth, we can simplify this recovery process to some degree. Let us assume that $f(x,y,t)$ has temporal bandwidth bounded by Ω_t rad/s. We note that the temporal bandwidth of f_d will also be bounded by Ω_t rad/s.

Let T_a denote a time interval no greater than the Nyquist limit ($\frac{\pi}{\Omega_t}$ seconds) suggested by the video's bandwidth, and assume that $T_a = VT$ for some integer $V \geq 1$. Then for any integer j, we can apply (3.6) and write

$$\begin{aligned} f_d(:,jT) &\approx \sum_{n \in \mathbb{Z}} f_d(:,nT_a) \gamma\left(\frac{jT - nT_a}{T_a}\right) = \sum_{n \in \mathbb{Z}} f_d(:,nT_a) \gamma\left(\frac{j}{V} - n\right) \\ &= \left[\cdots \ \gamma\left(\frac{j}{V} - 1\right) I_N \ \gamma\left(\frac{j}{V} - 2\right) I_N \ \gamma\left(\frac{j}{V} - 3\right) I_N \ \cdots\right] \begin{bmatrix} \vdots \\ f_d(:,T_a) \\ f_d(:,2T_a) \\ f_d(:,3T_a) \\ \vdots \end{bmatrix}, \end{aligned}$$

where I_N denotes the $N \times N$ identity matrix. Therefore,

$$\underbrace{\begin{bmatrix} f_d(:,T) \\ f_d(:,2T) \\ \vdots \\ f_d(:,MT) \end{bmatrix}}_{x_d} \approx \underbrace{\begin{bmatrix} \cdots & \gamma\left(\frac{1}{V}-1\right) I_N & \gamma\left(\frac{1}{V}-2\right) I_N & \cdots \\ \cdots & \gamma\left(\frac{2}{V}-1\right) I_N & \gamma\left(\frac{2}{V}-2\right) I_N & \cdots \\ & \vdots & \vdots & \vdots \\ \cdots & \gamma\left(\frac{M}{V}-1\right) I_N & \gamma\left(\frac{M}{V}-2\right) I_N & \cdots \end{bmatrix}}_{\Gamma} \underbrace{\begin{bmatrix} \vdots \\ f_d(:,T_a) \\ f_d(:,2T_a) \\ \vdots \end{bmatrix}}_{x_a}. \tag{3.10}$$

Assuming $\gamma(t)$ has temporal support within some reasonable bound, the matrix Γ above will have size $MN \times (\frac{MN}{V} + O(1))$ and so this allows a dimensionality reduction by a factor of V. In other words, using the interpolation matrix Γ, we can relate the raw video frames (acquired T seconds apart) to a reduced set of what we call *anchor frames*, which are spaced T_a seconds apart. In the equation above, the stack of anchor frames is denoted by x_a, and the length of x_a is approximately V times smaller than the length of x_d.

Putting all of this together, we have

$$y = \Phi x_d \approx \Phi\Gamma x_a, \tag{3.11}$$

where

$$\Phi\Gamma = \underbrace{\begin{bmatrix} \cdots & \gamma\left(\frac{1}{V}-1\right)\phi_1^T & \gamma\left(\frac{1}{V}-2\right)\phi_1^T & \gamma\left(\frac{1}{V}-3\right)\phi_1^T & \cdots \\ \cdots & \gamma\left(\frac{2}{V}-1\right)\phi_2^T & \gamma\left(\frac{2}{V}-2\right)\phi_2^T & \gamma\left(\frac{2}{V}-3\right)\phi_2^T & \cdots \\ & \vdots & \vdots & \vdots & \\ \cdots & \gamma\left(\frac{M}{V}-1\right)\phi_M^T & \gamma\left(\frac{M}{V}-2\right)\phi_M^T & \gamma\left(\frac{M}{V}-3\right)\phi_M^T & \cdots \end{bmatrix}}_{M \times (\frac{MN}{V}+O(1))}.$$

We have arrived at a CS problem in which the total number of unknowns has been reduced from MN to $\frac{MN}{V} + O(1)$. Indeed, the largest dimension of any matrix or vector in this formulation is now limited to $\frac{MN}{V} + O(1)$ instead of MN. Moreover, due to decay in γ, the matrix $\Phi\Gamma$ will be banded, with zeros in all positions sufficiently far from the diagonal. This facilitates storage of the matrix and reconstruction of very long video sequences by breaking the recovery problem into blocks. (However, we consider only reconstructing a single sequence of the video in this chapter.)

From the formulation above, we see that it is possible to focus the reconstruction process on recovery of a relatively low-rate stream of anchor frames x_a, rather than the high-rate stream of frames x_d measured by the imaging system. Of course, in simplifying the CS problem, we have changed the very unknowns that must be solved for. In many cases, we believe that it will suffice

merely to reconstruct and display the anchor frames themselves; however, we note that the raw video frames x_d can be estimated from the reconstructed anchor frames by using the interpolation equation (3.10).

If the anchor frames are defined at the video's temporal Nyquist rate, and if there are no additional assumptions made about the video, then one should not expect any temporal correlations to remain among the anchor frames. In many real world settings, however, there will be objects moving within the scene, and the smoothness of the object motion can lead to temporal correlations, e.g., that can be captured via motion-compensated transforms. Thus, in order to impose the strongest possible model on the vector of anchor frames, it may be helpful to look for sparsity in a motion-compensated wavelet transform.[2] In Section 3.4, we propose one method for doing this while confronting the chicken-or-egg problem.

3.3.2.3 Optimizing the spacing between anchor frames

The quality of the interpolation approximation described in (3.10) relies on the assumption that $T_a \leq \frac{\pi}{\Omega_t}$. However, in practical settings one may not know Ω_t exactly, and this raises the question of how to properly set T_a, the spacing between anchor frames. We note that when T_a is chosen to be too small a multiple of T (that is, when $V = \frac{T_a}{T}$ is small), the dimensionality of the reduced problem (3.11) may not be small enough to permit reconstruction. Thus, in practice one should generally aim to choose T_a small enough that the interpolation approximation (3.10) holds to a reasonable degree of accuracy but large enough that the number of unknowns is sufficiently reduced. Our simulations in Section 3.5.2 explore these trade-offs and demonstrate that it can indeed be possible to find a setting for T_a that meets both of these criteria.

3.4 Multiscale Reconstruction Algorithm

Our analysis in the previous sections has (*i*) revealed that videos that are sampled by imaging devices (including CS imaging devices) may have limited temporal bandwidth, (*ii*) characterized the tradeoffs between the spatial resolution of the camera, the speed of any moving objects, and the temporal bandwidth of the video, and (*iii*) explained how a relatively low-rate stream of "anchor frames" can be used to reduce the complexity of the reconstruction problem. In this section, we build on these insights and propose a complete algorithm for reconstructing a video from streaming compressive measurements. In order to construct an effective sparsifying basis for the video, this

[2] For promising experiments that involve a simplified version of our algorithm (one that uses anchor frames but not with a motion-compensated wavelet transform), we refer the reader to a companion technical report [134].

algorithm involves a motion-compensated wavelet transform, and in order to confront the chicken-or-egg problem, this algorithm is multiscale, employing anchor frames at a sequence of progressively finer scales and alternating between reconstructing an approximation of the video and estimating the motion vectors.

3.4.1 Problem Formulation

Following the setup in Section 3.3.2, we suppose that one linear measurement is collected from f_d every T seconds. Using (3.9), we can write the vector of M measurements as $y = \Phi x_d$. Using (3.10), we can write $x_d \approx \Gamma x_a$, where x_a is formed by stacking the anchor frames. For the sake of simplicity in formulating our algorithm, we specifically take $x_a = [f_d(:,T_a)^T \ f_d(:,2T_a)^T \ \cdots \ f_d(:,\frac{M}{V}T_a)^T]^T$, i.e., we truncate the stream of anchor frames at the beginning and end so that we consider exactly $N_a := \frac{M}{V}$ anchor frames, and we adapt Γ at the borders to account for this truncation. To model any interpolation errors, we introduce an error term $e \in \mathbb{R}^M$ and model the collected measurements as $y = \Phi\Gamma x_a + e$.

As a sparsifying basis to be used for reconstructing x_a, we employ the lifting-based invertible motion adaptive transform (LIMAT) [118]. Further details regarding LIMAT are given in Section 3.4.3 (and in [118]), but the key idea is this: Given a set of vectors describing the motion in a video, LIMAT yields a motion-compensated wavelet transform (across the temporal and spatial dimensions) intended to provide a sparse representation of that video. For a given set of motion vectors v, we let $\Psi_L(v)$ denote the resulting LIMAT transform. We will apply LIMAT transforms to videos of various sizes; in all cases, $\Psi_L(v)$ is a square matrix, but its dimension will depend on the context.

If one had access to the anchor frames x_a, it would be possible to examine the video and estimate the vectors v describing motions between frames. Alternatively, if one had access to the motion vectors v, one could write $x_a = \Psi_L(v)\alpha_a$ for some sparse coefficient vector α_a and solve a CS reconstruction problem such as

$$\widehat{\alpha}_a = \text{argmin}_{\alpha_a} \|\alpha_a\|_1 \text{ s.t. } \|y - \Phi\Gamma\Psi_L(v)\alpha_a\|_2 \leq \epsilon \tag{3.12}$$

to recover α_a and subsequently x_a from the compressive measurements [27]. In order to get around this chicken-or-egg problem, we propose a multiscale reconstruction approach.

3.4.2 The Multiscale Approach

Rather than reconstructing the full set of high-resolution anchor frames x_a directly, our algorithm aims to reconstruct a sequence of approximations to the anchor frames. These approximations begin at a low temporal and spatial resolution and progress to successively finer resolutions.

To set our notation, we use the variable s to denote scale, with $s = 1$ denoting the coarsest (low-resolution) scale and $s = S$ denoting the finest (high-resolution) scale for some positive integer S. For each $s = 1, 2, \ldots, S$, let $\mathcal{D}_s$ represent a $2^{2(s-S)}N \times N$ linear operator that, when applied to a video frame containing N pixels, performs a spatial low-pass filtering and downsampling operation to produce a low-resolution frame containing $2^{2(s-S)}N$ pixels. (When $s = S$, $\mathcal{D}_s$ is the identity; when $s = S - 1$, $\mathcal{D}_s$ averages over 2×2 blocks of pixels; when $s = S - 2$, $\mathcal{D}_s$ averages over 4×4 blocks of pixels; etc.) Recalling that

$$x_a = [f_d(:, T_a)^T \; f_d(:, 2T_a)^T \; \cdots \; f_d(:, N_a T_a)^T]^T \in \mathbb{R}^{N_a N},$$

we define for each $s = 1, 2, \ldots, S$, $N_s := 2^{3(s-S)} N_a N$ and

$$x_{a,s} := [\mathcal{D}_s f_d(:, T_a \cdot 2^{S-s})^T \; \mathcal{D}_s f_d(:, 2T_a \cdot 2^{S-s})^T \; \cdots \; \mathcal{D}_s f_d(:, N_a T_a)^T]^T \in \mathbb{R}^{N_s}.$$

The video $x_{a,s}$ contains $2^{s-S} N_a$ frames, each with $2^{2(s-S)}N$ pixels. It represents a lowpass filtered and downsampled version of the original set of anchor frames x_a. We note in particular that $x_{a,S} = x_a$.

Our reconstruction begins at the coarsest scale with an attempt to estimate $x_{a,1}$. Motivated by our temporal bandwidth analysis, we begin reconstruction at a low spatial resolution for the following reasons:

- Our theory predicts that at low spatial resolutions, the temporal bandwidth of the video will be small. This limits the amount by which the video can change between successive frames and justifies the temporal downsampling in the definition of $x_{a,1}$. This downsampling also leaves us with a relatively small number of frames that must be sparsely represented, and so this sequence can be reconstructed from a small number of measurements.

- The spatial downsampling leaves us with larger pixels in each frame, and so an object that traverses, say, P pixels per second in the high-resolution anchor frames will traverse only $2^{1-S}P$ pixels per second at scale $s = 1$. With relatively slow object motion, we can thus

obtain a reasonable estimate of $x_{a,1}$ *without employing motion compensation*, or equivalently, with LIMAT motion vectors v set to $\underline{0}$.

After we have reconstructed an estimate of $x_{a,1}$, it is then possible to compute a preliminary estimate of the motion in the video. Although the resulting motion vectors will have limited spatial and temporal accuracy (since $x_{a,1}$ does), they can be used to perform a motion-compensated reconstruction of $x_{a,2}$.[3] From this point, we iterate (as detailed below), alternating between video reconstruction and motion estimation, and proceeding to finer and finer scales. Between each pair of adjacent scales, we double the spatial resolution and (as suggested by our analysis) double the temporal resolution as well.

The following pseudo-code is an outline of our algorithm.

- **Step 0:** initialize scale $s \leftarrow 1$ and motion vectors $v_1 = \underline{0}$
- **Step 1:** solve

$$\widehat{\alpha}_{a,s} = \text{argmin}_{\alpha' \in \mathbb{R}^{N_s}} \|\alpha'\|_1 \text{ s.t. } \|y_s - \Phi_s \Gamma_s \Psi_L(v_s)\alpha'\|_2 \leq \epsilon_s, \tag{3.13}$$

  where, as explained below, y_s contains all measurements in y that are associated with scales $1, 2, \ldots, s$, Φ_s and Γ_s relate these measurements to $x_{a,s}$, and ϵ_s accounts for interpolation error
- **Step 2:** form $\widehat{x}_{a,s} = \Psi_L(v_s)\widehat{\alpha}_{a,s}$
- **Step 3:** if $s = S$, terminate algorithm; otherwise go to Step 4
- **Step 4:** given $\widehat{x}_{a,s}$ use motion estimation to compute the set of motion vectors v_{s+1}
- **Step 5:** set $s \leftarrow s + 1$ and go to Step 1

Our reconstruction algorithm places certain constraints on the measurement functions ϕ_m that can be used. In particular, note that for each $s \in \{1, 2, \ldots, S\}$ solving the optimization problem (3.13) requires a set of measurements y_s that are assumed to obey $y_s = \Phi_s \Gamma_s x_{a,s} + e_s$. To facilitate this, we assume that each measurement $y(m)$ is *associated* with some scale $s \in \{1, 2, \ldots, S\}$. A measurement $y(m)$ is said to be associated with scale s if $y(m)$ can be written as $y(m) = \langle \widetilde{\phi}_m, \mathcal{D}_s f_d(:, mT) \rangle$ for some $\widetilde{\phi}_m \in \mathbb{R}^{2^{2(s-S)}N}$. Stated differently, this requires that the frame $f_d(:, mT)$ be measured using a function that is piecewise constant over blocks of size $2^{S-s} \times 2^{S-s}$ pixels, i.e., that frame $f_d(:, mT)$ be measured with limited spatial resolution. This is easily accomplished using programmable measurement devices such as the single-pixel camera. In particular, we suggest using

[3] Typically we choose S small enough so that the frames in $x_{a,1}$ contain enough pixels to obtain a reasonable estimate of the motion vectors.

noiselets [37] for measurement functions, as they incorporate both randomness and multiple spatial resolutions.

The measurements y_s that we propose to use for solving (3.13) are merely a subset of the measurements in y. In particular, for each for $s = 1, 2, \ldots, S$, y_s contains all measurements in y that are associated with scales $1, 2, \ldots, s$. Thus, measurements from coarser scales are re-used for reconstruction at finer scales, and at the finest scale all measurements are used for reconstruction. Other terms appearing in (3.13) are Γ_s, an interpolation matrix that relates $x_{a,s}$ to a spatially lowpass-filtered stream of the original high-rate frames $[\mathcal{D}_s f_d(:,T)^T \ \mathcal{D}_s f_d(:,2T)^T \ \cdots \ \mathcal{D}_s f_d(:,MT)^T]^T$, and Φ_s, a measurement matrix populated with the vectors $\widetilde{\phi}_m$. The term ϵ_s appearing in (3.13) is to account for interpolation error. In our current simulations we assume that we know the amount of interpolation error as an oracle, and we set ϵ_s accordingly.

The specific number of measurements (call this M_s) that we associate with each scale s is a design parameter. We find it useful to allocate a generous portion of the measurement budget to M_1 for two reasons: (*i*) it is relatively cheap to reconstruct an accurate video at low resolution, and (*ii*) an accurate initial reconstruction is important to obtain an accurate initial estimate of the motion vectors. As we progress in the algorithm to finer scales, the number of unknowns that we are solving for increases by a factor of eight with each increment in scale. Therefore, it is also natural to increase the number of measurements M_s that we associate with each scale. Although the size of the video increases by a factor of eight between one scale and the next, however, the sparsity level may not necessarily increase by the same factor. To get a rough estimate of the increase in sparsity level, consider that within a 3D video, the boundary of a moving object will generally trace out a 2D surface. Using simple box-counting arguments, one would generally expect the sparsity level in an isotropic 3D dictionary (such as a 3D wavelet transform) to increase by a factor of four between adjacent scales. Based on this observation, we believe that an increase of M_s by (very roughly) a factor of four for every increment in scale would be a reasonable choice. Our simulations in Section 3.5.2 use measurement rates inspired by this $4\times$ rule of thumb and refined with a small amount of trial and error.

Finally, we comment on the allocation strategy of the multiscale measurements across the M frames, i.e., which of the M frames we associate with which scale. One straightforward allocation strategy would be a random one, e.g., randomly choose M_1 out of the M original frames and associate these with scale 1, randomly choose M_2 out of the $M - M_1$ remaining frames and associate these with scale 2, etc. However, we have found that in doing so, it is usually helpful to ensure that all anchor frames at a given scale s are automatically associated with scale s (or a coarser scale). The

remaining (non-anchor) frames are associated randomly.

3.4.3 Lifting-based Invertible Motion Adaptive Transform (LIMAT)

At each scale $s = 1, 2, \ldots, S$, we use LIMAT [118] as a sparsifying transform for the anchor frames $x_{a,s}$. Recall that $x_{a,s}$ contains $2^{s-S}N_a$ frames, each with $2^{2(s-S)}N$ pixels. To simplify notation in this section, set $n = 2^{s-S}N_a$ and for $k = 1, 2, \ldots, n$, let x_k denote frame k from the sequence $x_{a,s}$.

The lifting transform partitions the video into even frames $\{x_{2k}\}$ and odd frames $\{x_{2k+1}\}$ and attempts to predict the odd frames from the even ones using a forward motion compensation operator. This operator, which we denote by $\mathcal{F}$, takes as input one even frame and a collection of motion vectors denoted v_f that describe the anticipated motion of objects between that frame and its neighbor. For example, suppose that x_{2k} and x_{2k+1} differ by a 3-pixel shift that is captured precisely in v_f; then as a result $x_{2k+1} = \mathcal{F}(x_{2k}, v_f)$ exactly. Applying this prediction to each pair of frames and keeping only the prediction errors, we obtain a sequence of highpass residual detail frames (see (3.14) below). The prediction step is followed by an update step that uses an analogous backward motion compensation operator denoted $\mathcal{B}$ and motion vectors v_b. The combined lifting steps

$$h_k = x_{2k+1} - \mathcal{F}(x_{2k}, v_f) \tag{3.14}$$

$$l_k = x_{2k} + \frac{1}{2}\mathcal{B}(h_k, v_b) \tag{3.15}$$

produce an invertible transform between the original video and the lowpass $\{l_k\}$ and highpass $\{h_k\}$ coefficients. For maximum compression, the lifting steps can be iterated on pairs of the lowpass frames until there remains only one. Ideally, with perfect motion compensation, the $n - 1$ highpass frames will consist only of zeros, leaving only one frame of nonzero lowpass coefficients, and making the sequence significantly more compressible. As a final step, it is customary to apply the 2D discrete wavelet transform (DWT) to each lowpass and highpass frame to exploit any remaining spatial correlations.

In our proposed algorithm, we use block matching (BM) [16] to estimate motion between a pair of frames. The BM algorithm divides the reference frame into non-overlapping blocks. For each block in the reference frame the most similar block of equal size in the destination frame is found and the relative location is stored as a motion vector. There are several possible similarity measures; we use the ℓ_1 norm.

3.5 Discussion and Simulations

In this section, we conclude by describing the differences between our work and several other important ones in the literature. We also present simulations that demonstrate the performance of our algorithm.

3.5.1 Related Work

Our algorithm is intended for reconstructing a video from streaming compressive measurements; specifically, we assume that one compressive measurement is collected at each time instant. (The single-pixel camera is a prototypical example of an imaging device that produces such measurements.) We note that a preliminary version of our algorithm [104] was intended for a different measurement model—one in which multiple measurements are collected from each frame but at a much lower frame rate. For example, 1000 measurements might be collected from each of 30 frames per second. While acquiring such measurements may be possible using other compressive imaging architectures [112, 97], our particular interest in this chapter is on how to correctly deal with streaming measurements.

In addition to our preliminary algorithm [104], several others have also been proposed in the CS video literature that incorporate motion estimation and compensation. These are discussed below. We note that none of these algorithms were explicitly designed to handle streaming measurements. However, any of these algorithms could be modified to (approximately) operate using streaming measurements by partitioning the measurements into short groups and assuming that all measurements within a group come from the same frame. Such raw aggregation of measurements actually corresponds to using a rectangular interpolation kernel in our formulation (3.11). As one would expect, and as we will demonstrate in Section 3.5.2, one can achieve significantly better performance by employing smoother interpolation kernels. To the best of our knowledge we are the first to propose and justify the use of other interpolation kernels for aggregating measurements. We do note that one algorithm from the literature [117] does present a careful method for grouping the measurements in order to minimize the interpolation error when using a rectangular kernel.

One related algorithm [64] is based on the observation that nearby frames should have a sparse or compressible residual when one frame is subtracted from the other. The authors employ motion estimation and compensation to produce a sparse or compressible residual frame even in the presence of fast or global motion. More specifically, let x_1 and x_2 denote two frames in a video sequence. Initial estimates of x_1 and x_2 are obtained via independent reconstruction from the measurements $y_1 = \Phi x_1$ and $y_2 = \Phi x_2$, respectively. Motion vectors are then computed between these frame estimates, and

subsequently a motion-compensated frame, x_{mc}, is computed from x_1 and the estimated motion vectors. When motion estimation is accurate, the residual $x_{\mathrm{r}} = x_2 - x_{\mathrm{mc}}$ is likely to be highly sparse or compressible. The authors propose to reconstruct the residual frame, x_{r}, from the measurements $y_{\mathrm{r}} = y_2 - y_{\mathrm{mc}} = \Phi x_2 - \Phi x_{\mathrm{mc}}$. The estimate of the residual frame, $\widehat{x}_{\mathrm{r}}$, can be used to improve upon the previous estimate of x_2 via $\widehat{x}_2 = x_{\mathrm{mc}} + \widehat{x}_{\mathrm{r}}$. This procedure is then carried out for the next pair of frames x_2 and x_3. This method requires that the same matrix Φ be used for measuring each frame. In some cases, the use of such measurement matrices may decrease diversity in the measurements. As an extreme example, if a video contains no motion, the measurements from every frame will be identical.

Another related algorithm [123] has been proposed in the field of dynamic medical resonance imaging (MRI). This algorithm relies on the assumption that the locations of the significant coefficients of each video frame can be estimated via a motion-compensated frame. Similarly to the above, a motion-compensated frame, x_{mc}, is obtained from motion vectors computed between a pair of estimates of the frames x_1 and x_2. The initial estimates of the video frames are computed in a frame-by-frame fashion from their respective random measurements. When motion compensation is accurate, the indices of large coefficients of x_{mc} will provide an accurate prediction of the locations of the significant coefficients of x_2, and this knowledge can be used to reconstruct an accurate estimate of x_2. Once a new estimate of x_2 is formed, this procedure is repeated for the next frame x_3, and so on.

Another related algorithm [83] involves dividing a video sequence into several groups of pictures (GOPs), each of which is made up of a key frame followed by several non-key frames. The authors propose to reconstruct each key frame in a frame-by-frame fashion. Given the estimates of the key frames, estimates of the non-key frames are computed via a technique called motion-compensated interpolation. Refined estimates of the non-key frames are obtained in a frame-by-frame fashion, using the initial motion-compensated frame as the starting point of a gradient projection for sparse reconstruction (GPSR) solver. The authors propose a novel stopping criterion for the GPSR solver that helps find a solution that is not too different from the initial estimate. This procedure is then carried out for the next non-key frame, and so on.

Finally, one other related algorithm involves a dual-scale reconstruction of a video [117]. First, a sufficient number of low-resolution measurements are collected to permit a low-resolution preview of the video to be obtained using simple least-squares. Motion information is then extracted from this preview video, and this motion information is used to help reconstruct the high-resolution video sequence. An optimization problem minimizes the sum of the ℓ_1-norm of the expansion coefficients

of each individual frame in an appropriate sparsifying transform subject to a data fidelity constraint. Additionally, the minimization problem is subject to a constraint such as

$$\|\widehat{x}_i(x,y) - \widehat{x}_{i+1}(x+v_x, y+v_y)\|_2 \leq \epsilon, \ \forall i,$$

which ensures that the reconstructed video agrees with the motion vectors estimated from the preview video. While this method can be relatively successful in recovering videos from small numbers of measurements, according to the authors, the algorithm can have difficulty in reconstructing high spatial frequency components.

We would like to point out that all of the above methods require, for each frame, a number of measurements proportional to the sparsity level of that frame (after appropriate motion compensation and a spatial sparsifying transform such as a 2D wavelet transform). This is true simply because three of the above methods [64, 123, 83] involve reconstructing the video one or two frames at a time. The fourth of the above methods [117] does involve jointly reconstructing the ensemble of video frames. However, this algorithm still requires a total number of measurements proportional to the sum of the sparsity levels of the individual frames because that quantity is what is minimized in the ℓ_1 optimization procedure.

We argue that a temporal sparsifying transformation can help to decrease the sparsity level of a video signal and thus reduce the number of measurements that must be collected of that video. In particular, for videos with slowly moving objects or videos with stationary backgrounds, temporal redundancies may persist for longer than the duration of one or two frames. The above methods, however, are essentially employing a temporal transform with a temporal support of one or two frames. If one can successfully remove temporal redundancies over a longer temporal support, the overall sparsity level of the video will decrease, and this in turn will reduce the number of measurements that must be collected from each frame.

Some methods for CS video reconstruction have been proposed that do employ a 1D temporal sparsifying transform along with a 2D spatial sparsifying transform. In essence, each of these algorithms corresponds to applying a different 3D sparsifying transform Ψ. One algorithm uses the 3D-DWT for Ψ and reconstructs the entire video all at once [139]. Another approach, termed "C_n" and proposed for compressive coded aperture video reconstruction [96], relies on small inter-frame differences together with a spatial 2D-DWT to produce a sparse representation of the underlying video. Unfortunately, neither of these algorithms involves any motion compensation, and so these 3D transforms will not be successful in sparsifying videos containing any significant levels of motion.

We see that the two classes of methods described above have two distinct strengths. Algorithms in the first class employ motion information in the reconstruction to better reconstruct the video sequence, while algorithms in the second class employ a temporal transformation to remove the temporal redundancies over a longer temporal support. In this chapter (and in [104]), we use LIMAT which essentially combines these two strengths: LIMAT performs a full motion-compensated temporal decomposition that, when seeded with accurate motion vectors, can effectively remove temporal redundancies even in the presence of complex motion.

We would like to point out, however, that LIMAT is just one of many possible 3D sparsifying bases and that our algorithm can be modified to use other choices of basis as well. We also note that the frame-by-frame reconstruction methods [64, 123, 83] have the advantage of being more computationally efficient as they involve reconstruction problems of a smaller size. This may be an important factor in applications where computational resources are scarce and full temporal decompositions are not practical. We also reiterate that to some degree, the multiscale algorithm we present in this chapter is a proof-of-concept inspired by our temporal bandwidth analysis. We see our work as an addition to—not a replacement for—the nascent CS video literature, and we believe that the ideas we expound (such as using anchor frames to reduce the complexity of reconstruction) could be combined with some of the other existing ideas mentioned above.

Finally, to place our work in the proper context, we reiterate the differences between a standard ("non-compressive") video capture system and the CS-based ("compressive") video acquisition strategy discussed in this chapter. As we discussed in Section 3.1, the primary advantages of a compressive video system are twofold: first, it reduces the physical burden of measuring the incoming video, as it does not require a complete set of samples to be obtained for every frame, and second, it reduces the computational complexity of the encoding process, as it does not require any spatiotemporal transform to be implemented at the encoder. These advantages do not make a compressive video system appropriate for all situations, however. For example, it is well known in CS that for a given sparse signal, the requisite number of measurements is slightly higher (by a small multiplicative factor) than one would require if the relevant sparse components could be directly extracted from the signal. Standard video capture systems have the advantage of getting to "look at" the fully sampled video before identifying the critical features to be encoded. One would naturally expect, then, that for a given quality level, a traditional video encoder would require fewer bits than a compressive video encoder (one in which the CS measurements were quantized and converted to bits). Standard video capture systems also have a second main advantage: the decoding algorithm can be much simpler than in a compressive video system. Our comments on these points are not unique to our

particular CS framework; indeed they are fully consistent with observations made in another recent paper dealing with CS for video [109].[4] For our algorithm, however, a detailed comparison against a traditional encoder in terms of bit rate, power consumption, memory requirements, resiliency to errors, etc., is beyond the scope of this chapter.

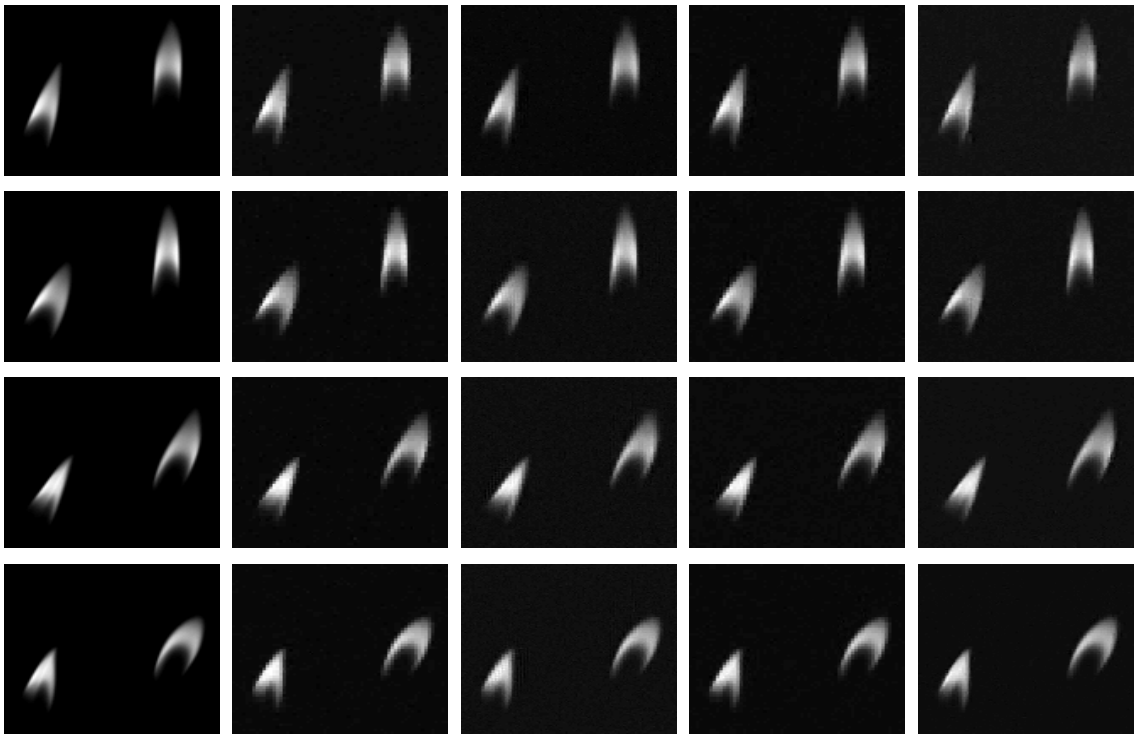
3.5.2 Simulations

In this section we present simulation results for our proposed algorithm. We compare our algorithm (using a linear interpolation kernel) to the 3D-DWT [139] and C_n [96] temporal sparsifying transforms using a rectangular interpolation kernel. We also compare to a modified version of our algorithm that uses LIMAT but with zero-motion vectors. The results demonstrate the benefits of combining motion compensation with a temporal sparsifying transform and the benefits of using a non-rectangular interpolation kernel.

We present results for two different test videos (courtesy of MERL), the *Candle* video and the *Pendulum + Cars* video. These are the same videos that we described in Section 3.2.3.2, but we have selected different time samples and cropped the frames to different image regions. Four of the original anchor frames of each video are presented in Figures 3.3(a) and 3.4(a). Each video consists of 512 frames; the frames in the *Candle* video are of size $N = 128 \times 128$, while the frames in the *Pendulum + Cars* video are of size $N = 256 \times 256$. The first test video contains two candle flames moving in front of a black (zero-valued) background. This video was acquired with a high-speed camera at 1000 frames per second. The flames warp into different shapes and sway from left to right as the video progresses. The typical speed of the moving edge of a flame is approximately 0.3 pixels per frame. The second video contains a white car that is moving in front of a "Newton's cradle" (a pendulum with several silver metal balls) and a stationary background. As the video progresses, the white car moves from right to left, and the two metal balls at each end of the Newton's cradle swing throughout the video sequence. This video was acquired with a high-speed camera at 250 frames per second. The motion of the car is mostly translational (it moves from right to left at a speed of roughly 0.025 pixels per frame), while the two metal balls swing at a much faster speed of roughly 1–1.5 pixels per frame.

In our simulations, we construct synthetic CS measurements from the original high-rate video sequences. In order to reasonably apply CS to either of these video sequences, one would need more

[4] The encoder proposed in [109] also has some philosophical similarities and differences with our work and others we have described. Like our work, the system in [109] can operate with frame-by-frame CS measurements and is intended to exploit spatial and temporal correlations in the video. Unlike our work, however, the system in [109] utilizes a special post-measurement encoding of the CS measurements to exploit temporal correlations. It also relies on raw aggregation (rectangular interpolation) of the measurements.

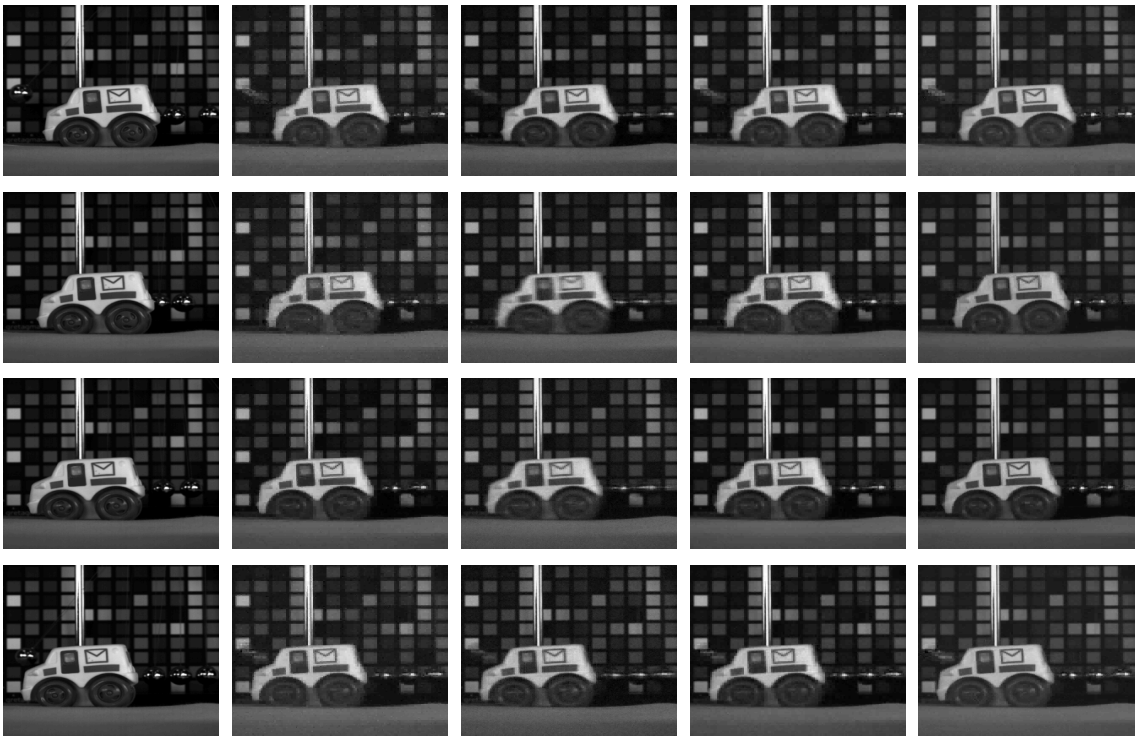


(a) Original frames (b) 3D-DWT (c) C_n with $n = 32$ (d) LIMAT with $v = 0$ (e) LIMAT

Figure 3.3: *Reconstructed anchor frames of* Candle *video. Out of 512 original frames, 32 anchor frames were reconstructed. From the top each row shows the 5th, 11th, 21st, and 28th anchor frame and its reconstruction via various reconstruction methods. From the left each column represents: (a) original frames, (b) 3D-DWT with rectangular interpolation kernel, PSNR = 33.67 dB, (c) C_n, $n = 32$, with rectangular interpolation kernel, PSNR = 33.68 dB, (d) LIMAT with rectangular interpolation kernel and zero-motion vectors, PSNR = 33.80 dB, (e) LIMAT with linear interpolation kernel, PSNR = 36.76 dB.*

than 1000 measurements per second, and thus with the available videos (acquired at 1000 and 250 frames per second) we would not have a sufficient number of measurements if we synthesized only one measurement from each frame. Thus, for the purposes of testing our algorithm, we collect more than one random measurement from each frame of the test video. (Still, the number of measurements we collect from each frame is moderate—120 per frame for the first video and 800 per frame for the second video.) Our setting remains very different from related ones [64, 123, 83], however, because we do not intend to reconstruct all of the high-rate frames directly (indeed, this would be difficult to do with so few measurements per frame); rather we will reconstruct a reduced set of anchor frames.

We begin by reconstructing both videos using $N_a = 32$ anchor frames. The total number of unknowns that we are solving for in case of the *Candle* video is $NN_a = 128 \times 128 \times 32$; in the case of the *Pendulum + Cars* video it is $NN_a = 256 \times 256 \times 32$. Note that the use of anchor frames reduces the reconstruction complexity by a factor of $V = 512/32 = 16$. For the *Candle* video, we divide the reconstruction into $S = 3$ scales. At scale 1 we reconstruct a low-resolution video of size



(a) Original frames (b) 3D-DWT (c) C_n with $n = 32$ (d) LIMAT with $v = 0$ (e) LIMAT

Figure 3.4: *Reconstructed anchor frames of* Pendulum + Cars *video. Out of 512 original frames, 32 anchor frames were reconstructed. From the top each row shows the 4th, 13th, 22nd, and 30th anchor frame and its reconstruction via various reconstruction methods. From the left each column represents: (a) original frames, (b) 3D-DWT with rectangular interpolation kernel, PSNR = 28.22 dB, (c) C_n, $n = 32$, with rectangular interpolation kernel, PSNR = 29.58 dB, (d) LIMAT with rectangular interpolation kernel and zero-motion vectors, PSNR = 29.81 dB, (e) LIMAT with linear interpolation kernel, PSNR = 31.38 dB.*

$N_1 := 32 \times 32 \times 8$, at scale 2 a low-resolution video of size $N_2 := 64 \times 64 \times 16$, and at scale 3 the full-resolution video of size $N_3 := 128 \times 128 \times 32$. Similarly, we divide the reconstruction into $S = 4$ scales for the *Pendulum + Cars* video. Thus, at scale 1 we reconstruct a low-resolution video of size $N_1 := 32 \times 32 \times 4$, at scale 2 a low-resolution video of size $N_2 := 64 \times 64 \times 8$, at scale 3 a low-resolution video of size $N_3 := 128 \times 128 \times 16$, and at scale 4 the full-resolution video of size $N_4 := 256 \times 256 \times 32$. For the *Candle* video we take 120 measurements from each of the 512 frames, and for the *Pendulum + Cars* video we take 800 measurements from each of the 512 frames. Out of the 512 frames in the video, for the *Candle* video, we allocate 54 frames to scale 1, 250 frames to scale 2, and 208 frames to scale 3. Thus, we have $M_1 = 54 \times 120$, $M_2 = 250 \times 120$, and $M_3 = 208 \times 120$ when reconstructing the video. Similarly, for the *Pendulum + Cars* video we allocate 5 frames to scale 1, 30 frames to scale 2, 130 frames to scale 3, and 347 frames to scale 4, so that $M_1 = 5 \times 800$, $M_2 = 30 \times 800$, $M_3 = 130 \times 800$, and $M_4 = 347 \times 800$ when reconstructing the video.

This means that when reconstructing the *Candle* video, at scale 1 we solve for N_1 unknowns

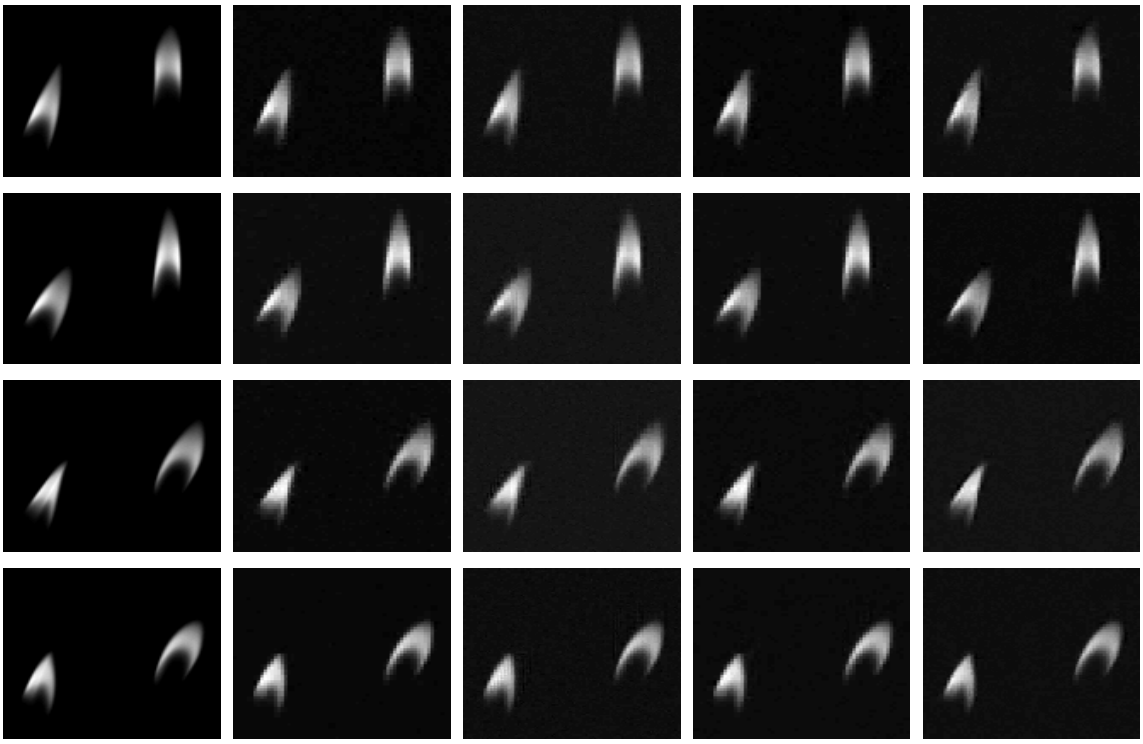
using M_1 measurements, and we have $M_1/N_1 \approx 0.79$. At scale 2, we solve for N_2 unknowns using $M_1 + M_2$ measurements, and we have $(M_1 + M_2)/N_2 \approx 0.56$ (recall that we re-use coarser scale measurements at the current scale). Finally, at scale 3, we solve for N_3 unknowns using $M_1+M_2+M_3$ measurements, and we have $(M_1+M_2+M_3)/N_3 \approx 0.12$. Similarly, for the *Pendulum + Cars* video, at scale 1 we have $M_1/N_1 \approx 0.98$,[5] at scale 2 we have $(M_1 + M_2)/N_2 \approx 0.85$, at scale 3 we have $(M_1 + M_2 + M_3)/N_3 \approx 0.50$, and at scale 4 we have $(M_1 + M_2 + M_3 + M_4)/N_4 \approx 0.20$.

Figures 3.3 and 3.4 show the original anchor frames and their reconstructions via the various reconstruction methods. The PSNR values shown in both figures have been computed across the 32 anchor frames. For each reconstruction method that we test, we use the same multiscale measurements described above; the only differences are in what kernel is used for interpolating the measurements and in what algorithm (i.e., sparsifying transform) is used for reconstruction. Columns (b) and (c) show the performance of the 3D-DWT [139] and C_n [96] temporal sparsifying transforms, respectively. For both we use a rectangular interpolation kernel. As discussed in Section 3.5.1, using a rectangular interpolation kernel is equivalent to aggregating the CS measurements and assigning them to the nearest anchor frame. Column (d) shows the performance of a modified version of our algorithm that uses LIMAT with zero-motion vectors and a rectangular interpolation kernel. Column (e) shows the performance of our full algorithm, using a linear interpolation kernel and using LIMAT with motion vectors estimated during the multiscale reconstruction procedure.

Using the same total number of measurements, we repeat the experiments above but using $N_a = 16$ anchor frames instead of 32 (we also use a different allocation of measurements across the scales). The results are presented in Figures 3.5 and 3.6. For all reconstruction methods—including ours—we see a decrease in the PSNR of the reconstruction. We believe that this is mainly due to the inevitable increase in interpolation error that occurs when the anchor frames are spaced further apart. In exchange for the decrease in PSNR, however, the complexity of the reconstruction problem has been reduced by a factor of 2.

In all experiments, we see that our algorithm gives the highest PSNR value. Our reconstruction algorithm also successfully recovers some features in the video frames that other methods cannot. For example, we can see that the flames in column (e) of the *Candle* video reconstructions have sharper and more detailed edges than those of other reconstructions shown in columns (b)–(d). We can also see that the edges of the car and the details of features of the wheels are more pronounced than those of the others. We note that objects with slow motion (e.g., cars) are most accurately

[5]Here we could consider taking a few more measurements so that $M_1/N_1 \geq 1$ and simply inverting the measurement matrix to obtain a low-resolution estimate of the video. However, as the measurements contain error due to interpolation, we believe it would be preferable to solve an ℓ_1-regularized problem instead.

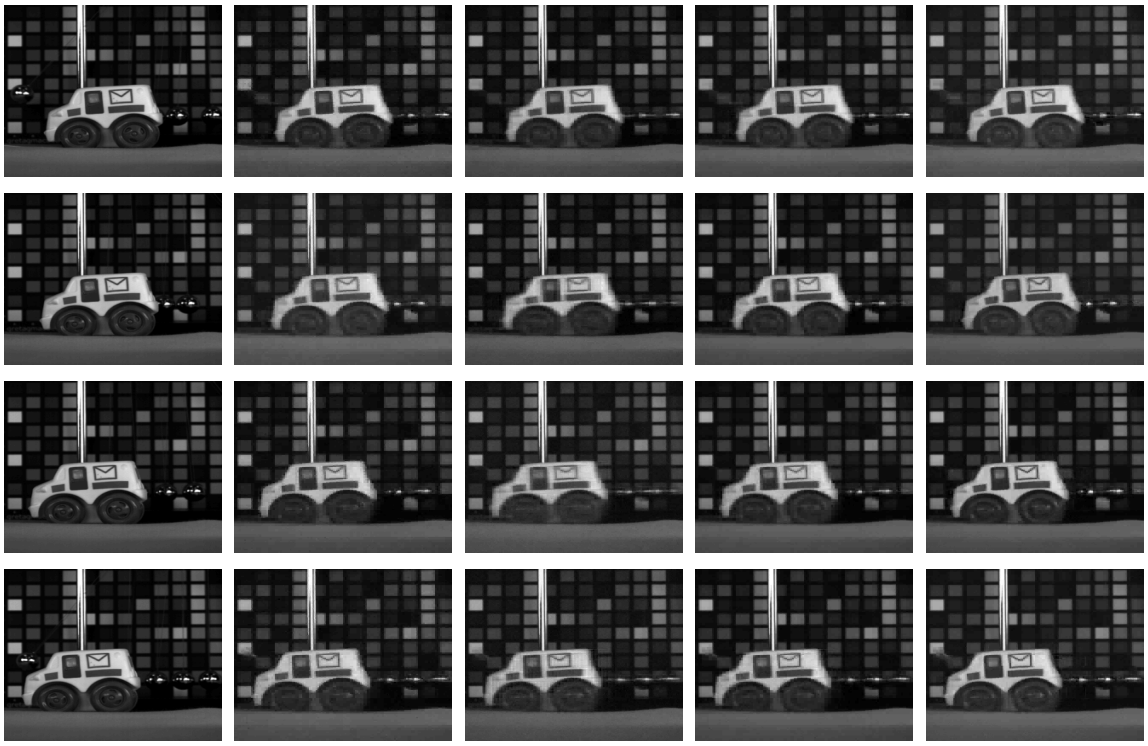


(a) Original frames (b) 3D-DWT (c) C_n with $n = 16$ (d) LIMAT with $v = 0$ (e) LIMAT

Figure 3.5: *Reconstructed anchor frames of* Candle *video. Out of 512 original frames, 16 anchor frames were reconstructed. From the top each row shows the 3rd, 6th, 12th, and 15th anchor frame and its reconstruction via various reconstruction methods. From the left each column represents: (a) original frames, (b) 3D-DWT with rectangular interpolation kernel, PSNR = 32.79 dB, (c) C_n, $n = 32$, with rectangular interpolation kernel, PSNR = 32.70 dB, (d) LIMAT with rectangular interpolation kernel and zero-motion vectors, PSNR = 32.64 dB, (e) LIMAT with linear interpolation kernel, PSNR = 35.49 dB.*

reconstructed, while objects that are moving fast (e.g., metal balls) are less accurately reconstructed. A different spacing of anchor frames could help to reconstruct more rapidly moving objects, and one could even consider adaptively changing the spacing of the anchor frames throughout the video. While such extensions are beyond the scope of this chapter, our companion technical report [134] does contain additional discussion on this front.

Overall, both visually and in terms of PSNR, we have seen the benefits of combining motion compensation with a temporal sparsifying transform and the benefits of using a non-rectangular interpolation kernel. Though not shown, we have also tested our algorithm using zero-motion LIMAT with a linear interpolation kernel. For $N_a = 32$, the PSNR for the reconstructed *Candle* video is 36.60 dB, and the PSNR for the reconstructed *Pendulum + Cars* video is 30.79 dB; for $N_a = 16$, the PSNR for the reconstructed *Candle* video is 34.84 dB, and the PSNR for the reconstructed *Pendulum + Cars* video is 29.35 dB. Thus, while both the motion compensation and the interpolation kernel are contributing to the gains that we see, the change in interpolation kernel seems to be a bit more



(a) Original frames (b) 3D-DWT (c) C_n with $n = 16$ (d) LIMAT with $v = 0$ (e) LIMAT

Figure 3.6: *Reconstructed anchor frames of* Pendulum + Cars *video. Out of 512 original frames, 16 anchor frames were reconstructed. From the top each row shows the 2nd, 6th, 11th, and 15th anchor frame and its reconstruction via various reconstruction methods. From the left each column represents: (a) original frames, (b) 3D-DWT with rectangular interpolation kernel, PSNR = 27.81 dB, (c) C_n, $n = 32$, with rectangular interpolation kernel, PSNR = 28.30 dB, (d) LIMAT with rectangular interpolation kernel and zero-motion vectors, PSNR = 28.48 dB, (e) LIMAT with linear interpolation kernel, PSNR = 29.64 dB.*

valuable.

The reconstruction time of our algorithm depends mainly on size of the unknown signal. Thus, the reconstruction will be faster at coarser scales in the algorithm and slower at finer scales. The reconstruction complexity also depends on the motion vectors, since the time to compute a matrix vector product increases with the number of non-zero motion vectors used in LIMAT. Thus, the reconstruction time when using LIMAT with zero-motion vectors will be comparable to the time required for, e.g., the 3D-DWT (which is based on orthonormal linear filters) but shorter than the time required for LIMAT with non-zero motion vectors. Using an off-the-shelf Windows-based PC with quad-core CPU, the total times for our algorithm to reconstruct 16 and 32 anchor frames of the *Candle* video were roughly 30 minutes and 230 minutes, respectively. To be more specific, for the reconstruction involving 32 anchor frames, the algorithm required 9 seconds at scale 1, 552 seconds at scale 2, and 13679 seconds at scale 3. For the sake of comparison, the 3D-DWT, C_n, and LIMAT with zero-motion vector algorithms took approximately 2204 seconds, 16106 seconds, and

2796 seconds, respectively. Reducing the computational complexity of our algorithm is an important problem for future research. We do note, though, that while our algorithm may not currently be feasible for real-time reconstruction purposes, as with CS-MUVI [117] it would be possible to view a low-resolution preview of the video in almost real-time.

CHAPTER 4

Compressed Sensing in Multiview Imaging and Reconstruction

4.1 Introduction

Armed with potentially limited communication and computational resources, designers of distributed imaging systems face increasing challenges in the quest to acquire, compress, and communicate ever richer and higher-resolution image ensembles. In this chapter, we consider multi-view imaging problems in which an ensemble of cameras collect images describing a common scene. To simplify the acquisition and encoding of these images, we study the effectiveness of non-collaborative compressed sensing (CS) [52, 26], also known as Distributed CS (DCS) [18], encoding schemes wherein each sensor directly and independently compresses its image using a small number of randomized measurements (see Figure 4.1). CS is commonly intended for the encoding of a single signal, and a rich theory has been developed for signal recovery from incomplete measurements by exploiting the assumption that the signal obeys a sparse model. In this chapter, we address the problem of how to recover an ensemble of images from a collection of image-by-image random measurements. To do this, we advocate the use of implicitly geometric models to capture the joint structure among the images.

While DCS encoding is non-collaborative, an effective DCS decoder should reconstruct all signals *jointly* to exploit their common structure. As we later discuss, most existing DCS algorithms for distributed imaging reconstruction rely fundamentally on sparse models to capture intra- and inter-signal correlations [18, 34, 96, 104]. What is missing from each of these algorithms, however, is an assurance that the reconstructed images have a global consistency, i.e., that they all describe a

This work is in collaboration with Michael B. Wakin [105], and builds upon earlier work that appeared in [137] by Michael B. Wakin.

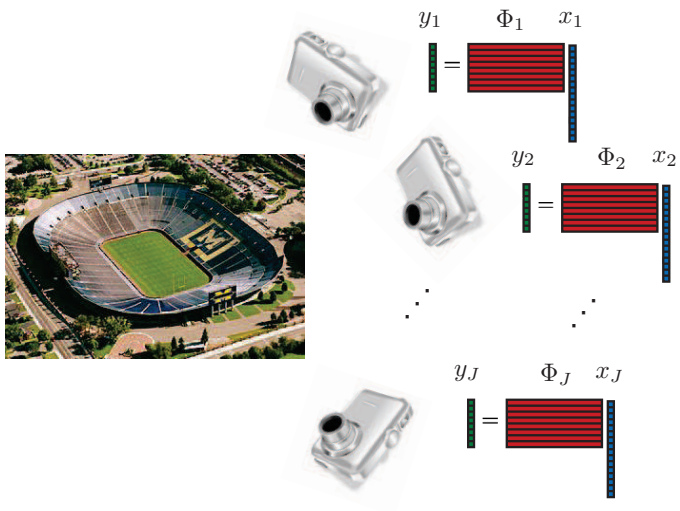


Figure 4.1: *Multi-view compressive imaging setup. A common scene is observed by J cameras from different positions. Each camera j encodes a small number of random measurements y_j of its observed image x_j, and a single decoder jointly reconstructs all images $\{x_j\}$ from the ensemble of compressive measurements $\{y_j\}$.*

common underlying scene. This may not only lead to possible confusion in interpreting the images, but more critically may also suggest that the reconstruction algorithm is failing to completely exploit the joint structure of the ensemble.

To better extend DCS techniques specifically to problems involving multi-view imaging, we propose in this chapter a general geometric framework in which many such reconstruction problems may be cast. We specifically focus on scenarios where a representation of the underlying scene is linearly related to the observations. This is mainly for simplicity, and there is plenty of room for the development of joint reconstruction algorithms given nonlinear mappings; however, we present a number of scenarios where a linear mapping can be found. For these problems, we explain how viewing the unknown images as living along a low-dimensional manifold within the high-dimensional signal space can inform the design of effective joint reconstruction algorithms. Such algorithms can build on existing sparsity-based techniques for CS but ensure a global consistency among the reconstructed images. We refine our discussion by focusing on two settings: far-field and near-field multi-view imaging. Finally, as a proof of concept, we demonstrate a "manifold lifting" algorithm in a specific far-field multi-view scenario where the camera positions are not known a priori and we only observe a small number of random measurements at each sensor. Even in such discouraging circumstances, by effectively exploiting the geometrical information preserved in the manifold model, we are able to accurately reconstruct both the underlying scene and the camera positions.

4.2 Problem Setup and Related Work

4.2.1 Multi-view Imaging using Image-by-Image Random Measurements

Let us now turn to the problem of distributed image compression for multi-view imaging. We imagine an ensemble of J distinct cameras that collect images $x_1, x_2, \ldots, x_J \in \mathbb{R}^N$ describing a common scene, with each image x_j taken from some camera position $\theta_j \in \Theta$. We would like to efficiently compress this ensemble of images, but as in any sensor network, we may be limited in battery power, computational horsepower, and/or communication bandwidth. Thus, although we may be able to posit sparse and manifold-based models for concisely capturing the intra- and inter-signal structures among the images in the ensemble, directly exploiting these models for the purpose of data compression may be prohibitively complex or require expensive collaboration among the sensors. This motivates our desire for an effective disjoint encoding strategy.

The encoding of multiple signals in distributed scenarios has long been studied under the auspices of the distributed source coding (DSC) community. The Slepian-Wolf framework [120] for lossless DSC states that two sources X_1 and X_2 are able to compress at their conditional entropy rate without collaboration and can be decoded successfully when the correlation model (i.e., the joint probability distribution $p(x_1, x_2)$) is known at the decoder. This work was extended to lossy coding by Wyner and Ziv when side information is available at the decoder [144], and in subsequent years, practical algorithms for these frameworks have been proposed based on channel coding techniques. However, one faces difficulties in applying these frameworks to multi-view imaging because the inter-image correlations are arguably better described geometrically than statistically. Several algorithms (e.g., [126, 67, 125]) have been proposed for combining these geometric and statistical frameworks, but fully integrating these concepts remains a very challenging problem.

As a simple alternative to these type of encoding schemes, we advocate the use of CS for distributed image coding, wherein for each sensor $j \in \{1, 2, \ldots, J\}$, the signal $x_j \in \mathbb{R}^N$ is independently encoded using an $M_j \times N$ measurement matrix Φ_j, yielding the measurement vector $y_j = \Phi_j x_j \in \mathbb{R}^{M_j}$. Such an encoding scheme is known as DCS [18]. While the primary motivation for DCS is to simplify the *encoding* of correlated high-resolution signals, one may of course bypass the potentially difficult *acquisition* of the high-resolution signals and directly collect the random measurements using CS hardware.

After the randomized encoding, the measurement vectors $y_1, y_2, \ldots, y_J$ are then transmitted to a central node for decoding. Indeed, DCS differs from single-signal CS only in the decoding process. Rather than recover the signals one-by-one from the measurement vectors, an effective DCS decoder

should solve a joint reconstruction problem, exploiting the intra- and inter-signal correlations among the signals $\{x_j\}$, while ensuring consistency with the measurements $\{y_j\}$.

The proper design of a DCS decoder depends very much on the type of data being collected and on the nature of the intra- and inter-signal correlations. Ideally, compared to signal-by-signal recovery, joint recovery should provide better reconstruction quality from a given set of measurement vectors, or equivalently, reduce the measurement burden needed to achieve a given reconstruction quality. For example, if each signal in the ensemble is K-sparse, we may hope to jointly recover the ensemble using fewer than the $\mathcal{O}(K \log(N/K))$ measurements per sensor that are required to reconstruct the signals separately. Like single-signal CS, DCS decoding schemes should be robust to noise and to dropped measurement packets. Joint reconstruction techniques should also be robust to the loss of individual sensors, making DCS well-suited for remote sensing applications.

4.2.2 Current Approaches to DCS Multi-view Image Reconstruction

For signals in general and images in particular, a variety of DCS decoding algorithms have been proposed to date. Fundamentally, all of these frameworks build upon the concept of sparsity for capturing intra- and inter-signal correlations.

One DCS modeling framework involves a collection of joint sparsity models (JSMs) [18]. In a typical JSM we represent each signal $x_j \in \mathbb{R}^N$ in terms of a decomposition $x_j = z_C + z_j$, where $z_C \in \mathbb{R}^N$ is a "common component" that is assumed to be present in all $\{x_j\}$, and $z_j \in \mathbb{R}^N$ is an "innovation component" that differs for each signal. Depending on the application, different sparsity assumptions may be imposed on z_C and z_j. In some cases these assumptions can dramatically restrict the space of possible signals. For example, all signals may be restricted to live within the same K-dimensional subspace. The DCS decoder then searches for a signal ensemble that is consistent with the available measurements and falls within the space of signals permitted by the JSM. For signal ensembles well-modeled by a JSM, DCS reconstruction can offer a significant savings in the measurement rates. While each sensor must take enough measurements to account for its innovation component z_j, all sensors can share the burden of measuring the common component z_C.

Unfortunately, the applicability of JSMs to multi-view imaging scenarios can be quite limited. While two cameras in very close proximity may yield images having sparse innovations relative to a common background, any significant difference in the camera positions will dramatically increase the complexity of the innovation components. Because conventional JSMs are not appropriate for capturing any residual correlation that may remain among these innovations, we would expect JSM-based recovery to offer very little improvement over independent CS recovery.

Recently, a significant extension of the JSM framework has been proposed specifically for multi-view compressive imaging [34]. This framework assumes that images of a common scene are related by local or global geometrical transformations and proposes an overcomplete dictionary of basis elements consisting of various geometrical transformations of a generating mother function. It is assumed that each image can be decomposed into its own subset of these atoms plus the geometrically transformed atoms of the neighboring images. The benefit of this approach is that information about one image helps reduce the uncertainty about which atoms should be used to comprise the neighboring images. Unfortunately, there seems to be a limit as to how much efficiency may be gained from such an approach. To reconstruct a given image, the decoder may be tasked with solving for, say, K sparse coefficients. While the correlation model may help reduce the measurement burden at that sensor below $\mathcal{O}(K \log(N/K))$, it is not possible to reduce the number of measurements below K. As we will later argue, however, there is reason to believe that alternative reconstruction techniques based on the underlying scene (rather than the images themselves) can succeed with even fewer than K measurements.

Other approaches for multi-view image reconstruction could draw naturally from recent work in CS video reconstruction by ordering the static images $\{x_j\}$ according to their camera positions and reconstructing the sequence as a sort of "fly-by" video. One approach for video reconstruction exploits the sparsity of inter-frame differences [96]. For multi-view imaging, this would correspond to a difference image $x_i - x_j$ having a sparse representation in some basis Ψ. Again, however, this condition may only be met if cameras i and j have very close proximity. We have also proposed a CS video reconstruction technique based on a motion-compensated temporal wavelet transform [104]. For multi-view imaging, we could modify this algorithm, replacing block-based motion compensation with disparity compensation. The challenge of such an approach, however, would be in finding the disparity information without prior knowledge of the images themselves. For video, we have addressed this challenge using a coarse-to-fine reconstruction algorithm that alternates between estimating the motion vectors and reconstructing successively higher resolution versions of the video using the motion-compensated wavelet transform.

What would still be missing from any of these approaches, however, is an assurance that the reconstructed images have a global consistency, i.e., that they all describe a common underlying scene. In the language of manifolds this means that the reconstructed images do not necessarily live on a common IAM defined by a hypothetical underlying scene. This may not only lead to possible confusion in interpreting the images, but more critically may also suggest that the reconstruction algorithm is failing to completely exploit the joint structure of the ensemble—the images are in fact

constrained to live in a much lower-dimensional set than the algorithm realizes.

4.3 Manifold Lifting Techniques for Multi-view Image Reconstruction

In light of the above observations, one could argue that an effective multi-view reconstruction algorithm should exploit the underlying geometry of the scene by using an inter-signal modeling framework that ensures global consistency. To inform the design of such an algorithm, we find it helpful to view the general task of reconstruction as what we term a *manifold lifting* problem: we would like to recover each image $x_j \in \mathbb{R}^N$ from its measurements $y_j \in \mathbb{R}^{M_j}$ ("lifting" it from the low-dimensional measurement space back to the high-dimensional signal space), while ensuring that all recovered images live along a common IAM.

Although this interpretation does not immediately point us to a general purpose recovery algorithm (and different multi-view scenarios could indeed require markedly different algorithms), it can be informative for a number of reasons. For example, as we have discussed in Sec. 2.2, manifolds can have stable embeddings under random projections. If we suppose that $\Phi_j = \Phi \in \mathbb{R}^{M \times N}$ for all j, then each measurement vector we obtain will be a point sampled from the embedded manifold $\Phi\mathcal{M} \subset \mathbb{R}^M$. From samples of $\Phi\mathcal{M}$ in $\mathbb{R}^M$, we would like to recover samples of (or perhaps all of) $\mathcal{M}$ in $\mathbb{R}^N$, and this may be facilitated if $\Phi\mathcal{M}$ preserves the original geometric structure of $\mathcal{M}$. In addition, as we have discussed in Sec. 2.1, many IAMs have a multiscale structure that has proved useful in solving non-compressive parameter estimation problems, and this structure may also be useful in solving multi-view recovery problems.

While this manifold-based interpretation may give us a geometric framework for signal modeling, it may not in isolation sufficiently capture all intra- and inter-signal correlations. Indeed, one cannot disregard the role that concise models such as sparsity may still play in an effective manifold lifting algorithm. Given an ensemble of measurements $y_1, y_2, \ldots, y_J$, there may be many candidates IAMs on which the original images $x_1, x_2, \ldots, x_J$ may live. In order to resolve this ambiguity, one could employ either a model for the intra-signal structure (such as sparsity) or a model for the underlying structure of the scene (again, possibly sparsity). To do the latter, one must develop a representation for the underlying scene or phenomenon that is being measured and understand the mapping between that representation and the measurements $y_1, y_2, \ldots, y_J$. To keep the problem simple, this mapping will ideally be linear, and as we discuss in this section, such a representation and linear mapping can be found in a number of scenarios.

To make things more concrete, we demonstrate in this section how the manifold lifting viewpoint

can inform the design of reconstruction algorithms in the context of two generic multi-view scenarios: far-field and near-field imaging. We also discuss how to address complications that can arise due to uncertainties in the camera positions. We hope that such discussions will pave the way for the future development of broader classes of manifold lifting algorithms.

4.3.1 Far-field Multi-view Imaging

We begin by considering the case where the cameras are far from the underlying scene, such as might occur in satellite imaging or unmanned aerial vehicle (UAV) remote sensing scenarios. In problems such as these, it may be reasonable to model each image $x_j \in \mathbb{R}^N$ as being a translated, rotated, scaled subimage of a larger fixed image. We represent this larger image as an element x drawn from a vector space such as $\mathbb{R}^Q$ with $Q > N$, and we represent the mapping from x to x_j (which depends on the camera position θ_j) as a linear operator that we denote as $R_{\theta_j} : \mathbb{R}^Q \to \mathbb{R}^N$. This operator R_{θ_j} can be designed to incorporate different combinations of translation, rotation, scaling, etc., followed by a restriction that limits the field of view.

This formulation makes clear the dependence of the IAM $\mathcal{M}$ on the underlying scene x: $\mathcal{M} = \mathcal{M}(x) = \{R_\theta x : \theta \in \Theta\} \subset \mathbb{R}^N$. Supposing we believe x to obey a sparse model and supposing the camera positions are known, this formulation also facilitates a joint recovery program that can ensure global consistency while exploiting the structure of the underlying scene. At camera j, we have the measurements $y_j = \Phi_j x_j = \Phi_j R_{\theta_j} x$. Therefore, by concatenating all of the measurements, we can write the overall system of equations as $y = \Phi_{\text{big}} R x$, where

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_J \end{bmatrix}, \; R = \begin{bmatrix} R_{\theta_1} \\ R_{\theta_2} \\ \vdots \\ R_{\theta_J} \end{bmatrix}, \text{ and } \Phi_{\text{big}} = \begin{bmatrix} \Phi_1 & 0 & \dots & 0 \\ 0 & \Phi_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Phi_J \end{bmatrix}. \tag{4.1}$$

Given y and $\Phi_{\text{big}} R$, and assuming x is sparse in some basis Ψ (such as the 2D wavelet domain), we can solve the usual optimization problem as stated in (2.1) (or (2.2) if the measurements are noisy). If desired, one can use the recovered image $\widehat{x}$ to obtain estimates $\widehat{x}_j := R_{\theta_j} \widehat{x}$ of the original subimages. These are guaranteed to live along a common IAM, namely $\mathcal{M}(\widehat{x})$.

4.3.2 Near-field Multi-view Imaging

Near-field imaging may generally be more challenging than far-field imaging. Defining a useful representation for the underlying scene may be difficult, and due to effects such as parallax and

occlusions, it may seem impossible to find a linear mapping from any such representation to the measurements. Fortunately, however, there are encouraging precedents that one could follow.

One representative application of near-field imaging is in Computed Tomography (CT). In CT, we seek to acquire a 3D volumetric signal x, but the signals x_j that we observe correspond to slices of the Fourier transform of x. (We may assume $y_j = x_j$ in such problems, and so the challenge is actually to recover $\mathcal{M}(x)$, or equivalently just x, rather than the individual $\{x_j\}$.) Given a fixed viewing angle θ_j, this relationship between x and x_j is linear, and so we may set up a joint recovery program akin to that proposed above for far-field imaging. Similar approaches have been used for joint recovery from undersampled frequency measurements in MRI [91].

For near-field imaging using visible light, there is generally no clear linear mapping between a 3D volumetric representation of the scene and the observed images x_j. However, rather than contend with complicated nonlinear mappings, we suggest that a promising alternative may be to use the *plenoptic function* [71] as a centralized representation of the scene. The plenoptic function f is a hypothetical 5D function used to describe the intensity of light that could be observed from any point in space, when viewed in any possible direction. The value $f(p_x, p_y, p_z, p_\theta, p_\phi)$ specifies the light intensity that would be measured by a sensor located at the position (p_x, p_y, p_z) and pointing in the direction specified by the spherical coordinates p_θ and p_ϕ. (Additional parameters such as color channel can be considered.) By considering only a bounded set of viewing positions, the plenoptic function reduces to a 4D function known as the *lumigraph* [71].

Any image $x_j \in \mathbb{R}^N$ of the scene has a clear relationship to the plenoptic function. A given camera j will be positioned at a specific point (p_x, p_y, p_z) in space and record light intensities arriving from a variety of directions. Therefore, x_j simply corresponds to a 2D "slice" of the plenoptic function, and once the camera viewpoint θ_j is fixed, the mapping from f to x_j is a simple linear restriction operator. Consequently, the structure of the IAM $\mathcal{M} = \mathcal{M}(f)$ is completely determined by the plenoptic function.

Plenoptic functions contain a rich geometric structure that we suggest could be exploited to develop sparse models for use in joint recovery algorithms. This geometric structure arises due to the geometry of objects in the scene: when a physical object having distinct edges is photographed from a variety of perspectives, the resulting lumigraph will have perpetuating geometric structures that encode the shape of the object under study. As a simple illustration[1], a Flatland-like scenario (imaging an object in the plane using 1D cameras) is shown in Figure 4.2(a). The resulting 2D

[1]The authors gratefully acknowledge Richard Baraniuk and Hyeokho Choi for many influential conversations concerning the lumigraph and for their help in developing the lumigraph experiments presented here.

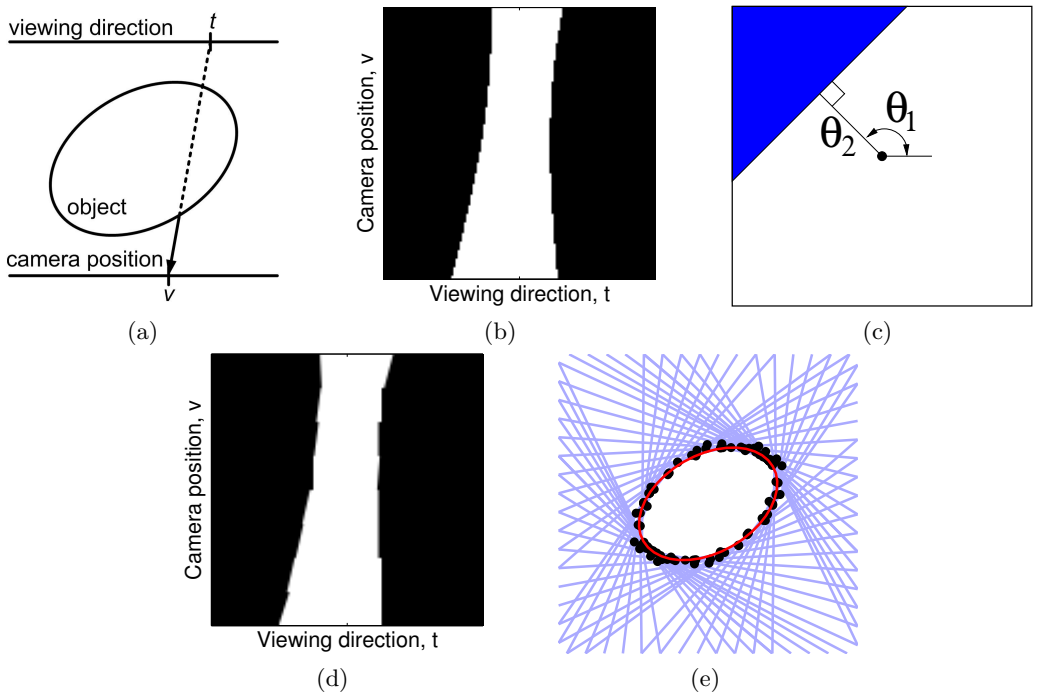


Figure 4.2: *Lumigraph geometry in compressive multi-view imaging. (a) Flatland-like illustration for collecting 1D images of an object in the 2D plane. At each camera position, all viewing directions may be considered. (b) Resulting* 128×128 *lumigraph for the ellipse-shaped object. Each row corresponds to a single "image". (In the real world each image is 2D and the full lumigraph is 4D.) The lumigraph can be repeated for viewing from all four sides of the object. (c) Wedgelets provide a simple parametric model for local patches of a 2D lumigraph; only two parameters are needed to describe the orientation and offset of the linear discontinuity. (d) Wedgelet-based lumigraph reconstruction from* $M = 5$ *compressive samples of each image (row of lumigraph). (e) Scene geometry estimated using local edge positions/orientations in the reconstructed lumigraph. Each blue line connects an estimated point on the object to a camera from which that point is visible. The true ellipse is shown in red.*

lumigraph is shown in Figure 4.2(b), where each row corresponds to a single "image". We see that geometric structures in the lumigraph arise due to shifts in the object's position as the camera viewpoint changes. For the 4D lumigraph these structures have recently been termed "plenoptic manifolds" [20] due to their own nonlinear, surface-like characteristics. If a sparse representation for plenoptic functions can be developed that exploits these geometric constraints, then it may be possible to recover plentopic functions from incomplete, random measurements using a linear problem formulation and recovery algorithms such as (2.1) or (2.2). One possible avenue to developing such a sparse representation could involve parameterizing local patches of the lumigraph using the wedgelet [51] or surflet [32] dictionaries. Wedgelets (see Figure 4.2(c)) can be tiled together to form piecewise linear approximations to geometric features; surflets offer piecewise polynomial approximations.

As a proof of concept, we present a simple experiment in support of this approach. For the lumigraph shown in Figure 4.2(b), which has $J = 128$ 1D "images" that each contain $N = 128$

pixels, we collect $M = 5$ random measurements from each image. From these measurements we attempt to reconstruct the entire lumigraph using wedgelets [32] following a multiscale technique outlined in Chapter 6 of [135]. The reconstructed lumigraph is shown in Figure 4.2(d) and is relatively accurate despite the small number of measurements.

Finally, to illustrate the rich interplay between geometry within the lumigraph and the underlying geometry of the scene, we show that it is actually possible to use the reconstructed lumigraph to estimate the underlying scene geometry. While we omit the precise details of our approach, the estimated wedgelets help us to infer three pieces of information: the positions of each local wedgelet patch in the v and t directions indicate a camera position and viewing direction, respectively, while the orientation of the wedgelet indicates a depth at which a point in the scene belongs to the object. Putting these estimates together, we obtain the reconstruction of the scene geometry shown in Figure 4.2(e). This promising proof of concept suggests that wedgelets or surflets could indeed play an important role in the future for developing improved concise models for lumigraph processing.

4.3.3 Dealing with Uncertainties in Camera Positions

In all of our discussions above, we have assumed the camera positions θ_j were known. In some situations, however, we may have only noisy estimates $\widehat{\theta}_j = \theta_j + n_j$ of the camera positions. Supposing that we can define linear mappings between the underlying scene and the images x_j, it is straightforward to extend the CS recovery problem to account for this uncertainty. In particular, letting R denote the concatenation of the mappings R_{θ_j} as in (4.1), and letting $\widehat{R}$ denote the concatenation of the mappings $R_{\widehat{\theta}_j}$ corresponding to the noisy camera positions, it follows that $y = \Phi_{\text{big}} R x = \Phi_{\text{big}} \widehat{R} x + n$ for some noise vector n, and so (2.2) can be used to obtain an approximation $\widehat{x}$ of the underlying scene. Of course, the accuracy of this approximation will depend on the quality of the camera position estimates.

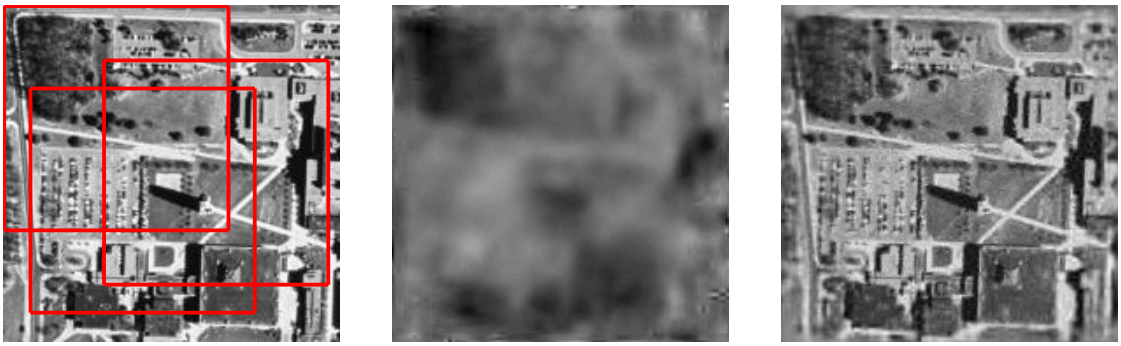
When faced with significant uncertainty about the camera positions, the multiscale properties of IAMs help us to conceive of a possible coarse-to-fine reconstruction approach. As in Sec. 2.1, let h_s denote a blurring kernel at scale s, and suppose for simplicity that $\Theta = \mathbb{R}$. Based on the arguments presented in [138], it follows that for most reasonable mappings $\theta \to x_\theta$, we will have $\| \frac{\partial (h_s * x_\theta)}{\partial \theta} \|_2 \to 0$ as $s \to 0$. What this implies is that, on manifolds of regularized images $\mathcal{M}_s = \{h_s * x_\theta : \theta \in \Theta\}$, the images will change slowly as a function of camera position, and so we can ensure that $h_s * (R_{\widehat{\theta}_j} x)$ is arbitrarily close to $h_s * (R_{\theta_j} x)$ by choosing s sufficiently small (a sufficiently "coarse" scale). Now, suppose that some elements of each y_j are devoted to measuring $h_s * x_j = h_s * (R_{\theta_j} x)$. We denote these measurements by $y_{j,s} = \Phi_{j,s}(h_s * x_j)$. In practice, we may replace the convolution operator

with a matrix H_s and collect $y_{j,s} = \Phi_{j,s} H_s x_j = \Phi_{j,s} H_s R_{\theta_j} x$ instead. Concatenating all of the $\{y_{j,s}\}_{j=1}^J$, we may then use the noisy position estimates to define operators $\{R_{\widehat{\theta}_j}\}$ and solve (2.2) as above to obtain an estimate $\widehat{x}$ of the scene. This estimate will typically correspond to a lowpass filtered version of x, since for many reasonable imaging models, we will have $h_s * (R_{\theta_j} x) \approx R_{\theta_j}(h'_s * x)$ for some lowpass filter h'_s, and this implies that $y_{j,s} \approx \Phi_{j,s} R_{\theta_j}(h'_s * x)$ contains only low frequency information about x.

Given this estimate, we may then re-estimate the camera positions by projecting the measurement vectors $y_{j,s}$ onto the manifold $\mathcal{M}(\widehat{x})$. (This may be accomplished, for example, using the parameter estimation techniques described in [138].) Then, having improved the camera position estimates, we may reconstruct a finer scale (larger s) approximation to the true images $\{x_j\}$, and so on, alternating between the steps of estimating camera positions and reconstructing successively finer scale approximations to the true images. This multiscale, iterative algorithm requires the sort of multiscale randomized measurements we describe above, namely $y_{j,s} = \Phi_{j,s}(h_s * x_j)$ for a sequence of scales s. In practice, the noiselet transform [37] offers one fast technique for implementing these measurement operators $\Phi_{j,s} H_s$ at a sequence of scales. Noiselet scales are also nested, so measurements at a scale s_1 can be re-used as measurements at any scale $s_2 > s_1$.

The manifold viewpoint can also be quite useful in situations where the camera positions are completely unknown, as they might be in applications such as cryo-electron microscopy (Cryo-EM) [119]. Because we anticipate that an IAM $\mathcal{M}$ will have a stable embedding $\Phi\mathcal{M}$ in the measurement space, it follows that the relative arrangement of the points $\{x_j\}$ on $\mathcal{M}$ will be preserved in $\Phi\mathcal{M}$. Since this relative arrangement will typically reflect the relative arrangement of the values $\{\theta_j\}$ in Θ, we may apply to the compressive measurements[2] any number of "manifold learning" techniques (such as ISOMAP [124]) that are designed to discover such parameterizations from unlabeled data. An algorithm such as ISOMAP will provide an embedding of J points in $\mathbb{R}^p$ whose relative positions can be used to infer the relative camera positions; a similar approach has been developed specifically for the Cryo-EM problem [119]. (Some side information may be helpful at this point to convert these relative position estimates into absolute position estimates.) Once we have these estimates, we may resort to the iterative refinement scheme described above, alternating between the steps of estimating camera positions and reconstructing successively finer scale approximations to the true images.

[2]We have found that this process also performs best using measurements of $h_s * x_j$ for s small because of the smoothness of the manifold $\mathcal{M}_s$ at coarse scales.



(a) (b) (c)

Figure 4.3: *(a) Setup for manifold lifting demonstration. The original image x (courtesy USGS) has size 192×192 and is observed by $J = 200$ satellites. The red boxes illustrate the limited field of view for a few such cameras. (b) Image-by-image reconstruction from random measurements, PSNR 14.4dB. (c) Joint reconstruction using our manifold lifting algorithm with unknown camera positions, PSNR 23.6dB.*

4.4 Manifold Lifting Case Study

4.4.1 Problem Setup

As a proof of concept, we now present a comprehensive multi-view reconstruction algorithm inspired by the manifold lifting viewpoint. We do this in the context of a far-field imaging simulation in which we wish to reconstruct a Q-pixel high-resolution image x of a large scene. Information about this scene will be acquired using an ensemble of J satellites, which will collect N-pixel photographs x_j of the scene from different positions and with limited but overlapping fields of view, as illustrated with red boxes in Figure 4.3(a).

We denote the vertical and horizontal position of satellite j by $\theta_j = (\theta_j^V, \theta_j^H) \in \mathbb{R}^2$. The satellite positions take real values and are chosen randomly except for the caveats that the fields of view all must fall within the square support of x and that each of the four corners of x must be seen by at least one camera. (These assumptions are for convenience but can be relaxed without major modifications to the recovery algorithm.) We let R_{θ_j} denote the $N \times Q$ linear operator that maps x to the image x_j. This operator involves a resampling of x to account for the real-valued position vector θ_j, a restriction of the field of view, and a spatial lowpass filtering and decimation, as we assume that x_j has lower resolution (larger pixel size) than x.

In order to reduce data transmission burdens, we suppose that each satellite encodes a random set of measurements $y_j = \Phi_j x_j \in \mathbb{R}^{M_j}$ of its incident image x_j. Following the discussion in Section 4.3.3, these random measurements are collected at a sequence of coarse-to-fine scales $s_1, s_2, \ldots, s_T$ using noiselets. (The noiselet measurements can actually be collected using CS imaging hardware [55], bypassing the need for a conventional N-pixel sensor.) We concatenate all of the measurement

vectors $\{y_{j,s_i}\}_{i=1}^{T}$ into the length-M_j measurement vector $y_j = \Phi_j x_j$. Finally, we assume that all satellites use the same set of measurement functions, and so we define $M := M_1 = M_2 = \cdots = M_J$ and $\Phi := \Phi_1 = \Phi_2 = \cdots = \Phi_J$.

Our decoder will be presented with the ensemble of the measurement vectors $y_1, y_2, \ldots, y_J$ but *will not* be given any information about the camera positions (save for an awareness of the two caveats mentioned above) and will be tasked with the challenge of recovering the underlying scene x. Although it would be interesting to consider quantization in the measurements, it is beyond the scope of this chapter and we did not implement any quantization steps in the following simulations.

4.4.2 Manifold Lifting Algorithm

We combine the discussions provided in Sections 4.3.1 and 4.3.3 to design a manifold lifting algorithm that is specifically tailored to this problem.

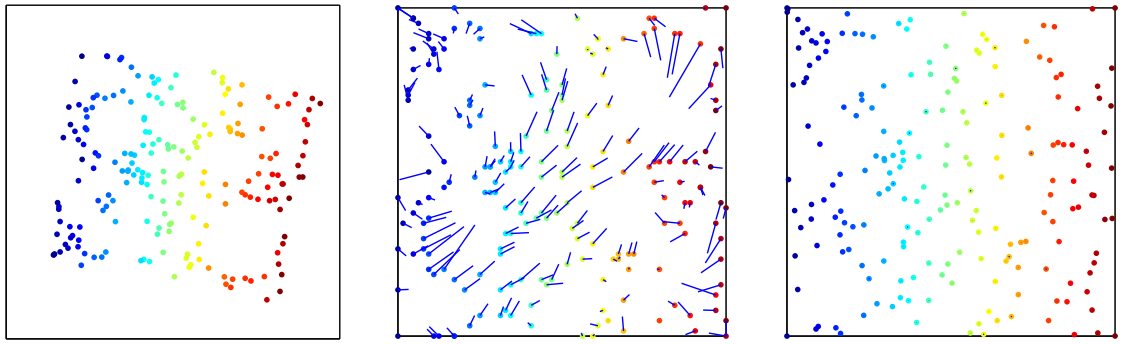
4.4.2.1 Initial estimates of satellite positions

The algorithm begins by obtaining a preliminary estimate of the camera positions. To do this, we extract from each y_j the measurements corresponding to the two or three coarsest scales (i.e., y_{j,s_1}, y_{j,s_2}, and possibly y_{j,s_3}), concatenate these into one vector, and pass the ensemble of such vectors (for all $j \in \{1, 2, \ldots, J\}$) to the ISOMAP algorithm. ISOMAP then delivers an embedding of points $v_1, v_2, \ldots, v_J$ in $\mathbb{R}^2$ that best preserves pairwise geodesic distances compared to the input points; an example ISOMAP embedding is shown in Figure 4.4(a). What can be inferred from this embedding are the relative camera positions; a small amount of side information is required to determine the proper scaling, rotation, and (possible) reflection of these points to correctly align them with an absolute coordinate system. Assuming that we know the correct vertical and horizontal reflections, after reflecting these camera positions correctly, we then rotate and scale them to fill the square support of x.

4.4.2.2 Iterations

Given the initial estimates $\{\widehat{\theta}_j\}$ of our camera positions, we can then define the operators $\{R_{\widehat{\theta}_j}\}$ and consequently $\widehat{R}$. By concatenating the measurement vectors and measurement matrices, initially only those at the coarsest scale (i.e., y_{j,s_1} across all j), we write the overall system of equations as $y = \Phi\widehat{R}x + n$ as in Sec. 4.3.1, and solve for

$$\widehat{\alpha} = \text{argmin}_\alpha \|\alpha\|_1 \text{ subject to } \|y - \Phi_{\text{big}}\widehat{R}\Psi\alpha\|_2 \leq \epsilon,$$



(a) (b) (c)

Figure 4.4: *(a) Initial ISOMAP embedding $v_1, v_2, \ldots, v_J$ of the measurement vectors. (b) Initial estimates $\{\widehat{\theta}_j\}$ of camera positions after rotating and scaling the $\{v_j\}$. (c) Final camera position estimates after running the manifold lifting algorithm. In (b) and (c), the colored points represent the estimated camera positions (color coded by the true θ_j^H value), while the blue vectors represent the error with respect to the true (but unknown) camera position.*

where Ψ is a wavelet basis and ϵ is chosen[3] to reflect the uncertainty in the camera positions θ_j. Given $\widehat{\alpha}$, we can then compute the corresponding estimate of the underlying scene as $\widehat{x} = \Psi\widehat{\alpha}$.

After we obtain the estimate $\widehat{x}$, we refine the camera positions by registering the measurement vectors y_j with respect to this manifold. In other words, we solve the following optimization problem:

$$\widehat{\theta}_j = \text{argmin}_\theta \|y_j - \Phi R_\theta \widehat{x}\|_2,$$

where again in each y_j we use only the coarse scale measurements. To solve this problem we use the multiscale Newton algorithm proposed in [138].

With the improved estimates $\widehat{\theta}_j$, we may then refine our estimate of $\widehat{x}$ but can do so by incorporating finer scale measurements. We alternate between the steps of reconstructing the scene $\widehat{x}$ and re-estimating the camera positions $\widehat{\theta}_j$, successively bringing in the measurements $y_{j,s_2}, y_{j,s_3}, \ldots, y_{j,s_T}$. (At each scale it may help to alternate once or twice between the two estimation steps before bringing in the next finer scale of measurements. One can also repeat until convergence or until reaching a designated stopping criterion.) Finally, having brought in all of the measurements, we obtain our final estimate $\widehat{x}$ of the underlying scene.

4.4.2.3 Experiments

We run our simulations on an underlying image x of size $Q = 192 \times 192$ that is shown in Figure 4.3(a). We suppose that x corresponds to 1 square unit of land area. We observe this scene

[3]In our experiments, we choose the parameter ϵ as somewhat of an oracle, in particular as $1.1\|y - \Phi_{\text{big}}\widehat{R}x\|_2$. In other words, this is slightly larger than the error that would result if we measured the true image x but with the wrong positions as used to define $\widehat{R}$. This process should be made more robust in future work.

using $J = 200$ randomly positioned cameras, each with a limited field of view. Relative to x, each field of view is of size 128×128, corresponding to 0.44 square units of land area as indicated by the red boxes in Figure 4.3(a). Within each field of view, we observe an image x_j of size $N = 64 \times 64$ pixels that has half the resolution (twice the pixel size) compared to x. The total number of noiselet scales for an image of this size is 6. For each image we disregard the coarsest noiselet scale and set $s_1, s_2, \ldots, s_5$ corresponding to the five finest noiselet scales. For each image we collect 96 random noiselet measurements: 16 at scale s_1, and 20 at each of the scales $s_2, \ldots, s_5$. Across all scales and all cameras, we collect a total of $96 \cdot 200 = 19200 \approx 0.52Q$ measurements.

Based on the coarse scale measurements, we obtain the ISOMAP embedding $v_1, v_2, \ldots, v_J$ shown in Figure 4.4(a). After rotating and scaling these points, the initial estimates $\{\widehat{\theta}_j\}$ of camera positions are shown in Figure 4.4(b). These initial position estimates have a mean absolute error of 1.8 and 2.0 pixels (relative to the resolution of x) in the vertical and horizontal directions, respectively. Figure 4.4(c) shows the final estimated camera positions after all iterations of our manifold lifting algorithm. These estimates have a mean absolute error of 0.0108 and 0.0132 pixels in the vertical and horizontal directions, respectively. The final reconstruction $\widehat{x}$ obtained using these estimated camera positions is shown in Figure 4.3(c). We note that the border areas are not as accurately reconstructed as the center region because fewer total measurements are collected near the borders of x. The scale-by-scale progression of the reconstruction of x and the estimated camera positions are shown in Figure 4.5. Figure 4.5(a) shows the reconstructed images of x at each scale $s_1, s_2, \ldots, s_5$, where the left most image is the reconstruction at the coarsest scale s_1 and the right most image is the reconstructed image at the finest scale s_5. Figure 4.5(b) shows the corresponding camera position estimates that were used in the reconstruction of the images in Figure 4.5(a). As we have mentioned above, it can help to alternate between reconstruction of the image and estimation of the camera positions at the same scale more than once before moving on to the next finer scale. In this particular simulation, we have alternated between reconstruction and camera position estimation 3 to 4 times at each scale but the finest and 6 times at the finest scale.

In order to assess the effectiveness of our algorithm, we compare it to three different reconstruction methods. In all of these methods we assume that the exact camera positions are known and we keep the total number of measurements fixed to 19200. First, we compare to image-by-image CS recovery, in which we reconstruct the images x_j independently from their random measurements y_j and then superimpose and average them at the correct positions. As expected, and as shown in Figure 4.3(b), this does not yield a reasonable reconstruction because there is far too little data collected (just 96 measurements) about any individual image to reconstruct it in isolation. Thus,

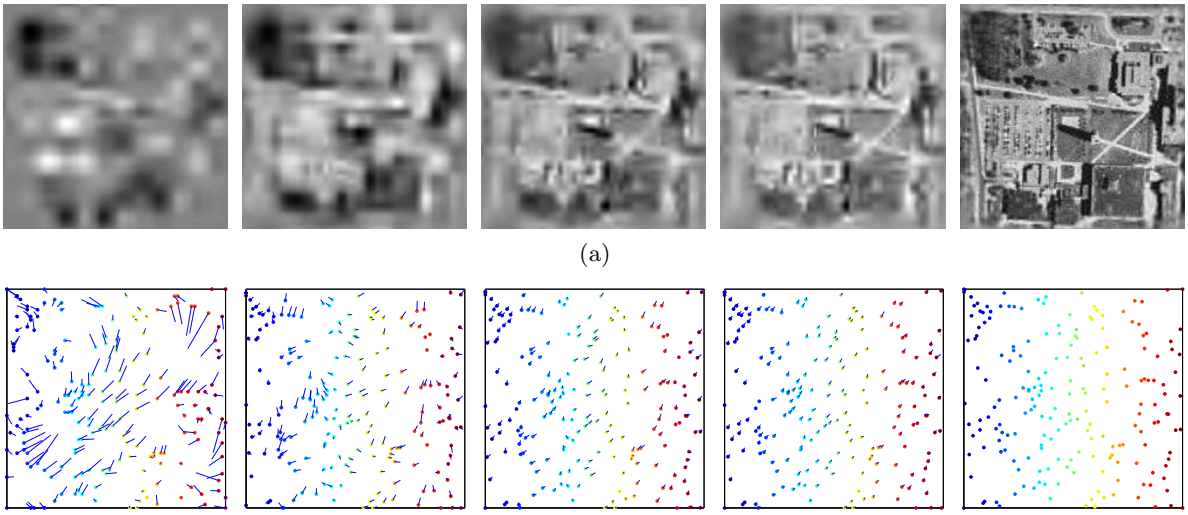


Figure 4.5: *(a) Scale-by-scale reconstruction of the underlying image proceeding from the coarsest scale s_1 on the left to the finest scale s_5 on the right. (b) The corresponding camera position estimates used in the reconstruction of the images in (a) proceeding from the coarsest scale s_1 on the left to the finest scale s_5 on the right.*

we see the dramatic benefits of joint recovery.

Second, for the sake of completeness, we compare to a non-distributed encoding scheme in which one measures the entire image x using a fully populated $19200 \times N$ Gaussian random matrix. Figure 4.6(a) shows the reconstructed image obtained using a single invocation of ℓ_1-minimization. Perhaps surprisingly, the reconstruction quality is actually inferior to that obtained using the manifold lifting algorithm with distributed measurements (shown in Figure 4.6(c)). This is somewhat counterintuitive since one would expect that the spatially limited measurement functions would have inferior isometry properties compared to global measurement functions. Although we do not have a concrete theoretical explanation for this phenomenon, we believe that this difference in reconstruction quality is mainly due to the multiscale nature of measurement functions employed in our manifold lifting example. To support this argument with an experiment, we run the manifold lifting algorithm with spatially limited but non-multiscale measurement functions: for each window, we measure a total of 96 noiselet measurements at the finest scale only, where previously these 96 measurements were spread across several scales. In this case, the reconstructed image has a PSNR of 19.8dB, which is worse than that obtained using a global Gaussian measurement matrix. This is consistent with our intuition that, when using measurements with limited spatial support, one could pay a penalty in terms of reconstruction quality.

Third, we compare to another alternative encoding scheme, where rather than encode 96 random noiselet measurements of each image, we encode the 96 largest wavelet coefficients of the image in the

(a) (b) (c)

Figure 4.6: *(a) Reconstruction using fully dense Gaussian random matrix, PSNR 21.9dB. (b) Joint reconstruction using transform coding measurements with known camera positions, PSNR 22.8dB. (c) Joint reconstruction using random measurements with known camera positions, PSNR 24.7dB.*

Haar wavelet basis. (We choose Haar due to its similarity with the noiselet basis, but the performance is similar using other wavelet bases.) This is a rough approximation for how a non-CS transform coder might encode the image, and for the encoding of a single image in isolation, this is typically a more efficient encoding strategy than using random measurements. (Recall that for reconstructing a single signal, one must encode about $K \log(N/K)$ random measurements to obtain an approximation comparable to K-term transform coding.) However, when we concatenate the ensemble of encoded wavelet coefficients and solve (2.1) to estimate $\widehat{x}$, we see from the result in Figure 4.6(b) that the reconstructed image has lower quality than we obtained using a manifold lifting algorithm based on random measurements, even though the camera positions were *unknown* for the manifold lifting experiment. In a sense, by using joint decoding, we have reduced the CS overmeasuring factor from its familiar value of $\log(N/K)$ down to something below 1! We believe this occurs primarily because the images $\{x_j\}$ are highly correlated, and the repeated encoding of large wavelet coefficients (which tend to concentrate at coarse scales) results in repeated encoding of redundant information across the multiple satellites. In other words, it is highly likely that prominent features will be encoded by many satellites over and over again, whereas other features may not be encoded at all. As a result, by examining Figure 4.6(b) we see that strong features such as streets and the edges of buildings (which have large wavelet coefficients) are relatively more accurately reconstructed than, for example, forests or cars in parking lots (which have smaller wavelet coefficients). Random measurements capture more diverse information within and among the images. To more clearly illustrate the specific benefit that random measurements provide over transform coding (for which the camera positions were known), we show in Figure 4.6(c) a reconstruction obtained using random measurements with known camera positions.

J	Independent CS	TC w/ cam.	ML w/ cam.	ML w/out cam.	
				PSNR	$\frac{1}{J}\sum_j \lvert\theta_j - \widehat{\theta}_j\rvert$
200	14.4	22.8	24.7	23.6	(0.0108, 0.0132)
150	13.7	22.9	24.6	23.7	(0.0110, 0.0148)
100	15.1	23.5	25.1	23.9	(0.0177, 0.0121)
70	15.6	23.7	24.6	23.8	(0.0059, 0.0143)

Table 4.1: *Reconstruction results with varying numbers of camera positions J. From left to right, the columns correspond to the PSNR (in dB) of image-by-image CS reconstruction from random measurements, joint reconstruction from transform coding measurements with known camera positions, joint reconstruction from random measurements with known camera positions, and joint reconstruction from random measurements with unknown camera positions. The final subcolumn lists the mean absolute error of the estimated camera positions in the vertical and horizontal directions, respectively.*

Finally, we carry out a series of simulations with the same image x using different numbers J of camera positions. We keep the total number of measurements (19200) and the sizes of the subimages (64×64) constant. The results are summarized in Table 4.1. In all cases, our manifold lifting algorithm without knowledge of the camera positions outperforms transform coding with knowledge of the camera positions. We do note that as J decreases, the performance of transform coding improves. This is likely because each satellite now has more measurements to devote to encoding information about the underlying scene, and there are fewer total cameras to encode redundant information.

CHAPTER 5

Concentration of Measure for Block Diagonal Matrices with Applications to Compressive Signal Processing

5.1 Introduction

Recent technological advances have enabled the sensing and storage of massive volumes of data from a dizzying array of sources. While access to such data has revolutionized fields such as signal processing, the limits of some computing and storage resources are being tested, and front-end signal acquisition devices are not always able to support the desire to measure in increasingly finer detail. To confront these challenges, many signal processing researchers have begun investigating compressive linear operators $\Phi : \mathbb{R}^N \to \mathbb{R}^M$ for high resolution signals $x \in \mathbb{R}^N$ ($M < N$), either as a method for simple dimensionality reduction or as a model for novel data acquisition devices [52, 26]. Because of their universality and amenability to analysis, randomized compressive linear operators (i.e., random matrices with $M < N$) have drawn particular interest.

The theoretical analysis of random matrices often relies on the general notions that these matrices are well-behaved most of the time and that we can bound the probability with which they perform poorly. Frequently, these notions are formalized using some form of the *concentration of measure phenomenon* [87], a powerful characterization of the tendency of certain functions of high-dimensional random processes to concentrate sharply around their mean. As one important example of this phenomenon, it is known that for any fixed signal $x \in \mathbb{R}^N$, if Φ is an $M \times N$ matrix populated with independent and identically distributed (i.i.d.) random entries drawn from a suitable distribution, then with high probability Φ will approximately preserve the norm of x. More precisely, for many random distributions for Φ, the probability that $\left| \|\Phi x\|_2^2 - \|x\|_2^2 \right|$ will exceed a small fraction

This work is in collaboration with Han Lun Yap, Christopher J. Rozell, and Michael B. Wakin. Jae Young Park and Han Lun Yap are both primary authors to this work that appeared in [107]. This work also builds upon previous work that appeared in [140, 113] in collaboration with the same authors.

of $\|x\|_2^2$ decays exponentially in the number of measurements M.

As we discuss in Section 5.2, such concentration results have a number of favorable implications. Among these is the Johnson-Lindenstrauss (JL) lemma [81, 1, 40], which states that when applied to a finite set of points $Q \subset \mathbb{R}^N$, a randomized compressive operator Φ can provide a stable, distance preserving embedding of Q in the measurement space $\mathbb{R}^M$. This enables the efficient solution of a broad variety of signal processing problems by permitting these problems to be solved in the low-dimensional observation space (such as finding the nearest neighbor to a point x in a database Q). Such concentration results have also been used to prove that certain families of random matrices can satisfy the Restricted Isometry Property (RIP) [30, 14, 101] (see Definition 2.3). In the field of Compressed Sensing (CS), the RIP is commonly used as a sufficient condition to guarantee that a sparse signal x can be recovered from the measurements Φx.

Despite the utility of norm preservation in dimensionality reduction, concentration analysis to date has focused almost exclusively on dense matrices that require each measurement to be a weighted linear combination of all entries of x. Dense random matrices are often either impractical because of the resources required to store and work with a large unstructured matrix (e.g., one with i.i.d. entries), or unrealistic as models of acquisition devices with architectural constraints preventing such global data aggregation. For example, in a distributed sensing system, communication constraints may limit the dependence of each measurement to only a subset of the data. For a second example, applications involving streaming signals [9, 23] often have datarates that necessitate operating on local signal blocks rather than the entire signal simultaneously.

In such scenarios, the data may be divided naturally into discrete subsections (or blocks), with each block acquired via a local measurement operator. To see the implications of this, let us model a signal $x \in \mathbb{R}^{NJ}$ as being partitioned into J blocks $x_1, x_2, \ldots, x_J \in \mathbb{R}^N$, and for each $j \in \{1, 2, \ldots, J\}$, suppose that a local measurement operator $\Phi_j : \mathbb{R}^N \to \mathbb{R}^{M_j}$ collects the measurements $y_j = \Phi_j x_j$. Concatenating all of the measurements into a vector $y \in \mathbb{R}^{\sum_j M_j}$, we then have

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_J \end{bmatrix}}_{y:\ (\sum_j M_j)\times 1} = \underbrace{\begin{bmatrix} \Phi_1 & & & \\ & \Phi_2 & & \\ & & \ddots & \\ & & & \Phi_J \end{bmatrix}}_{\Phi:\ (\sum_j M_j)\times NJ} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_J \end{bmatrix}}_{x:NJ\times 1}. \tag{5.1}$$

In cases such as these, we see that the overall measurement operator Φ will have a characteristic block diagonal structure. In some scenarios, the local measurement operator Φ_j may be unique for each block, and we say that the resulting Φ has a *Distinct Block Diagonal* (DBD) structure. In other scenarios it may be appropriate or necessary to repeat a single operator across all blocks (such that $\Phi_1 = \Phi_2 = \cdots = \Phi_J$); we call the resulting Φ a *Repeated Block Diagonal* (RBD) matrix.

Starting from this block diagonal matrix structure, the main contributions of this chapter are a derivation of concentration of measure bounds for DBD and RBD matrices and an extensive exploration of the implications and utility of these bounds for the signal processing community. Specifically, in Section 5.3 we derive concentration of measure bounds both for DBD matrices populated with i.i.d. subgaussian[1] random variables and for RBD matrices populated with i.i.d. Gaussian random variables. In contrast to the signal agnostic concentration of measure bounds for i.i.d. dense matrices, these bounds are signal dependent; in particular, the probability of concentration depends on the "diversity" of the component signals $x_1, x_2, \ldots, x_J$ being well-matched to the measurement matrix (we make this precise in Section 5.3). As our analytic discussion and supporting simulations show, these measures of diversity have clear intuitive interpretations and indicate that, for signals with the most favorable characteristics, the concentration of measure probability for block diagonal matrices can scale exactly as for an i.i.d. dense random matrix.

Sections 5.4 and 5.5 are devoted to a detailed investigation of the utility of these non-uniform concentration results for signal processing practitioners. Specifically, in Section 5.4, we extend our concentration results to formulate a modified version of the JL lemma appropriate for block diagonal matrices. We also explain how this lemma can be used to guarantee the performance of various compressive-domain signal inference and processing algorithms such as signal detection and estimation. Given the applicability of these results for providing performance guarantees in these tasks, a natural question is whether there are large classes of signals that have the diversity required to make block diagonal matrices perform well. In Section 5.5 we provide several examples of signal families that are particularly favorable for measurement via DBD or RBD matrices.

5.2 Background and Related Work

In this section, we begin with a definition of subgaussian random variables, describe more formally several existing concentration of measure results for random matrices, and review some standard applications of these results in the literature.

[1] Subgaussian random variables [25, 132] are precisely defined in Section 5.2.1, and can be thought of as random variables from a distribution with tails that can be bounded by a Gaussian.

5.2.1 Subgaussian Random Variables

In fields such as CS, the Gaussian distribution is often invoked for probabilistic analysis thanks to its many convenient properties. Gaussian random variables, however, are just one special case in a much broader class of *subgaussian* distributions [25, 132]. Where possible, we state our results in terms of subgaussian random variables, which are defined below using standard notation from the literature.

Definition 5.1. *[132]* A random variable W is subgaussian if $\exists a > 0$ such that

$$(\mathbf{E}|W|^p)^{1/p} \leq a\sqrt{p} \text{ for all } p \geq 1.$$

The quantity $\|W\|_{\psi_2} := \sup_{p\geq 1} p^{-1/2}(\mathbf{E}|W|^p)^{1/p}$ is known as the subgaussian norm of W.

We restrict our attention to zero-mean subgaussian random variables in this chapter. Examples of such random variables include zero-mean Gaussian random variables, ± 1 Bernoulli random variables (each value with probability $\frac{1}{2}$), and uniform random variables on $[-1, 1]$.

For a given subgaussian random variable W, the variance $\mathrm{Var}(W)$ is a constant multiple of $\|W\|_{\psi_2}^2$ with $0 \leq \frac{\mathrm{Var}(W)}{\|W\|_{\psi_2}^2} \leq \sqrt{2}$; the exact value of $\frac{\mathrm{Var}(W)}{\|W\|_{\psi_2}^2}$ depends on the specific distribution for W under consideration (Gaussian, Bernoulli, etc.). We also note that in some cases, we will consider independent realizations of a subgaussian random variable W that are normalized to have different variances. However, it is useful to note that for any scalar $\alpha > 0$, $\frac{\mathrm{Var}(\alpha W)}{\|\alpha W\|_{\psi_2}^2} = \frac{\alpha^2 \mathrm{Var}(W)}{\alpha^2 \|W\|_{\psi_2}^2} = \frac{\mathrm{Var}(W)}{\|W\|_{\psi_2}^2}$.

5.2.2 Concentration Inequalities

Concentration analysis to date has focused almost exclusively on dense random matrices populated with i.i.d. entries drawn from some distribution. Commonly, when Φ has size $M \times N$ and the entries are drawn from a suitably normalized distribution, then for any fixed signal $x \in \mathbb{R}^N$ the goal is to prove for any $\epsilon \in (0, 1)$ that

$$P(\left|\|\Phi x\|_2^2 - \|x\|_2^2\right| > \epsilon \|x\|_2^2) \leq 2e^{-Mc_0(\epsilon)}, \tag{5.2}$$

where $c_0(\epsilon)$ is some constant (depending on ϵ) that is typically on the order of ϵ^2. When discussing bounds such as (5.2) where the probability of failure scales as e^{-X}, we refer to X as the *concentration exponent*.

The past several years have witnessed the derivation of concentration results for a variety of (ultimately related) random distributions for Φ. A uniform concentration result of the form (5.2)

was originally derived for dense Gaussian matrices populated with entries having mean zero and variance $\frac{1}{M}$ [76]; one straightforward derivation of this uses standard tail bounds for chi-squared random variables [1]. Using slightly more complicated arguments, similar concentration results were then derived for Bernoulli matrices populated with random $\frac{\pm 1}{\sqrt{M}}$ entries (each with probability $\frac{1}{2}$) and for a "database-friendly" variant populated with random $\{\frac{3}{\sqrt{M}}, 0, -\frac{3}{\sqrt{M}}\}$ entries (with probabilities $\{\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\}$) [1]. Each of these distributions, however, is itself subgaussian, and more recently it has been shown that uniform concentration results of the form (5.2) in fact holds for *all* subgaussian distributions having variance $\frac{1}{M}$ [101, 47].[2] Moreover, it has been shown that a subgaussian distribution is actually necessary for deriving a uniform concentration result of the form (5.2) for a dense random matrix populated with i.i.d. entries [47].

Concentration inequalities have also been derived in the literature for certain structured (non-i.i.d.) dense random matrices. Examples include matrices populated with random orthogonal rows [65] or matrices constructed by combining a structured matrix with certain random operations [5]. A concentration bound also holds for the randomized RIP matrices [6, 85] that we discuss in the final paragraph of Section 5.2.3 below.

5.2.3 Applications of Concentration Inequalities

While nominally a concentration result of the form (5.2) appears to guarantee only that the norm of a particular signal x is preserved in the measurements Φx, in fact such a result can be used to guarantee that the information required to discriminate x from *other signals* may actually be preserved in Φx. Indeed, one of the prototypical applications of a concentration result of the form (5.2) is to prove that with high probability, Φ will preserve distances between various signals of interest.

Definition 5.2. Consider two sets $U, V \subset \mathbb{R}^N$. We say that a mapping Φ provides a *stable embedding of* (U, V) *with conditioning* δ if

$$(1-\delta)\|u-v\|_2^2 \leq \|\Phi(u-v)\|_2^2 \leq (1+\delta)\|u-v\|_2^2 \tag{5.3}$$

holds for all $u \in U$ and $v \in V$.

To relate this concept to norm preservation, note that for a fixed $x \in \mathbb{R}^N$, a randomized operator that satisfies (5.2) will, with high probability, provide a stable embedding of $(\{x\}, \{0\})$ with

[2] This fact also follows from our Theorem 5.3 by considering the special case where $J = 1$.

conditioning ϵ. As we discuss in Section 5.4, this fact is useful for studying the performance of a compressive-domain signal detector [43].

However, much richer embeddings may also be considered. For example, suppose that a signal family of interest $Q \subset \mathbb{R}^N$ consists of a finite number of points. If a randomized operator Φ satisfies (5.2) for each vector of the form $u - v$ for $u, v \in Q$, then it follows from a simple union bound that with probability at least $1 - 2|Q|^2 e^{-Mc_0(\epsilon)}$, Φ will provide a stable embedding of (Q, Q) with conditioning ϵ. From this fact one obtains the familiar JL lemma [81, 1, 40], which states that this stable embedding will occur with high probability if $M = O\left(\frac{\log(|Q|)}{c_0(\epsilon)}\right)$. Thus, the information required to discriminate between signals in Q is preserved in the compressive measurements. This fact is useful for studying the performance of a compressive-domain nearest-neighbor search [76], a compressive-domain signal classifier [43], and various other signal inference strategies that we discuss in Section 5.4.

In certain cases (particularly when dealing with randomized operators that satisfy (5.2) uniformly over all signals $x \in \mathbb{R}^N$), it is possible to significantly extend embedding results far beyond the JL lemma. For example, for a set Q consisting of all signals with sparsity K in some basis for $\mathbb{R}^N$, one can couple the above union bound approach with some elementary covering arguments [14, 101] to show that if $M = O\left(K \log(N/K)\right)$, then Φ will provide a stable embedding of (Q, Q) with high probability. This guarantee is known as the RIP (see Definition 2.3) in CS, and from the RIP, one can derive deterministic bounds on the performance of CS recovery algorithms such as ℓ_1 minimization [27]. Concentration of measure type results have also been used to prove the RIP for random matrices with subexponential columns [2], and a concentration result of the form (5.2) has also been used to probabilistically analyze the performance of ℓ_1 minimization [47]. Finally, we note that one can also generalize sparsity-based embedding arguments to the case where Q is a low-dimensional submanifold of $\mathbb{R}^N$ [15].

In a different direction of interest regarding stable embeddings of finite collections of points, we note that several authors have also recently shown that the direction of implication between the JL lemma and the RIP can be reversed. Specifically, it has been shown [6, 85] that randomizing the column signs of a matrix that satisfies the RIP results in a matrix that also satisfies the JL lemma. One of the implications of this is that a computationally efficient stable embedding can be achieved by randomizing the column signs of a partial Fourier matrix [31, 72, 114].

5.3 Non-uniform Concentration of Measure Inequalities

In this section we state our concentration of measure results for Distinct Block Diagonal (DBD) and Repeated Block Diagonal (RBD) matrices. For each type of matrix we provide a detailed examination of the derived concentration rates and use simulations to demonstrate that our results do indeed capture the salient signal characteristics that affect the concentration probability. We also discuss connections between the concentration probabilities for the two matrix types.

5.3.1 Distinct Block Diagonal (DBD) Matrices

5.3.1.1 Analytical Results

Before stating our first result, we define the requisite notation. For a given signal $x \in \mathbb{R}^{NJ}$ partitioned into J blocks of length N as in (5.1), we define a vector describing the energy distribution across the blocks of x: $\gamma = \gamma(x) := \left[\|x_1\|_2^2 \ \|x_2\|_2^2 \ \cdots \ \|x_J\|_2^2\right]^T \in \mathbb{R}^J$. Also, letting $M_1, M_2, \ldots, M_J$ denote the number of measurements to be taken of each block, we define a $J \times J$ diagonal matrix containing these numbers along the diagonal: $\mathbf{M} := \text{diag}(M_1, M_2, \ldots, M_J)$. Finally, for a given signal $x \in \mathbb{R}^{NJ}$ and measurement allocation $\mathbf{M}$, we define the quantities

$$\Gamma_2(x, \mathbf{M}) := \frac{\|\gamma\|_1^2}{\|\mathbf{M}^{-1/2}\gamma\|_2^2} = \frac{\left(\sum_{j=1}^J \|x_j\|_2^2\right)^2}{\sum_{j=1}^J \frac{\|x_j\|_2^4}{M_j}}, \ \Gamma_\infty(x, \mathbf{M}) := \frac{\|\gamma\|_1}{\|\mathbf{M}^{-1}\gamma\|_\infty} = \frac{\sum_{j=1}^J \|x_j\|_2^2}{\max_j \frac{\|x_j\|_2^2}{M_j}}. \tag{5.4}$$

Using this notation, our first result concerning the concentration of DBD matrices is captured in the following theorem.

Theorem 5.3. *Suppose $x \in \mathbb{R}^{NJ}$, and for each $j \in \{1, 2, \ldots, J\}$ suppose that $M_j > 0$. Let ϕ denote a subgaussian random variable with mean 0, variance 1, and subgaussian norm $\|\phi\|_{\psi_2}$. Let $\{\Phi_j\}_{j=1}^J$ be random matrices drawn independently, where each Φ_j has size $M_j \times N$ and is populated with i.i.d. realizations of the renormalized random variable $\frac{\phi}{\sqrt{M_j}}$, and let Φ be a $\left(\sum_{j=1}^J M_j\right) \times NJ$ DBD matrix composed of $\{\Phi_j\}_{j=1}^J$ as in (5.1). Then*

$$P\left(\left|\|\Phi x\|_2^2 - \|x\|_2^2\right| > \epsilon\|x\|_2^2\right) \leq 2\exp\left\{-C_1 \min\left(\frac{C_2^2\epsilon^2}{\|\phi\|_{\psi_2}^4}\Gamma_2(x, \mathbf{M}), \frac{C_2\epsilon}{\|\phi\|_{\psi_2}^2}\Gamma_\infty(x, \mathbf{M})\right)\right\}, \tag{5.5}$$

where C_1 and C_2 are absolute constants.

Proof: See Appendix A.

From the tail bound (5.5), it is easy to deduce that the concentration probability of interest

decays exponentially as a function of $\epsilon^2\Gamma_2(x,\mathbf{M})$ in the case where $0 \leq \epsilon \leq \frac{\|\phi\|_{\psi_2}^2\Gamma_\infty(x,\mathbf{M})}{C_2\Gamma_2(x,\mathbf{M})}$ and exponentially as a function of $\epsilon\Gamma_\infty(x,\mathbf{M})$ in the case where $\epsilon > \frac{\|\phi\|_{\psi_2}^2\Gamma_\infty(x,\mathbf{M})}{C_2\Gamma_2(x,\mathbf{M})}$. One striking thing about Theorem 5.3 is that, in contrast to analogous concentration of measure results for dense matrices with i.i.d. subgaussian entries,[3] the concentration rate depends explicitly on the signal x being measured.

To elaborate on this point, since we are frequently concerned in practice with applications where ϵ is small, let us focus on the first case of (5.5), when the concentration exponent scales with $\Gamma_2(x,\mathbf{M})$. In this case, we see that larger values of $\Gamma_2(x,\mathbf{M})$ promote sharper concentration of $\|\Phi x\|_2^2$ about its mean $\|x\|_2^2$. Using elementary inequalities relating the ℓ_1 and ℓ_2 norms, one can bound the range of possible Γ_2 values by $\min_j M_j \leq \Gamma_2(x,\mathbf{M}) \leq \sum_{j=1}^J M_j$. The worst case, $\Gamma_2(x,\mathbf{M}) = \min_j M_j$, is achieved when all of the signal energy is concentrated into exactly one signal block where the fewest measurements are collected, i.e., when $\|x_j\|_2^2 = 0$ except for a single index $j' \in \{\arg\min_j M_j\}$ (where $\{\arg\min_j M_j\}$ is the set of indices where $\{M_j\}$ is minimum). In this case the DBD matrix exhibits significantly worse performance than a dense i.i.d. matrix of the same size $(\sum_{j=1}^J M_j)\times NJ$, for which the concentration exponent would scale with $\sum_{j=1}^J M_j$. This makes intuitive sense, as this extreme case would correspond to only one block of the DBD matrix sensing all of the signal energy. On the other hand, the best case, $\Gamma_2(x,\mathbf{M}) = \sum_{j=1}^J M_j$, is achieved when the number of measurements collected for each block is proportional to the signal energy in that block. In other words, letting $\mathrm{diag}(\mathbf{M})$ represent the diagonal of $\mathbf{M}$, when $\mathrm{diag}(\mathbf{M}) \propto \gamma$ (i.e., when $\mathrm{diag}(\mathbf{M}) = C\gamma$ for some constant $C > 0$) the concentration exponent scales with $\sum_{j=1}^J M_j$ just as it would for a dense i.i.d. matrix of the same size. This is in spite of the fact that the DBD matrix has many fewer nonzero elements.

The probability of concentration behaves similarly in the second case of (5.5), where the concentration exponent scales with $\Gamma_\infty(x,\mathbf{M})$. One can bound the range of possible Γ_∞ values by $\min_j M_j \leq \Gamma_\infty(x,\mathbf{M}) \leq \sum_{j=1}^J M_j$. The lower bound is again achieved when $\|x_j\|_2^2 = 0$ except for a single index $j' \in \{\arg\min_j M_j\}$, and the upper bound again is achieved when $\mathrm{diag}(\mathbf{M}) \propto \gamma$.

The above discussion makes clear that the concentration performance of a DBD matrix can vary widely depending on the signal being measured. In particular, DBD matrices can perform as well as dense i.i.d. matrices if their measurement allocation is well matched to the energy distribution of the signal. Such a favorable event can occur either (i) by design, if a system designer has some operational knowledge of the energy distributions to expect, or (ii) by good fortune, if favorable

[3] The uniformity of such concentration results comes not from the fact that these matrices are dense but rather that that are populated with i.i.d. random variables; certain structured dense random matrices (such as partial Fourier matrices) could have signal-dependent concentration inequalities.

signals happen to arrive that are well matched to a fixed system design. We note that even in the former situation when the general energy distribution across blocks is known, this does not imply that the designer has a priori knowledge of the signal being sensed. Furthermore, even when significant information about the signal (or a finite class of signals) is known, there may still be much to learn by actually measuring the signal. For example, Section 5.4 outlines several interesting signal inference problems that benefit from a norm-preservation guarantee for a known signal (or finite signal family). Also, in the second of these situations, it may not be unreasonable to expect that a fixed measurement allocation will be well matched to an unknown signal most of the time. For example, in Section 5.5 we describe several realistic signal classes that are favorably matched to fixed systems that have equal measurement allocations ($M_1 = M_2 = \cdots = M_J$).

Two final comments are in order. First, while Theorem 5.3 was derived by considering all signal blocks to be of equal length N, one can see by a close examination of the proof that the same theorem in fact holds for signals partitioned into blocks of unequal lengths. Second, it is instructive to characterize the range of ϵ for which the two cases of Theorem 5.3 are relevant; we do so in the following lemma, which can be proved using standard manipulations of the ℓ_1, ℓ_2, and ℓ_∞ norms.

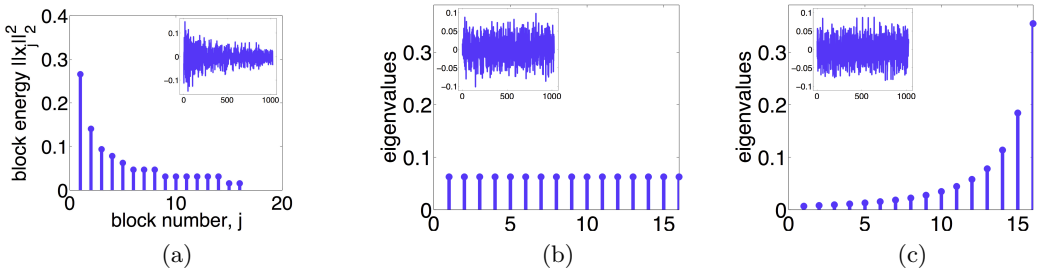
Lemma 5.4. *If $J \geq 2$, the point of demarcation between the two cases of Theorem 5.3 obeys*

$$\frac{\|\phi\|_{\psi_2}^2 \cdot 2(\sqrt{J}-1)}{C_2(J-1)} \frac{\min_j \sqrt{M_j}}{\max_j \sqrt{M_j}} \leq \frac{\|\phi\|_{\psi_2}^2 \Gamma_\infty(x, \mathbf{M})}{C_2 \Gamma_2(x, \mathbf{M})} \leq \frac{\|\phi\|_{\psi_2}^2}{C_2}.$$

Examining the bound above, we note that for $J \geq 2$ it holds that $\frac{2(\sqrt{J}-1)}{J-1} \geq \frac{1}{\sqrt{J}}$. Thus, as an example, when $M_1 = M_2 = \cdots = M_J$, the first ("small ϵ") case of Theorem 5.3 is guaranteed to at least permit $\epsilon \in \left[0, \frac{\|\phi\|_{\psi_2}^2}{C_2\sqrt{J}}\right]$. We note further that when the measurement matrix is well-matched to the signal characteristics, the first case of Theorem 5.3 permits ϵ as large as $\frac{\|\phi\|_{\psi_2}^2}{C_2}$, which is independent of J.

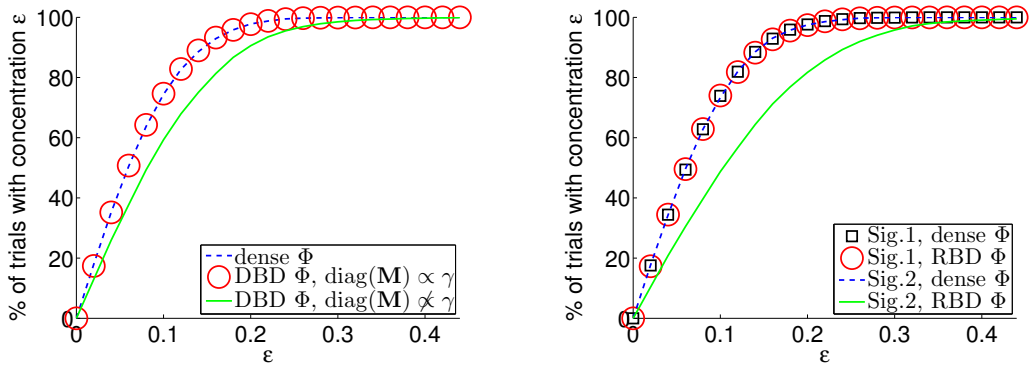
5.3.1.2 Supporting Experiments

While the quantity $\Gamma_2(x, \mathbf{M})$ plays a critical role in our analytical tail bound (5.5), it is reasonable to ask whether this quantity actually plays a central role in the empirical concentration performance of DBD matrices. We explore this question with a series of simulations. To begin, we randomly construct a signal of length 1024 partitioned into $J = 16$ blocks of length $N = 64$. The energy distribution γ of the signal x is plotted in Figure 5.1(a) (and the signal x itself is plotted in the top right corner). For this simulation, to ensure $\text{diag}(\mathbf{M}) \propto \gamma$ with integer values for the M_j, we



(a) (b) (c)

Figure 5.1: *(a) Test signal for concentration in a DBD matrix. The main panel plots the energy distribution γ when the signal is partitioned into $J = 16$ blocks of length $N = 64$; the subpanel plots the length-1024 signal x itself. (b),(c) Test signals for concentration in a RBD matrix. The main panels plot the eigenvalue distributions λ for Sig. 1 and Sig. 2, respectively, when partitioned into $J = 16$ blocks of length $N = 64$; the subpanels plot the length-1024 signals themselves.*



(a) (b)

Figure 5.2: *Comparisons of concentration behavior for various matrix and signal types; each figure shows the percentage of trials for which $(1-\epsilon) \leq \|\Phi x\|_2/\|x\|_2 \leq (1+\epsilon)$ as a function of ϵ. (a) Performance of the dense and DBD matrices on the signal shown in Figure 5.1(a). (b) Performance of the dense and RBD matrices on the signals shown in Figure 5.1(b),(c).*

begin by constructing $\mathbf{M}$ (populated with integers) and then normalize each block of a randomly generated signal to set γ accordingly.

Fixing this signal x, we generate a series of 10000 random 64×1024 matrices Φ using zero-mean Gaussian random variables for the entries. In one case, the matrices are fully dense and the entries of each matrix have variance $1/64$. In another case, the matrices are DBD with $\mathrm{diag}(\mathbf{M}) \propto \gamma$ and the entries in each block have variance $1/M_j$. Thus, we have $\Gamma_2(x, \mathbf{M}) = \sum_{j=1}^{J} M_j$ and our Theorem 5.3 gives the same concentration bound for this DBD matrix as for the dense i.i.d. matrix of the same size. For each type of matrix, Figure 5.2(a) shows the percentage of trials for which $(1-\epsilon) \leq \|\Phi x\|_2/\|x\|_2 \leq (1+\epsilon)$ as a function of ϵ, and indeed, the curves for the dense and DBD matrices are indistinguishable.

Finally, we consider a third scenario in which we construct 10000 random 64×1024 DBD matrices as above but with an equal number of measurements in each block. In other words, we set all $M_j = 4$, and obtain measurement matrices that are no longer matched to the signal energy distribution. We

quantify this mismatch by noting that $\Gamma_2(x, 4 \cdot I_{J\times J}) = 32.77 < \sum_{j=1}^{J} M_j$. Again, Figure 5.2(a) shows the concentration success probability over these 10000 random matrices. It is evident that these mismatched DBD matrices provide decidedly less sharp concentration of $\|\Phi x\|_2$.

5.3.2 Repeated Block Diagonal (RBD) Matrices

5.3.2.1 Analytical Results

We now turn our attention to the concentration performance of the more restricted RBD matrices. Before stating our result, let us again define the requisite notation. Given a signal $x \in \mathbb{R}^{NJ}$ partitioned into J blocks of length N, we define the $J \times N$ matrix of concatenated signal blocks

$$X := [x_1 \; x_2 \; \cdots \; x_J]^T, \tag{5.6}$$

and we denote the non-negative eigenvalues of the $N \times N$ symmetric matrix $A = X^T X$ as $\{\lambda_i\}_{i=1}^{N}$. We let $\lambda = \lambda(x) := [\lambda_1, \ldots, \lambda_N]^T \in \mathbb{R}^N$ be the vector composed of these eigenvalues. We let $M := M_1 = M_2 = \cdots = M_J$ denote the number of measurements to be taken in each block. Finally, for a given signal $x \in \mathbb{R}^{NJ}$ and per-block measurement rate M, we define the quantities

$$\Lambda_2(x, M) := \frac{M\|\lambda\|_1^2}{\|\lambda\|_2^2} \text{ and } \Lambda_\infty(x, M) := \frac{M\|\lambda\|_1}{\|\lambda\|_\infty}. \tag{5.7}$$

Equipped with this notation, our main result concerning the concentration of RBD matrices is as follows.

Theorem 5.5. *Suppose $x \in \mathbb{R}^{NJ}$. Let $\widetilde{\Phi}$ be a random $M \times N$ matrix populated with i.i.d. zero-mean Gaussian entries having variance $\sigma^2 = \frac{1}{M}$, and let Φ be an $MJ \times NJ$ block diagonal matrix as defined in* (5.1)*, with $\Phi_j = \widetilde{\Phi}$ for all j. Then*

$$P(\left|\|\Phi x\|_2^2 - \|x\|_2^2\right| > \epsilon\|x\|_2^2) \leq 2\exp\left\{-C_1 \min\left(C_3^2\epsilon^2\Lambda_2(x, M),\; C_3\epsilon\Lambda_\infty(x, M)\right)\right\}, \tag{5.8}$$

where C_1 and C_3 are absolute constants.

Proof: See Appendix B.

From (5.8), one can deduce that the concentration probability of interest decays exponentially as a function of $\epsilon^2\Lambda_2(x, M)$ in the case where $0 \leq \epsilon \leq \frac{\Lambda_\infty(x,M)}{C_3\Lambda_2(x,M)}$ and exponentially as a function of $\epsilon\Lambda_\infty(x, M)$ in the case where $\epsilon > \frac{\Lambda_\infty(x,M)}{C_3\Lambda_2(x,M)}$. Thus, we see that the concentration rate again depends explicitly on the signal x being measured.

Again, since we are frequently concerned in practice with applications where ϵ is small, let us focus on the first case of (5.8), when the concentration exponent scales with $\Lambda_2(x, M)$. It follows from the standard relation between ℓ_1 and ℓ_2 norms that $M \leq \Lambda_2(x, M) \leq M \min(J, N)$. One extreme, M, is achieved when $A = \sum_j x_j x_j^T$ has only one nonzero eigenvalue, implying that the blocks x_j are the same modulo a scaling factor. In this case, the RBD matrix exhibits significantly worse performance than a dense i.i.d. matrix of the same size $MJ \times NJ$, for which the concentration exponent would scale with MJ rather than M. However, this diminished performance is to be expected since the same $\widetilde{\Phi}$ is used to measure each identical signal block.

The other extreme, $\Lambda_2(x, M) = M \min(J, N)$ is favorable as long as $J \leq N$, in which case the concentration exponent scales with MJ just as it would for a dense i.i.d. matrix of the same size. For this case to occur, A must have J nonzero eigenvalues and they must all be equal. By noting that the nonzero eigenvalues of $A = X^T X$ are the same as those of the Grammian matrix $G = XX^T$, we conclude that this most favorable case can occur only when the signal blocks are mutually orthogonal and have the same energy. Alternatively, if the signal blocks span a K-dimensional subspace of $\mathbb{R}^N$ we will have $M \leq \Lambda_2(x, M) \leq MK$. All of this can also be seen by observing that calculating the eigenvalues of $A = X^T X$ is equivalent to running Principal Component Analysis (PCA) [77] on the matrix X comprised of the J signal blocks. Said another way, an RBD matrix performs as well as a dense i.i.d. matrix of the same size when the signal has uniform energy distribution across its blocks (as in the DBD case) *and* has sufficient variation in the directions exhibited by the blocks.

We note that there is a close connection between the diversity measures $\Gamma_2(x, \mathbf{M})$ and $\Lambda_2(x, M)$ that is not apparent at first glance. For a fair comparison, we assume in this discussion that $\mathbf{M} := \text{diag}(M, M, \ldots, M)$. Now, note that $\|\lambda\|_1^2 = \|\gamma\|_1^2$ and also that $\|\lambda\|_2^2 = \|A\|_F^2 = \|XX^T\|_F^2 = \sum_{i=1}^J \|x_i\|_2^4 + 2\sum_{i>j}(x_i^T x_j)^2 = \|\gamma\|_2^2 + 2\sum_{i>j}(x_i^T x_j)^2$. Using these two relationships, we can rewrite $\Lambda_2(x, M)$ as

$$\Lambda_2(x, M) = \frac{M\|\lambda\|_1^2}{\|\lambda\|_2^2} = \frac{M\|\gamma\|_1^2}{\|\gamma\|_2^2 + 2\sum_{i>j}(x_i^T x_j)^2} \leq \frac{M\|\gamma\|_1^2}{\|\gamma\|_2^2} = \Gamma_2(x, \mathbf{M}). \tag{5.9}$$

From this relationship we see that Λ_2 and Γ_2 differ only by the additional inner-product term in the denominator of Λ_2, and we also see that $\Lambda_2 = \Gamma_2$ if and only if the signal blocks are mutually orthogonal. This more stringent condition for RBD matrices—requiring more intrinsic signal diversity—is expected given the more restricted structure of the RBD matrices.

5.3.2.2 Supporting Experiments

While the quantity $\Lambda_2(x, M)$ plays a critical role in our analytical upper bound (5.8) on the concentration tail probabilities, it is reasonable to ask whether this quantity actually plays a central role in the empirical concentration performance of RBD matrices. We explore this question with a series of simulations. To begin, we randomly construct a signal of length 1024 partitioned into $J = 16$ blocks of length $N = 64$, and we perform Gram-Schmidt orthogonalization to ensure that the J blocks are mutually orthogonal and have equal energy. The nonzero eigenvalues of $A = X^T X$ are shown in the plot of λ in Figure 5.1(b) (and the signal x itself, denoted "Sig. 1", is plotted in the top left corner).

As we have discussed above, for signals such as Sig. 1 we should have $\Lambda_2(x, M) = MJ$, and Theorem 5.5 suggests that an RBD matrix can achieve the same concentration rate as a dense i.i.d. matrix of the same size. Fixing this signal, we generate a series of 10000 random 64×1024 matrices Φ populated with zero-mean Gaussian random variables. In one case, the matrices are dense and each entry has variance $1/64$. In another case, the matrices are RBD, with a single 4×64 block repeated along the main diagonal, comprised of i.i.d. Gaussian entries with variance $\frac{1}{4}$. For each type of matrix, Figure 5.2(b) shows the percentage of trials for which $(1-\epsilon) \leq \|\Phi x\|_2 / \|x\|_2 \leq (1+\epsilon)$ as a function of ϵ. As anticipated, we can see that the curves for the dense and RBD matrices are indistinguishable.

In contrast, we also construct a second signal x (denoted "Sig. 2") that has equal energy between the blocks but has non-orthogonal components (resulting in non-uniform λ); see Figure 5.1(c). This signal was constructed to ensure that the sorted entries of λ exhibit an exponential decay. Due to the non-orthogonality of the signal blocks, we see for this signal that $\Lambda_2(x, M) = 21.3284$ which is approximately 3 times less than the best possible value of $MJ = 64$. Consequently, Theorem 5.5 provides a much weaker concentration exponent when this signal is measured using an RBD matrix than when it is measured using a dense i.i.d. matrix. As shown in Figure 5.2(b), we see that the concentration performance of the full dense matrix is agnostic to this new signal structure, while the concentration is clearly not as sharp for the RBD matrix.

5.4 Applications

As discussed briefly in Section 5.2.3, a concentration of measure inequality—despite nominally pertaining to the norm preservation of a single signal—can lead to a number of guarantees for problems involving multi-signal embeddings and signal discrimination. In this section, we extend

our concentration bounds to formulate a modified version of the JL lemma appropriate for block diagonal matrices. We also survey a collection of compressive-domain inference problems (such as detection and classification) in which such a result can be leveraged. For simplicity we will focus on DBD matrices in this section, but parallel results can be derived in each case for RBD matrices. Given the nonuniform nature of our concentration bounds, the performance of algorithms for solving these problems will depend on the signals under consideration, and so, in Section 5.5 we provide several examples of signal classes that are particularly favorable for measurement via DBD or RBD matrices.

5.4.1 Stable Embeddings and the Johnson-Lindenstrauss Lemma

For a given signal $x \in \mathbb{R}^{NJ}$ and measurement allocation $\mathbf{M}$, let us define $\widetilde{\Gamma}_2(x, \mathbf{M}) := \frac{\Gamma_2(x,\mathbf{M})}{\sum_{j=1}^J M_j}$ and $\widetilde{\Gamma}_\infty(x, \mathbf{M}) := \frac{\Gamma_\infty(x,\mathbf{M})}{\sum_{j=1}^J M_j}$. Note that both quantities are upper bounded by 1, with equality achieved for signals best matched to $\mathbf{M}$ as discussed in Section 5.3.1. Using this notation, Theorem 5.3 allows us to immediately formulate a version of the JL lemma appropriate for DBD matrices.

Theorem 5.6. *Let U, V be two finite subsets of $\mathbb{R}^{NJ}$, let Φ be a randomly generated DBD matrix as described in Theorem 5.3 with measurement allocation $\mathbf{M}$, and let $\rho \in (0, 1)$ be fixed. If*

$$\sum_{j=1}^J M_j \geq \frac{\log|U| + \log|V| + \log(2/\rho)}{C_1 \min\left(\frac{C_2^2\delta^2}{\|\phi\|_{\psi_2}^4} \min_{u\in U, v\in V} \widetilde{\Gamma}_2(u-v, \mathbf{M}), \frac{C_2\delta}{\|\phi\|_{\psi_2}^2} \min_{u\in U, v\in V} \widetilde{\Gamma}_\infty(u-v, \mathbf{M})\right)}, \tag{5.10}$$

then with probability exceeding $1-\rho$, Φ will provide a stable embedding of (U, V) with conditioning δ. Alternatively, under the same conditions, with probability exceeding $1-\rho$ the matrix Φ will provide a stable embedding of (U, V) with conditioning

$$\begin{aligned} &\widetilde{\delta}(U, V, \mathbf{M}, \rho) \\ &\quad := \frac{\|\phi\|_{\psi_2}^2}{C_2} \max\left(\sqrt{\frac{\log|U| + \log|V| + \log(2/\rho)}{C_1 \min_{u\in U, v\in V} \Gamma_2(u-v, \mathbf{M})}}, \frac{\log|U| + \log|V| + \log(2/\rho)}{C_1 \min_{u\in U, v\in V} \Gamma_\infty(u-v, \mathbf{M})}\right). \end{aligned} \tag{5.11}$$

Proof. Taking the union bound over all $u \in U$ and $v \in V$ and using (5.5), we then know that the desired (near) isometry holds over all difference vectors $u - v$ except with probability bounded by

$$2|U||V| \exp\left(-C_1 \min\left(\frac{C_2^2\delta^2}{\|\phi\|_{\psi_2}^4} \min_{u\in U, v\in V} \Gamma_2(u-v, \mathbf{M}), \frac{C_2\delta}{\|\phi\|_{\psi_2}^2} \min_{u\in U, v\in V} \Gamma_\infty(u-v, \mathbf{M})\right)\right). \tag{5.12}$$

Ensuring that (5.10) holds guarantees that the above failure probability is less than ρ. The bound in

(5.11) follows from (5.10) and the observation that $\min(a\delta^2, b\delta) = c$ implies that $\delta = \max\left(\sqrt{\frac{c}{a}}, \frac{c}{b}\right)$.

□

Similar theorems can be formulated for RBD matrices, as well as for stable embeddings of a signal subspace rather than just a finite family of signals. Equation (5.10) gives a lower bound on the total number of measurements to guarantee a stable embedding with conditioning δ. One can see that the denominator on the right-hand side will scale with $\delta^2 \cdot \min_{u\in U, v\in V} \widetilde{\Gamma}_2(u-v, \mathbf{M})$ when $0 \leq \delta \leq \frac{\|\phi\|_{\psi_2}^2 \min_{u\in U, v\in V} \widetilde{\Gamma}_\infty(u-v,\mathbf{M})}{C_2 \min_{u\in U, v\in V} \widetilde{\Gamma}_2(u-v,\mathbf{M})}$ and with $\delta \cdot \min_{u\in U, v\in V} \widetilde{\Gamma}_\infty(u-v, \mathbf{M})$ when $\delta > \frac{\|\phi\|_{\psi_2}^2 \min_{u\in U, v\in V} \widetilde{\Gamma}_\infty(u-v,\mathbf{M})}{C_2 \min_{u\in U, v\in V} \widetilde{\Gamma}_2(u-v,\mathbf{M})}$. Thus, focusing just on cases where δ is small, in order to guarantee a stable embedding with a moderate number of measurements, we require $\widetilde{\Gamma}_2(u-v, \mathbf{M})$ to be sufficiently close to 1 over all $u \in U$ and $v \in V$. Equivalently, if $\Gamma_2(u-v, \mathbf{M})$ is uniformly close to $\sum M_j$ over all $u \in U$ and $v \in V$, the conditioning $\widetilde{\delta}$ provided in (5.11) is comparable to what would be achieved with a dense i.i.d. random matrix of the same size. In Section 5.5, we provide several examples of signal classes of U and V for which it may be reasonable to expect such uniformly favorable Γ_2 (or Λ_2) values.

The attentive reader may notice that the failure probability in (5.12) is in fact loose, since we have bounded the sum of the individual failure probabilities by $|U|\,|V|$ times the worse case failure probability. Due to the exponential form of these probabilities, however, it seems that the worse case probability—even if it is rare—will typically dominate this sum. Therefore, in most applications we do not expect that it is possible to significantly improve over the bounds provided in (5.12) and thus (5.10). Unfortunately, it appears that this fact would forbid the derivation of a sharp RIP bound for block diagonal matrices via the elementary covering arguments mentioned briefly in Section 5.2.3.[4] However, such an RIP result, which would guarantee the stable embedding of an infinite family of signals that are sparse in a particular basis, is not necessary for problems that require embeddings of only finite signal families or appropriate for problems where the signals may not all be sparse in the same basis (e.g., using the RIP to derive embedding guarantees such as Theorem 5.6 could require many more measurements than using a concentration bound).

Indeed, ensuring a stable embedding of even a finite signal class is very useful for guaranteeing the performance of many types of compressive-domain signal inference and processing algorithms. In the following subsections, we explore two canonical tasks (detection and classification) in detail to show how signal characteristics affect one's ability to solve these problems using measurements

[4] As an aside, since the submission of the corresponding journal paper [107] of this chapter, some of the authors (with an additional collaborator) have shown that using tools from the theory of empirical processes [114], it is possible to derive RIP bounds for DBD matrices [145] that are in fact dependent on the basis used for sparse representation of the signals. This does not make the present results obsolete, however: neither our concentration bounds nor our measurement bound (5.10) follow as a consequence of the RIP result.

acquired via a block diagonal matrix. Performing tasks such as these directly in the measurement space not only reduces the data acquisition burden but can also reduce the computational burden far below what is required to solve these problems in the high dimensional ambient signal space.

Before concluding this subsection, we briefly note that there are several other tasks (aside from detection and classification) that can be performed in the measurement space when a block diagonal matrix provides a stable embedding of a finite signal family. For one example, when a block diagonal matrix Φ provides a stable embedding of $(\mathcal{S}, \{x\})$ for some signal database $\mathcal{S}$ and query signal x, it is possible to solve the approximate nearest neighbor problem [76] (finding the closest point in $\mathcal{S}$ to x) in the compressed domain without much loss of precision. Another potential application in compressive signal processing involves a simple compressive-domain linear estimator [43]. When Φ provides a stable embedding of $(\mathcal{L}, \mathcal{X} \cup -\mathcal{X})$ for some sets $\mathcal{L}$ and $\mathcal{X}$, then for any $\ell \in \mathcal{L}$ and $x \in \mathcal{X}$, we can estimate the value of $\langle \ell, x \rangle$ from the measurements $\langle \Phi\ell, \Phi x \rangle$. Signal families $\mathcal{L}$ and $\mathcal{X}$ whose sum and difference vectors $\ell \pm x$ have favorable Γ_2 values will have favorable and predictable estimation performance. Finally, a similar result also discussed in [43] shows that *filtering* vectors in order to separate signal and interference subspaces is possible when the difference vectors between these subspaces are stably embedded by Φ.

5.4.2 Signal Detection in the Compressed Domain

While the canonical CS results center mostly around reconstructing signals from compressive measurements, there is a growing interest in forgoing this recovery process and answering certain signal processing questions directly in the compressed domain. One such problem that can be solved is binary detection, where one must decide whether or not a known template signal was present when the noisy compressive measurements were collected [74, 44, 43, 116]. In particular, let $s \in \mathbb{R}^{NJ}$ denote a known signal, and suppose that from the measurement vector y, we wish to decide between two hypotheses:

$$\mathcal{H}_0 : \ y = z \text{ or } \mathcal{H}_1 : \ y = \Phi s + z,$$

where Φ is a known compressive measurement matrix, and z is a vector of i.i.d. zero-mean Gaussian noise with variance σ^2. If one were designing a measurement matrix specifically for the purpose of detecting this signal, then the optimal choice of Φ would be the matched filter, i.e., $\Phi = s^T$. However, implementing a measurement system that is designed specifically for a particular s restricts its capabilities to detecting that signal only, which could require a hardware modification every time

s changes. A more generic approach would be to design Φ randomly (perhaps with a block diagonal structure out of necessity or due to efficiency considerations) and then use the acquired measurements y to test for one or more candidate signals s.

Given the measurements y, a Neyman-Pearson (NP) optimal detector [43] maximizes the probability of detection, $P_D := P\{\mathcal{H}_1 \text{ chosen } | \mathcal{H}_1 \text{ is true}\}$, subject to a given limitation on the probability of a false alarm, $P_F = P\{\mathcal{H}_1 \text{ chosen } | \mathcal{H}_0 \text{ is true}\}$. The optimal decision for this problem is made based on whether or not the sufficient statistic $t := y^T \Phi s$ surpasses a threshold κ, i.e., $t \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \kappa$, where κ is chosen to meet the constraint $P_F \leq \alpha$ for a specified α. As can be seen from the analysis in [43], the performance of such a detector depends on $\|\Phi s\|_2$. In effect, if Φ "loses" signal energy for some signals the detector performance will suffer, and if Φ "amplifies" signal energy for some signals the detector performance will improve. A stable embedding of any signal the detector may encounter, however, guarantees consistent performance of the detector.

Theorem 5.7. *Suppose $\mathcal{S}$ is a finite set of signals and let Φ be a randomly generated DBD matrix as described in Theorem 5.3 with a number of measurements denoted by the matrix $\mathbf{M}$. Fix $0 < \rho < 1$ and pick α such that $P_F \leq \alpha$. Then with probability exceeding $1 - \rho$, any signal $s \in \mathcal{S}$ can be detected with probability of detection bounded by*

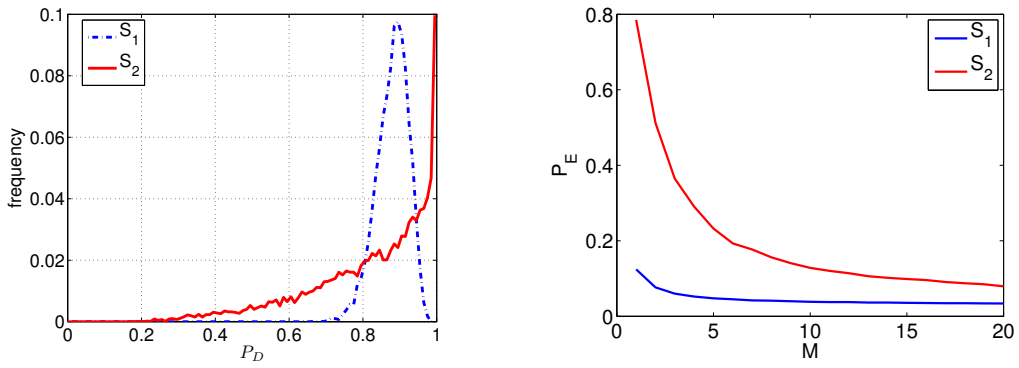
$$Q\left(Q^{-1}(\alpha) - \sqrt{1 - \widetilde{\delta}(\mathcal{S}, \{0\}, \mathbf{M}, \rho)}\sqrt{\frac{\|s\|_2^2}{\sigma^2}}\right) \leq P_D(\alpha) \leq Q\left(Q^{-1}(\alpha) - \sqrt{1 + \widetilde{\delta}(\mathcal{S}, \{0\}, \mathbf{M}, \rho)}\sqrt{\frac{\|s\|_2^2}{\sigma^2}}\right),$$

where $Q(\alpha) = \frac{1}{\sqrt{2\pi}} \int_\alpha^\infty e^{-\frac{u^2}{2}} du$ and where $\widetilde{\delta}(\mathcal{S}, \{0\}, \mathbf{M}, \rho)$ is as defined in (5.11).

The proof of this theorem follows by combining the fact that

$$P_D(\alpha) = Q\left(Q^{-1}(\alpha) - \frac{\|\Phi s\|_2}{\sigma}\right)$$

(see [43]) with (5.11) and the monotonicity of the function $Q(\cdot)$. While achieving the best possible P_D for a given P_F is of course desirable for a detector, another very important consideration for a system designer is the reliability and consistency of that system. Large fluctuations in performance make it difficult to ascribe meaning to a particular detection result and to take action with a certain level of confidence. The theorem above tells us that the consistency of the detector performance is tied to how reliably Φ preserves the norm of signals in $\mathcal{S}$. Examining this relationship, it is clear that more favorable values of $\Gamma_2(s, \mathbf{M})$ for a signal class result in tighter bounds on $P_D(\alpha)$ and therefore



(a) (b)

Figure 5.3: *(a) Histogram of P_D for 10000 signals with uniform energy across blocks (signal class $\mathcal{S}_1$) and for 10000 signals with decaying energy across blocks (signal class $\mathcal{S}_2$) when measured with a DBD matrix. The compressive NP detector has the constraint $P_F \leq \alpha = 0.1$. (b) Plots of the probability of misclassification P_E over a range of values of $M = 1, \cdots, 20$. The first class of signals $\mathcal{S}_1$ are sparse in the frequency domain. The second class of signals $\mathcal{S}_2$ are nonzero only on a single block in the time domain. While P_E decreases with increasing M for both classes of signals, classification performs better for the signals in $\mathcal{S}_1$, which are more amenable to a stable embedding with a DBD matrix.*

in stronger consistency guarantees for the detector.

To illustrate this fact with an example, we create a single DBD measurement matrix $\Phi \in \mathbb{R}^{MJ \times NJ}$ having an equal number of measurements $M_j = M$ per block. We take $M = 4$, $J = 16$ and $N = 64$, and we draw the nonzero entries of Φ as i.i.d. Gaussian random variables with variance $1/M$. We test the detection performance of this matrix by drawing 10000 unit-norm test signals randomly from two classes: $\mathcal{S}_1$, in which signals have uniform energy across their blocks, and $\mathcal{S}_2$, in which signals have decaying energy across their blocks. We choose the noise variance σ^2 such that each test signal s has a constant signal-to-noise ratio of $10 \log_{10}\left(\frac{\|s\|_2^2}{\sigma^2}\right) = 8$dB. Because of the construction of $\mathcal{S}_1$, $\Gamma_2(s, \mathbf{M})$ attains its maximum value of MJ for all signals $s \in \mathcal{S}_1$, resulting a small conditioning $\widetilde{\delta}$ and a tight bound on P_D. In contrast, $\mathcal{S}_2$ will have a smaller value of $\Gamma_2(s, \mathbf{M})$, resulting in larger values of $\widetilde{\delta}$ and much looser bounds on P_D. We choose the maximum probability of failure to be $\alpha = 0.1$ and use the equation $P_D(\alpha) = Q\left(Q^{-1}(\alpha) - \frac{\|\Phi s\|_2}{\sigma}\right)$ to calculate the expected P_D for each signal.

Figure 5.3(a) shows the histogram of P_D for the signals in $\mathcal{S}_1$ and $\mathcal{S}_2$ when measured with a DBD matrix. We see that for the uniform energy signals in $\mathcal{S}_1$, the detector performance is indeed tightly clustered around $P_D = 0.9$; one can see that this is the point of concentration predicted by Theorem 5.7 since $Q\left(Q^{-1}(0.1) - \sqrt{10^{8/10}}\right) \approx 0.8907$. Thus for this class of signals, the detector performance is consistent and we can be assured of a favorable P_D when using the detector for all signals in $\mathcal{S}_1$. However, when using the DBD matrix on the signal class $\mathcal{S}_2$, the P_D values are widely spread out compared to those for $\mathcal{S}_1$, despite the fact that the average P_D is

nearly the same. Although some individual signals may have above average performance because the measurement matrix happened to amplify their energies, other signals may have very poor performance because the measurement matrix significantly attenuated their energies. Thus this experiment shows how the signal statistics affect the performance reliability in compressive detection tasks when the measurements matrices have block diagonal structure.

5.4.3 Classification in the Compressed Domain

Rather than determining the presence or absence of a fixed candidate signal, some scenarios may require the classification of a signal among multiple hypotheses [44, 43]. In particular, let $s_1, s_2, \ldots, s_R \in \mathbb{R}^{NJ}$ denote known signals, and suppose that from the measurement vector y, we wish to decide between R hypotheses:

$$\mathcal{H}_i : y = \Phi s_i + z, \text{ for } i = 1, 2, \ldots, R,$$

where Φ is a known compressive measurement matrix, and z is a vector of i.i.d. zero-mean Gaussian noise with variance σ^2.

It is straightforward to show that when each hypothesis is equally likely, the classifier with minimum probability of error selects the hypothesis that minimizes the sufficient statistic $t_i := \|y - \Phi s_i\|_2^2$. As can be imagined, the performance of such a classifier depends on how well Φ preserves pairwise distances among the signals $\{s_i\}$. If a situation were to occur where $\frac{\|\Phi s_p - \Phi s_q\|_2}{\|s_p - s_q\|_2}$ was small for some p, q, then s_p could easily be mistaken for s_q in the measurements y. Therefore, having a stable embedding can again be particularly useful for guaranteeing consistent and predictable performance.

Theorem 5.8. *Let $\mathcal{S}$ denote a fixed set of signals with $|\mathcal{S}| = R < \infty$ and fix $0 < \rho < 1$. Suppose Φ is a randomly generated DBD matrix as described in Theorem 5.3 with a number of measurements denoted by the matrix $\mathbf{M}$. Assume we receive the measurements $y = \Phi s_{i^*} + z$ for some $i^* \in \{1, 2, \ldots, R\}$ and $z \sim \mathcal{N}(0, \sigma^2 I)$. Then, with probability at least*

$$1 - \left(\frac{R-1}{2}\right) \exp\left\{ -\frac{d^2\left(1 - \widetilde{\delta}(\mathcal{S}, \mathcal{S}, \mathbf{M}, \rho)\right)}{8\sigma^2} \right\} - 2\rho,$$

we have $i^* = \arg\min_{i \in \{1,\ldots,R\}} t_i$ and thus the signal s_{i^*} can be correctly classified. Here $d := \min_{i,j} \|s_i - s_j\|_2$ denotes the minimum separation among the signals in $\mathcal{S}$ and $\widetilde{\delta}(\mathcal{S}, \mathcal{S}, \mathbf{M}, \rho)$ is as defined in (5.11).

The proof of this theorem again follows by combining bounds from [43] with (5.11). From this theorem it follows that, if Φ is a block diagonal matrix, signal families $\mathcal{S}$ whose difference vectors $s_p - s_q$ have favorable Γ_2 values will have consistent and predictable classification performance.

The following simulation demonstrates the potential for predictable classification of signals acquired using compressive block diagonal matrices. We again consider DBD matrices having an equal number of measurements $M_j = M$ per block, and we consider signals having $J = 16$ blocks of length $N = 64$. We first create a favorable class of unit-norm signals $\mathcal{S}_1$ with $R = J$ elements such that each signal has just 4 nonzero DFT coefficients at randomly chosen frequencies. To ensure that the signals are real, we restrict the coefficients on conjugate pairs of frequencies to have complex conjugate values. We also ensure that no frequencies are repeated amongst the signals in $\mathcal{S}_1$. As we show in Section 5.5.2, frequency sparse signals with randomly selected frequency support will have large Γ_2 values with high probability; therefore the difference of any two signals from $\mathcal{S}_1$ will also have a large Γ_2 value with high probability. We also create a second class of unit-norm signals $\mathcal{S}_2$ with $R = J$ elements such that each signal s_r for $r = 1, 2, \ldots, R$ has nonzero (and randomly selected) values only on its r-th block and is zero everywhere else. Difference signals from this class will have small Γ_2 values since their energies across the blocks are not uniform.

For each M ranging from 1 to 20, we create 1000 instances of a random DBD matrix Φ of size $MJ \times NJ$. For each Φ and for each signal class $\mathcal{S}_1$ and $\mathcal{S}_2$, we identify the indices i_1, i_2 that minimize $\|\Phi s_{i_1} - \Phi s_{i_2}\|_2$. The signals corresponding to s_{i_1} and s_{i_2} will be among the most difficult to classify since they each have a close neighbor (either Φs_{i_2} or Φs_{i_1}, respectively) after projection by Φ. Then for each of these signals $\{s_{i_j}\}_{j=1,2}$, we create 1000 noisy measurement vectors $y = \Phi s_{i_j} + z$ with $z \sim \mathcal{N}(0, \sigma^2 I)$ and with σ chosen such that $10 \log_{10}\left(\frac{d^2}{\sigma^2}\right) = 15\text{dB}$. Finally, we let $p = \arg\min t_i$ be the output of the classifier and calculate the probability of misclassification, $P_E(M)$, for each M as the proportion of occurrences of $p \neq i_1$ or $p \neq i_2$, respectively, over the combined 1000 instances of noise z and 1000 instances of Φ.

Figure 5.3(b) plots $P_E(M)$ for both classes of signals. As expected, the curve for $\mathcal{S}_1$ is lower than that for $\mathcal{S}_2$ since the signals in $\mathcal{S}_1$ are expected to have a stable embedding with a tighter conditioning. Both curves also show a decreasing trend for increasing M (although it is much more obvious for signal class $\mathcal{S}_2$) as should be expected. Lastly, we see that $P_E(M)$ saturates at a certain level with increasing M. This is also predicted by Theorem 5.8, where the smallest upper bound that can be provided for P_E is given by $\frac{R-1}{2} \exp\left(-\frac{d^2}{8\sigma^2}\right) > 0$. With the parameters used in this experiment, it can be calculated that the smallest theoretical upper bound for P_E is approximately 0.144. This shows that Theorem 5.8 may be slightly pessimistic.

5.5 Favorable Signal Classes

The various compressive signal processing guarantees presented in Section 5.4 are built upon the premise that a DBD matrix provides a stable embedding of the signals of interest; as we have noted, these arguments can be extended to RBD matrices as well. Our analysis has also indicated that such stable embeddings are most easily realized with matrices that are well matched to the energy distribution (and sometimes orthogonality) of the signal blocks. In many applications, however—perhaps for the sake of generality, or because little advance knowledge of the signals is available—it may be most natural to use a fixed and equal allocation of measurements. Fortunately, there are a number of interesting signal families (and, in some cases, the corresponding difference signals) that provide favorable Γ_2 values for "uniform" DBD matrices where all M_j are equal (for all j, we suppose $M_j = M$ for some M) and in some cases also provide favorable Λ_2 values for RBD matrices. In this section we survey several such examples.

5.5.1 Delay Networks and Multiview Imaging

One favorable signal class for uniform DBD and RBD matrices can occasionally arise in certain sensor network or multi-view imaging scenarios where signals with steeply decaying autocorrelation functions are measured under small perturbations. Consider for example a distributed sensor network of J sensors where we would like to detect the presence of a known signal given the observations from each sensor. Due to limited resources, each sensor uses random measurement matrices $\Phi_1, \Phi_2, \ldots, \Phi_J$ to efficiently capture the underlying information with only a few random projections. Suppose that the received signals $x_1, x_2, \ldots, x_J \in \mathbb{R}^N$ represent observations of some common known underlying prototype signal $w \in \mathbb{R}^N$. However, due to the configurations of the sensors, these observations occur with different delays or translations. More formally, we might consider the one-dimensional delay parameters $\delta_1, \delta_2, \ldots, \delta_J \in \mathbb{Z}$ and have that for each j, $x_j(n) = w(n - \delta_j)$. Then, denoting the measurements at sensor j as $y_j = \Phi_j x_j$ it is straightforward to see that the overall system of equations can be represented with a DBD matrix, or when $\Phi_1 = \Phi_2 = \cdots = \Phi_J$ with an RBD matrix.

Assuming w is suitably zero-padded so that border and truncation artifacts can be neglected, we will have $\|x_j\|_2 = \|w\|_2$ for all x_j; this gives $\Gamma_2([x_1^T \; x_2^T \; \cdots x_J^T]^T, \mathbf{M}) = MJ$, which is the ideal case for observation with a uniform DBD matrix. This suggests that the outputs from distributed network systems can be highly amenable to the sort of compressive-domain signal inference and processing tasks described Section 5.4.

Moreover, the correlations among the components x_j can be characterized in terms of the autocorrelation function R_w of w: we will have $\langle x_i, x_j \rangle = \sum_{n=1}^{N} x_i(n)x_j(n) = \sum_{n=1}^{N} w(n-\delta_i)w(n-\delta_j)$, which neglecting border and truncation artifacts will simply equal $R_w(|\delta_i - \delta_j|)$. Therefore, signals w that exhibit strong decay in their autocorrelation function will be natural candidates for observation with RBD matrices as well. For example, equation (5.9) gives

$$\Lambda_2([x_1^T \; x_2^T \; \cdots x_J^T]^T, M) = \frac{MJ^2\|w\|_2^4}{J\|w\|_2^4 + 2\sum_{i>j} R_w(|\delta_i - \delta_j|)^2}.$$

When $R_w(|\delta_i - \delta_j|)$ is small for most i and j, the quantity $\Lambda_2([x_1^T \; x_2^T \; \cdots x_J^T]^T, M)$ is near its optimal value of MJ.

5.5.2 Frequency Sparse Signals

Signals having sparse frequency spectra arise in many different applications involving communications intelligence systems and RF sensor networks. Based on time-frequency uncertainty principles and the well-known incoherence of sinusoids and the canonical basis (i.e., "spikes and sines") [127, 29], it is natural to expect that most signals that are sparse in the frequency domain should have their energy distributed relatively uniformly across blocks in the time domain. In the following theorem, we make formal the notion that frequency sparse signals are indeed likely to be favorable for measurement via uniform DBD matrices, producing values of $\Gamma_2(x, \mathbf{M})$ within a log factor of its maximum possible.

Theorem 5.9. *Let $N, \beta > 1$ be fixed, suppose $N' = NJ > 512$, and let $\mathbf{M} = \mathrm{diag}\{M, M, \ldots, M\}$ be a DBD measurement allocation with M fixed. Let $\Omega \subset [1,\ N']$ be of size $S \leq N$ generated uniformly at random. Then with probability at least $1 - O(J(\log(N'))^{1/2}(N')^{-\beta})$,[5] every signal $x \in \mathbb{C}^{N'}$ whose DFT coefficients are supported on Ω will have:*[6]

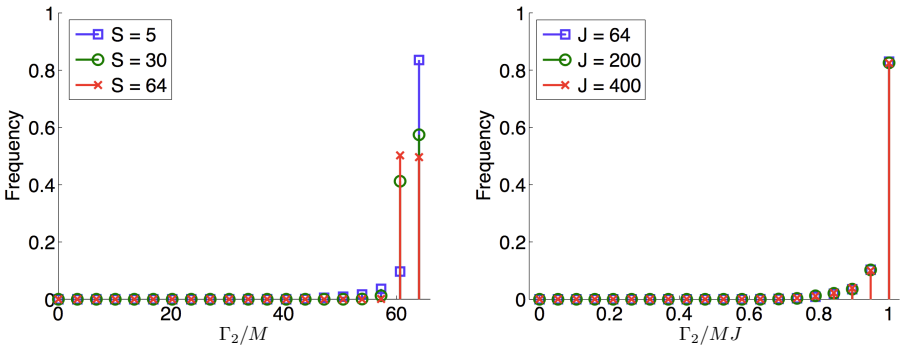
$$\Gamma_2(x, \mathbf{M}) \geq MJ \cdot \min\left\{ \frac{0.0779}{(\beta+1)\log(N')}, \frac{1}{\left(\sqrt{6(\beta+1)\log N'} + \frac{(\log N')^2}{N}\right)^2} \right\}. \tag{5.13}$$

Proof: See Appendix C.

Note that as N' grows, the lower bound on $\Gamma_2(x, \mathbf{M})$ scales as $\frac{MJ}{\frac{1}{N^2}\log^4(N')}$, which (treating the

[5] The $O(\cdot)$ notation is with respect to N'. With the component length N fixed, this means that only the number of blocks J is growing with increasing N'.

[6] We consider complex-valued signals for simplicity and clarity in the exposition. A result with the same spirit that holds with high probability can be derived for strictly real-valued signals, but this comes at the cost of a more cumbersome derivation.



(a) (b)

Figure 5.4: *Histograms of the normalized quantity Γ_2 for frequency sparse signals. (a) The distribution of $\frac{\Gamma_2}{M}$ for randomly generated frequency sparse signals of length $N' = N \times J = 64 \times 64$ for sparsity levels $S \in \{5, 30, 64\}$. Note that $\frac{\Gamma_2}{M}$ accumulates near its upper bound of $J = 64$ for all three sparsity levels. (b) The distribution of $\frac{\Gamma_2}{MJ}$ for randomly generated frequency sparse signals with $S = 5$ and the number of signal blocks $J \in \{64, 200, 400\}$. Note that $\frac{\Gamma_2}{MJ}$ accumulates near its upper bound of 1.*

fixed value N as a constant) is within $\log^4(N')$ of its maximum possible value of MJ. Thus the concentration exponent for *most* frequency sparse signals when measured by a uniform DBD matrix will scale with $\epsilon^2 MJ/\log^4(N')$ for small ϵ. Also note that the failure probability in the above theorem can be bounded by $O(\frac{1}{N'^{\beta-2}})$ since both J and $\sqrt{\log(N')}$ are less than N'.

To explore the analysis above we use two illustrative simulations. For the first experiment, we generate 5000 signals with length $N' = NJ = 64 \times 64 = 4096$, using three different sparsity levels $S \in \{5, 30, 64\}$. The DFT coefficient locations are selected uniformly at random, and the corresponding nonzero coefficient values are drawn from the i.i.d. standard Gaussian distribution. Figure 5.4(a) plots the ratio $\frac{\Gamma_2(x,\mathbf{M})}{M}$, showing that this quantity is indeed near the upper bound of $J = 64$, indicating a favorable energy distribution. This gives support to the fact that the theoretical value of $\Gamma_2(x, \mathbf{M})$ predicted in Theorem 5.9 does not depend strongly on the exact value of S. For the second experiment, we fix the sparsity at $S = 5$ and vary the signal block length $J \in \{64, 200, 400\}$ (and thus the total signal length $N' = NJ$ changes as well). For each J we generate 5000 random signals in the same manner as before and plot in Figure 5.4(b) the distribution of $\frac{\Gamma_2(x,\mathbf{M})}{MJ}$. It is clear that this value concentrates near its upper bound of 1, showing the relative accuracy of the prediction that $\frac{\Gamma_2}{M}$ scales linearly with increasing J. While some of the quantities in Theorem 5.9 appear pessimistic (e.g., the scaling with $\log^4(N')$), these simulations confirm the intuition derived from the theorem that frequency sparse signals should indeed have favorable energy distributions and therefore favorable concentration properties when measured with DBD matrices.

Because differences between frequency sparse signals are themselves sparse in the frequency domain, it follows immediately that not only do frequency sparse signals x have favorable $\Gamma_2(x, \mathbf{M})$

values for uniform DBD matrices, but also that most differences $x_1 - x_2$ between frequency sparse signals have favorable $\Gamma_2(x_1 - x_2, \mathbf{M})$ values. Thus, when measured by a uniform DBD matrix, many families of frequency sparse signals are likely to perform favorably and predictably in the compressive signal processing scenarios outlined in Section 5.4.

Importantly, Theorem 5.9 can also allow us to guarantee the stable embedding of certain infinite collections of frequency sparse signals. In particular, for any sparse support Ω on which (5.13) holds uniformly, one can apply standard covering arguments (as discussed briefly in Section 5.2.3) to guarantee that with a moderate total number of measurements $MJ = O\left(\frac{|\Omega|}{\epsilon^2}\log^4(N')\right)$, a uniform DBD matrix will simultaneously approximately preserve the norm of all frequency signals supported on Ω. This fact allows one to consider compressive-domain interference cancellation (as discussed in Section 5.4.1 and in [43]) from a set of frequency sparse signals, where the set of possible interferers live in a known subspace of frequency sparse signals.

5.5.3 Difference Signals

In applications such as classification, we require a stable embedding of difference vectors between signals in a certain signal class. It is interesting to determine what signal families in addition to frequency sparse signals will give rise to difference signals that have favorable values of Γ_2 (for uniform DBD matrices) or Λ_2 (for RBD matrices).

We provide a partial answer to this question by briefly exemplifying a signal family that is favorable for measurement via uniform DBD matrices. Consider a set $Q \subset \mathbb{R}^{JN}$ of signals that, when partitioned into J blocks of length N, satisfy both of the following properties: (i) each $x \in Q$ has uniform energy across the J blocks, i.e., $\|x_1\|_2^2 = \|x_2\|_2^2 = \cdots = \|x_J\|_2^2 = \frac{1}{J}\|x\|_2^2$, and (ii) each $x \in Q$ has highly correlated blocks, i.e., for some $a \in (0, 1]$, $\langle x_i, x_j \rangle \geq a\frac{1}{J}\|x\|_2^2$ for all $i, j \in \{1, 2, \ldots, J\}$. The first of these conditions ensures that if $\mathbf{M} = \text{diag}\{M, M, \ldots, M\}$, then each $x \in Q$ will have $\Gamma_2(x, \mathbf{M}) = MJ$ and thus be highly amenable to measurement by a uniform DBD matrix. The second condition, when taken in conjunction with the first, ensures that all difference vectors of the form $x - y$ where $x, y \in Q$ will also be highly amenable to measurement by a uniform DBD matrix. In particular, for any $i, j \in \{1, 2, \ldots, J\}$, one can show that

$$\left|\|x_i - y_i\|_2^2 - \|x_j - y_j\|_2^2\right| \leq \frac{4\sqrt{2}\|x\|_2\|y\|_2\sqrt{1-a}}{J},$$

meaning that the energy differences in each block of the difference signals can themselves have small differences. One implication of this bound is that as $a \to 1$, $\Gamma_2(x - y, \mathbf{M}) \to MJ$.

Signal families of the form specified above—with uniform energy blocks and high inter-block correlations—may generally arise as the result of observing some phenomenon that varies slowly as a function of time or of sensor position. As an empirical demonstration, let us consider a small database of eight real-world video signals frequently used as benchmarks in the video compression community, where we will treat each video frame as a signal block.[7] We truncate each video to have $J = 150$ frames, each of size $N = 176 \times 144 = 25344$ pixels, and we normalize each video (not each frame) to have unit energy. Because the test videos are real-world signals, they do not have perfectly uniform energy distribution across the frames, but we do observe that most frame energies are concentrated around $\frac{1}{J} = 0.00667$. For each video, we present in Table 5.1 the minimum and maximum inner products $\langle x_i, x_j \rangle$ over all $i \neq j$, and we also list the quantity $\frac{\Gamma_2(x,\mathbf{M})}{M}$ for each video.

Video name	Akiyo	Bridge close	Bridge far	Carphone
$\max\langle x_i, x_j \rangle$	0.00682	0.00668	0.00668	0.00684
$\min\langle x_i, x_j \rangle$	0.00655	0.00664	0.00665	0.00598
Γ_2/M	149.9844	149.9998	149.9999	149.9287
Video name	Claire	Coastguard	Foreman	Miss America
$\max\langle x_i, x_j \rangle$	0.00690	0.00742	0.00690	0.00695
$\min\langle x_i, x_j \rangle$	0.00650	0.00562	0.00624	0.00606
Γ_2/M	149.9782	149.2561	149.9329	149.9301

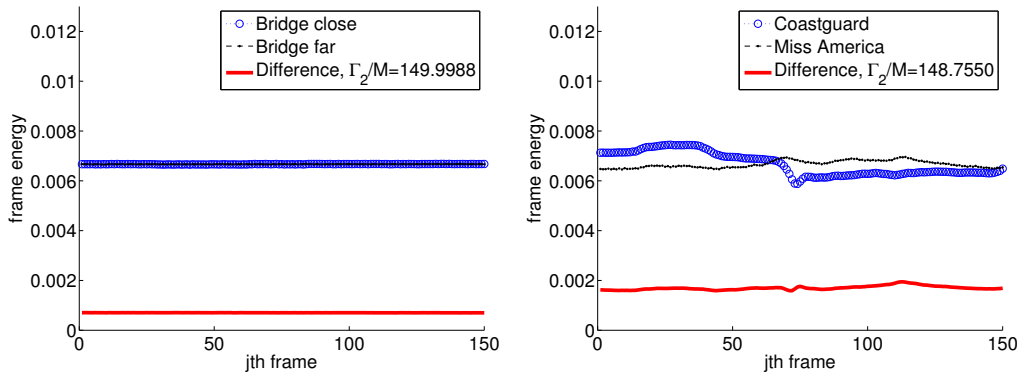
Table 5.1: *The maximum and minimum inner products between all pairs of distinct frames in each video, and the quantity Γ_2/M for each video. The best possible value of Γ_2/M is $J = 150$.*

As we can see, the minimum inner product for each video is indeed quite close to 0.00667, suggesting from the arguments above that the pairwise differences between various videos are likely to have highly uniform energy distributions. To verify this, we compute the quantity $\frac{\Gamma_2}{M}$ for all possible $\binom{8}{2}$ pairwise difference signals. As we are limited in space, we present in Figure 5.5 plots of the energies $\|x_j\|_2^2$, $\|y_j\|_2^2$, and $\|x_j - y_j\|_2^2$ as a function of the frame index j for the pairs of videos x, y that give the best (highest) and the worst (smallest) values of $\frac{\Gamma_2(x-y,\mathbf{M})}{M}$. We see that even the smallest $\frac{\Gamma_2}{M}$ value is quite close to the best possible value of $\frac{\Gamma_2}{M} = 150$. All of this suggests that the information required to discriminate or classify various signals within a family such as a video database may be well preserved in a small number of random measurements collected by a uniform DBD matrix.

5.5.4 Random Signals

Our discussions above have demonstrated that favorable Γ_2 values (for uniform DBD matrices) and Λ_2 values (for RBD matrices) can arise for signals in a variety of practical scenarios. This is no

[7] Videos were obtained from `http://trace.eas.asu.edu/yuv/`.



(a) (b)

Figure 5.5: *Plots of the energy distributions of individual videos and of their differences for the best video pair and the worst video pair among all possible* $\binom{8}{2}$ *possible video pairs. (a) The difference of the video pair, "Bridge close" and "Bridge far", giving the best value of* $\Gamma_2(x-y,\mathbf{M})/M = 149.9988$. *(b) The difference of the video pair, "Coastguard" and "Miss America", giving the worst value of* $\Gamma_2(x-y,\mathbf{M})/M = 148.7550$.

accident. Indeed, as a blanket statement, it is true that a large majority of all signals $x \in \mathbb{R}^{JN}$, when partitioned into a sufficiently small number of blocks J and measured uniformly, will have favorable values of both Γ_2 and Λ_2. One way of formalizing this fact is with a probabilistic treatment such as that given in the following lemma.

Lemma 5.10. *Let* ϕ *denote a subgaussian random variable with mean 0, variance* σ^2, *and subgaussian norm* $\|\phi\|_{\psi_2}$, *and suppose* $x \in \mathbb{R}^{NJ}$ *is populated with i.i.d. realizations of* ϕ. *Let* $\mathbf{M} = \text{diag}\{M, M, \ldots, M\}$ *with* M *fixed. Pick* $\epsilon \leq \frac{\|\phi\|_{\psi_2}^2}{C_2}$ *and suppose that* $J \leq \frac{C_1 C_2^2 N \epsilon^2}{2\|\phi\|_{\psi_2}^4 \log(12/\epsilon)}$, *where* C_1, C_2 *are absolute constants as given in Theorem 5.3. Then, with probability at least* $1 - 2\exp\left(-\frac{1}{2}\frac{C_1 C_2^2 N \epsilon^2}{\|\phi\|_{\psi_2}^4}\right)$, *we have*

$$\Gamma_2(x,\mathbf{M}) \geq \Lambda_2(x,M) \geq M + M\left(\frac{1-\epsilon}{1+\epsilon}\right)^2 (J-1).$$

Proof: See Appendix D.

We see from Lemma 5.10 that when random vectors are partitioned into a sufficiently small number of blocks, these signals will have their norms preserved giving rise to $\Gamma_2(x,\mathbf{M})$ and $\Lambda_2(x,M)$ values close to their optimal value of MJ. To give some illustrative numbers, numerical simulations showed that over 10000 random draws of Gaussian i.i.d. signals with $J = 16$ and $N = 64$ the average value of $\Gamma_2(x,\mathbf{M})/M$ was 15.5 and the average value of $\Lambda_2(x,M)/M$ was 12.6, which are both large fractions of the maximum possible value of 16. We also note that by using the same argument we can show that the differences of random signals will exhibit large Γ_2 and Λ_2 values. One possible use of this lemma could be in studying the robustness of block diagonal measurement systems to noise in the signal. The lemma above tells us that when restrictions are met on the number of blocks, random noise will tend to yield blocks that are nearly orthogonal and have highly uniform energies,

thereby guaranteeing that they will not have their energy amplified by the matrix.

5.6 Conclusion

In this chapter we have derived concentration of measure inequalities for compressive DBD and RBD matrices. Our experimental results confirm what our theoretical bounds suggest: that the actual probability of concentration depends on the degree of alignment between the allocation of the measurements and the energy distribution (and sometimes orthogonality) of the signal blocks. However, in situations where one can optimize the measurement allocation in anticipation of certain signal characteristics or where a fixed system may be measuring certain favorable classes of signals, we have shown that the highly structured DBD and RBD matrices can provide concentration performance that is on par with the dense i.i.d. matrices often used in CS. We have highlighted a number of compressive signal processing applications that benefit from having a stable embedding of a finite signal family, and we have presented a modified JL lemma for block diagonal matrices that reflects the number of measurements required to ensure such a stable embedding. Finally, we have surveyed a number of signal classes whose blockwise energy distribution and/or orthogonality makes them well-suited to measurement via uniform DBD matrices or via RBD matrices. Despite not leading to state-of-the-art RIP bounds, we conclude that our nonuniform concentration results can provide a valuable tool for understanding and possibly mitigating the potential pitfalls of working with highly constrained block diagonal matrices.

There are many natural questions that arise from these results and are suitable topics for future research. For example, it would be natural to consider whether the concentration results for Gaussian RBD matrices could be extended to more general subgaussian RBD matrices (to match the distribution used in our DBD analysis), or whether strong RIP results can be established for RBD matrices. Also, as more applications are identified in the future, it will be important to examine the diversity characteristics of a broader variety of signal classes to determine their favorability for measurement via block diagonal matrices. Additionally, it would be interesting to examine whether the concentration of measure result for RBD matrices could prove useful in the analysis in the multiple measurement vector (MMV) problem [60, 45] that arises, for example, in array signal processing.

As a final note, we briefly mention that our concentration bounds for block diagonal matrices can actually be useful for studying certain *other* types of structured matrices that arise in linear systems applications. In particular, these results can be applied to derive concentration bounds and RIP results for compressive Toeplitz matrices that arise in problems such as channel sensing

and for compressive observability matrices that arise in the analysis of linear dynamical systems. Although space limitations prevent us from detailing these results here, we refer the interested reader to [146, 141] for more information.

CHAPTER 6

Sketched SVD: Recovering Spectral Features from Compressive Measurements

6.1 Introduction

Consider a collection of data arranged in a matrix X of size $N \times n$. Each column represents a length-N signal (or image, frequency counts of terms in a particular document, etc.) and there are n such observations. The singular value decomposition (SVD) of X,

$$X = U\Sigma V^T,$$

carries important information about the structure of the data set, especially when the rank k of X is small. In particular, the columns of U (known as the *left* singular vectors of X) span the principal directions of the data set and can be used as basis vectors for building up typical signals, and the diagonal entries of Σ (known as the singular values of X) reflect the energy of the data set in each of these directions. The extraction of these features is commonly known as Principal Component Analysis (PCA), and PCA is a fundamental and commonly used tool in data analysis and compression. This exact same process can be viewed through a slightly different lens when one imagines the columns of X as independent realizations of a length-N random vector x. Computing the left singular vectors of X is equivalent to computing the eigenvectors of XX^T, which (up to rescaling) is the $N \times N$ sample covariance matrix of the data. In this context, PCA is also known as the Karhunen-Loeve (KL) Transform, and the KL Transform is a fundamental and commonly used tool in statistics.

There are in fact a number of applications where the *right* singular vectors V of a data matrix X

This work is in collaboration with Michael B. Wakin and Anna C. Gilbert [69].

are more important, or equivalently, where the eigenvectors of $X^T X$ carry the structure of interest. For example, the product ΣV^T gives a low-dimensional embedding of the data set that preserves distances and angles between the n data vectors. This embedding can be used for clustering or categorizing the signals [103, 19]; for example, this is used for comparing documents in latent semantic analysis. The right singular vectors of X can also be viewed as the result of applying the KL Transform to the rows, rather than the columns, of X. In this sense, the columns of V describe the inter-signal (rather than intra-signal) statistical correlations. In cases where the column index corresponds to a distinct sensor position, or a vertex in a graph, etc., this correlation structure can carry important structural information [122, 35].

Unfortunately, many of the applications in which we seek the right singular vectors of X (equivalently, the eigenvectors of $X^T X$) are those in which the data is simply too large, too distributed, or generated too quickly for us to store the data or to process it efficiently in one, centralized location. There are, however, settings in which the data sets—while large—have low intrinsic dimension or are of low rank. Let us suppose that the length of each data vector N is much larger than the number of observations n, and suppose that X has rank $k \leq n$. The data may or may not be generated in a dynamic, streaming fashion and it may or may not be collected in a distributed fashion amongst n sensors.

We wish to design a joint observation process (which can be distributed amongst n sensors) that maintains a "sketch" of the data stream and a reconstruction process that, at a central location, reconstructs *not* an approximation of the original data, but rather a good approximation to the singular values σ_j and the (right) singular vectors v_j of the original matrix X. The sketch of the data stream should be a linear, non-adaptive procedure, one that is efficient to update, and one that uses as few observations of the data matrix as possible so that as little communication as possible is required from the sensors to the central processing entity. Because the procedure is linear and non-adaptive, we can represent the sketch as a matrix-matrix product $\Phi X = Y$, where the observation matrix Φ is of size $m \times N$ and the sketch Y is of size $m \times n$. We want m as small as possible. From the sketch matrix Y, we want to produce estimates σ'_j of the k non-zero singular values and estimates v'_j of the associated (right) singular vectors of X such that

$$(1-\epsilon)^{1/2}\sigma_j \leq \sigma'_j \leq (1+\epsilon)^{1/2}\sigma_j$$

and

$$\|v_j - v'_j\|_2 \leq \epsilon\gamma,$$

where γ is a function of the smallest gap between the singular values of X.

Most of the current work on low-rank matrix approximations, robust PCA [28], rank-revealing QR factorizations (see [73] and the references therein for a comprehensive survey), etc. has focused on obtaining a good approximation $\widehat{X}$ to the original data matrix X, albeit one that is parsimonious. Some of the measurement schemes [28] sketch both the row and column space of X, collecting measurements $Y = \Phi X \Phi^T$, and some sketch the column space only so as to derive a few orthogonal vectors that span the column space [73]. These results are of the form

$$\|X - \widehat{X}\|_{\mathcal{X}} \leq (1 + \epsilon)\|X\|_{\mathcal{X}}$$

for some norm $\mathcal{X}$ (typically, the Frobenius norm, but others as well). Our goal is an approximation to the *singular vectors* and *singular values* of X themselves, directly, and while one could apply standard perturbation theory techniques to compare the singular vectors of an approximation $\widehat{X}$ to those of the original data X, the error guarantees would be rather poor. Furthermore, while one could ask about preserving the subspaces spanned by the singular vectors, there are many applications from PCA to data clustering, image segmentation, graph embedding, and modal analysis in which the individual singular vectors are critical for data analysis tasks, and data reconstruction is not necessarily required.

Our algorithmic approach is straightforward. We sketch one side of the $N \times n$ data matrix X, maintaining a sketch matrix $Y = \Phi X$ of size $m \times n$. (The fact that the sketch is one-sided allows it to be computed sensor-by-sensor in distributed data collection settings.) We then compute the SVD of the sketch matrix Y, using standard (iterative) SVD algorithms. Our analysis is quite different from that of most randomized linear algebra methods. We assume that the sketching matrix Φ is randomly generated and satisfies the distributional Johnson-Lindenstrauss (JL) property (see Definition 6.1) so that with high probability it acts as a near isometry on the column span of X, and we then exploit *relative error* (as opposed to absolute error) perturbation analysis for *deterministic* (as opposed to random) matrices to obtain our results. As detailed in our main result, Theorem 6.2, we can obtain accurate relative estimates for the singular values, and in some cases we can obtain accurate estimates for the singular vectors as well. However, we struggle to achieve high accuracy in the singular vectors when the corresponding singular values are close. This is a consequence of well-studied perturbation theory and seems inherent in our approach.

One major application of our work is to determining structural graph properties of streaming graph data, albeit for low-rank graphs (ones with many connected components). Recent work on the

structure and evolution of online social networks [86] suggests that a significant fraction of vertices in such networks participate in isolated communities, "small groups who interact with one another but not with the network at large."

In Section 6.2, we set the stage for our mathematical problem and in Section 6.3, we outline the related work, including an overview of the relative error perturbation techniques from linear algebra that we will use. In Section 6.4, we present our main result which we apply, in Section 6.5, to the spectral analysis of streaming graphs.

6.2 Problem Setup

Let X denote the $N \times n$ real-valued data matrix. For example, one may envision that each column represents a length-N time series signal collected from one of n sensors. Let us assume that $N \geq n$, and that $\text{rank}(X) = k \leq n$. We write the *truncated* SVD of X as $X = U_X \Sigma_X V_X^T$, where unlike the full SVD, U_X is $N \times k$, $\Sigma_X = \text{diag}(\sigma_1, \ldots, \sigma_k)$ with $\sigma_1 \geq \cdots \geq \sigma_k > 0$, and V_X is $n \times k$. Our goal in this chapter is to estimate the singular values Σ_X and the right singular vectors V_X from a low-dimensional sketch obtained by left-multiplying X with a compressive matrix. In particular, we let Φ denote a sketching matrix of dimension $m \times N$, and we denote the sketch of X by $Y = \Phi X$. We are specifically interested in cases where $m < N$ and thus Y is a shorter matrix than X. We also note that the sketching matrix Φ can be applied individually to each column of X, and these sketched columns can be concatenated to form the $m \times n$ matrix Y. In other words, the sketching can be performed sensor-by-sensor.

Our algorithm for estimating Σ_X and V_X from Y is explained in Section 6.4.1. This algorithm is very simple and is based on the idea that under a suitable choice of Φ, the singular values and right singular vectors of Y will approximate the singular values and right singular vectors of X. In order to state our results, we write the truncated SVD of Y as $Y = U_Y \Sigma_Y V_Y^T$, where U_Y is $m \times k$, $\Sigma_Y = \text{diag}(\sigma'_1, \ldots, \sigma'_k)$ with $\sigma'_1 \geq \cdots \geq \sigma'_k \geq 0$, and V_Y is $n \times k$. (We will be interested in cases where $m \geq k$, and typically Y will have rank k just like X.) It will also be useful for us to write the eigendecompositions of $X^T X$ and $Y^T Y$ as $X^T X = V_X \Lambda_X V_X^T$ and $Y^T Y = V_Y \Lambda_Y V_Y^T$, respectively, where $\Lambda_X = \text{diag}(\lambda_1, \ldots, \lambda_k)$ and $\Lambda_Y = \text{diag}(\lambda'_1, \ldots, \lambda'_k)$. (Although it is not common practice we will use the "truncated" eigendecomposition so that V_X and V_Y are both of dimension $n \times k$ and Λ_X and Λ_Y are both of dimension $k \times k$; when $k = n$ we will have the usual eigendecomposition.) Note that $\lambda_j = \sigma_j^2$ and $\lambda'_j = \sigma'^2_j$ for $j = 1, \ldots, k$.

To ensure that the spectral information about X is preserved in the sketched matrix $Y = \Phi X$,

we rely on randomness to construct the sketching matrix Φ. Any random distribution that can be used to construct a Johnson-Lindenstrauss (JL) embedding can be used to generate Φ.

Definition 6.1. An $m \times N$ random matrix Φ is said to satisfy the *distributional JL property* if for any fixed $x \in \mathbb{R}^N$, and any $0 < \epsilon < 1$,

$$\Pr\left[\left|\|\Phi x\|_2^2 - \|x\|_2^2\right| > \epsilon \|x\|_2^2\right] \leq 2e^{-mf(\epsilon)},$$

where $f(\epsilon) > 0$ is a constant depending only on ϵ.

For most random matrices satisfying the distributional JL property, the functional dependence on ϵ, $f(\epsilon)$, is quadratic in ϵ as $\epsilon \to 0$. For compactness, we suppress this except where necessary for quantifying the number of measurements or the run time of our algorithm. There are a variety of random matrix constructions known to possess the distributional JL property. Notably, random matrices populated with independent and identically distributed (i.i.d.) subgaussian entries (see Definition 5.1) will possess this property [41]. Subgaussian distributions include suitably scaled Gaussian and ± 1 Bernoulli random variables. Other examples of non-i.i.d. JL matrices and discussions of the function $f(\epsilon)$ are contained in works such as [80, 39]. There are many papers that address the sparsity of a JL matrix, the speed of its application to a vector, the minimum number of rows it must possess, the minimum amount of randomness necessary to generate such a matrix, etc. For our results, a random matrix satisfying the distributional JL property is sufficient and, depending on the particular application (streaming versus static data), we want either fast update times (i.e., a sparse transform) or a fast transform. We appeal to a long line of work in assessing the qualities of such transforms and in constructing them, either randomly or deterministically.

Finally, we note that a primary objective of this work is to quantify the amount of perturbation of the right singular vectors of X under the random measurement operator Φ. For $j = 1, \ldots, k$, let us denote the jth column of V_X as v_j and the jth column of V_Y as v'_j. Our bounds concern the quantity $\|v_j - v'_j\|_2$. However, we note that the right singular vectors v_j and v'_j are each unique only up to multiplication by -1. Thus, without loss of generality, we will assume that the sign of each v'_j is chosen so that it is positively correlated with v_j. That is, we will assume for each j that $\langle v_j, v'_j \rangle \geq 0$.

6.3 Related Work

In order to quantify the amount of change in the singular values and the right singular vectors, we approach this problem from the matrix perturbation theoretic perspective. To see the connection to matrix perturbation theory, let us write

$$Y^T Y = X^T \Phi^T \Phi X = V_X \Sigma_X U_X^T \Phi^T \Phi U_X \Sigma_X V_X^T.$$

Defining $\Delta_\Phi := \Phi^T \Phi - I$, $Y^T Y$ becomes

$$Y^T Y = V_X \Sigma_X U_X^T (I + \Delta_\Phi) U_X \Sigma_X V_X^T = V_X \Sigma_X^2 V_X^T + V_X \Sigma_X U_X^T \Delta_\Phi U_X \Sigma_X V_X^T.$$

Given this equation, we can think of the symmetric matrix $Y^T Y$ as being the summation of some original symmetric matrix

$$A := V_X \Sigma_X^2 V_X^T$$

and a perturbation matrix

$$E := V_X \Sigma_X U_X^T \Delta_\Phi U_X \Sigma_X V_X^T.$$

Roughly speaking, when E is small in some sense, it would be reasonable to expect $Y^T Y$ to have approximately the same spectral information as A. Thus we can think about our problem as quantifying the amount of change between the eigenvalues and eigenvectors of A (which equal the squared singular values and the right singular vectors of X) and those of $Y^T Y$ under the additive perturbation E.

In the following sections, we briefly review some of the important results in the matrix perturbation theory literature and also discuss the connection of our problem to the simultaneous iteration method, an important algorithm for computing the eigendecomposition of a matrix.

6.3.1 Absolute Bounds

There is an extensive literature in the field of matrix perturbation theory quantifying the amount of change in the eigenvalues and eigenvectors of a symmetric $n \times n$ matrix A when it undergoes an additive perturbation. A perturbed matrix, $\widetilde{A}$, may be written in the form of $\widetilde{A} = A + E$, where E is the perturbation matrix that is being added to A. It is well-known that the eigenvalues of A and those of $\widetilde{A}$ will be close to one another when the amount of perturbation E is small (typically with respect to the 2-norm of E). Let us denote the jth largest eigenvalues of A and $\widetilde{A}$ as λ_j and $\widetilde{\lambda}_j$,

respectively. Then, it is known [70] that for $j = 1, \ldots, n$,

$$|\widetilde{\lambda}_j - \lambda_j| \leq \|E\|_2. \tag{6.1}$$

Thus we can see that the distance between each perturbed eigenvalue and the corresponding original eigenvalue will depend on the amount of perturbation, i.e., $\|E\|_2$.

To discuss the perturbation in the eigenvectors let us first represent the eigenvectors of A as v_j such that $Av_j = \lambda_j v_j$. Similarly, let us define the eigenvectors of $\widetilde{A}$ as $\widetilde{v}_j$ such that $\widetilde{A}\widetilde{v}_j = \widetilde{\lambda}_j \widetilde{v}_j$. It is well known that for general matrices A and E, the eigenvectors v_j and $\widetilde{v}_j$ may vary drastically even when the amount of perturbation is small. In other words, $\|\widetilde{v}_j - v_j\|_2$ can be large even when $\|E\|_2$ is small. To see why, let us for example look at the case when two eigenvalues, λ_1 and λ_2, of A are equal to each other. For such a case, we know that the eigenvectors corresponding to those eigenvalues, v_1 and v_2, will not be unique: any linear combination of the two eigenvectors will also be a valid eigenvector corresponding to the same eigenvalue. A perturbation to this matrix will generally cause λ_1 and λ_2 to split into two eigenvalues $\widetilde{\lambda}_1$ and $\widetilde{\lambda}_2$, each of which will satisfy equation (6.1) above. If $\widetilde{\lambda}_1$ and $\widetilde{\lambda}_2$ are distinct, the corresponding eigenvectors $\widetilde{v}_1$ and $\widetilde{v}_2$ will now be uniquely identified. Since v_1 and v_2 were not unique, it is possible for $\widetilde{v}_1$ to differ from any particular choice of v_1 and for $\widetilde{v}_2$ to differ from any particular choice of v_2.

The perturbation in the eigenvectors, however, is not completely arbitrary. It is known that the angle between the space spanned by v_1 and v_2 and the space spanned by $\widetilde{v}_1$ and $\widetilde{v}_2$ will be small if E is small. To state this result more concretely, let us represent the eigendecompositions of A and $\widetilde{A}$ as

$$A = V\Lambda V^T = (V_1 \; V_2) \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}$$

and

$$\widetilde{A} = \widetilde{V}\widetilde{\Lambda}\widetilde{V}^T = \begin{pmatrix} \widetilde{V}_1 & \widetilde{V}_2 \end{pmatrix} \begin{pmatrix} \widetilde{\Lambda}_1 & 0 \\ 0 & \widetilde{\Lambda}_2 \end{pmatrix} \begin{pmatrix} \widetilde{V}_1^T \\ \widetilde{V}_2^T \end{pmatrix}.$$

The eigenvalue matrices are such that,

$$\Lambda_1 = \mathrm{diag}(\lambda_1, \ldots, \lambda_p), \;\; \Lambda_2 = \mathrm{diag}(\lambda_{p+1}, \ldots, \lambda_n),$$
$$\widetilde{\Lambda}_1 = \mathrm{diag}(\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_p), \;\; \widetilde{\Lambda}_2 = \mathrm{diag}(\widetilde{\lambda}_{p+1}, \ldots, \widetilde{\lambda}_n)$$

for an arbitrarily chosen $p \in \{2, \ldots, n-1\}$.

It is possible to quantify how close the spaces spanned by the columns of V_1 and $\widetilde{V}_1$ are. In order to provide a measure of closeness between the two spaces, the following notion of angle matrix was defined in [46]:

$$\Theta(X_1, X_2) := \arccos\left((X_1^T X_1)^{-\frac{1}{2}} X_1^T X_2 (X_2^T X_2)^{-1} X_2^T X_1 (X_1^T X_1)^{-\frac{1}{2}}\right)^{-\frac{1}{2}},$$

where X_1 and X_2 are two matrices of the same dimension $n \times p$ with $n > p$ and full column rank. The singular values of $\Theta(X_1, X_2)$ are the angles required to rotate the space spanned by X_1 onto that of X_2. Going back to our notation of V_1, V_2, $\widetilde{V}_1$, and $\widetilde{V}_2$, it was shown [46] that

$$\|\sin\Theta(V_1, \widetilde{V}_1)\| = \|\widetilde{V}_2^T V_1\|$$

for any unitarily invariant norm. This fact can be used to bound the angle between V_1 and $\widetilde{V}_1$. In particular, if $\kappa := \min|\lambda(\Lambda_1) - \lambda(\widetilde{\Lambda}_2)| > 0$, then

$$\|\sin\Theta(V_1, \widetilde{V}_1)\|_F = \|\widetilde{V}_2^T V_1\|_F \leq \frac{\|E\|_F}{\kappa}.$$

Once again we note that the above bound relies on the absolute separation between eigenvalues (in contrast with the relative separation, which appears in Section 6.3.2). The above can be further generalized to any invariant norm. Detailed discussion on this subject can be found in [46].

6.3.2 Relative Bounds

The perturbation results discussed above are in terms of the absolute differences between eigenvalues. These types of results are most useful for ensuring the preservation of the largest eigenvalues but least useful for ensuring the relative preservation of the smallest eigenvalues; a small *absolute* change in a small eigenvalue could actually correspond to a large *relative* change in that eigenvalue. The absolute perturbation results are the best we can do when the perturbation to A is completely arbitrary. However, when the perturbation exhibits some structure one can do much better than what the absolute error bounds indicate.

Consider the class of perturbations that take the form $\widetilde{A} = A + E = D^T A D$, where D is nonsingular. It was shown [57] that in this case a relative perturbation bound for the eigenvalues is given by

$$|\widetilde{\lambda}_j - \lambda_j| \leq |\lambda_j| \|D^T D - I\|_2,$$

where the factor $\|D^T D - I\|_2$ represents how close D is to being an orthonormal matrix. In the extreme case when D has orthonormal columns we will have that $\widetilde{\lambda}_j = \lambda_j$ as expected.

Similarly, the angle $0 \leq \theta_j \leq \frac{\pi}{2}$ between the jth eigenvector and its corresponding perturbed eigenvector has been shown [57] to satisfy

$$\sin\theta_j \leq \frac{\|D^T D\|_2 \|(DD^T)^{-1} - I\|_2}{\rho_j(A) - \|D^T D - I\|_2} + \|D - I\|_2,$$

provided that $\rho_j(A) > \|D^T D - I\|_2$, where the jth relative gap, $\rho_j(A)$, of the eigenvalues of A is defined as

$$\rho_j(A) = \min_{i \neq j} \frac{|\lambda_i - \lambda_j|}{|\lambda_j|}.$$

We can see that for the type of perturbation described above we obtain a much stronger perturbation result that depends on the relative gap between the eigenvalues. There are many variants of relative perturbation results that have been proposed to date [89, 88, 57, 17] that differ from one another depending on the underlying matrix A (e.g., whether it is a symmetric matrix, positive definite matrix, indefinite matrix, etc.) and also on the type of perturbation.

6.3.3 Relation to Simultaneous Iteration

Our problem also has a close connection to various algorithms for computing the eigenvalues and eigenvectors of symmetric matrices. One algorithm that we shall focus on is the simultaneous iteration method [115, 36]. This method is best suited for cases when we are interested in computing the top few eigenvalues and their corresponding eigenvectors and when the underlying matrix is sparse.

To state the algorithm explicitly, let us set some notation. Let A be an $n \times n$ positive-semidefinite matrix with eigendecomposition $A = V \Lambda V^T$. We let λ_j and v_j denote the jth eigenvalue of A and its corresponding eigenvector. We also assume that the eigenvalues are ordered in descending order such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. We then pick a set of trial vectors and denote them as $p_1, p_2, \ldots, p_k$, where the number of trial vectors k depends on how many eigenvectors we wish to compute. The trial vectors can be any set of orthonormal vectors such that

$$\text{span}(p_1, \ldots, p_k) \bigcap \text{span}(v_{k+1}, \ldots, v_n) = \{0\}. \tag{6.2}$$

One possible choice of trial vectors is a set of k orthonormal vectors that are chosen randomly. Let us stack the trial vectors into columns of a matrix and denote it as $P = [p_1, \ldots, p_k]$. Given this

notation the simultaneous iteration method is carried out as follows:[1]

1. $W^{(0)} \leftarrow AP$
2. for $i = 1, 2, \ldots$
   (a) $Q^{(i)}R^{(i)} \leftarrow W^{(i)}$ via the QR-decomposition
   (b) $W^{(i+1)} \leftarrow AQ^{(i)}$
   (c) if stopping criterion is not met, set $i \leftarrow i + 1$ and go back to step (a), otherwise output $Q^{(i)}$.

As we can see, the simultaneous iteration method iteratively refines the set of eigenvectors of A. Denoting the jth column of $Q^{(i)}$ as $q_j^{(i)}$, it is known [115] that

$$\|q_j^{(i)} - v_j\|_2 = \mathcal{O}(p_j^i),$$

where $p_j = \max\{\lambda_{j+1}/\lambda_j, \lambda_j/\lambda_{j-1}\}$. From this we can see that the rate of convergence of the eigenvectors depends on the ratio between the eigenvalues. Put differently, an eigenvector that corresponds to an eigenvalue with a favorable eigenvalue ratio p_j will converge faster to the true eigenvector. This is also similar in spirit to the relative perturbation results that we discussed above in that p_j provides relative measure of the closeness between the eigenvalues, and the accuracy between v_j and $q_j^{(i)}$ depends on p_j.

Lastly, let us focus on the very first step in the simultaneous iteration algorithm, in which we multiply the original matrix A with k (potentially randomly chosen) vectors. Interestingly, this is similar in spirit to our algorithm, except that our data matrix X is not necessarily square or positive-semidefinite, and we do not require the rows of Φ to be orthonormalized. We would like to note that—when they are used in the simultaneous iteration method—random trial vectors are merely chosen as a way to satisfy the condition (6.2). We believe, however, that randomness will also help to better preserve the true eigenvectors in the first iteration.

6.3.4 Randomized Algorithms for Linear Algebra

In a similar vein, there have been a large number of results on what we will refer to as randomized algorithms for linear algebra. The monograph [93] covers a number of these methods and references.

There are several lines of work that are closely related to our results. The first involves the spectral analysis of random matrices and the application to algorithmic tasks such as information

[1] There are a few variants of the simultaneous iteration method. Here, we use the algorithm presented in [115].

retrieval and spectral random graph analysis. Two representative works are [10, 38] which build random matrix models (or random perturbations of random matrices) for data and graphs and then use those models to find the approximate spectral structures in the data (or the SVDs). Many of the perturbation results used in these papers fall into our category of absolute bounds.

A second line of work is that of robust PCA or low-rank matrix completion, of which [28] is just one example (there are many other such papers). In this problem, the data matrix X is either sampled or random linear measurements of the form $\Phi X \Phi^T$ are obtained, from which a sparse, low-rank approximation $\widehat{X}$ to the original data matrix is produced. The primary goal of this line of work is to approximate the original data with a parsimonious representation. Our work, in contrast, aims to recover or to approximate the parsimonious representation itself.

A third line of work is the recovery of principal components of a data matrix X from compressive projection measurements [111, 63]. Briefly speaking, in these works, a rectangular data matrix X of size $p \times n$ is considered where $p < n$, i.e., X has more columns than rows. Each column of X represents a data sample, and the objective in these works is to compute the *left* singular vectors of X from compressive projections of the columns of X. However, different random projection matrices are applied to different columns of X. The key differences between the work proposed in [111, 63] and our work are that 1) we are interested in the right singular vectors of X, 2) the data matrix X in our problem is assumed to have more rows than columns, 3) our measurement matrix is a JL matrix, and 4) we apply the same random matrix to every column of X.

Finally, we emphasize the distinction between subspace approximation and the approximation of the singular vectors themselves. Feldman et al. [61] give coreset and sketching algorithms for approximating subspaces spanned by portions of the data set. This problem is also similar to the work of Halko et al. [73], in which one constructs a basis for an approximate subspace spanned by the columns of X from a sketch $Y = \Phi X$ of the data. Finally, we note that the work of Drineas et al. [54] that approximates leverage scores of a matrix is similar in nature to ours but does not produce approximate singular vectors.

6.4 Main Result

6.4.1 Proposed Algorithm and Estimation Bounds

Recall the problem setup discussed in Section 6.2. Our algorithm for estimating Σ_X and V_X from the sketched matrix $Y = \Phi X$ is stated in Algorithm 1. This algorithm is very simple: we simply return the truncated singular values and right singular vectors of Y.

Algorithm 1: Pseudo-code for sketched SVD.

Input: Sketched matrix $Y = \Phi X$
Output: $\widehat{\Sigma}_X$ and $\widehat{V}_X$ (estimates of the singular values and right singular vectors, respectively, of X)
$(U_Y, \Sigma_Y, V_Y) \leftarrow \text{SVD}(Y)$
$\widehat{\Sigma}_X \leftarrow \Sigma_Y$
$\widehat{V}_X \leftarrow V_Y$

The computational complexity of our algorithm can be divided into two parts. The first part concerns the complexity of computing the sketch Y from Φ and X. If both Φ and X are available at a central processing node, Y can be computed simply by multiplying Φ and X; as discussed in Section 6.2, there may be a fast algorithm for doing this, depending on the structure of Φ. As we have noted, however, it is also possible to compute the sketch column-by-column by applying Φ separately to each column of X; when data is collected in a distributed fashion, this may be the natural way to construct a sketch. Let us denote the computational complexity computing Y as $T_1(m, N, n)$. In distributed scenarios, we will have $T_1(m, N, n) = nT'(m, N)$, where $T'(m, N)$ denotes the computational complexity of one matrix-vector multiplication with Φ.

The second part concerns the complexity of computing the SVD of Y. Using standard techniques, computing the SVD of an $m \times n$ matrix with $m \geq n$ requires $\mathcal{O}(mn^2)$ operations. (When k is very small compared to n, we may have $m < n$ and the SVD of Y can be computed even more efficiently than this.) Combining this fact with the bound on m provided in our main result (see (6.3)) and assuming $f(\epsilon)$ is quadratic in ϵ, the computational complexity of this second part will be $\mathcal{O}(n^2 k \epsilon^{-2}(\log(1/\epsilon) + \log(1/\delta))$, where δ denotes the failure probability. One can add this cost to $T_1(m, N, n)$ to determine the overall computational complexity, although it is important to stress that the computation of Y may be streaming or distributed over many sensors, while the computation of the SVD of Y may be performed all at once at a central computing node.

We are now ready to state our main result.

Theorem 6.2. *Let X be an $N \times n$ matrix with $N \geq n$ and rank$(X) = k \leq n$, and let $X = U_X \Sigma_X V_X^T$ denote the truncated SVD of X as explained in Section 6.2. Let $\epsilon \in (0, 1)$ denote a distortion factor and $\delta \in (0, 1)$ denote a failure probability, and suppose Φ is an $m \times N$ random matrix that satisfies the distributional JL property with*

$$m \geq \frac{k \log(42/\epsilon) + \log(2/\delta)}{f(\epsilon/\sqrt{2})}. \tag{6.3}$$

Let $Y = \Phi X$ denote the sketched matrix, and let $\widehat{\Sigma}_X = \Sigma_Y$ and $\widehat{V}_X = V_Y$ denote the estimated

singular vectors and right singular values of X returned by Algorithm 1. Then with probability exceeding $1-\delta$, rank$(Y) = k$ and both of the following statements hold:

1. *(Preservation of singular values) For all $j = 1, \ldots, k$,*

$$(1-\epsilon)^{1/2} \leq \frac{\sigma'_j}{\sigma_j} \leq (1+\epsilon)^{1/2},$$

where $\sigma_1 \geq \cdots \geq \sigma_k \geq 0$ denote the singular values of X (the diagonal entries of Σ_X) and $\sigma'_1 \geq \cdots \geq \sigma'_k \geq 0$ denote the singular values of Y (the diagonal entries of $\widehat{\Sigma}_X$).

2. *(Preservation of right singular vectors) For all $j = 1, \ldots, k$,*

$$\|v_j - v'_j\|_2 \leq \min\left\{\sqrt{2},\ \frac{\epsilon\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}} \max_{i \neq j} \frac{\sqrt{2}\sigma_i\sigma_j}{\min\limits_{c\in[-1,1]}\{|\sigma_i^2 - \sigma_j^2(1+c\epsilon)|\}}\right\},$$

where $v_1, \ldots, v_k$ denote the right singular vectors of X (the columns of V_X) and $v'_1, \ldots, v'_k$ denote the right singular vectors of Y (the columns of $\widehat{V}_X$).

Proof: See Section 6.4.2.

Corollary 6.3. *When Φ is generated randomly from an i.i.d. subgaussian distribution (suitably scaled) or some other random distribution satisfying the distributional JL property with quadratic $f(\cdot)$, the bounds in Theorem 6.2 will hold with $m = \mathcal{O}(k\epsilon^{-2}(\log(1/\epsilon) + \log(1/\delta))$.*

This result states that from Y we can obtain accurate relative estimates for the singular values of X, and in some cases we can obtain accurate estimates for the right singular vectors of X as well. However, we struggle to achieve high accuracy in the singular vectors when the corresponding singular values are close. This is a consequence of well-studied perturbation theory (recall the role that the relative gap played in Section 6.3.2) and seems inherent in our approach.

We note that, naturally, we could obtain similar results for preserving the left singular vectors of X were we to sketch its rows, rather than its columns. We also note that when the exact rank k of the data matrix is unknown (or if the rank of X is not necessarily below n), substituting n for k in the measurement bound (6.3) yields a guarantee that applies to any $N \times n$ matrix X with $N \geq n$.

6.4.2 Proof of Theorem 6.2

In this section we prove Statements 1 and 2 within Theorem 6.2. As noted in Section 6.2, it will be useful for us to write the truncated eigendecompositions of X^TX and Y^TY as $X^TX = V_X\Lambda_X V_X^T$

and $Y^TY = V_Y \Lambda_Y V_Y^T$, respectively, where $\Lambda_X = \text{diag}(\lambda_1, \ldots, \lambda_k)$ and $\Lambda_Y = \text{diag}(\lambda'_1, \ldots, \lambda'_k)$. Recall that $\lambda_j = \sigma_j^2$ and $\lambda'_j = \sigma_j'^2$ for $j = 1, \ldots, k$.

6.4.2.1 Proof of Statement 1

In order to prove Theorem 6.2, we require the following result, adapted from Lemma 5.1 in [14] and Theorem 4.3 in [42].

Lemma 6.4 ([14, 42])**.** *Let $\mathcal{X}$ denote a k-dimensional subspace of $\mathbb{R}^N$. Let $\epsilon \in (0,1)$ denote a distortion factor and $\delta \in (0,1)$ denote a failure probability, and suppose Φ is an $m \times N$ random matrix that satisfies the distributional JL property with*

$$m \geq \frac{k \log(42/\epsilon) + \log(2/\delta)}{f(\epsilon/\sqrt{2})}.$$

Then with probability exceeding $1 - \delta$,

$$\sqrt{1-\epsilon}\|x\|_2 \leq \|\Phi x\|_2 \leq \sqrt{1+\epsilon}\|x\|_2,$$

for all $x \in \mathcal{X}$.

To see how this lemma can help us guarantee the preservation of the singular values of X (or, equivalently, the eigenvalues of X^TX), we begin by noting that

$$Y^TY = X^T\Phi^T\Phi X = V_X \Sigma_X U_X^T \Phi^T \Phi U_X \Sigma_X V_X^T.$$

We define a new $k \times k$ matrix

$$M := V_X^T Y^T Y V_X = \Sigma_X U_X^T \Phi^T \Phi U_X \Sigma_X$$

and represent its eigendecompostion as $M = V_M \Lambda_M V_M^T$. Noting that $Y^TY = V_Y \Sigma_Y^2 V_Y^T$, we have

$$M = V_X^T Y^T Y V_X = V_X^T V_Y \Sigma_Y^2 V_Y^T V_X.$$

From this we can infer that $\Lambda_M = \Sigma_Y^2$, i.e., M has the same set of eigenvalues as Y^TY. Thus, we turn our attention to proving that the eigenvalues of M approximate the eigenvalues of X^TX. Let

us define $\Delta_\Phi := \Phi^T\Phi - I$ and consider the ratio

$$\frac{x^T M x}{x^T \Sigma_X^2 x} = \frac{x^T \Sigma_X U_X^T \Phi^T \Phi U_X \Sigma_X x}{x^T \Sigma_X^2 x} = \frac{x^T \Sigma_X (I + U_X^T \Delta_\Phi U_X) \Sigma_X x}{x^T \Sigma_X^2 x}$$
$$= 1 + \frac{x^T \Sigma_X U_X^T \Delta_\Phi U_X \Sigma_X x}{x^T \Sigma_X^2 x}.$$

We will be interested in the range of values that the fraction $(x^T M x)/(x^T \Sigma_X^2 x)$ can take over all nonzero $x \in \mathbb{R}^k$. We note that for any vector $x \in \mathbb{R}^k$ we can associate a vector $y := \Sigma_X x \in \mathbb{R}^k$ and write

$$\frac{x^T M x}{x^T \Sigma_X^2 x} = 1 + \frac{y^T U_X^T \Delta_\Phi U_X y}{y^T y}.$$

To bound the range of values that this quantity can take, it suffices to consider all vectors $y \in \mathbb{R}^k$ with unit norm. Thus, we focus on the quantity

$$1 + y^T U_X^T \Delta_\Phi U_X y = 1 + y^T U_X^T (\Phi^T \Phi - I) U_X y = y^T U_X^T \Phi^T \Phi U_X y = \|\Phi U_X y\|_2^2. \tag{6.4}$$

Our next step is to apply Lemma 6.4 on the subspace $\mathcal{X} = \text{colspan}(U_X)$, using the fact that $\|U_X y\|_2 = \|y\|_2 = 1$. This tells us that with a probability of at least $1 - \delta$,

$$1 - \epsilon \leq \|\Phi U_X y\|_2^2 \leq 1 + \epsilon,$$

holds for all unit norm vectors $y \in \mathbb{R}^k$. Combining this inequality with (6.4) we get

$$-\epsilon \leq y^T U_X^T \Delta_\Phi U_X y \leq \epsilon, \tag{6.5}$$

which implies that for any nonzero $x \in \mathbb{R}^k$,

$$1 - \epsilon \leq \frac{x^T M x}{x^T \Sigma_X^2 x} \leq 1 + \epsilon. \tag{6.6}$$

In order to complete the proof we use the following lemma, which is a simplification of Lemma 1 in [17].

Lemma 6.5 (Lemma 1, [17])**.** *Let H be a diagonal matrix and suppose δH has the property that for all nonzero x,*

$$g_l \leq \frac{x^T (H + \delta H) x}{x^T H x} \leq g_u,$$

where $0 < g_l \leq g_u$. *Then*

$$g_l \leq \frac{\lambda_i(H + \delta H)}{\lambda_i(H)} \leq g_u$$

for all i, *where* $\lambda_i(Z)$ *denotes the* i*th eigenvalue of the matrix* Z.

Applying Lemma 6.5 to (6.6) with $H = \Sigma_X^2$ (which is diagonal) and $H + \delta H = M$ completes the proof of Statement 1 of Theorem 6.2 and also implies that $\text{rank}(Y) = k$.

6.4.2.2 Proof of Statement 2

In order to prove Statement 2 of Theorem 6.2, we require the following important theorem.

Theorem 6.6 (Theorem 1, [99]). *Let* $H = U\Gamma U^*$ *and* $\widetilde{H} = H + \delta H = \widetilde{U}\widetilde{\Gamma}\widetilde{U}^*$ *be* $p \times p$ *positive definite matrices. Assume that* U *and* $\widetilde{U}$ *are unitary and that* $\Gamma = diag(\gamma_1, \ldots, \gamma_p)$ *and* $\widetilde{\Gamma} = diag(\widetilde{\gamma}_1, \ldots, \widetilde{\gamma}_p)$ *are diagonal. Let* $S = U^*\widetilde{U}$, *and assume*

$$\eta = \|H^{-\frac{1}{2}} \delta H H^{-\frac{1}{2}}\| < 1,$$

where $H^{-\frac{1}{2}} = U\Gamma^{-1/2}U^*$. *Then for any* j *and for any set* $\mathscr{T}$ *not containing* j *we have,*

$$\left(\sum_{i \in \mathscr{T}} |s_{ij}|^2 \right)^{1/2} \leq \min \left\{ 1, \ \max_{i \in \mathscr{T}} \frac{\gamma_i^{1/2} \widetilde{\gamma}_j^{1/2}}{|\gamma_i - \widetilde{\gamma}_j|} \frac{\eta}{\sqrt{1 - \eta}} \right\},$$

and, in particular, for any $i \neq j$,

$$|s_{ij}| \leq \min \left\{ 1, \ \frac{\gamma^{1/2} \widetilde{\gamma}_j^{1/2}}{|\gamma_i - \widetilde{\gamma}_j|} \frac{\eta}{\sqrt{1 - \eta}} \right\}.$$

To prove Statement 2, we continue from the proof of Statement 1. In particular, we suppose (6.5) holds for all unit norm vectors $y \in \mathbb{R}^k$ (recall that this event happens with probability at least $1 - \delta$). Our first goal will be to prove that this implies that

$$\|U_X^T \Delta_\Phi U_X\|_2 \leq \epsilon. \tag{6.7}$$

To see why this follows, let us for notational simplicity denote $A = U_X^T \Delta_\Phi U_X$. Note that A is a symmetric matrix. Then, (6.5) says that $-\epsilon \leq y^T A y \leq \epsilon$ holds for all unit norm vectors $y \in \mathbb{R}^k$. Note that this is equivalent to $-\epsilon \leq \frac{y^T A y}{y^T y} \leq \epsilon$. This fraction is the well known Rayleigh quotient. It can be shown that the range of values that the Rayleigh quotient takes is confined between the minimum and the maximum eigenvalues of A. Let us denote the maximum and minimum eigenvalues of A as

$\alpha_{\max}$ and $\alpha_{\min}$, respectively. Since equation (6.5) says that the Rayleigh quotient is in between $-\epsilon$ and ϵ we can infer that $-\epsilon \leq \alpha_{\min} \leq \frac{y^T A y}{y^T y} \leq \alpha_{\max} \leq \epsilon$. Thus, $\|A\|_2 = \max\{|\alpha_{\min}|, |\alpha_{\max}|\} \leq \epsilon$ and so we have proved that (6.7) holds.

To quantify $\|v_j - v'_j\|_2$, we look at a different yet equivalent quantity that may simplify the problem. Let us again look at the matrix M that we introduced in the proof of Statement 1. We have seen that there is a close connection between the eigenvalues of M and those of $Y^T Y$. We now show that in order to prove that the eigenvectors of $Y^T Y$ approximate those of $X^T X$, it suffices to study the eigenvectors of M.

Remembering that

$$M = V_X^T Y^T Y V_X = V_X^T V_Y \Sigma_Y^2 V_Y^T V_X,$$

we can see that the eigenvectors of M are closely related to the right singular vectors of X and Y. Specifically, we have that $V_M = V_X^T V_Y$, and denoting the jth eigenvector of M as $\widetilde{v}_j$, it is easy to see that $\widetilde{v}_j = V_X^T v'_j$. This implies that $\langle \widetilde{v}_j, e_j \rangle = \langle v_j, v'_j \rangle$ for $j = 1, \ldots, k$, where e_j represents the jth canonical basis vector. Furthermore, we note that $\text{colspan}(V_Y) = \text{rowspan}(Y) = \text{rowspan}(X)$ since every row in Y is a linear combination of the rows in X and since we have argued above that $\text{rank}(Y) = \text{rank}(X)$. From this (and the fact that $v'_j \in \text{colspan}(V_Y)$) it follows that $\|\widetilde{v}_j\|_2 = \|V_X^T v'_j\|_2 = 1$. Now, using the relation $\langle \widetilde{v}_j, e_j \rangle = \langle v_j, v'_j \rangle$ and the facts that $\|\widetilde{v}_j\|_2 = \|e_j\|_2 = \|v_j\|_2 = \|v'_j\|_2 = 1$, we see that $\|\widetilde{v}_j - e_j\|_2 = \|v_j - v'_j\|_2$. To make sense out of the quantity $\|\widetilde{v}_j - e_j\|_2$, let us examine the expression $M = \Sigma_X^2 + \Sigma_X U_X^T \Delta_\Phi U_X \Sigma_X$. We can view M as the sum of a diagonal matrix Σ_X^2 and a perturbation matrix $\Sigma_X U_X^T \Delta_\Phi U_X \Sigma_X$. The eigenvectors of Σ_X^2 are the canonical basis vectors e_j. Therefore, the quantity $\|\widetilde{v}_j - e_j\|_2$ reflects the amount of perturbation in the eigenvectors of M. This is why, to bound $\|v_j - v'_j\|_2$, it suffices to focus on the perturbation analysis of M.

We apply Theorem 6.6 as follows: As we have discussed, we will quantify $\|v_j - v'_j\|_2$ via $\|\widetilde{v}_j - e_j\|_2$. Let us set the original unperturbed matrix as $H = \Sigma_X^2$ and the perturbation to this matrix as $\delta H = \Sigma_X U_X^T \Delta_\Phi U_X \Sigma_X$, such that

$$\widetilde{H} = H + \delta H = \Sigma_X (I + U_X^T \Delta_\Phi U_X) \Sigma_X = M,$$

and both H and $\widetilde{H}$ are $k \times k$. Clearly, H is positive definite since it is a diagonal matrix with all positive entries along the diagonal (because $\text{rank}(X) = k$). To check that M is positive definite, note that $M = \Sigma_X U_X^T \Phi^T \Phi U_X \Sigma_X$ is of the form $M = B^T B$, where $B = \Phi U_X \Sigma_X$ is an $m \times k$ matrix with $m \geq k$. The fact that B has full column rank will follow because all diagonal entries of

Σ_X are nonzero (again, because $\text{rank}(X) = k$) and because in the proof of Statement 1 we applied Lemma 6.4 on the subspace $\mathcal{X} = \text{colspan}(U_X)$. Because B has full column rank, M will be positive definite.

We further have that

$$\eta = \|H^{-\frac{1}{2}}\delta H H^{-\frac{1}{2}}\|_2 = \|\Sigma_X^{-1}\delta H \Sigma_X^{-1}\|_2 = \|U_X^T \Delta_\Phi U_X\|_2.$$

Then, applying (6.7), $\eta = \|U_X^T\Delta_\Phi U_X\|_2 \leq \epsilon$. Let us set $S = I^T V_M = V_M$ and denote the jth eigenvalue of M as $\widetilde{\lambda}_j$. Then, straightforward application of Theorem 6.6 yields

$$\left(\sum_{i\neq j}|s_{ij}|^2\right)^{1/2} \leq \min\left\{1,\ \max_{i\neq j}\frac{\sigma_i\widetilde{\lambda}_j^{1/2}}{|\sigma_i^2-\widetilde{\lambda}_j|}\frac{\eta}{\sqrt{1-\eta}}\right\} \leq \min\left\{1,\ \max_{i\neq j}\frac{\sigma_i\widetilde{\lambda}_j^{1/2}}{|\sigma_i^2-\widetilde{\lambda}_j|}\frac{\epsilon}{\sqrt{1-\epsilon}}\right\}.$$

As we have discussed in Section 6.2 we assume that $s_{jj} = \langle \widetilde{v}_j, e_j\rangle = \langle v_j, v_j'\rangle \geq 0$. Then,

$$\begin{aligned}
\|v_j - v_j'\|_2 = \|\widetilde{v}_j - e_j\|_2 &= \sqrt{\|\widetilde{v}_j\|_2^2 - 2\langle\widetilde{v}_j, e_j\rangle + \|e_j\|_2^2}, \\
&= \sqrt{2}\sqrt{1-s_{jj}}, \\
&= \sqrt{2}\sqrt{1-\sqrt{1-\sum_{i\neq j}|s_{ij}|^2}}, \\
&\leq \sqrt{2}\sqrt{1-\sqrt{1-\min\left\{1,\ \max_{i\neq j}\frac{\sigma_i^2\widetilde{\lambda}_j}{(\sigma_i^2-\widetilde{\lambda}_j)^2}\frac{\epsilon^2}{1-\epsilon}\right\}}}, \\
&\leq \sqrt{2}\min\left\{1,\ \max_{i\neq j}\frac{\sigma_i\widetilde{\lambda}_j^{1/2}}{|\sigma_i^2-\widetilde{\lambda}_j|}\frac{\epsilon}{\sqrt{1-\epsilon}}\right\},
\end{aligned}$$

where the last inequality is due to the fact that $1-\sqrt{1-x} \leq x$ for $0 \leq x \leq 1$.

To write the above only in terms of the unperturbed singular values, σ_j, we make use of Statement 1 and the fact that $\widetilde{\lambda}_j = (\sigma_j')^2$ for $j = 1, \ldots, k$ to obtain

$$\|v_j - v_j'\|_2 \leq \min\left\{\sqrt{2}, \frac{\epsilon\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}}\max_{i\neq j}\frac{\sqrt{2}\sigma_i\sigma_j}{\min\limits_{c\in[-1,1]}\{|\sigma_i^2-\sigma_j^2(1+c\epsilon)|\}}\right\}.$$

6.5 Application to Spectral Analysis of Streaming Graphs

In this section, we apply our data analysis framework to streaming graphs, a model of data collection where edges of a graph are updated dynamically. We consider a scenario in which edges

are inserted and deleted over an observation period, and our goal is to maintain a small data structure that encodes the graph information so that we may analyze the spectrum of the graph quickly at any point during or after the sequence of edge updates. Spectral graph analysis has a multitude of applications including graph embedding, graph isomorphism testing, data clustering/segmentation (of which there are yet many more applications!), numerical linear algebra, etc. We refer the reader to just a few in [19, 103, 122, 35]. Determining the spectrum of a graph is at the heart of many modern data analysis and graphical information processing algorithms.

Let $G = (V, E)$ be a graph with vertex set V and undirected, unweighted edges E. Let A denote the symmetric binary adjacency matrix of G, denote by d_v the degree of a vertex $v \in V$, and define the graph Laplacian as

$$L_G(u, v) = \begin{cases} d_v & \text{if } u = v \\ -1 & \text{if } u \text{ and } v \text{ are adjacent} \\ 0 & \text{otherwise.} \end{cases}$$

There is a compact definition of L_G using the adjacency matrix: $L_G = \text{diag}(d_v) - A$.

Let X be the incidence matrix of the graph G. This matrix has $N = |E|$ rows and $n = |V|$ columns and to define each entry of X, consider an edge (u, v) between vertices u and v. Since the graph is undirected, the ordering of the vertices is chosen arbitrarily. Then,

$$X_{(u,v),u} = 1 \quad \text{and} \quad X_{(u,v),v} = -1.$$

It is well-known that the rank of the graph is the rank of the incidence matrix and that this value is $|V| - c$ where c is the number of connected components in G. If the graph G is weighted, we replace the ± 1's with the appropriate weights in the incidence matrix.

From the definitions of the graph Laplacian and the incidence matrix, it is clear that $L_G = X^T X$. The singular values of X are, therefore, related to the eigenvalues of L_G in a straightforward fashion:

$$\sigma_i(X) = \sqrt{\lambda_i(L_G)}.$$

Furthermore, the *right* singular vectors V of $X = U_X \Sigma_X V_X^T$ are the eigenvectors of the Laplacian. Thus, it is sufficient to compute (good) approximations to the singular values and the right singular vectors of X to obtain (good) approximations to the top eigenvalues of L_G and the corresponding eigenvectors. From standard spectral graph theory, we know that the eigenvalues λ_i of the Laplacian

satisfy $\lambda_1 \geq \cdots \geq \lambda_n = 0$ and, with the assumption that G has c connected components,

$$\lambda_1 \geq \cdots \geq \lambda_{n-c} > \lambda_{n-c+1} = \cdots = \lambda_n = 0.$$

In particular, $\operatorname{rank}(X) = \operatorname{rank}(L_G) = n - c$.

Next, we define the streaming graph model. Following [4, 3], we define a dynamic graph stream as a stream of edge updates (both insertions and deletions). This is a faithful model of the evolution of an online social network, for example, in which users connect and disconnect to other users over time [86].

Definition 6.7 (Dynamic graph stream)**.** A stream $S = \langle a_1, \ldots, a_T \rangle$ where $a_t = (j_t, k_t, \Delta_t) \in [n] \times [n] \times \mathbb{R}$ defines a weighted graph $G = (V, E)$ where $V = [n]$ and the weight of an edge (j, k) is given by

$$A(j,k) = \sum_{t:(j_t,k_t)=(j,k) \text{ or } (k,j)} \Delta_t.$$

We assume that at any update time t, the adjacency matrix A is well-formed; that the edge weight is non-negative; and that the graph has no self-loops.

In this prototype application, the stream of edge updates S defines the edge-vertex incidence matrix X of the graph G. The matrix X has $N = \binom{n}{2}$ rows and n columns, and for each stream item, we update two entries in X as

$$X_{(j_t,k_t),j_t} = X_{(j_t,k_t),j_t} + \Delta_t,$$
$$X_{(j_t,k_t),k_t} = X_{(j_t,k_t),k_t} - \Delta_t.$$

We collect sketches of each column of X and aggregate them into a matrix Y. Denoting the jth column of X by x_j, the jth column of the sketched matrix is given by $y_j = \Phi x_j$. We can update the sketch in a streaming fashion. Upon receipt of a stream item (u_t, v_t, Δ_t), we update y_{u_t} and y_{v_t}:

$$y_{u_t} = y_{u_t} + \Delta_t \phi_{(u_t,v_t)},$$
$$y_{v_t} = y_{v_t} - \Delta_t \phi_{(u_t,v_t)},$$

where ϕ_j denotes the jth column of Φ.

Our main result, Theorem 6.2, tells us that a sketch of the matrix X is sufficient to recover information about its singular value decomposition.

Corollary 6.8. *Assume that the undirected, weighted graph $G = (V, E)$ is presented in a streaming fashion so that its incidence matrix X has $n = |V|$ columns, $N = \binom{n}{2}$ rows, and rank $k \leq n-1$. Let $\epsilon \in (0,1)$ denote a distortion factor and $\delta \in (0,1)$ denote a failure probability, and suppose Φ is an $m \times N$ random matrix that satisfies the distributional JL property with*

$$m \geq \frac{k \log(42/\epsilon) + \log(2/\delta)}{f(\epsilon/\sqrt{2})}.$$

Let $Y = \Phi X$ denote an $m \times n$ sketch of X maintained in the streaming graph model, and let $\widehat{\Sigma}_X = \Sigma_Y$ and $\widehat{V}_X = V_Y$ denote the estimated singular vectors and right singular values of X returned by Algorithm 1. Then with probability at least $1 - \delta$, the following statements hold:

1. *(Preservation of eigenvalues) For all $j = 1, \ldots, k$,*

$$1 - \epsilon \leq \frac{\lambda'_j}{\lambda_j} \leq 1 + \epsilon$$

   where λ_j denote the true eigenvectors of the graph Laplacian L_G and λ'_j denote the estimated eigenvalues obtained by squaring the diagonal entries of $\widehat{\Sigma}_X$.

2. *(Preservation of eigenvectors) For all $j = 1, \ldots, k$,*

$$\|v_j - v'_j\|_2 \leq \min\left\{\sqrt{2},\ \frac{\epsilon\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}} \max_{i \neq j} \frac{\sqrt{2}\lambda_i^{1/2}\lambda_j^{1/2}}{\min_{c \in [-1,1]}\{|\lambda_i - \lambda_j(1 + c\epsilon)|\}}\right\},$$

   where v_j are the eigenvectors of the graph Laplacian L_G and v'_j denote the estimated eigenvectors obtained from the columns of $\widehat{V}_X$.

Because the adjacency matrix A of G has at most $|V|^2$ non-zero entries, this result is useful only when the rank k of L_G is significantly smaller than $n = |V|$, the number of vertices; or, equivalently, when G has many connected components. In this case, the size of the sketch is smaller than that of the adjacency matrix. In summary, for highly disconnected graphs presented in a streaming fashion, we can recover the approximate eigenvalues and eigenvectors of the Laplacian. The sparsity of the matrix Φ and the speed with which we can update the sketch matrix Y under a stream of updates are functions of the quality of the JL transform. The structural evolution of online social networks [86] suggests that it is reasonable to assume that the underlying graph has a significant fraction of vertices in small, disconnected components so that the graph is essentially a low-rank graph.

6.6 Conclusion

We present a data collection and analysis scheme that permits the distributed collection of data X by resource constrained sensors in a network and the central computation of the spectral decomposition of $X^T X$ or the right singular vectors of the data itself. The algorithm returns not an approximation to the original data, but a good approximation to the singular values σ_j and the right singular vectors v_j of the data. This data collection and analysis framework makes a small number of linear, non-adaptive measurements of the data. The number of measurements each sensor makes is comparable to the rank of the data and, if the data are full rank, the number of measurements at each sensor is comparable to the total number of sensors. This efficient data collection is especially important for sensors that are severely resource constrained and cannot store or transmit a large amount of data to a central device; we believe that one possible application of such an algorithm would be in operational modal analysis of structures (buildings, bridges, etc.) [108].

CHAPTER 7

Compressed Sensing in Structural Health Monitoring Systems

7.1 Introduction

7.1.1 Structural health monitoring systems

Over the past decade, more than 5 million commercial buildings[1], 130 million housing units[2], and 0.6 million bridges[3] have been built in the United States. This trend will continue and engineers and architects will continue to build ever more complex and larger structures in the future. Although buildings are built safe and robust at the time of construction, structural damage caused over time by continuous use is inevitable. In order to maintain safely operable structures for as long as possible, periodic inspections are a must. The purpose of these inspections is to locate damaged areas such as cracks or wear and tear in structures. Once damage is detected and considered repairable the structure will undergo repairs to maintain safely operable conditions and to extend its life expectancy. If, on the other hand, the damage is considered irreparable and the state of the structure inoperable, the structure must be taken out of service immediately.

Due to the large sizes and complex natures of structures, the task of inspection is labor intensive, costly and time consuming. In light of these problems, there have been significant efforts in the structural engineering community to automate the task of health monitoring of structures. Structural Health Monitoring (SHM) systems are precisely designed to address this issue. Throughout the SHM literature, researchers in the field have proposed numerous methods to make this task as efficient

This work is in collaboration with Michael B. Wakin and Anna C. Gilbert.

[1] 2003 Commercial Buildings Energy Consumption SurveyOverview of Commercial Buildings Characteristics, http://www.eia.gov/consumption/commercial/data/2003/pdf/a5.pdf

[2] http://factfinder2.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=AHS_2011_C01AH&prodType=table

[3] http://www.infrastructurereportcard.org/fact-sheet/bridges

and as accurate as possible. Although the details of each of the proposed SHM systems vary, there are common features that are shared among many methods [121, 49].

A typical SHM system monitors in-service structures in real-time. To do so, it makes use of a network of sensors installed on structures to collect relevant vibration data for damage detection. These may include acceleration data, strain data, or displacement data. The acquired data from each sensor are transmitted over a network to the central data repository where damage detection algorithms are run to detect, localize, classify, etc., possible damage in structures.

An important part of damage detection is the identification of modal parameters, which is called modal analysis. It is carried out to infer modal properties such as modal frequencies, mode shapes and modal damping ratio. These parameters describe the vibrational characteristics when external forces such as wind, earth quakes, or vehicle loadings are subjected to structures. For example, when a structure is forced to vibrate close to a modal frequency, the shape of vibration of the structure will be dominated by the corresponding mode shape. This vibration will eventually die out in the absence of external force, and the size of damping ratio will determine how fast the vibration will come to an end.

Many damage detection algorithms make use of these parameters to detect, localize, and assess the severity of damage. Briefly speaking, damage detection methods rely on the notion that structural properties will change once the structures become damaged, thus causing the modal parameters to change. Assuming that one has the modal parameters at the time when the structure was healthy, these are compared to the current estimates of modal parameters to judge whether or not damage has occurred. A comprehensive survey of damage detection methods is presented in [121, 49].

7.1.2 The usefulness of wireless SHM systems

In the early designs of SHM systems, sensors were linked via coaxial cables that provided reliable communication to the central data repository. In addition, power cables were coupled with the coaxial cables providing requisite power to run the sensors. However, due to the large sizes and complex designs of structures, installing the sensors together with the coaxial cables and power cables was expensive, difficult, and time-consuming. Hence, it was generally the case that the deployment of dense networks of sensors were too expensive, allowing only a few sensors to be installed. A dense network of sensors is generally desired as it allows for more accurate damage detection analysis.

As a way to overcome this issue researchers have proposed to deploy wireless sensors on structures to acquire and transmit data to the central repository. With the rapid advancement of wireless

technology and the ability to build sensors at low cost, it has become possible to deploy a much denser network of sensors for a given budget. Therefore, the use of wireless sensors in place of wired ones provides the advantages of low installation cost, rapid installation time and density of sensors.

The challenges of wireless sensor networks in SHM are quite different from those of wired sensor networks. Contrary to a wired sensor network, a primary challenge in a wireless network is to maximize the life expectancy of the battery that is used to power each sensor in the network. There are important questions that need to be asked when designing a wireless sensor from the battery saving point of view: Should we compress the time data before we transmit it to the central data repository? This will allow us to transmit less information at the cost of additional power for processing. Or, would we save more battery by sending the raw data itself without processing the data at all?

In [92], the authors presented thorough answers to these questions in real world applications for a certain wireless sensor. The authors found that on average, significant savings in battery can be expected by locally compressing and processing the data first compared to sending the raw time data. The main reason for this was that the radio that transmits the data drains much more power (almost 3 times as much) than the processor that carries out the local computation. As long as the execution time of the local algorithm is reasonably short, it would be more beneficial to first process and compress, then transmit less information.

Another factor that contributes to the severe draining of the battery is the sampling interval. Obviously, the faster and/or the longer we sample the more power is required. As we know, the Nyquist-Shannon sampling theorem tells us that the sampling frequency should be proportional to the maximum frequency content in the signal. For the purpose of modal analysis the highest frequency content will be determined via the highest modal frequency of the underlying structure. Intuitively, structures that are stiff and light will generally posses high modal frequencies and for such structures we may have to sample at relatively high frequencies. The length of total sampling interval also plays an important role in optimizing the battery life. Obviously, it would be best to sample only for the requisite amount of time and turn the sensor off once we have all the information we need.

In light of these observations, we believe that wireless sensors equipped with Compressed Sensing (CS) architectures will be a perfect fit to address the concerns in the conventional wireless systems. To see how, let us highlight the advantages that a CS wireless sensor could bring to this scenario. CS in wireless sensors will be able to minimize the battery usage by directly obtaining the compressed time data, bypassing the need for any local processing. This also allows us to sample at a rate lower

than the Nyquist rate providing a further reduction in battery usage. As a result, each sensor will transmit minimal amount of data (typically proportional to the information level of each signal) to the data repository, which in turn will reduce the power drained by the transmission radio. Lastly, all of the sensors will acquire the time data in a completely disjoint fashion eliminating the need for any additional communication amongst sensors. This, again, will help reduce the power drained by the transmission radio. Mathematically, denoting the continuous-time displacement signal at each sensor as $v_n(t)$, where $n = 1, \ldots, N$ and N denotes the number of nodes, a CS architecture would simply sample at time points $t_1, \ldots, t_M$ and transmit the resulting vector $\{v_n\} = \{v_n(t_1), \ldots, v_n(t_M)\}$. The compression would come from the fact that M is smaller than the number of Nyquist samples. Typically, one would choose $t_1, \ldots, t_M$ randomly.

The aforementioned ideal attributes of CS equipped sensors come with a price. Because we have significantly undersampled the original signal at each node, the decoder has to face a severely ill-posed system of equations. There have been several papers presenting the application of CS to SHM systems. In [98], the authors have implemented a prototype wireless sensor, which takes compressed measurements locally once the wireless sensor acquires the time data. By sending both compressed measurements and the raw time measurements to the central node, the authors evaluate the performance of CS reconstruction of the raw data. It is shown that a number of measurements $M \approx 0.8L$ is needed for an accurate reconstruction of the raw data, where L represents the length of the original signal. Once the time data are reconstructed, conventional modal analysis techniques are carried out that aid in damage detection. In [148], the authors also reconstruct the original time data from CS measurements obtained at each sensor. Similarly to the work in [98], the authors claim that many measurements were required to reach an accurate level of reconstruction.

The main reason for the above methods to require such large fractions of measurements is because each signal was simply not sparse enough in the Discrete Wavelet Transform (DWT) or the Discrete Fourier Transform (DFT). The lack of sparsity of each individual signal suggests that the model of sparsity of each signal may not be an appropriate model for the types of signals that arise in modal analysis. A joint signal model for the signal entire ensemble may serve as a better model to exploit the abundant correlations that potentially exist across the signals.

A timely question to ask at this point is whether or not we truly need to reconstruct the underlying signal at all. The only reason to reconstruct the signals in the above methods is to utilize conventional modal analysis tools that all rely on signals that are sampled at the Nyquist rate. Although this approach to first reconstruct then extract modal parameters is a viable option, there are two reasons to contemplate this question. The first reason is the potential loss of performance. The frequency

content within the signals plays an important role in the subsequent modal analysis algorithm. For example, a popular modal analysis algorithm proposed in [24] is very sensitive to the accuracy of each frequency component of the signal. As we anticipate noise in our acquired signals the reconstruction of each signal will also be noisy. These errors will propagate to the modal analysis step, which could potentially lead to misidentification and errors in modal parameters.

The second reason is the added computational complexity. Taking the above method for example, using an off-the-self CS reconstruction algorithm presented in [102], the total computational complexity for N signals will scale as $NL\log^2(L)$. Added to this will be the additional computation required for the subsequent modal analysis step. One can view this step of reconstruction as being somewhat wasteful as this is carried out only to utilize one of the conventional modal analysis algorithms.

The main objective of this chapter is to propose a novel method that directly extracts the mode shapes from CS measurements without ever reconstructing the individual time signals. Our proposed method differentiates itself from the previously proposed CS based methods in that it exploits the joint signal structure that can be observed through equations describing a simplified (no damping and free-decay) Multiple-Degree-Of-Freedom (MDOF) system. Our method is as simple as computing the Singular Value Decomposition (SVD) of the signal matrix obtained by stacking each $\{v_n\}$ into a matrix. We evaluate the performance of this method when $t_1, \ldots, t_M$ are uniform time points, and also when $t_1, \ldots, t_M$ are random time points. For each sampling method, we give sufficient conditions on the required sampling rate, the total sampling time span, and the total number of measurements for the accurate recovery of mode shape vectors. Our analysis reveals the surprising fact that the requisite sampling rate for the uniform measurement case is lower than the Nyquist rate, but that the required number of measurements is structure dependent. The analysis on the random measurements shows that we can achieve the same recovery guarantee as the uniform sampling case at the expense of more but structure independent number of measurements. At the end of this chapter, we present promising simulation results that show that our algorithm is able to accurately estimate the mode shapes given only a small fraction of measurements of the original signal.

7.2 Background

In this section, we give an introduction to the frequently used mathematical model that governs the motion of structures. We begin with the simplest system called the Single-Degree-Of-Freedom

(SDOF) system, and then move on to the MDOF system. Following what is standard in structural dynamics community, we will use $\{x\}$ to denote a vector x and $[A]$ to denote a matrix A. We will denote the lth entry of $\{x\}$ as $\{x\}(j)$, and denote $[A]_{l,n}$ as the lth row and nth column entry of $[A]$. Furthermore, we will reserve $i = \sqrt{-1}$ to denote the imaginary unit and $[A]^*$ to denote the conjugate transpose of $[A]$.

7.2.1 Single-degree-of-freedom system

An SDOF system under no external force can be described by the following differential equation,

$$m\ddot{x}(t) + C\dot{x}(t) + kx(t) = 0, \tag{7.1}$$

where m, C, and k denote the mass, damping, and stiffness parameters of the underlying system. To solve for the signal $x(t)$ that satisfies the above equation, let us assume a solution of the form $x(t) = Ae^{st}$, where $A, s \in \mathbb{C}$. Then, $\ddot{x}(t) = As^2e^{st}$, $\dot{x}(t) = Ase^{st}$, $x(t) = Ae^{st}$, and by plugging in these expressions into equation (7.1) we get,

$$\left(ms^2 + Cs + k\right) x(t) = 0.$$

Since the above needs to be satisfied for all t, it must be that $ms^2 + Cs + k = 0$, and it is easy to see that $s = \frac{-C \pm \sqrt{C^2 - 4mk}}{2m}$. In the structural dynamics community it is customary to rewrite this solution as,

$$s = -\xi\omega_0 \pm \omega_0\sqrt{\xi^2 - 1}, \tag{7.2}$$

where $\omega_0 = \sqrt{\frac{k}{m}}$, $\xi = \frac{C}{2m\omega_0}$ represent the natural frequency and damping ratio, respectively. As we can see, the natural frequencies and damping ratios will always be positive, and depending on the value of ξ, s may be real or complex and there may be one or two possible solutions.

In this chapter, we will be restricting ourselves to the case when there is no damping, i.e., $C = 0$ and thus $\xi = 0$. From equation (7.2) we can see that when $\xi = 0$, we will have two purely imaginary solutions $s_1 = i\omega_0$, and $s_2 = -i\omega_0$. Thus, $x_1(t) = Ae^{i\omega_0 t}$, and $x_2(t) = Be^{-i\omega_0 t}$ are both eligible solutions to equation (7.1). In fact, any solution to the above equation can be expressed as linear combinations of $x_1(t)$ and $x_2(t)$, such that the general equation can be written as

$$x(t) = Ae^{i\omega_0 t} + Be^{-i\omega_0 t}.$$

We can also express the above as,

$$x(t) = Ae^{i\omega_0 t} + Be^{-i\omega_0 t} = (A+B)\cos(\omega_0 t) + i(A-B)\sin(\omega_0 t).$$

Furthermore, noting that $x(0) = A + B$, and $\dot{x}(0) = i(A-B)\omega_0$,

$$x(t) = x(0)\cos(\omega_0 t) + \frac{\dot{x}(0)}{\omega_0}\sin(\omega_0 t).$$

Since the signals that we acquire from a sensor are real, we want to deal with real valued signals $x(t)$, and this means it must be that $B = A^*$. Denoting $A = a + ib$, we can once again rewrite the above as,

$$x(t) = 2a\cos(\omega_0 t) - \frac{2b}{\omega_0}\sin(\omega_0 t).$$

Finally, because any linear combination of sines and cosines with the same frequency is also a sine wave with the same frequency, we can rewrite the above as,

$$x(t) = \rho\sin(\omega_0 t + \theta),$$

where

$$\rho = 2\sqrt{a^2 + \frac{b^2}{\omega_0^2}}, \text{ and } \theta = \begin{cases} \arcsin\left(\frac{a}{\sqrt{a^2+\frac{b^2}{\omega_0^2}}}\right) & b \leq 0 \\ \pi - \arcsin\left(\frac{a}{\sqrt{a^2+\frac{b^2}{\omega_0^2}}}\right) & b > 0. \end{cases} \tag{7.3}$$

7.2.2 Multiple-degree-of-freedom system

Similarly to the SDOF system, an N-degree MDOF system[4] can be formulated as

$$[M]\{\ddot{u}(t)\} + [C]\{\dot{u}(t)\} + [K]\{u(t)\} = \{0(t)\},$$

where $[M]$ is an $N \times N$ diagonal mass matrix, $[C]$ is a symmetric $N \times N$ damping matrix, $[K]$ is an $N \times N$ symmetric stiffness matrix, and $\{u(t)\}$ is an $N \times 1$ vector of displacement signals. Note that $\{u(t)\} = \{u_1(t), \ldots, u_N(t)\}$, and each $\{u(t)\}(l) = u_l(t)$, where $l = 1, \ldots, N$, is a displacement

[4]Theoretically, a structure will have infinitely many degrees of freedom. However, the number of mode shapes and their corresponding modal frequencies and modal damping ratios that we are able to detect is equal to the number of sensor nodes placed on structures. In the following, whenever we are dealing with an N-degree MDOF system we are implicitly assuming that we have N sensor nodes deployed on the structure.

signal. One can view $u_l(t)$ as the signal being observed at the lth sensor node. Let us again consider an undamped system and set $[C] = [0]$. This simplifies the above equation to

$$[M]\{\ddot{u}(t)\} + [K]\{u(t)\} = \{0(t)\}.$$

Let us assume $\{u(t)\} = \{\phi\}Ae^{i\omega t}$ to be a solution to the above equation, where $\omega, t \in \mathbb{R}$. Note that $\{\phi\}$ is an $N \times 1$ spatial vector that is independent of time and we define it so that it has unit energy, i.e., $\|\{\phi\}\|_2 = 1$ (we can assume this as the normalization can be absorbed into the scalar variable A). Plugging in the appropriate derivative to the above expression we get,

$$\left(-[M]\omega^2 + [K]\right)\{\phi\}Ae^{i\omega t} = \{0(t)\}.$$

Since the above must hold for all values of t it must be that,

$$\left([K] - \omega^2[M]\right)\{\phi\} = \{0\}.$$

This equation represents a generalized eigenvalue problem and our objective is to find pairs of ω^2 and $\{\phi\}$ that satisfy the above equation. Notice the similarity to the conventional eigenvalue problem which corresponds to the case when $[M] = I$. To solve the above problem, as in the conventional eigenvalue problem, one starts off by computing the generalized eigenvalue ω that satisfies

$$\det\left([K] - \omega^2[M]\right) = 0.$$

Assuming that this does not vanish as a function of ω^2 and M is full rank, the above will represent an Nth order polynomial and there will be N generalized eigenvalues ω^2 as solutions to this polynomial. Each ω^2 when plugged back in to the above equation will have a corresponding generalized eigenvector $\{\phi\}$ satisfying the above equation. Thus, there will be N generalized eigenvalues, i.e., modal frequencies, $\omega_1, \ldots, \omega_N$ and their corresponding generalized eigenvectors, i.e., mode shape vectors, $\{\phi_1\}, \ldots, \{\phi_N\}$. Note that each modal frequency must be real and $\omega_n > 0$. Also, without loss of generality, let us assume that the frequencies are sorted, i.e., $\omega_1 \geq \omega_2 \geq \cdots \geq \omega_N > 0$.

Going back to our assumption about the solution to the MDOF system equation it is clear that each $\{u_n(t)\} = \{\phi_n\}A_n e^{i\omega_n t}$ will be valid solutions, where $n = 1, \ldots, N$. Furthermore, as in the SDOF system $\{\phi_n\}B_n e^{-i\omega_n t}$ will also be valid solutions and thus for each n a complete solution will

be of the form

$$\{u_n(t)\} = \{\phi_n\} \left(A_n e^{i\omega_n t} + B_n e^{-i\omega_n t} \right). \tag{7.4}$$

We want $\{u_n(t)\}$ to be real, which we can get when $\{\phi_n\}$ is real (there exists a real eigenvector that satisfies the above equation given that the mass and stiffness matrices are real and symmetric) and $B_n = A_n^*$. Thus, as with the SDOF case we can rewrite the solution as

$$\{u_n(t)\} = \{\phi_n\}\rho_n \sin(\omega_n t + \theta_n),$$

where ρ_n and θ_n are as defined in (7.3).

Finally, it is easy to see that all linear combinations of $\{u_n(t)\}$ are valid solutions to the MDOF system equation and thus the general solution is of the form

$$\{u(t)\} = \sum_{n=1}^{N} \{\phi_n\}\rho_n \sin(\omega_n t + \theta_n). \tag{7.5}$$

In the structural dynamics community this is also known as the modal superposition equation.

7.3 Problem Formulation

7.3.1 The analytic signal of $\{u(t)\}$

In order to simplify the upcoming analysis, we are going use what is known as the analytic signal of $\{u(t)\}$ [22]. The definition of an analytic signal is given as,

Definition 7.1 (Definition 1.2.1, [22])**.** A signal $v(t)$ is said to be *analytic* iff

$$V(f) = 0 \text{ for } f < 0,$$

where $V(f)$ is the Fourier transform of $v(t)$.

Essentially, an analytic signal can be obtained by removing all the negative frequencies in the original signal. The analytic signal is a frequently used representation in mathematics, signal processing, and communications, prevalently for easing the mathematical manipulations.

To see what the analytic signal of $\{u(t)\}$ is, let us take a step back and look at each entry in

$\{u(t)\}$, i.e., $u_l(t)$. Each $u_l(t)$ can be written as

$$u_l(t) = \sum_{n=1}^{N} \{\phi_n\}(l)\rho_n \sin(\omega_n t + \theta_n) = \sum_{n=1}^{N} \{\phi_n\}(l)(A_n e^{i\omega_n t} + A_n^* e^{-i\omega_n t}).$$

Note that the variable A_n and A_n^* corresponds to what we have seen in equation (7.4). Then, it is easy to see that the analytic signal of $u_l(t)$, represented as $v_l(t)$, is simply

$$v_l(t) = \sum_{n=1}^{N} \{\phi_n\}(l) A_n e^{i\omega_n t},$$

which only contains non-negative frequencies, i.e., $e^{i\omega_n t}$. Then, the analytic signal of the entire vector $\{u(t)\}$, denoted as $\{v(t)\}$, can be written as

$$\{v(t)\} = \sum_{n=1}^{N} \{\phi_n\} A_n e^{i\omega_n t}.$$

Note that $\{v(t)\}$ is no longer real but complex. The practical implementation of obtaining an analytic signal involves the application of a Hilbert transform. However, detailed discussion on this matter is out of scope of this chapter and we will refer interested readers to [84] for more detail. For the remainder of this chapter, we will assume that we have successfully extracted the analytic signal from each $u_l(t)$. Thus, all the derivations from here onwards will be in terms of $\{v(t)\}$.

7.3.2 The relationship to singular value decomposition

We can write the modal superposition equation in a matrix-vector multiplication format such as

$$\{v(t)\} = \sum_{n=1}^{N} \{\phi_n\} A_n e^{i\omega_n t} = \underbrace{[\{\phi_1\}, \{\phi_2\}, \ldots, \{\phi_N\}]}_{[\Phi]} \underbrace{\begin{bmatrix} A_1 & 0 & \ldots & 0 \\ 0 & A_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \ldots & A_N \end{bmatrix}}_{[\Gamma]} \underbrace{\begin{Bmatrix} e^{i\omega_1 t} \\ e^{i\omega_2 t} \\ \vdots \\ e^{i\omega_N t} \end{Bmatrix}}_{\{s(t)\}}$$

$$= [\Phi][\Gamma]\{s(t)\}, \tag{7.6}$$

where $[\Phi]$ denotes the $N \times N$ mode shape vector matrix with orthonormal columns, $[\Gamma]$ is an $N \times N$ diagonal matrix with diagonal entries A_n, and $\{s(t)\}$ is an $N \times 1$ vector known as the modal

coordinate vector. The above equation establishes an explicit relationship of the observed time data $\{v(t)\}$ to the modal parameters such as the mode shape vector and modal frequencies. This equation plays the key role of motivating the use of the SVD as a way to extract the modal parameters.

In order to see how the SVD can be used to extract the modal parameters, let us suppose that we sample each row of $\{v(t)\}$ in time at M distinct points $t_1, \ldots, t_M$. Let us denote the resulting $N \times M$ data matrix as

$$[V] = \begin{bmatrix} v_1(t_1) & v_1(t_2) & \ldots & v_1(t_M) \\ v_2(t_1) & v_2(t_2) & \ldots & v_2(t_M) \\ \vdots & & & \\ v_N(t_1) & v_N(t_2) & \ldots & v_N(t_M) \end{bmatrix} \in \mathbb{R}^{N \times M}.$$

The sampling of $\{v(t)\}$ at $t_1, \ldots, t_M$ implies the sampling of $\{s(t)\}$ at the exact same time points which results in

$$[S] = \begin{bmatrix} e^{i\omega_1 t_1} & e^{i\omega_1 t_2} & \ldots & e^{i\omega_1 t_M} \\ e^{i\omega_2 t_1} & e^{i\omega_2 t_2} & \ldots & e^{i\omega_2 t_M} \\ \vdots & & & \\ e^{i\omega_N t_1} & e^{i\omega_N t_2} & \ldots & e^{i\omega_N t_M} \end{bmatrix} \in \mathbb{R}^{N \times M}.$$

Overall, this means that after sampling $\{v(t)\}$ at $t_1, \ldots, t_M$, equation (7.6) becomes

$$[V] = [\Phi][\Gamma][S]. \tag{7.7}$$

The above equation makes explicit the relationship between SVD and the modal parameters. At this point, we know that $[\Phi]$ is a square matrix with orthonormal columns, and $[\Gamma]$ is a diagonal matrix. If $[S]$ is a matrix with orthogonal (or orthonormal) rows, then equation (7.7) describes nothing but the SVD of $[V]$. In that case, we can obtain the modal parameters by simply computing the SVD of $[V]$.

7.3.3 The implications of sampling

Note that we have not made any assumptions on how we select the sampling times $t_1, \ldots, t_M$. In fact, it is easy to see that the orthogonality of the rows in $[S]$ will depend highly on how we sample

$\{s(t)\}$. A natural example that comes to mind is when the rows of $[S]$ are the Discrete Fourier Transform (DFT) vectors. As we know, any subset of the DFT vectors are perfectly orthogonal to one another. Then, the SVD of $[V]$ would exactly recover the modal parameters. This particular example corresponds to the case when the modal frequencies lie on a grid such that $\omega_n = \frac{2\pi k_n}{MT_s}$, where $k_n \in \{1, \ldots, N\}$ and T_s represents the sampling interval, and we sample at $t_m = T_s(m-1)$, where $m \in \{1, \ldots, M\}$, assuming $N \leq M$. This, however, is an unrealistic model for the purpose of modal analysis specifically because the modal frequencies will typically not lie on a grid as explained above.

If we drop the assumption that the modal frequencies lie on a grid, the problem becomes much more complicated and it is very likely for the rows of $[S]$ to be correlated to each other. However, it seems possible to carefully design a sampling scheme that would make the rows nearly orthogonal. Then, intuitively, we would expect for the SVD of V to give us mode shape vectors that are good approximations to the true ones. All of this will be discussed in the following sections.

7.4 Main Results

In this section, we present our two main results. For each result, we propose to use Algorithm 2 to recover the mode shape vectors. We propose two sampling schemes, uniform and random sampling, and for each sampling scheme we provide a sufficient condition for the accurate recovery of mode shapes via Algorithm 2.

7.4.1 Proposed method

Our proposed method to recover the mode shape vectors is very simple and is described in Algorithm 2.

Algorithm 2: Pseudo-code for mode shape estimation

Input: Data matrix $[V]$
Output: $[\hat{\Phi}]$ (left singular vectors of $[V]$)
SVD(V) = $[\hat{\Phi}][\hat{\Gamma}][\hat{S}]$

As described above, our method simply computes the SVD of $[V]$ and returns the matrix of left singular vectors $[\hat{\Phi}] = [\{\hat{\phi}_1\}, \ldots, \{\hat{\phi}_N\}]$ as estimates of the true mode shape matrix $[\Phi]$. One point to note about the algorithm is that because $[V]$ is $N \times M$, where we assume $M > N$, the dimensions of each $[\hat{\Phi}]$, $[\hat{\Gamma}]$, and $[\hat{S}]$ will be $N \times N$, $N \times M$, and $M \times M$, respectively. This is unlike the dimensions of their supposedly respective counterparts given in equation (7.7). Actually, by noting that only N

diagonal entries in $[\hat{\Gamma}]$ are non-zero we can compute the truncated SVD to obtain a decomposition with the desired dimensions.

7.4.2 Uniform sampling

Equipped with the above algorithm, let us now suppose that we sample each complex exponential $e^{i\omega_n t}$ at uniform sampling interval denoted as T_s. The sampling times will be denoted as $t_m = (m-1)T_s$, $m = 1, \ldots, M$. Note that this means that we are sampling within the time span $[0, t_{\max}]$, where $t_{\max} := (M-1)T_s$. For this uniform sampling scenario, we can establish the following theorem.

Theorem 7.2. *Let $[V] = [\Phi][\Gamma][S]$ be as given in (7.7) describing an N-degree of freedom system sampled according to the uniform sampling scheme described above. Let us define the minimum and maximum spacing of the modal frequencies as $\delta_{\min} := \min_{l \neq n} |\omega_l - \omega_n|$, and $\delta_{\max} := \max_{l \neq n} |\omega_l - \omega_n|$, respectively. Let us sample for a time span of at least*

$$t_{\max} \geq \frac{2\pi(\log(\lfloor N/2 \rfloor + 1.01))}{\epsilon \delta_{\min}},$$

with sampling interval $T_s = \frac{\pi}{\delta_{\max}}$. Or, equivalently, let us take number of measurements satisfying,

$$M \geq \frac{2(\log\lfloor N/2 \rfloor + 1.01)}{\epsilon} \frac{\delta_{\max}}{\delta_{\min}} + 1,$$

taken at the sampling interval $T_s = \frac{\pi}{\delta_{\max}}$. Then, the mode shape estimates $[\hat{\Phi}]$ obtained via Algorithm 2 satisfy the following bound,

$$\|\{\phi_n\} - \{\hat{\phi}_n\}\|_2 \leq \min \left\{ \sqrt{2},\ \frac{\epsilon\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}} \max_{l \neq n} \frac{\sqrt{2}\gamma_l \gamma_n}{\min\limits_{c \in [-1,1]} \{|\gamma_l^2 - \gamma_n^2(1+c\epsilon)|\}} \right\}, \tag{7.8}$$

where $\gamma_n := |A_n|$ are the magnitudes of the diagonal entries of $[\Gamma]$.

Note from equation (7.8) that the error in the estimate will be small when ϵ is small. The variable ϵ essentially captures how close the rows of $[S]$ are to being orthogonal. Furthermore, the bigger the smallest separation between γ_n, the better our estimate. Turning our attention to the sampling parameters, the above theorem essentially tells us that we need to sample for a time span that is inversely proportional to the minimum spacing between the modal frequencies. Thus, the smaller the minimum spacing between the modal frequencies, the longer we have to sample to get an accurate estimate. Also, the maximum spacing between modal frequencies determines how fast we need to sample, i.e., $T_s = \frac{\pi}{\delta_{\max}}$. Comparing this sampling interval to the Nyquist sampling interval

which would be $T_0 = \frac{\pi}{2\max_n \omega_n}$, it is interesting to note that $T_s > T_0$. This suggests that for the purpose of mode shape extraction, we can sample at a rate lower than the Nyquist rate and still accurately recover the mode shapes. Finally, note that the condition on M is fairly satisfactory as it essentially scales with $\frac{\log(N)}{\epsilon}$. The only disturbing factor is the ratio $\frac{\delta_{\max}}{\delta_{\min}}$, which suggests that the required number of measurements is structure dependent.

7.4.3 Random sampling

Let us now consider a simple random sampling scheme, where we sample M time points $t_1, \ldots, t_M$ uniformly at random in the time interval $[0, t_{\max}]$. Then, the following theorem can be established.

Theorem 7.3. *Let $[V] = [\Phi][\Gamma][S]$ be as given in (7.7) describing an N-degree of freedom system sampled according to the uniform sampling scheme described above. Let us define the minimum spacing of the modal frequencies as $\delta_{\min} := \min_{l \neq n} |\omega_l - \omega_n|$. Let us sample for a time span of at least*

$$t_{\max} \geq \frac{4(\log(\lfloor N/2 \rfloor) + 1.01)}{\epsilon^2 \delta_{\min}},$$

and within this time span let us take number of measurements satisfying,

$$M > \frac{\log(N)}{D((1+\epsilon)/N || (1+\epsilon^2)/N)},$$

where $D(a||b) := a(\log(a) - \log(b)) + (1-a)(\log(1-a) - \log(1-b))$ for $a, b \in [0,1]$, is known as the binary information divergence, or the Kullback-Leibler divergence. Then, there exists a constant $c_1 > 0$ such that with probability at least $1 - e^{-c_1 M}$ we will have that the mode shape estimates $[\hat{\Phi}]$ obtained via Algorithm 2 satisfy the following bound,

$$\|\{\phi_n\} - \{\hat{\phi}_n\}\|_2 \leq \min \left\{ \sqrt{2},\ \frac{\epsilon\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}} \max_{l \neq n} \frac{\sqrt{2}\gamma_l \gamma_n}{\min\limits_{c \in [-1,1]} \{|\gamma_l^2 - \gamma_n^2(1+c\epsilon)|\}} \right\}, \tag{7.9}$$

where $\gamma_n := |A_n|$ are the magnitudes of the diagonal entries of $[\Gamma]$.

The above theorem looks very similar to that of the uniform sampling case. The recovery guarantee and the required time span is essentially the same. The only major difference is the requisite number of measurements, which no longer has the dependency on $\frac{\delta_{\max}}{\delta_{\min}}$ and scales slightly worse than $\frac{\log(N)}{\epsilon}$.

7.5 Proof of Main Results

In this section we provide the proofs of the main results. To do so, we take a perturbation theoretic viewpoint. We first describe how we can formulate our problem as a perturbation problem and then provide separate proofs for each theorem.

7.5.1 Perturbation analysis

We start with the equation,

$$[V] = [\Phi][\Gamma][S]$$

where we assume that we sampled at some arbitrary times $t_1, \ldots, t_M$.

To carry out perturbation analysis let us note that

$$[V][V]^* = [\Phi][\Gamma][S][S]^*[\Gamma]^*[\Phi]^*,$$

where

$$[S][S]^* = \begin{bmatrix} 1 & \frac{1}{M}\sum_{m=1}^{M} e^{j(\omega_1-\omega_2)t_m} & \ldots & \frac{1}{M}\sum_{m=1}^{M} e^{j(\omega_1-\omega_N)t_m} \\ \frac{1}{M}\sum_{m=1}^{M} e^{j(\omega_2-\omega_1)t_m} & 1 & \ldots & \frac{1}{M}\sum_{m=1}^{M} e^{j(\omega_2-\omega_N)t_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{M}\sum_{m=1}^{M} e^{j(\omega_N-\omega_1)t_m} & \frac{1}{M}\sum_{m=1}^{M} e^{j(\omega_N-\omega_2)t_m} & \ldots & 1 \end{bmatrix}.$$

Thus, we can decompose this product as, $[S][S]^* = [I] + [\Delta]$, where $[\Delta]$ contains the off-diagonal entries of $[S][S]^*$. Then,

$$[V][V]^* = \underbrace{[\Phi][\Gamma][\Gamma]^*[\Phi]^*}_{H} + \underbrace{[\Phi][\Gamma][\Delta][\Gamma]^*[\Phi]^*}_{\delta H}.$$

The above expression allows us to view $[V][V]^*$ as the summation of the original matrix H and the perturbation matrix δH. To quantify how much the mode shapes $[\Phi]$ change due to δH, we are going to use Theorem 6.2 presented in Chapter 6. In order to use this theorem all we need to do is

to quantify $\eta = \|[H]^{-\frac{1}{2}}[\delta H][H]^{-\frac{1}{2}}\|_2$, where $[H]^{-\frac{1}{2}} = [\Phi]\left([\Gamma][\Gamma]^H\right)^{-\frac{1}{2}}[\Phi]^*$. But

$$\begin{aligned}
\eta &= \|[H]^{-\frac{1}{2}}[\delta H][H]^{-\frac{1}{2}}\|_2 = \|[\Phi]\left([\Gamma][\Gamma]^*\right)^{-\frac{1}{2}}[\Phi]^*[\Phi][\Gamma][\Delta][\Gamma]^*[\Phi]^*[\Phi]\left([\Gamma][\Gamma]^*\right)^{-\frac{1}{2}}[\Phi]^*\|_2, \\
&= \|\left([\Gamma][\Gamma]^*\right)^{-\frac{1}{2}}[\Gamma][\Delta][\Gamma]^*\left([\Gamma][\Gamma]^*\right)^{-\frac{1}{2}}\|_2, \\
&= \max_{x\neq 0} \frac{x^H\left([\Gamma][\Gamma]^*\right)^{-\frac{1}{2}}[\Gamma][\Delta]^*[\Gamma]^*\left([\Gamma][\Gamma]^*\right)^{-\frac{1}{2}}\left([\Gamma][\Gamma]^*\right)^{-\frac{1}{2}}[\Gamma][\Delta][\Gamma]^*\left([\Gamma][\Gamma]^*\right)^{-\frac{1}{2}}x}{x^Hx}, \\
&= \max_{x\neq 0} \frac{x^H\left([\Gamma][\Gamma]^*\right)^{-\frac{1}{2}}[\Gamma][\Delta]^*[\Delta][\Gamma]^*\left([\Gamma][\Gamma]^*\right)^{-\frac{1}{2}}x}{x^Hx}, \\
&= \max_{y\neq 0} \frac{y^H[\Delta]^*[\Delta]y}{y^Hy}, \qquad \text{let } y = [\Gamma]^*\left([\Gamma][\Gamma]^*\right)^{-\frac{1}{2}}x \text{ and note } \; y^*y = x^*x \\
&= \|[\Delta]\|_2,
\end{aligned}$$

which in turn tells us that we need to quantify $\|[\Delta]\|_2$. Since the exact expression of $\|[\Delta]\|_2$ is difficult to obtain we are going to compute an upper bound of $\|[\Delta]\|_2$ instead. To bound $\|[\Delta]\|_2$ let us take a step back and note that $\lambda_n([S][S]^*) = \lambda_n([I]+[\Delta]) = 1+\lambda_n([\Delta])$, where use $\lambda_n([A])$ to denote the nth largest eigenvalue of $[A]$. This can be seen by looking at the eigendecomposition $[S][S]^* = [U_S][\Lambda_S][U_S]^* = [I]+[\Delta]$ and writing $[\Lambda_S] = [I]+[U_S]^*[\Delta][U_S]$. Note that $[U_S]^*[\Delta][U_S]$ must be a diagonal matrix, which corresponds to the eigenvalues of $[\Delta]$.

Once we can find an upper and lower bound on the eigenvalues of $[S][S]^*$ such that $\lambda_d \leq \lambda_n([S][S]^*) \leq \lambda_u$ holds for all n, then we can easily see that $\lambda_d - 1 \leq \lambda_n([\Delta]) \leq \lambda_u - 1$, which in turn implies that $\|[\Delta]\|_2 = \max_n|\lambda_n([\Delta])| \leq \max\{|\lambda_u - 1|, |\lambda_d - 1|\}$. Therefore, our strategy is to bound $\lambda_n([S][S]^*)$ from below and above in order to bound $\|[\Delta]\|_2$. In the following sections we establish this result for both the random and uniform sampling cases. Our main results follow directly from the respective results on $\lambda_n([S][S]^*)$ and the above arguments.

7.5.2 Proof for random sampling

Let us first consider how to establish a bound on $\|[\Delta]\|_2$ if we were to sample $t_1, \ldots, t_M$ at uniformly random in the time interval $[0, t_{\max}]$. We can establish the following bound on the eigenvalues of $[S][S]^*$.

Theorem 7.4. *Let $[S][S]^*$ be the matrix as described above. Given that we sample*

$$M > \frac{\log(N)}{D((1+\epsilon)/N||(1+\epsilon^2)/N)},$$

and choose

$$t_{\max} \geq \frac{4}{\epsilon^2 \delta_{\min}}(\log(\lfloor N/2 \rfloor) + 1.01),$$

where $\delta_{\min} := \min_{l \neq n} |\omega_l - \omega_n|$*, there exists a constant* $c_1 > 0$ *such that with probability at least* $1 - e^{-c_1 M}$ *we will have that,*

$$1 - \epsilon < \lambda_n \left([S][S]^* \right) < 1 + \epsilon, \quad \forall n.$$

Proof. To bound $\|[S][S]^*\|_2$ we are going to use a slightly modified theorem that appeared in [128]. Let us first state the original theorem.

Theorem 7.5. *[[128], Theorem 5.1] Consider a sequence* $\{Z_m : m = 1, \ldots, M\}$ *of independent, d-dimensional, random, self-adjoint matrices that satisfy*

$$Z_m \succeq 0 \text{ and } \lambda_{\max}(Z_m) \leq 1 \text{ almost surely.}$$

Compute the minimum and maximum eigenvalues of the average expectation,

$$\bar{\mu}_{\min} := \lambda_{\min} \left(\frac{1}{M} \sum_{m=1}^{M} \mathbf{E} Z_m \right) \qquad \bar{\mu}_{\max} := \lambda_{\max} \left(\frac{1}{M} \sum_{m=1}^{M} \mathbf{E} Z_m \right).$$

Then

$$P \left\{ \lambda_{\min} \left(\frac{1}{M} \sum_{m=1}^{M} Z_m \right) \leq \alpha \right\} \leq d e^{-M D(\alpha || \bar{\mu}_{\min})} \text{ for } 0 \leq \alpha \leq \bar{\mu}_{\min}$$

$$P \left\{ \lambda_{\max} \left(\frac{1}{M} \sum_{m=1}^{M} Z_m \right) \geq \alpha \right\} \leq d e^{-M D(\alpha || \bar{\mu}_{\max})} \text{ for } \bar{\mu}_{\max} \leq \alpha \leq 1.$$

The binary information divergence $D(a||b) := a(\log(a) - \log(b)) + (1-a)(\log(1-a) - \log(1-b))$ *for* $a, b \in [0, 1]$.

As the exact expression of $\bar{\mu}_{\max}$ and $\bar{\mu}_{\min}$ is difficult to compute, we are going to need the following corollary that only requires an upper bound on $\bar{\mu}_{\max}$ and a lower bound on $\bar{\mu}_{\min}$.

Corollary 7.6. *Consider a sequence* $\{\hat{Z}_m : m = 1, \ldots, M\}$ *of independent, d-dimensional, random, self-adjoint matrices that satisfy*

$$\hat{Z}_m \succeq 0 \text{ and } \lambda_{\max}(\hat{Z}_m) \leq c \text{ almost surely.}$$

Then for any $\tilde{\mu}_{\min}$ *and* $\tilde{\mu}_{\max}$ *such that,*

$$\tilde{\mu}_{\min} \leq \lambda_{\min}\left(\frac{1}{M}\sum_{m=1}^{M}\mathbf{E}\hat{Z}_m\right), \text{ and } \tilde{\mu}_{\max} \geq \lambda_{\max}\left(\frac{1}{M}\sum_{m=1}^{M}\mathbf{E}\hat{Z}_m\right),$$

we have

$$P\left\{\lambda_{\min}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{Z}_m\right) \leq \alpha\right\} \leq de^{-MD(\alpha/c||\tilde{\mu}_{\min}/c)} \text{ for } 0 \leq \alpha \leq \tilde{\mu}_{\min}, \tag{7.10}$$

$$P\left\{\lambda_{\max}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{Z}_m\right) \geq \alpha\right\} \leq de^{-MD(\alpha/c||\tilde{\mu}_{\max}/c)} \text{ for } \tilde{\mu}_{\max} \leq \alpha \leq c. \tag{7.11}$$

The binary information divergence $D(a||b) := a(\log(a) - \log(b)) + (1-a)(\log(1-a) - \log(1-b))$ *for* $a, b \in [0,1]$.

Proof. Let us denote $\hat{Z}_m = cZ_m$, where Z_m is such that $Z_m \succeq 0$ and $\lambda_{\max}(Z_m) \leq 1$. This simply implies $\hat{Z}_m \succeq 0$ and $\lambda_{\max}(\hat{Z}_m) \leq c$, and that $\hat{\mu}_{\max} := \lambda_{\max}(\frac{1}{M}\sum_{m=1}^{M}\mathbf{E}\hat{Z}_m) = c\lambda_{\max}(\frac{1}{M}\sum_{m=1}^{M}\mathbf{E}Z_m) = c\bar{\mu}_{\max}$. To derive the probability for $\hat{Z}_m$, let us look at the following chain of inequalities,

$$\begin{aligned}
P\left\{\lambda_{\max}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{Z}_m\right) \geq \alpha\right\} &= P\left\{c\lambda_{\max}\left(\frac{1}{M}\sum_{m=1}^{M}Z_m\right) \geq \alpha\right\}, \\
&= P\left\{\lambda_{\max}\left(\frac{1}{M}\sum_{m=1}^{M}Z_m\right) \geq \alpha/c\right\}, \\
&\leq de^{-MD(\alpha/c||\bar{\mu}_{\max})} = de^{-MD(\alpha/c||\hat{\mu}_{\max}/c)}.
\end{aligned}$$

The range of α that admits the above probability is $\bar{\mu}_{\max} \leq \alpha/c \leq 1$, which is equivalent to $\hat{\mu}_{\max} \leq \alpha \leq c$. Similarly, we have $\hat{\mu}_{\min} := \lambda_{\min}(\frac{1}{M}\sum_{m=1}^{M}\mathbf{E}\hat{Z}_m) = c\lambda_{\min}(\frac{1}{M}\sum_{m=1}^{M}\mathbf{E}Z_m) = c\bar{\mu}_{\min}$ for the minimum eigenvalue. Then,

$$\begin{aligned}
P\left\{\lambda_{\min}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{Z}_m\right) \leq \alpha\right\} &= P\left\{c\lambda_{\min}\left(\frac{1}{M}\sum_{m=1}^{M}Z_m\right) \leq \alpha\right\}, \\
&= P\left\{\lambda_{\min}\left(\frac{1}{M}\sum_{m=1}^{M}Z_m\right) \leq \alpha/c\right\}, \\
&\leq de^{-MD(\alpha/c||\bar{\mu}_{\min})} = de^{-MD(\alpha/c||\hat{\mu}_{\min}/c)}.
\end{aligned}$$

Again, the above is valid for $0 \leq \alpha/c \leq \bar{\mu}_{\min}$ and equivalently $0 \leq \alpha \leq \hat{\mu}_{\min}$.

Next we need to show that the above probability is also valid for an upper bound $\tilde{\mu}_{\max} \geq \hat{\mu}_{\max}$ and a lower bound $\tilde{\mu}_{\min} \leq \hat{\mu}_{\max}$. We show this for the maximum eigenvalue by noting the fact that the information divergence (also known as the Kullback-Leibler divergence) $D(a||b)$ is a decreasing function with respect to the second argument b when $a \geq b$. Similarly, for the minimum eigenvalue we use the fact that $D(a||b)$ is an increasing function with respect to b when $a \leq b$.

Let us first look at the derivative of $D(a||b)$ with respect to b.

$$\frac{dD(a||b)}{db} = -\frac{a}{b} + \frac{1-a}{1-b} = \frac{-a(1-b)+b(1-a)}{b(1-b)} = \frac{b-a}{b(1-b)}.$$

Note that the above derivative will be negative when $a \geq b$ and positive when $b \geq a$, which is what we wanted to show. This essentially means that for any $b \leq b^{'} \leq 1$ as long as $a \geq b$, we will have $D(a||b) \geq D(a||b^{'})$. Similarly, for any $0 \leq b^{'} \leq b$, as long as $a \leq b$ we will have $D(a||b) \geq D(a||b^{'})$. This completes the proof. □

In order to apply Corollary 7.6, let us write

$$[S][S]^* = \frac{1}{M}\sum_{m=1}^{M} \begin{Bmatrix} e^{i\omega_1 t_m} \\ \vdots \\ e^{i\omega_N t_m} \end{Bmatrix} \left\{e^{-i\omega_1 t_m}, \ldots, e^{-i\omega_N t_m}\right\}$$

Let us define the vector $\{S_m\} = \{e^{-i\omega_1 t_m}, \ldots, e^{-i\omega_N t_m}\}^*$, where $\|\{S_m\}\|_2^2 = N$. Let us set the matrix Z_m that appears in the above theorem as $Z_m = \{S_m\}\{S_m\}^*$. As a result, Z_m will be i.i.d. positive semi-definite matrix, i.e., $Z_m \succeq 0$, of rank 1 with $\lambda_{\max}(Z_m) = \|\{S_m\}\|_2^2 = N$. We wish to compute

$$\lambda_{\min}\left(\frac{1}{M}\sum_{m=1}^{M}\mathbf{E}\{S_m\}\{S_m\}^*\right) = \lambda_{\min}\left(\mathbf{E}\{S_m\}\{S_m\}^*\right),$$

$$\lambda_{\max}\left(\frac{1}{M}\sum_{m=1}^{M}\mathbf{E}\{S_m\}\{S_m\}^*\right) = \lambda_{\max}\left(\mathbf{E}\{S_m\}\{S_m\}^*\right),$$

or the appropriate lower and upper bound on the above quantities. Note that

$$\mathbf{E}\{S_m\}\{S_m\}^* = \mathbf{E}\begin{bmatrix} 1 & e^{i(\omega_1-\omega_2)t_m} & \dots & e^{i(\omega_1-\omega_N)t_m} \\ e^{-i(\omega_1-\omega_2)t_m} & 1 & \dots & e^{i(\omega_2-\omega_N)t_m} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-i(\omega_1-\omega_N)t_m} & e^{-i(\omega_2-\omega_N)t_m} & \dots & 1 \end{bmatrix}$$

$$= \begin{cases} 1 & \text{when } l = n \\ e^{i(\omega_l-\omega_n)\frac{t_{\max}}{2}}\operatorname{sinc}((\omega_l-\omega_n)\frac{t_{\max}}{2}) & \text{when } l \neq n. \end{cases}$$

The eigenvalues can also be written as $\lambda_n\left(\mathbf{E}\{S_m\}\{S_m\}^*\right) = \lambda_n\left(I + [\Delta_S]\right) = 1 + \lambda_n([\Delta_S])$, where $[\Delta_S]$ is the off-diagonal matrix of $\mathbf{E}\{S_m\}\{S_m\}^*$. According to Gershgorin's circle theorem [68] we know that every eigenvalue of $\mathbf{E}\{S_m\}\{S_m\}^*$ must lie within at least one of the Gershgorin's disk. As $[\Delta_S]$ has zero diagonal entries, every Gershgorin's disk must be centered at zero. Essentially, all we need to do is to find the Gershgorin's disk that has the largest radius. Then, it will follow that the largest radius corresponds to the bounds on the minimum and maximum eigenvalues.

The lth row of $[\Delta_S]$ will be associated with the following disk,

$$|\lambda([\Delta_S])| \leq \sum_{n=1,n\neq l}^{N} |\operatorname{sinc}((\omega_l-\omega_n)\frac{t_{\max}}{2})| \leq \sum_{n=1,n\neq l}^{N} |\frac{2}{(\omega_l-\omega_n)t_{\max}}|.$$

At this point, we do not know which row will correspond to the largest radius as we do not know all the separation of the modal frequencies $\omega_l - \omega_n$. We tackle this problem by considering the worst case upper bound on each disk radii. Then, the radius of the disk with the largest upper bound will correspond to the lower and upper bound on the smallest and largest eigenvalues.

To do so, let us assume that there is some minimum spacing between the modal frequencies that we denote as $\delta_{\min} := \min_{l,n\in\{1,\dots,N\}} |\omega_l - \omega_n| > 0$. The worst case bound on each of the radii will be when all the modal frequencies are separated equally by the minimum spacing $\delta_{\min}$. To show how we can use this fact, let us focus on the first row such that $l = 1$. In this case we find that,

$$|\lambda([\Delta_S])| \leq \sum_{n=2}^{N} |\operatorname{sinc}((\omega_1-\omega_n)\frac{t_{\max}}{2})| \leq \sum_{n=2}^{N} \frac{2}{|\omega_1-\omega_n|t_{\max}} \leq \frac{2}{t_{\max}}\sum_{n=1}^{N-1}\frac{1}{\delta_{\min}n}$$
$$= \frac{2}{\delta_{\min}t_{\max}}\sum_{n=1}^{N-1}\frac{1}{n}.$$

It is easy to see that the largest upper bound on the radius will be due to the middle row of $[\Delta_S]$. Let us denote the index of the middle row as $l^{'}$ (when N is even we can take either of $N/2$ or $N/2+1$ as the middle row) then,

$$|\lambda([\Delta_S])| \leq \sum_{n=1,n\neq l'}^{N} |\operatorname{sinc}((\omega_{l'}-\omega_n)\frac{t_{\max}}{2})| \leq \sum_{n=1,n\neq l'}^{N} |\frac{2}{(\omega_{l'}-\omega_n)t_{\max}}|$$
$$\leq \frac{2}{\delta_{\min}t_{\max}} \sum_{n=1}^{\lfloor N/2 \rfloor} \frac{2}{n}.$$

In summary, all the eigenvalues of $[\Delta_S]$ will lie within the following disk,

$$|\lambda([\Delta_S])| \leq \frac{4}{\delta_{\min}t_{\max}} \sum_{n=1}^{\lfloor N/2 \rfloor} \frac{1}{n}.$$

The summation term in the above bound is also known as the Harmonic number. We can simplify the above expression a little by using the following upper bound on the Harmonic number.

Theorem 7.7 ([133], Theorem 1)**.** *For any natural number $n \geq 1$, the following inequality is valid:*

$$\frac{1}{2N+\frac{1}{1-\gamma}-2} \leq \sum_{n=1}^{N} \frac{1}{n} - \log(N) - \gamma < \frac{1}{2N+\frac{1}{3}}.$$

The constant $\gamma = 0.57721\cdots$ is known as Euler's constant. The constants $\frac{1}{1-\gamma}-2 = 0.3652\cdots$ and $\frac{1}{3}$ are the best possible, and equality holds only for $N = 1$.

Using this theorem we have

$$|\lambda([\Delta_S])| \leq \frac{4}{\delta_{\min}t_{\max}} \sum_{n=1}^{\lfloor N/2 \rfloor} \frac{1}{n} < \frac{4(\log(\lfloor N/2 \rfloor)+\gamma+\frac{1}{2\lfloor N/2 \rfloor+\frac{1}{3}})}{\delta_{\min}t_{\max}} < \frac{4(\log(\lfloor N/2 \rfloor)+\gamma+3/7)}{\delta_{\min}t_{\max}}$$
$$< \frac{4(\log(\lfloor N/2 \rfloor)+1.01)}{\delta_{\min}t_{\max}}.$$

Collecting everything together, we will have for all n,

$$1 - \frac{4(\log(\lfloor N/2 \rfloor)+1.01)}{\delta_{\min}t_{\max}} < \lambda_n\left(\mathbf{E}\{S_m\}\{S_m\}^*\right) = 1 + \lambda_n([\Delta_S]) < 1 + \frac{4(\log(\lfloor N/2 \rfloor)+1.01)}{\delta_{\min}t_{\max}}$$

For the above bound to be useful we require that the right hand side to be less than or equal to n. In fact, we need much more than this. Let us choose

$$\delta_{\min}t_{\max} \geq \frac{4}{\epsilon^2}(\log(\lfloor N/2 \rfloor)+1.01),$$

where $0 < \epsilon < 1$. Then, we have that

$$\tilde{\mu}_{\min} := 1 - \epsilon^2 < \lambda_n(\mathbf{E}\{S_m\}\{S_m\}^*) < 1 + \epsilon^2 =: \tilde{\mu}_{\max}.$$

Note that $\tilde{\mu}_{\max} \leq 1 + \epsilon$ and $\tilde{\mu}_{\min} \geq 1 - \epsilon$. Then, according to the above theorem, inequality (7.11) will hold for any $1+\epsilon^2 \leq \alpha \leq N$, which will always include $\alpha = 1+\epsilon$. Similarly, inequality (7.10) will hold for any $0 \leq \alpha \leq 1-\epsilon^2$, which will always include $\alpha = 1-\epsilon$. Substituting the appropriate values of $\alpha = 1 \pm \epsilon$, $\tilde{\mu}_{\max}$, and $\tilde{\mu}_{\min}$ into Corollary 7.6, with probability at least $1 - Ne^{-MD((1+\epsilon)/N||(1+\epsilon^2)/N)}$ we will have

$$1 - \epsilon < \lambda_{\min}([S][S]^*) \leq \lambda_n([S][S]^*) \leq \lambda_{\max}([S][S]^*) < 1 + \epsilon.$$

All that remains is to show that there exists a constant $c_1 > 0$ such that

$$Ne^{-MD((1+\epsilon)/N||(1+\epsilon^2)/N)} \leq e^{-c_1 M}.$$

It is easy to see that this can be accomplished by choosing

$$M > \frac{\log(N)}{D((1+\epsilon)/N||(1+\epsilon^2)/N)}.$$

This conditions means, $D((1+\epsilon)/N||(1+\epsilon^2)/N) - \frac{\log(N)}{M} > 0$, and if we choose $c_1 > 0$ sufficiently small we can have that $D((1+\epsilon)/N||(1+\epsilon^2)/N) - \frac{\log(N)}{M} > c_1 > 0$. With simple manipulation we can see that this will guarantee $Ne^{-MD((1+\epsilon)/N||(1+\epsilon^2)/N)} < e^{-c_1 M}$. □

7.5.3 Proof for uniform sampling

For the uniform sampling scenario, we can establish the following theorem on the eigenvalues of $[S][S]^*$.

Theorem 7.8. *Let $[S][S]^*$ be the matrix as described above. Let us define the minimum and maximum spacing of the modal frequencies as $\delta_{\min} := \min_{l,n} |\omega_l - \omega_n|$, and $\delta_{\max} := \max_{l,n} |\omega_l - \omega_n|$, respectively. Let us sample for a time span of at least*

$$t_{\max} \geq \frac{2\pi(\log(\lfloor N/2 \rfloor + 1.01))}{\epsilon \delta_{\min}}$$

with sampling interval $T_s = \frac{\pi}{\delta_{\max}}$. *Or, in other words, let us take at least*

$$M \geq \frac{2(\log\lfloor N/2 \rfloor + 1.01)}{\epsilon} \frac{\delta_{\max}}{\delta_{\min}} + 1$$

measurements with sampling interval $T_s = \frac{\pi}{\delta_{\max}}$. *This will ensure the following bound on the eigenvalues of* $[S][S]^*$*:*

$$1 - \epsilon \leq \lambda_n([S][S]^*) \leq 1 + \epsilon.$$

Proof. Let us recall that the off-diagonal matrix of $[S][S]^*$, denoted as $[\Delta]$, has the following entries,

$$[\Delta] = \begin{pmatrix} 0 & \frac{1}{M}\sum_{m=0}^{M-1} e^{j(\omega_1-\omega_2)mT_s} & \dots & \frac{1}{M}\sum_{m=0}^{M-1} e^{j(\omega_1-\omega_N)mT_s} \\ \frac{1}{M}\sum_{m=0}^{M-1} e^{-j(\omega_1-\omega_2)mT_s} & 0 & \dots & \frac{1}{M}\sum_{m=0}^{M-1} e^{j(\omega_2-\omega_N)mT_s} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{M}\sum_{m=0}^{M-1} e^{-j(\omega_1-\omega_N)mT_s} & \frac{1}{M}\sum_{m=0}^{M-1} e^{-j(\omega_2-\omega_N)mT_s} & \dots & 0 \end{pmatrix}$$

We can evaluate each off-diagonal entry $[\Delta]_{l,n}$ as

$$\begin{aligned}[\Delta]_{l,n} &= \frac{1}{M}\sum_{m=0}^{M-1} e^{j(\omega_p-\omega_q)mT_s} = \frac{1}{M}\frac{1-e^{j(\omega_l-\omega_n)MT_s}}{1-e^{j(\omega_l-\omega_n)T_s}} \\ &= e^{j(\omega_l-\omega_n)T_s(M-1)/2}\frac{\sin((\omega_l-\omega_n)\frac{MT_s}{2})}{M\sin((\omega_l-\omega_n)\frac{T_s}{2})} = e^{j(\omega_l-\omega_n)T_s(M-1)/2}\frac{\sin(|\omega_l-\omega_n|\frac{MT_s}{2})}{M\sin(|\omega_l-\omega_n|\frac{T_s}{2})}.\end{aligned}$$

The above expression looks quite similar to a sinc function. In fact the real part of the above is known as the periodic sinc function or the Dirichlet function and is defined as

$$\text{psinc}(x) = \frac{\sin(M\frac{x}{2})}{M\sin(\frac{x}{2})}.$$

More specifically,

$$\text{psinc}(x) = \begin{cases} \frac{\sin(M\frac{x}{2})}{M\sin(\frac{x}{2})}, \text{when } x \neq 2\pi k,\ k = 0, \pm 1, \pm 2, \dots \\ (-1)^{k(M-1)}, \text{when } x = 2\pi k,\ k = 0, \pm 1, \pm 2, \dots . \end{cases}$$

As its name implies, the psinc function is a periodic function where the period is equal to 2π when M is odd, and 4π when M is even. Every time x is equal to an integer multiple of 2π, the psinc

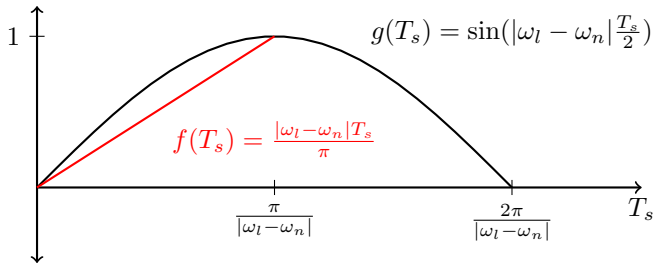


Figure 7.1: A plot of $g(T_s) = \sin(|\omega_l - \omega_n|\frac{T_s}{2})$ in black and $f(T_s) = \frac{|\omega_l-\omega_n|T_s}{\pi}$ in red. This plot illustrates that $g(T_s) \geq f(T_s)$ when $T_s \leq \frac{\pi}{|\omega_l-\omega_n|}$.

function will reach its maximum value.

Again, we are going to bound the eigenvalues of $[\Delta]$ using the Gershgorin's disk theorem. The theorem says that every eigenvalue of $[\Delta]$ will lie in at least one Gershgorin's disk. This disk is of the form,

$$|\lambda([\Delta])| \leq \sum_{n=1}^{N} |[\Delta]_{l,n}| = \sum_{n=1}^{N} \left| \frac{\sin(|\omega_l - \omega_n|\frac{MT_s}{2})}{M\sin(|\omega_l - \omega_n|\frac{T_s}{2})} \right|.$$

We need to find the disk with the maximum radius and thus it is our next objective to compute an upper bound on the radius of all disks.

As the psinc function fluctuates between low and high values depending on the combination of $\omega_l - \omega_n$, T_s, and M, we have to restrict ourselves to only certain values of $\omega_l - \omega_n$, T_s, and M, to make sure that we can guarantee the evaluation of psinc to be small. Note that all ω_n for $n = 1, \ldots, N$ are arbitrary and unknown but fixed and therefore cannot be controlled. Thus we have to find a condition on T_s and/or M.

Let us first look at each $[\Delta]_{l,n}$. One way to find an upper bound of $[\Delta]_{l,n}$ is to note that the denominator can be lower bounded by a linear function for a certain range of T_s. More specifically, note that for $T_s \leq \frac{\pi}{|\omega_l-\omega_n|}$ we have that,

$$\sin\left(|\omega_l - \omega_n|\frac{T_s}{2}\right) \geq \frac{|\omega_l - \omega_n|T_s}{\pi}.$$

This is illustrated in Figure 7.1. Applying this lower bound to each $|[\Delta]_{l,n}|$ we get,

$$|[\Delta]_{l,n}| = \left| \frac{\sin(|\omega_l - \omega_n|\frac{MT_s}{2})}{M\sin(|\omega_l - \omega_n|\frac{T_s}{2})} \right| \leq \left| \frac{\pi\sin(|\omega_l - \omega_n|\frac{MT_s}{2})}{|\omega_l - \omega_n|MT_s} \right| = \frac{\pi}{2}\left|\text{sinc}(|\omega_l - \omega_n|\frac{MT_s}{2})\right|,$$

when $T_s \leq \frac{\pi}{|\omega_l-\omega_n|}$. We can see that the off-diagonal entries behave essentially like a sinc function when $T_s \leq \frac{\pi}{|\omega_l-\omega_n|}$. To further upper bound the above we simply note that $\text{sinc}(x) \leq 1/|x|$, and

apply this to get

$$|[\Delta]_{p,q}| \leq \frac{\pi}{|\omega_p - \omega_q| M T_s}, \quad \text{when } T_s \leq \frac{\pi}{|\omega_p - \omega_q|}.$$

As we need to make sure that the above holds for all pairs of p and q we need to choose a sampling interval such that

$$T_s \leq \frac{\pi}{\delta_{\max}}. \tag{7.12}$$

Overall, given that we sample with T_s satisfying inequality (7.12), we will have that

$$|\lambda([\Delta])| \leq \sum_{n=1}^{N} \frac{\pi}{|\omega_l - \omega_n| M T_s}.$$

Following the same arguments as we have seen for random sampling this will give us

$$|\lambda([\Delta])| \leq \frac{\pi}{\delta_{\min} M T_s} \sum_{n=1}^{\lfloor N/2 \rfloor} \frac{2}{n} < \frac{2\pi(\log(\lfloor N/2 \rfloor) + 1.01)}{\delta_{\min} M T_s} < \frac{2\pi(\log(\lfloor N/2 \rfloor) + 1.01)}{\delta_{\min}(M-1) T_s}.$$

Now, note that $(M-1)T_s = t_{\max}$ and if we choose $t_{\max}$ so that

$$t_{\max} \geq \frac{2\pi(\log(\lfloor N/2 \rfloor) + 1.01)}{\delta_{\min} \epsilon},$$

we will have that

$$|\lambda([\Delta])| < \epsilon.$$

In summary, when we sample at sampling interval satisfying $T_s \leq \frac{\pi}{\delta_{\max}}$ and a time span satisfying $t_{\max} \geq \frac{2\pi(\log(\lfloor N/2 \rfloor)+1.01)}{\delta_{\min}\epsilon}$, we will have

$$1 - \epsilon < \lambda_n([S][S]^*) = 1 + \lambda_n([\Delta]) < 1 + \epsilon.$$

Or, in other words, if we set $T_s = \frac{\pi}{\delta_{\max}}$, and remembering that $t_{\max} = (M-1)T_s$, this means that we need to sample at least

$$M \geq \frac{2(\log(\lfloor N/2 \rfloor) + 1.01)}{\epsilon} \frac{\delta_{\max}}{\delta_{\min}} + 1,$$

to achieve the above eigenvalues guarantee. Note that this is the smallest number of measurements we need since we set T_s as large as possible. If we were to reduce the sampling interval we would need to take more measurements to cover the same time span $t_{\max}$.

□

7.6 Experimental Results

In order to see how the proposed method performs, we present simulation results with real data collected from a bridge in Ypsilanti, MI[5]. On this bridge, there are $N = 18$ wireless nodes each of which are equipped with an accelerometer. Each sensor measures acceleration data and transmits them to the central node for analysis.

We compare the proposed method to the method presented in [98, 148]. In the following we term this method "CS+FDD". This method reconstructs each signal $\{u_l\}$ from random Gaussian measurements, $\{y_l\} = [\Phi]\{u_l\}$, one-by-one and feeds them in to the Frequency Domain Decomposition (FDD) method presented in [24] for modal analysis. The reconstruction solves the following problem,

$$\min_{\{\alpha_l\}} \|\{\alpha_l\}\|_1 \text{ s.t. } \{y_l\} = [\Phi][\Psi]\{\alpha_l\}, \tag{7.13}$$

where $\{u_l\} = [\Psi]\{\alpha_l\}$ and $[\Psi]$ represents the discrete Wavelet transform (our simulations were carried out with DWT, but similar results were obtained with DFT). To solve this problem, we use the SPGL1 solver [131, 130].

For both the proposed method and the CS+FDD method, we took $M = 50$ measurements for each $\{u_l\}$ using the same measurement matrix $[\Phi]$. Since we do not know of the true mode shapes of our dataset we benchmark the mode shapes returned via FDD. There are three dominant mode shapes identified via FDD in this data set.

To assess the closeness between two mode shapes we compute the correlation (COR) between two vectors $\{a\}$ and $\{b\}$,

$$COR(\{a\},\{b\}) = \frac{|\langle\{a\},\{b\}\rangle|}{\|\{a\}\|_2\|\{b\}\|_2} \in [0,1]. \tag{7.14}$$

This function will equal to 1 when $\{a\}$ and $\{b\}$ are equal and 0 when they are orthogonal. The results are presented in Figure 7.2. As we can see, the mode shapes of CS+FDD, plotted in black, are not highly correlated with the mode shapes of FDD, plotted in blue. This is because there were not enough measurements to accurately reconstruct the original signal $\{u_l\} \in \mathbb{R}^L$, which in turn

[5]We would like to thank Sean O'Connor and Prof. Jerome P. Lynch at the University of Michigan for helpful discussions on modal analysis and providing us with real datasets to carry out the experiments presented in this chapter.

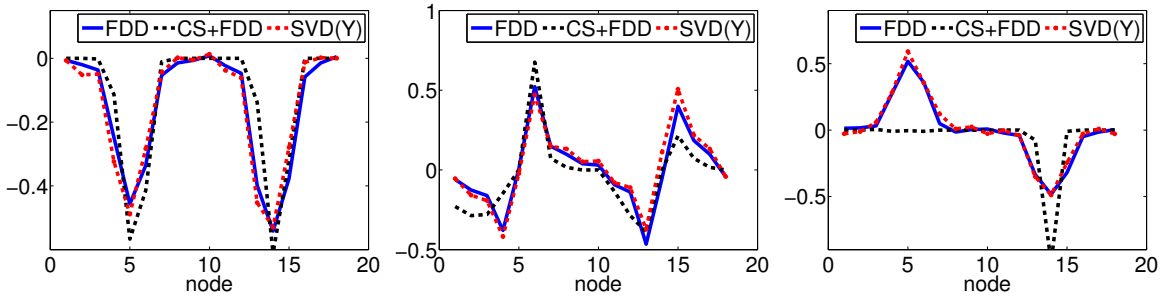


Figure 7.2: *Comparison of FDD with CS+FDD and our proposed method SVD(Y). Each mode shape returned by CS+FDD and SVD(Y) is compared against the corresponding mode shape returned by FDD. The result is evaluated by computing the correlation (COR) between a pair of mode shapes. (a) CS+FDD:* $COR = 0.9382$*, SVD(Y):* $COR = 0.9867$ *(b) CS+FDD:* $COR = 0.8726$*, SVD(Y):* $COR = 0.9701$*, and (c) CS+FDD:* $COR = 0.5209$*, SVD(Y):* $COR = 0.9882$.

resulted in bad estimates of mode shapes by FDD. In fact, the number of measurements we took corresponded to $M = 50 \approx 0.02L$ and each signal was simply not sparse enough for an accurate recovery. On the other hand, given the same set of measurements our proposed method, plotted in red, was able to extract highly correlated mode shapes.

The simulation results indeed support our theoretical results that the SVD of the data matrix Y should return accurate estimates of the true mode shape vectors. We would like to emphasize the fact that the dataset that we have used for this simulation contained noise and were acquired from a real structure *with* damping. Despite these conditions, the fact that our method was able to successfully identify the mode shape vectors suggest that our theoretical findings are potentially extendable to more complicated scenarios.

CHAPTER 8

Conclusion

In this dissertation, we have applied CS to various multi-signal applications. We gave thorough treatments of the sensing, joint signal model, and reconstruction algorithm aspects that need to be considered for a successful application of CS in these multi-signal environments.

8.1 Joint signal model and reconstruction algorithm

8.1.1 Video application

In the video application, based on the observation that many videos should have limited temporal bandwidth we proposed an algorithm that only reconstructs a small set of anchor frames of the video. The proposed algorithm is multiscale and iterative in nature and accurately compensates for the motion between anchor frames by using LIMAT as the sparsifying transform in the reconstruction step. By alternating between video reconstruction and motion estimation our algorithm is able to successfully reconstruct the anchor frames. Simulation results show that our algorithm outperforms other sparsifying transforms proposed in the literature.

There are various future directions one can think of based on our work. For example, although we have reported results in terms of the reconstruction quality of the anchor frames themselves, it is worth reiterating that from the estimated anchor frames $\widehat{x}_a$ one can, if desired, also obtain an estimate of the original high-rate video sequence via $\widehat{x}_d = \Gamma \widehat{x}_a$. As an example, we construct an estimate of the full 512-frame *Pendulum + Cars* video sequence by interpolating the 32 anchor frames that were reconstructed in Figure 3.4. In Figure 8.1, we plot in blue the PSNR for each of the 512 reconstructed frames. Among these 512 frames, of course, are the 32 anchor frames that were reconstructed in the first place; the PSNR for each of these anchor frames is marked with a red square. We note that the PSNR of the full reconstructed video sequence is 31.31 dB, which is only

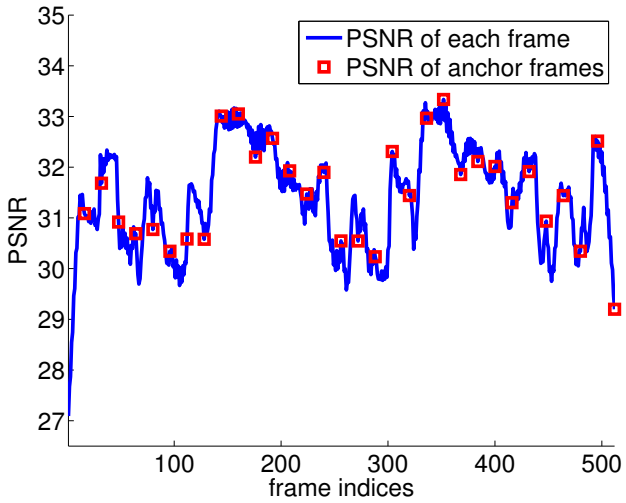


Figure 8.1: *Frame-by-frame PSNR values of high-rate* Pendulum + Cars *video estimate. This video was obtained by interpolating the 32 anchor frames that were reconstructed with our algorithm. Four of the reconstructed anchor frames are illustrated in Figure 3.4(e). The blue line shows the PSNR values of each frame, and the red squares mark the PSNR values of each anchor frame. The PSNR of the full reconstructed video sequence is* 31.31 *dB, which is only slightly lower than the PSNR of the reconstructed anchor frames.*

slightly lower than the PSNR of the reconstructed anchor frames. This confirms that the anchor frames capture much of the critical information in the video, and of course reconstructing the anchor frames was an essential step in recovering the full video, because the full video (with $256 \times 256 \times 512$ unknowns) would be very difficult to reconstruct directly from just 800×512 measurements.

Based on the above observation, our work can be of great value to conventional high-rate video acquisition systems that suffer from an explosion of high-resolution video frames. In such systems a video sequence of only a few seconds or minutes can amount to tera-bytes of data. It is easy to see how the entire process from video acquisition to data storage can be extremely challenging even for the most state-of-the-art computers. Our analysis of temporal complexity of videos provides a interesting guideline on how to make each step in the process more efficient. For example, our analysis on the limited temporal complexity suggests that the temporal bandwidth is essentially limited to a bandwidth that is proportional to the maximum motion content in the video. Therefore, the speed of motion essentially dictates the information level of the video, and depending on the motion content and the desired spatial resolution one may be able to sample at a much lower rate reducing the overall complexity of the system.

Another direction of future work is to establish the overall and/or scale-by-scale reconstruction guarantees of the multiscale algorithm using LIMAT as the sparsifying transform. This guarantee will depend on how accurately we can extract the motion vectors after each scale and also on how these motion vectors may affect the coherence of the columns of the LIMAT matrix. A good starting

point would be to study the effects of motion vectors on the coherence, i.e., what kind of motion vectors will cause the coherence to be high and what kind of motion will cause the coherence to be low. Based on these findings we would then need to think about how the combination of a block diagonal matrix with the LIMAT matrix will affect the reconstruction performance. Since both block diagonal matrices and LIMAT are dependent on the underlying signal it would be important to devise an appropriate assumption about the signal to carry out the analysis.

8.1.2 Multi-view application

In the multi-view imaging application, we have discussed how non-collaborative CS measurement schemes can be used to simplify the acquisition and encoding of multi-image ensembles. We have presented a geometric framework in which many multi-view imaging problems may be cast and explained how this framework can inform the design of effective manifold lifting algorithms for joint reconstruction. We conclude with a few remarks concerning practical and theoretical aspects of the manifold lifting framework.

First, let us briefly discuss the process of learning camera positions when they are initially completely unknown. In our satellite experiments, we have observed that the accuracy of the ISOMAP embedding depends on the relative size of the subimages x_j to the underlying scene x, with larger subimages leading us to higher quality embeddings. As the size of the subimages decreases, we need more and more camera positions to get a reasonable embedding, and we can reach a point where even thousands of camera positions are insufficient. In such cases, and in applications not limited to satellite imaging, it may be possible to get a reliable embedding by grouping local camera positions together. On a different note, once an initial set of camera position estimates have been obtained, it may also be possible to build on an idea suggested in [75] and seek a refinement of these position estimates that minimize the overall ℓ_1 norm of the reconstructed image. A multiscale approach could again help such a technique converge if the initial estimates are far off.

Second, an interesting open question is whether the measurement matrices utilized in DCS multi-view imaging scenarios satisfy the RIP with respect to some reconstruction basis Ψ. Establishing an RIP bound would give a guide for the requisite number of measurements (ideally, at each scale) and also give a guarantee for reconstruction accuracy. Although we do not yet have a definitive answer to this question, we suggest that there may be promising connections between these matrices and other structured matrices that have been studied in the CS literature. For example, the measurement matrix $\Phi_{\text{big}}R$ employed in the satellite experiment is closely related to a partial circulant matrix, where the relative shifts between the rows represent the relative offsets between the camera posi-

tions. RIP results have been established for circulant matrices [12] that are generated by a densely populated random row vector. In our case, $\Phi_{\text{big}}R$ has more of a block circulant structure because it is generated by the submatrices Φ_j, and so there may also be connections with the analysis in [96]. However, each row of $\Phi_{\text{big}}R$ will contain a large number of zeros, and it is conceivable that this could degrade the isometric property of $\Phi_{\text{big}}R$. We believe, though, that by collecting multiple measurements from each camera, we are compensating for this degradation. Other possible directions for analysis could be to build on the concentration of measure bounds recently established for block diagonal matrices [140] and Toeplitz matrices [116].

Finally, another open question in the manifold lifting framework is what could be said about the uniqueness of $\mathcal{M}(x)$ given samples of $\Phi\mathcal{M}(x)$. When all points on the manifold $\mathcal{M}(x)$ are K-sparse, the RIP can be one avenue to proving uniqueness, but since our objective is to sample fewer than $\mathcal{O}(K\log(N/K))$ measurements for each signal, a stronger argument would be preferable. By considering the restricted degrees of freedom that these signal ensembles have, it seems reasonable to believe that we can in fact establish a stronger result.

8.1.3 Structural health monitoring application

In this work, we identified the joint signal structure of signal ensembles that arise in SHM systems from equations describing an MDOF system. This allowed us to relate the acquired measurements from each sensor node to the underlying modal parameters via a simple SVD of the data matrix. We provided sufficient conditions for the accurate recovery of mode shapes via the SVD for uniform and random sampling schemes. Our experimental results on a real dataset[1] showed that the proposed method is able to successfully recover the underlying mode shapes.

An immediate extension of our work would be to consider the problem of extracting the modal frequencies in addition to the mode shapes. Essentially, the SVD of $[V]$ indirectly reveals an estimate of the modal frequencies. More specifically, let us consider the case when $\{v(t)\}$ is sampled according to the uniform sampling scheme. Then the matrix $[\hat{S}]$ that is a byproduct obtained via SVD of $[V]$ implicitly contains the information of the modal frequencies because each row of $[\hat{S}]$ can be written as $[\hat{S}]_{l,:} = \{e^{i\omega_l t_1}, \ldots, e^{i\omega_l t_M}\} + \{n\}$, where $\{n\} \in \mathbb{R}^M$ represents an error component. A straightforward way to estimate the frequency content of each $[\hat{S}]_{l,:}$, is to use Fourier transform and locate the peaks in the spectrum. However, it is possible that such a method may not work well (due to potential

[1] As we had limited access to real datasets we were only able to test our method against one particular dataset. To get a better idea of how our method performs given this dataset we could use cross-validation and test our method on various subsamples of $\{u_1\}, \ldots, \{u_{18}\}$. We could also create synthetic vibration data by simulating a system of ordinary differential equations.

aliasing) for frequencies that are higher than $\delta_{\max}/2$. In such a case, random sampling schemes together with sparsity enforced reconstructions may serve as a better method for extracting modal frequencies.

Other extensions are also possible such as considering damping in an MDOF system, and including a forcing term in the equation. Each extension will lead to a more realistic modeling of the underlying structural dynamics that will lead to a more precise and accurate algorithm.

Finally, all of our theory is based on the fact that we are sampling each vibration signal at the same time points $t_1, \ldots, t_M$. This facilitated the extraction of important geometrical information such as the right singular vectors of the signal matrix. Another interesting alternative to this sampling method would be to use different sampling times at each sensor node. This may potentially increase the diversity in the measurements. However, to utilize these measurements in the extraction of the modal parameters we have to design a completely new algorithm since the signal matrix can no longer be related to the modal parameters via the SVD. The key to designing such algorithm would be to identify an appropriate signal model to fully exploit the added diversity in the measurements.

8.2 Measurement matrices in multi-signal scenarios

We have also analyzed important characteristics of CS measurement matrices. In particular, we have studied the CoM behavior of BD matrices, and also established that random matrices are able to closely preserve the singular values and right-singular vectors of matrices in the relative sense. Although the meaning and purpose of each result differs, the common thread between the two results is that they are both applicable in multi-signal scenarios. Interestingly, when each signal $x_j \in \mathbb{R}^N$ is measured with the same measurement matrix Φ, i.e., $y_i = \Phi x_i \ i = 1, \ldots, J$, the overall equation can be represented with an RBD matrix but also via $Y = \Phi X$, where $Y = [y_1, \ldots, y_J]$ and $X = [x_1, \ldots, x_J]$. Our results provide a useful guideline that if we desire a CoM guarantee it is best when the signals have equal energies and are orthogonal to one another, but if we want to obtain accurate right singular vectors of X it is best when the singular values of X are sharply decaying.

A curious line of future work is to establish a unifying theory on the advantages/disadvantages, the gains/losses, etc., of using the same versus different measurement matrices for the acquisition of multiple signals. In the applications that we have discussed in this dissertation we have seen both cases. For example, in CS video we have used different measurement matrices to measure each video frame. In both multiview imaging and SHM systems we have used the same measurement matrices to measure each image/vibration signal. It is easy to see that using different measurement

matrices will help us acquire more diverse information about the underlying signals. For example, in the extreme case when $x_1 = \cdots = x_J$, each corresponding measurement vector $y_1, \ldots, y_J$ will give us more information about x_1. However, for the same example, using the same measurement matrix to acquire $x_1, \ldots, x_J$, each corresponding measurement vector $y_1, \ldots, y_J$ will give us the same information over and over again. On the other hand, using the same measurement matrix can give us important insights into the geometry that relates the signal ensemble. For example, in both multiview imaging and SHM systems it was crucial for us to use the same measurement matrix to estimate the initial camera positions and the mode shape vectors, respectively.

An interesting application of our work on the singular values and singular vectors is to the recovery of a low-rank, $N \times N$ symmetric matrix X. Let us write the eigendecomposition of X as $X = U\Lambda U^T$, where we assume that Λ only has a few non-zero eigenvalues that are sharply decaying, or sufficiently separated. Suppose we wish to recover X from $Y = \Phi X$. A simple way to achieve this is to compute the SVD of Y to get estimates of $\hat{U}$ and $\hat{\Lambda}$. This can be used to build $\hat{X} = \hat{U}\hat{\Lambda}\hat{U}^T$. In some sense, it is remarkable how we can recover X without exploiting any sparsity within and/or across the signals in X, which are what most reconstruction methods rely on in the multi-signal CS literature.

APPENDICES

APPENDIX A

Appendix A: Proof of Theorem 5.3

Proof. Let $y = \Phi x$. For each matrix Φ_j, we let $[\Phi_j]_{i,n}$ denote the n^{th} entry of the i^{th} row of Φ_j. Further, we let $y_j(i)$ denote the i^{th} component of measurement vector y_j, and we let $x_j(n)$ denote the n^{th} entry of signal block x_j.

We begin by characterizing the point of concentration. One can write $y_j(i) = \sum_{n=1}^{N} [\Phi_j]_{i,n} x_j(n)$, and so it follows that $\mathbf{E}y_j^2(i) = \mathbf{E}\left(\sum_{n=1}^{N} [\Phi_j]_{i,n} x_j(n)\right)^2$. Since the $[\Phi_j]_{i,n}$ are zero-mean and independent, all cross product terms are equal to zero, and therefore we can write $\mathbf{E}y_j^2(i) = \mathbf{E}\sum_{n=1}^{N} [\Phi_j]_{i,n}^2 x_j^2(n) = \sigma_j^2 \|x_j\|_2^2 = \frac{1}{M_j}\|x_j\|_2^2$. Combining all of the measurements, we then have $\mathbf{E}\|y\|_2^2 = \sum_{j=1}^{J}\sum_{i=1}^{M_j} \mathbf{E}y_j^2(i) = \sum_{j=1}^{J}\sum_{i=1}^{M_j} \frac{\|x_j\|_2^2}{M_j} = \sum_{j=1}^{J} \|x_j\|_2^2 = \|x\|_2^2$.

Now, we are interested in the probability that $\left|\|y\|_2^2 - \|x\|_2^2\right| > \epsilon\|x\|_2^2$. Since $\mathbf{E}\|y\|_2^2 = \|x\|_2^2$, this is equivalent to the condition that $\left|\|y\|_2^2 - \mathbf{E}\|y\|_2^2\right| > \epsilon\mathbf{E}\|y\|_2^2$. For a given $j \in \{1, 2, \ldots, J\}$ and $i \in \{1, 2, \ldots, M_j\}$, all $\{[\Phi_j]_{i,n}\}_{n=1}^{N}$ are i.i.d. subgaussian random variables with subgaussian norms equal to $\|\frac{\phi}{\sqrt{M_j}}\|_{\psi_2} = \frac{\|\phi\|_{\psi_2}}{\sqrt{M_j}}$. From above, we know that $y_j(i)$ can be expressed as a linear combination of these random variables, with weights given by the entries of x_j. As with Gaussian random variables, linear combinations of i.i.d. subgaussian random variables are also subgaussian. In particular, from [132, Lemma 9] it follows that each $y_j(i)$ is a subgaussian random variable with subgaussian norm $\|y_j(i)\|_{\psi_2} \leq c_1 \frac{\|\phi\|_{\psi_2}}{\sqrt{M_j}}\|x_j\|_2$, where c_1 is an absolute constant.

In order to obtain a concentration bound for $\|y\|_2^2$, we require the following important theorem regarding sums of squares of subgaussian random variables.

Theorem 1. [132] *Let $X_1, \ldots, X_L$ be independent subgaussian random variables with subgaussian norms $\|X_i\|_{\psi_2}$ for all $i = 1, \ldots, L$. Let $T = \max_i \|X_i\|_{\psi_2}^2$. Then for every $t \geq 0$ and every $a =$*

$(a_1, \ldots, a_L) \in \mathbb{R}^L$, *we have*

$$P\left\{\left|\sum_{i=1}^{L} a_i(X_i^2 - \mathbf{E}X_i^2)\right| \geq t\right\} \leq 2\exp\left\{-C_1 \min\left(\frac{t^2}{16T^2\|a\|_2^2}, \frac{t}{4T\|a\|_\infty}\right)\right\},$$

where $C_1 > 0$ *is an absolute constant.*

Proof. From [132, Lemma 14], we know that each $X_1^2, \ldots, X_L^2$ is a subexponential random variable with subexponential norm $\|X_i^2\|_{\psi_1} \leq 2\|X_i\|_{\psi_2}^2$. For each $i = 1, 2, \ldots, L$, we define $Y_i = X_i^2 - \mathbf{E}X_i^2$ is a zero-mean subexponential random, and from [132, Remark 18], it follows that $\|Y_i\|_{\psi_1} \leq 2\|X_i^2\|_{\psi_1}$. The theorem follows by applying [132, Proposition 16] to the sum $\sum_{i=1}^{L} a_i Y_i$ with $K = 4T \geq \max_i \|Y_i\|_{\psi_1}$. □

Now, let us define $\widetilde{y}_j(i) := \frac{y_j(i)}{\|y_j(i)\|_{\psi_2}}$ so that $\|\widetilde{y}_j(i)\|_{\psi_2} = 1$, and note that

$$P\left(\left|\|y\|_2^2 - \mathbf{E}\|y\|_2^2\right| > \epsilon\|x\|_2^2\right) = P\left(\left|\sum_j \sum_i \|y_j(i)\|_{\psi_2}^2 \left(\widetilde{y}_j^2(i) - \mathbf{E}\widetilde{y}_j^2(i)\right)\right| > \epsilon\|x\|_2^2\right).$$

We apply Theorem 1 to the subgaussian random variables $\widetilde{y}_j(i)$ (over all i, j) with weights $a_j(i) = \|y_j(i)\|_{\psi_2}^2$. Letting a denote a vector of length $\sum_j M_j$ containing these weights, we have that

$$\begin{aligned}\|a\|_2^2 &= \sum_j \sum_i a_j^2(i) = \sum_j \sum_i \|y_j(i)\|_{\psi_2}^4 \leq c_1^4\|\phi\|_{\psi_2}^4 \sum_j \sum_i \|x_j\|_2^4/M_j^2 \\ &= c_1^4\|\phi\|_{\psi_2}^4 \sum_j \|x_j\|_2^4/M_j = c_1^4\|\phi\|_{\psi_2}^4 \|\mathbf{M}^{-1/2}\gamma\|_2^2,\end{aligned}$$

and

$$\|a\|_\infty = \max_{i,j} a_j(i) = \max_{i,j} \|y_j(i)\|_{\psi_2}^2 \leq c_1^2\|\phi\|_{\psi_2}^2 \max_j \|x_j\|_2^2/M_j = c_1^2\|\phi\|_{\psi_2}^2 \|\mathbf{M}^{-1}\gamma\|_\infty.$$

Further note that $\|x\|_2^2 = \|\gamma\|_1$ and $\|x\|_2^4 = \|\gamma\|_1^2$. We complete the proof by substituting these quantities into Theorem 1 with $T = 1$ and $t = \epsilon\|x\|_2^2$ and by taking $C_2 = \frac{1}{4c_1^2}$. □

APPENDIX B

Appendix B: Proof of Theorem 5.5

In order to prove Theorem 5.5, we will require the following two lemmas.

Lemma B.1. *Suppose $x \in \mathbb{R}^{NJ}$ and $\widetilde{\Phi}$ is an $M \times N$ matrix where $\widetilde{\Phi}^T = [\phi_1 \ \phi_2 \ \cdots \ \phi_M]$ with each $\phi_i \in \mathbb{R}^N$. Let Φ be an $MJ \times NJ$ RBD matrix as defined in (5.1) with all $\Phi_j = \widetilde{\Phi}$. If $y = \Phi x$, then $\|y\|_2^2 = \sum_{i=1}^M \phi_i^T A \phi_i$, where $A = X^T X$ with X defined in (5.6).*

Proof of Lemma B.1.

$$\begin{aligned}\|y\|_2^2 = x^T \Phi^T \Phi x = \sum_{j=1}^J x_j^T \widetilde{\Phi}^T \widetilde{\Phi} x_j = \sum_{j=1}^J x_j^T \left(\sum_{i=1}^M \phi_i \phi_i^T \right) x_j \\ = \sum_{i=1}^M \phi_i^T \left(\sum_{j=1}^J x_j x_j^T \right) \phi_i = \sum_{i=1}^M \phi_i^T A \phi_i.\end{aligned}$$

□

Lemma B.2. *Suppose $z \in \mathbb{R}^N$ is a random vector with i.i.d. Gaussian entries each having zero-mean and variance σ^2. For any symmetric $N \times N$ matrix A with eigenvalues $\{\lambda_i\}_{i=1}^N$, there exists a collection of independent, zero-mean Gaussian random variables $\{w_i\}_{i=1}^N$ with variance σ^2 such that $z^T A z = \sum_{i=1}^N \lambda_i w_i^2$.*

Proof of Lemma B.2. Because A is symmetric, it has an eigen-decomposition $A = V^T D V$, where D is a diagonal matrix of its eigenvalues $\{\lambda_i\}_{i=1}^N$ and V is an orthogonal matrix of eigenvectors. Then we have $z^T A z = (Vz)^T D (Vz) = \sum_{i=1}^N \lambda_i w_i^2$, where $w = Vz$ and $w = [w_1, \ w_2, \ \cdots, \ w_N]^T$. Since V is an orthogonal matrix, $\{w_i\}_{i=1}^N$ are i.i.d. Gaussian random variables with zero-mean and variance σ^2. □

Proof of Theorem 5.5. Let $y = \Phi x$. We first calculate $\mathbf{E}\|y\|_2^2$ to determine the point of concentration. Applying Lemma B.1 to y and Lemma B.2 with $z = \phi_i$ for each $i = 1, 2, \ldots, M$, we have $\|y\|_2^2 = \sum_{i=1}^M \phi_i^T A \phi_i = \sum_{i=1}^M \sum_{j=1}^N \frac{\lambda_j}{M} w_{i,j}^2$, where each $\{w_{i,j}\}_{i,j}$ is an independent Gaussian random variable with zero mean and unit variance. After switching the order of the summations and observing that $\mathrm{Tr}(X^T X) = \mathrm{Tr}(XX^T)$ where $\mathrm{Tr}(\cdot)$ is the trace operator, we have $\mathbf{E}\|y\|_2^2 = \sum_{j=1}^N \frac{\lambda_j}{M} \sum_{i=1}^M \mathbf{E} w_{i,j}^2 = \sum_{j=1}^N \lambda_j = \mathrm{Tr}(XX^T) = \|x\|_2^2$.

Having established the point of concentration for $\|y\|_2^2$, let us now compute the probability that $\left| \|y\|_2^2 - \|x\|_2^2 \right| > \epsilon \|x\|_2^2$. Since $\mathbf{E}\|y\|_2^2 = \|x\|_2^2$, this is equivalent to the condition that $\left| \|y\|_2^2 - \mathbf{E}\|y\|_2^2 \right| > \epsilon \mathbf{E}\|y\|_2^2$. We again apply Theorem 1 to establish a concentration result. To do so, note that each $w_{i,j}$ is a subgaussian random variable with the same subgaussian norm $\|w\|_{\psi_2} := \|w_{i,j}\|_{\psi_2}$; because these variables are Gaussian with unit variance, it is also known [132] that there exists an absolute constant c_2 such that $\|w\|_{\psi_2} \leq c_2$. Let us define $\widetilde{w}_{i,j} := \frac{w_{i,j}}{\|w\|_{\psi_2}}$ so that $\|\widetilde{w}_{i,j}\|_{\psi_2} = 1$, and note that

$$P(|\|y\|_2^2 - \mathbf{E}\|y\|_2^2| > \epsilon \|x\|_2^2) = P\left(\left| \sum_j \sum_i \frac{\|w\|_{\psi_2}^2}{M} \lambda_j (\widetilde{w}_{i,j}^2 - \mathbf{E}\widetilde{w}_{i,j}^2) \right| > \epsilon \|x\|_2^2 \right).$$

We apply Theorem 1 to the subgaussian random variables $\widetilde{w}_{i,j}$ (over all i, j) with weights $a_j(i) = \frac{\|w\|_{\psi_2}^2}{M} \lambda_j$. Letting a denote a vector of length MJ containing these weights, we have that $\|a\|_2^2 = \sum_j \sum_i a_j^2(i) = \sum_j \sum_i \frac{\|w\|_{\psi_2}^4}{M^2} \lambda_j^2 = \frac{\|w\|_{\psi_2}^4}{M} \sum_j \lambda_j^2 = \frac{\|w\|_{\psi_2}^4}{M} \|\lambda\|_2^2 \leq \frac{c_2^4}{M} \|\lambda\|_2^2$ and $\|a\|_\infty = \max_{i,j} a_j(i) = \frac{\|w\|_{\psi_2}^2}{M} \max_j \lambda_j = \frac{\|w\|_{\psi_2}^2}{M} \|\lambda\|_\infty \leq \frac{c_2^2}{M} \|\lambda\|_\infty$. Note that $\|x\|_2^2 = \mathrm{Tr}(X^T X) = \|\lambda\|_1$ and $\|x\|_2^4 = \|\lambda\|_1^2$ since the eigenvalues $\{\lambda_j\}_{j=1}^N$ are non-negative. We complete the proof by substituting these quantities into Theorem 1 with $T = 1$ and $t = \epsilon \|x\|_2^2$ and by taking $C_3 = \frac{1}{4c_2^2}$.

□

APPENDIX C

Appendix C: Proof of Theorem 5.9

Our result follows from an application of the following.

Theorem 2. [29, Theorem 3.1] *Let $x \in \mathbb{C}^{N'}$ and $\beta > 1$. Suppose $N' > 512$ and choose N_T and N_Ω such that:*

$$N_T + N_\Omega \leq \frac{0.5583N'/q}{\sqrt{(\beta+1)\log(N')}} \text{ and } N_T + N_\Omega \leq \frac{\sqrt{2/3}N'\left(\frac{1}{q} - \frac{(\log N')^2}{N'}\right)}{\sqrt{(\beta+1)\log(N')}}.$$

Fix a subset T of the time domain with $|T| = N_T$. Let Ω be a subset of size N_Ω of the frequency domain generated uniformly at random. Then with probability at least $1 - O((\log(N'))^{1/2}N'^{-\beta})$, every signal x supported on Ω in the frequency domain has most of its energy in the time domain outside of T. In particular, $\|x_T\|_2^2 \leq \frac{\|x\|_2^2}{q}$, where x_T denotes the restriction of x to the support T.

Proof of Theorem 5.9. First, observe that $\|\gamma\|_1^2 = \|x\|_2^4$ and $\|\gamma\|_2^2 = \sum_{k=1}^J \|x_k\|_2^4$. Next, apply Theorem 2 with $N_\Omega = S$ and $N_T = N = N'/J$, being careful to select a value for q such that (C.1) is satisfied. In particular, we require

$$\frac{1}{q} \geq \frac{(N+S)\sqrt{(\beta+1)\log N'}}{0.5583N'} \text{ and } \frac{1}{q} \geq \frac{\frac{(N+S)}{\sqrt{2/3}}\sqrt{(\beta+1)\log N'} + (\log N')^2}{N'}.$$

This is satisfied if we choose

$$q \leq \min\left\{\frac{0.5583N'}{(N+S)\sqrt{(\beta+1)\log N'}}, \frac{N'}{\frac{(N+S)}{\sqrt{2/3}}\sqrt{(\beta+1)\log N'} + (\log N')^2}\right\}. \tag{C.1}$$

Choosing any q satisfying (C.1), we have that with *failure probability* at most

$O((\log(N'))^{1/2}(N')^{-\beta})$, $\|x_k\|_2^2 \leq \frac{\|x\|_2^2}{q}$ for each $k = 1, 2, \ldots, J$, implying that each block individually is favorable. Taking a union bound for all k to cover each block, we have that with total failure probability at most $O(J(\log(N'))^{1/2}(N')^{-\beta})$, $\|\gamma\|_2^2 = \sum_{k=1}^{J} \|x_k\|_2^4 \leq \frac{J\|x\|_2^4}{q^2}$. Thus with this same failure probability, $\frac{\Gamma_2}{M} = \frac{\|\gamma\|_1^2}{\|\gamma\|_2^2} \geq \frac{q^2}{J}$. Combining with (C.1) and using the fact that $S < N$, we thus have:

$$\begin{aligned}
\frac{\Gamma_2}{M} &\geq \min\left\{ \frac{0.5583^2 N^2 J}{(N+S)^2(\beta+1)\log N'}, \frac{N^2 J}{\left(\frac{(N+S)}{\sqrt{2/3}}\sqrt{(\beta+1)\log N'} + (\log N')^2\right)^2} \right\} \\
&\geq \min\left\{ \frac{(0.5583^2/2^2) J}{(\beta+1)\log(N')}, \frac{J}{\left(\frac{2}{\sqrt{2/3}}\sqrt{(\beta+1)\log N'} + \frac{(\log N')^2}{N}\right)^2} \right\}.
\end{aligned}$$

□

APPENDIX D

Appendix D: Proof of Lemma 5.10

Proof. Let X be the $J \times N$ matrix as defined in (5.6). Without loss of generality, we suppose the nonzero eigenvalues $\{\lambda_i\}_{i=1}^{\min(J,N)}$ of $X^T X$ are sorted in order of decreasing magnitude, and we let $\lambda_{\max} := \lambda_1$ and $\lambda_{\min} := \lambda_{\min(J,N)}$. We can lower bound Λ_2 in terms of these extremal eigenvalues by writing

$$\begin{aligned} \Lambda_2 = \frac{M\|\lambda\|_1^2}{\|\lambda\|_2^2} = M\frac{\sum_i \lambda_i^2 + \sum_i \sum_{j\neq i} \lambda_i\lambda_j}{\sum_i \lambda_i^2} &\geq M + M\frac{\lambda_{\min}}{\lambda_{\max}}\frac{\sum_i \sum_{j\neq i}\lambda_i}{\sum_i \lambda_i} \\ &= M + M\frac{\lambda_{\min}}{\lambda_{\max}}(J-1). \end{aligned} \tag{D.1}$$

Assume that $\epsilon \leq \frac{\|\phi\|_{\psi_2}^2}{C_2}$, and let us define the following events:

$$\begin{aligned} A &= \left\{ N\sigma^2(1-\epsilon)^2 \leq \frac{\|X^T z\|_2^2}{\|z\|_2^2} \leq N\sigma^2(1+\epsilon)^2,\ \forall z \in \mathbb{R}^J \right\}, \\ B &= \left\{\lambda_{\max} \leq N\sigma^2(1+\epsilon)^2\right\} \bigcap \left\{\lambda_{\min} \geq N\sigma^2(1-\epsilon)^2\right\}, \\ C &= \left\{ \frac{\lambda_{\min}}{\lambda_{\max}} \geq \left(\frac{1-\epsilon}{1+\epsilon}\right)^2 \right\}, \\ D &= \left\{ \Lambda_2 \geq M + M\left(\frac{1-\epsilon}{1+\epsilon}\right)^2 (J-1) \right\}. \end{aligned}$$

These events satisfy $A = B \subseteq C \subseteq D$, where the last relation follows from (D.1). It follows that $P(D^c) \leq P(A^c)$, where A^c represents the complement of event A. Because X^T is populated with i.i.d. subgaussian random variables, it follows as a corollary of Theorem 5.3 (by setting $M \leftarrow N$ and $J \leftarrow 1$ in the context of that theorem) that for any $z \in \mathbb{R}^J$ and $\epsilon \leq \frac{\|\phi\|_{\psi_2}^2}{C_2}$, $P(\left|\|X^T z\|_2^2 - N\sigma^2\|z\|_2^2\right| >$

$\epsilon N\sigma^2\|z\|_2^2) \leq 2\exp\left(-\frac{C_1C_2^2N\epsilon^2}{\|\phi\|_{\psi_2}^4}\right)$. Thus, for an upper bound for $P(A^c)$, we can follow the straightforward arguments in [14, Lemma 5.1] and conclude that $P(A^c) \leq 2\left(\frac{12}{\epsilon}\right)^J \exp\left(-\frac{C_1C_2^2N\epsilon^2}{\|\phi\|_{\psi_2}^4}\right)$. Thus by choosing $J \leq \frac{C_1C_2^2N\epsilon^2}{2\|\phi\|_{\psi_2}^4 \log(12/\epsilon)}$, we see that $P(D^c) \leq 2\exp\left(-\frac{1}{2}\frac{C_1C_2^2N\epsilon^2}{\|\phi\|_{\psi_2}^4}\right)$. Finally, the fact that $\Gamma_2 \geq \Lambda_2$ follows from (5.9). $\square$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] D. Achlioptas. Database-friendly random projections. In *Proc. 20th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Syst. (PODS)*, pages 274–281, New York, NY, USA, 2001. ACM.

[2] R. Adamczak, A. E. Litvak, A. Pajor, and N. Tomczak-Jaegermann. Restricted isometry property of matrices with independent columns and neighborly polytopes by random sampling. *Arxiv preprint arXiv:0904.4723*, 2009.

[3] K. J. Ahn, S. Guha, and A. McGregor. Analyzing graph structure via linear measurements. In *SODA '12: Proc. of the 23rd Annual ACM-SIAM Symp. on Discrete Algorithms*, 2012.

[4] K. J. Ahn, S. Guha, and A. McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *PODS '12: Proc. of the 31st Symp. on Principles of Database Syst.*, May 2012.

[5] N. Ailon and B. Chazelle. The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39:302–322, May 2009.

[6] N. Ailon and E. Liberty. Almost optimal unrestricted fast Johnson-Lindenstrauss transform. *Arxiv preprint arXiv:1005.5513*, 2010.

[7] T. Ajdler, L. Sbaiz, and M. Vetterli. The plenacoustic function and its sampling. *IEEE Trans. on Signal Process.*, 54(10):3790–3804, 2006.

[8] T. Ajdler, L. Sbaiz, and M. Vetterli. Dynamic measurement of room impulse responses using a moving microphone. *The J. of the Acoustical Society of America*, 122(3):1636–1645, 2007.

[9] M. S. Asif, D. Reddy, P. T. Boufounos, and A. Veeraraghavanc. Streaming compressive sensing for high-speed periodic videos. In *Proc. Int. Conf. Image Process. (ICIP)*, 2009.

[10] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *STOC '01: Proc. of the 33rd Annual ACM Symp. on Theory of Computing*, 2001.

[11] P. K. Baheti and M. A. Neifeld. Random projections based feature-specific structured imaging. *Opt. Express*, 16(3):1764–1776, February 2008.

[12] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak. Toeplitz-structured compressed sensing matrices. In *Proc. IEEE/SP Workshop Stat. Signal Process. (SSP)*, 2007.

[13] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Trans. Inf. Theory*, 56(4):1982–2001, 2010.

[14] R. G. Baraniuk, M. A. Davenport, R. A. DeVore, and M. B. Wakin. A simple proof of the restricted isometry property for random matrices. *Constr. Approx.*, 28(3):253–263, 2008.

[15] R. G. Baraniuk and M. B. Wakin. Random projections of smooth manifolds. *Found. Comput. Math.*, 9(1):51–77, 2009.

[16] A. Barjatya. Block matching algorithms for motion estimation. 2004. Technical report, Utah State University.

[17] J. Barlow and J. Demmel. Computing accurate eigensystems of scaled diagonally dominant matrices. Technical report, 1980.

[18] D. Baron, M. B. Wakin, M. Duarte, S. Sarvotham, and R. G. Baraniuk. Distributed compressed sensing. Rice University Technical Report TREE-0612, November 2006.

[19] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Inf. Process. Syst.*, 14:585–591, 2001.

[20] J. Berent and P. L. Dragotti. Plenoptic manifolds. *IEEE Signal Process. Mag.*, 24(6), 2007.

[21] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Independent Component Anal. and Signal Separation*, 27(3):265–274, 2009.

[22] B. Boashash. *Time frequency signal analysis and processing: A comprehensive reference.* Elsevier, 2003.

[23] P. Boufounos and M. S. Asif. Compressive sampling for streaming signals with sparse frequency content. In *Proc. Conf. Inf. Sci. and Syst. (CISS)*, 2010.

[24] R. Brincker, L. Zhang, and P. Andersen. Modal identification from ambient responses using frequency domain decomposition. *Proc. of the 18th Int. Modal Anal. Conf.*, 1:62530, 2000.

[25] V. V. Buldygin and Y. V. Kozachenko. Sub-gaussian random variables. *Ukrainian Math. J.*, 32:483–489, November 1980.

[26] E. J. Candès. Compressive sampling. In *Proc. Int. Congress Math.*, volume 3, pages 1433–1452, Madrid, Spain, August 2006.

[27] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Math.*, 346(9-10):589–592, 2008.

[28] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. of the ACM*, 58(3), 2011.

[29] E. J. Candès and J. Romberg. Quantitative robust uncertainty principles and optimally sparse decompositions. *Found. Comp. Math.*, 6(2):227–254, 2006.

[30] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inf. Theory*, 51(12):4203–4215, 2005.

[31] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inf. Theory*, 52(12):5406 –5425, 2006.

[32] V. Chandrasekaran, M. B. Wakin, D. Baron, and R. G. Baraniuk. Representation and compression of multi-dimensional piecewise functions using surflets. *IEEE Trans. Inf. Theory*, 55(1):374–400, 2009.

[33] M. Chen, J. Silva, J. W. Paisley, C. Wang, D. B. Dunson, and L. Carin. Corrections to "compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds". *IEEE Trans. on Signal Process.*, 59(3):1329, 2011.

[34] X. Chen and P. Frossard. Joint reconstruction of compressed multi-view images. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, 2009.

[35] F. R. K. Chung. *Spectral graph theory*, volume 92. Amer. Math. Society, 1997.

[36] M. Clint and A. Jennings. The evaluation of eigenvalues and eigenvectors of real symmetric matrices by simultaneous iteration. *Comput. J.*, 13(1):76–80, 1970.

[37] R. Coifman, F. Geshwind, and Y. Meyer. Noiselets. *Appl. Comput. Harmon. Anal.*, 10(1):27–44, 2001.

[38] A. Dasgupta, J. E. Hopcroft, and F. McSherry. Spectral analysis of random graphs with skewed degree distributions. In *FOCS '04: Proc. of the 45th Annual IEEE Symp. on Found. of Comput. Sci.*, 2004.

[39] A. Dasgupta, R. Kumar, and T. Sarlos. A sparse Johnson-Lindenstrauss transform. In *STOC '10: Proc. of the 42nd ACM Symp. on Theory of Computing*, 2010.

[40] S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss lemma. *Random Struct. Algor.*, 22(1):60–65, 2002.

[41] M. A. Davenport. Concentration of measure and sub-gaussian distributions, 2009. Available online at `http://cnx.org/content/m32583/latest/`.

[42] M. A. Davenport. *Random Observations on Random Observations: Sparse Signal Acquisition and Processing*. PhD thesis, Rice University, 2010.

[43] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk. Signal processing with compressive measurements. *IEEE J. Sel. Topics Signal Process.*, 4(2):445–460, 2010.

[44] M. A. Davenport, M. F. Duarte, M. B. Wakin, J. N. Laska, D. Takhar, K. F. Kelly, and R. G. Baraniuk. The smashed filter for compressive classification and target recognition. In *Proc. IS&T/SPIE Symp. Electronic Imaging: Comput. Imaging*, 2007.

[45] M. E. Davies and Y. C. Eldar. Rank awareness in joint sparse recovery. *Arxiv preprint arXiv:1004.4529*, 2010.

[46] C. Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation, III. *SIAM J. Numer. Anal.*, 7, 1970.

[47] R. DeVore, G. Petrova, and P. Wojtaszczyk. Instance-optimality in probability with an l1-minimization decoder. *Appl. and Comput. Harm. Anal.*, 27(3):275–288, 2009.

[48] M. Do, D. Marchand-Maillet, and M. Vetterli. On the bandwidth of the plenoptic function. *IEEE Trans. on Image Process.*, 21(2), 2012.

[49] S. W. Doebling, C. R. Farrar, M. B. Prime, and D. W. Shevitz. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review. *Distribution*, 213(Pt 10):1762–70, 1996.

[50] D. W. Dong and J. J. Atick. Statistics of natural time-varying images. *Network: Comput. in Neural Syst.*, 6(3):345–358, 1995.

[51] D. L. Donoho. Wedgelets: Nearly-minimax estimation of edges. *Ann. Statist.*, 27:859–897, 1999.

[52] D. L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, April 2006.

[53] D. L. Donoho and C. Grimes. Image manifolds which are isometric to Euclidean space. *J. of Math. Imaging Comp. Vision*, 23(1):5–24, July 2005.

[54] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *CoRR*, abs/1109.3843, 2011.

[55] M. F. Duarte, M. A. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Process. Mag.*, 25(2):83–91, 2008.

[56] M. F. Duarte, S. Sarvotham, M. B. Wakin, D. Baron, and R. G. Baraniuk. Joint sparsity models for distributed compressed sensing. In *Online Proc. of the Workshop on Signal Process. with Adaptative Sparse Structured Representations (SPARS)*, Rennes, France, 2005.

[57] S. C. Eisenstat and I. C. F. Ipsen. Relative perturbation techniques for singular value problems. *SIAM J. Numer. Anal.*, 32(6):1972–1988, 1995.

[58] Y. C. Eldar, P. Kuppinger, and H. Bölcskei. Compressed sensing of block-sparse signals: Uncertainty relations and efficient recovery. *CoRR*, abs/0906.3173, 2009.

[59] Y. C. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Trans. Inf. Theory*, 55(11):5302–5316, 2009.

[60] Y. C. Eldar and H. Rauhut. Average case analysis of multichannel sparse recovery using convex relaxation. *IEEE Trans. Inf. Theory*, 56(1):505–519, 2009.

[61] D. Feldman, M. Monemizadeh, C. Sohler, and D. P. Woodruff. Coresets and sketches for high dimensional subspace approximation problems. In *SODA '10: Proc. of the 21st Annual ACM-SIAM Symp. on Discrete Algorithms*, 2010.

[62] R. Fergus, A. Torralba, and W. T. Freeman. Random lens imaging. Technical Report MIT CSAIL TR 2006-058, Massachusetts Institute of Technology, 2006.

[63] J. E. Fowler. Compressive-projection principal component analysis. *IEEE Trans. on Image Process.*, 18(10):2230–2242, 2009.

[64] J. E. Fowler, S. Mun, and E. W. Tramel. Block-based compressed sensing of images and video. *Found. Trends Signal Process.*, 4(4):297–416, 2012.

[65] P. Frankl and H. Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *J. Combinatorial Theory, Series B*, 44(3):355–362, 1988.

[66] M. E. Gehm, R. John, D. J. Brady, R. M. Willett, and T. J. Schulz. Single-shot compressive spectral imaging with a dual-disperser architecture. *Opt. Express*, 15(21):14013–14027, October 2007.

[67] N. Gehrig and P. L. Dragotti. Geometry-driven distributed compression of the plenoptic function: performance bounds and constructive algorithms. *IEEE Trans. on Image Process.*, 18:457–470, March 2009.

[68] S. Gerschgorin. Über die abgrenzung der eigenwerte einer matrix. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, pages 749–754, 1931.

[69] A. C. Gilbert, J. Y. Park, and M. B. Wakin. Sketched SVD: Recovering spectral features from compressive measurements. *CoRR*, abs/1211.0361, 2012.

[70] G. H. Golub and C. F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)*. The Johns Hopkins University Press, 3rd edition, 1996.

[71] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proc. Ann. Conf. Comput. Graphics Interactive Tech. (SIGGRAPH)*, 1996.

[72] O. Guédon, S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann. Subspaces and orthogonal decompositions generated by bounded orthogonal systems. *Positivity*, 11(2):269–283, 2007.

[73] N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[74] J. D. Haupt and R. D. Nowak. Compressive sampling for signal detection. In *Proc. Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*, 2007.

[75] C. Huff and R. Muise. Wide-area surveillance with multiple cameras using distributed compressive imaging. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conf. Series*, volume 8055, 2011.

[76] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. ACM Symp. Theory of Comput.*, pages 604–613, 1998.

[77] A. J. Izenman. *Modern Multivariate Statistical Techniques*. Springer, 2008.

[78] L. Jacques, P. Vandergheynst, A. Bibet, V. Majidzadeh, A. Schmid, and Y. Leblebici. CMOS compressed imaging by random convolution. In *ICASSP*, pages 1113–1116, 2009.

[79] J. Jain and A. Jain. Displacement measurement and its application in interframe image coding. *IEEE Trans. Commun.*, 29(12):1799–1808, December 1981.

[80] T. S. Jayram and D. Woodruff. Optimal bounds for Johnson-Lindenstrauss transforms and streaming problems with sub-constant error. In *SODA '11: Proc. of the 22nd Annual ACM-SIAM Symp. on Discrete Algorithms*, 2011.

[81] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Proc. Conf. Modern Anal. and Probability*, 1984.

[82] R. Kainkaryam, A. Bruex, A. Gilbert, J. Schiefelbein, and P. Woolf. poolMC: Smart pooling of mRNA samples in microarray experiments. *BMC Bioinformatics*, 11(1):299, 2010.

[83] L. W. Kang and C. S. Lu. Distributed compressive video sensing. In *Proc. Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP)*, 2009.

[84] F. W. King. *Hilbert Transforms*. Cambridge University Press, 2009.

[85] F. Krahmer and R. Ward. New and improved Johnson-Lindenstrauss embeddings via the Restricted Isometry Property. *Arxiv preprint arXiv:1009.0744*, 2010.

[86] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In Philip S. S. Yu, Jiawei Han, and Christos Faloutsos, editors, *Link Mining: Models, Algorithms, and Applications*, pages 337–357. Springer New York, 2010.

[87] M. Ledoux. The concentration of measure phenomenon. *Math. Surveys and Monographs, AMS*, 2001.

[88] R. C. Li. Relative perturbation theory: (II) Eigenspace and singular subspace variations. *SIAM J. Matrix Anal. Appl.*, 20:471–492, 1996.

[89] R. C. Li. Relative perturbation theory: (I) Eigenvalue and singular value variations. *SIAM J. Matrix Anal. Appl.*, 19:956–982, 1998.

[90] M. Lustig, D. Donoho, and J. M. Pauly. Sparse mri: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

[91] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly. Compressed sensing MRI. In *IEEE Signal Process. Mag.*, volume 25, pages 72–82, 2008.

[92] J. P. Lynch. An overview of wireless structural health monitoring for civil structures. *Philosophical Trans. of the Royal Society A: Math., Physical and Engineering Sci.*, 365(1851):345–372, 2007.

[93] M. W. Mahoney. Randomized algorithms for matrices and data. *Found. Trends Mach. Learn.*, 3(2):123–224, 2011.

[94] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 2008.

[95] H. Mansour, H. Wason, T. T.Y. Lin, and F. J. Herrmann. Randomized marine acquisition with compressive sampling matrices. *Geophysical Prospecting*, 60(4):648–662, 2012.

[96] R. Marcia and R. Willett. Compressive coded aperture video reconstruction. In *Proc. European Signal Process. Conf. (EUSIPCO)*, 2008.

[97] R. F. Marcia, Z. T. Harmany, and R. M. Willett. Compressive coded aperture imaging. In *Proc. SPIE Electronic Imaging*, 2009.

[98] D. Mascarenas, D. Hush, J. Theiler, and C. Farrar. The application of compressed sensing to detecting damage in structures. In *Proc. 8th Int. Workshop on Structural Health Monitoring*, 2011.

[99] R. Mathias and K. Veselić. A relative perturbation bound for positive definite matrices. *Linear Algebra and its Applications*, 270:315 – 321, 1998.

[100] E. Meijering and M. Unser. A note on cubic convolution interpolation. *IEEE Trans. on Image Process.*, 12(4):477–479, April 2003.

[101] S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann. Uniform uncertainty principle for Bernoulli and subgaussian ensembles. *Const. Approx.*, 28(3):277–289, 2008.

[102] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Appl. and Comput. Harm. Anal.*, 26(3):301 – 321, 2009.

[103] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in Neural inf. Process. Syst.*, 2:849–856, 2002.

[104] J. Y. Park and M. B. Wakin. A multiscale framework for compressive sensing of video. In *Proc. Picture Coding Symp. (PCS)*, 2009.

[105] J. Y. Park and M. B. Wakin. A geometric approach to multi-view compressive imaging. *EURASIP J. on Advances in Signal Process.*, 2012:1–15, 2012.

[106] J. Y. Park and M. B. Wakin. Multiscale algorithm for reconstructing videos from streaming compressive measurements. *Journal of Electronic Imaging*, 22(2):021001–021001, 2013.

[107] J. Y. Park, H. L. Yap, C. J. Rozell, and M. B. Wakin. Concentration of measure for block diagonal matrices with applications to compressive signal processing. *IEEE Trans. on Signal Process.*, 59(12):5859–5875, 2011.

[108] B. Peeters and G. De Roeck. Stochastic system identification for operational modal analysis: A review. *J. of Dynamic Syst., Measurement, and Control*, 123(4):659–667, 2001.

[109] S. Pudlewski, A. Prasanna, and T. Melodia. Compressed-sensing-enabled video streaming for wireless multimedia sensor networks. *IEEE Trans. Mobile Comput.*, 11(6):1060–1072, 2012.

[110] G. Puy, Y. Wiaux, R. Gruetter, J. P. Thiran, D. Van De Ville, and P. Vandergheynst. Spread spectrum for accelerated acquisition in magnetic resonance imaging, 2010.

[111] H. Qi and S. Hughes. Invariance of principal components under low-dimensional random projection of the data. *IEEE Int. Conf. on Image Process. (ICIP)*, October 2012.

[112] R. Robucci, J. D. Gray, L. K. Chiu, J. Romberg, and P. Hasler. Compressive sensing on a CMOS separable-transform image sensor. *Proc. of the IEEE*, 98(6):1089–1101, 2010.

[113] C. J. Rozell, H. L. Yap, J. Y. Park, and M. B. Wakin. Concentration of measure for block diagonal matrices with repeated blocks. In *Proc. Conf. Inf. Sci. and Syst. (CISS)*, February 2010.

[114] M. Rudelson and R. Vershynin. On sparse reconstruction from Fourier and Gaussian measurements. *Commun. on Pure and Appl. Math.*, 61(8):1025–1045, 2008.

[115] H. Rutishauser. Computational aspects of F. L. Bauer's simultaneous iteration method. *Numerische Math.*, 13:4–13, 1969. 10.1007/BF02165269.

[116] B. M. Sanandaji, T. L. Vincent, and M. B. Wakin. Concentration of measure inequalities for compressive Toeplitz matrices with applications to detection and system identification. In *Proc. IEEE Conf. Decision and Control (CDC)*, 2010.

[117] A. C. Sankaranarayanan, C. Studer, and R. G. Baraniuk. CS-MUVI: Video compressive sensing for spatial-multiplexing cameras. In *IEEE Int. Conf. on Comput. Photography*, Seattle, WA, April 2012.

[118] A. Secker and D. Taubman. Lifting-based invertible motion adaptive transform framework for highly scalable video compression. *IEEE Trans. Image Process.*, 12(12):1530–1542, December 2003.

[119] A. Singer, R. R. Coifman, F. J. Sigworth, D. W. Chester, and Y. Shkolnisky. Detecting consistent common lines in cryo-em by voting. *J. of Struct. Biol.*, November 2009.

[120] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Inf. Theory*, 19(4):471–480, January 2003.

[121] H. Sohn, C. R Farrar, F. Hemez, and J. Czarnecki. A review of structural health monitoring literature 1996-2001. *Structural Health Monitoring*, pages 1–7, 2001.

[122] D. A. Spielman. Spectral graph theory. *Lecture Notes, Yale University*, 2009.

[123] R. Sundaresan, Y. Kim, M. S. Nadar, and A. Bilgin. Motion-compensated compressed sensing for dynamic imaging. *Proc. SPIE*, 7798, 2010.

[124] J. B. Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

[125] I. Tošić and P. Frossard. Geometry-based distributed scene representation with omnidirectional vision sensors. *IEEE Trans. on Image Process.*, 17(7):1033–1046, 2008.

[126] I. Tošić and P. Frossard. Distributed multi-view image coding with learned dictionaries. In *Proc. Int. ICST Mobile Multimedia Commun. Conf.*, 2009.

[127] J. A. Tropp. On the linear independence of spikes and sines. *J. Fourier Anal. and Applications*, 14(5):838–858, 2008.

[128] J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Found. Comput. Math.*, 12(4):389–434, 2012.

[129] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory*, 53:4655–4666, 2007.

[130] E. van den Berg and M. P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction, June 2007. http://www.cs.ubc.ca/labs/scl/spgl1.

[131] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM J. on Sci. Comput.*, 31(2):890–912, 2008.

[132] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing: Theory and Applications*. Cambridge University Press. to appear.

[133] M. B. Villarino. Sharp Bounds for the Harmonic Numbers. *ArXiv Mathematics e-prints*, 2005.

[134] M. B. Wakin. A study of the temporal bandwidth of video and its implications in compressive sensing. Colorado School of Mines Technical Report 2012-08-15, available at `http://inside.mines.edu/~mwakin/papers/blvideo-mbw-tr-2012-08-15.pdf`.

[135] M. B. Wakin. *The Geometry of Low-Dimensional Signal Models*. PhD thesis, Department of Electrical and Computer Engineering, Rice University, Houston, TX, 2006.

[136] M. B. Wakin. Manifold-based signal recovery and parameter estimation from compressive measurements. 2008.

[137] M. B. Wakin. A manifold lifting algorithm for multi-view compressive imaging. In *Proc. Picture Coding Symp. (PCS)*, 2009.

[138] M. B. Wakin, D. Donoho, H. Choi, and R. G. Baraniuk. The multiscale structure of non-differentiable image manifolds. In *Proc. Wavelets XI at SPIE Optics and Photonics*, August 2005.

[139] 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. (PCS)*, Beijing, China, April 2006.

[140] M. B. Wakin, J. Y. Park, H. L. Yap, and C. J. Rozell. Concentration of measure for block diagonal measurement matrices. In *Proc. Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*, March 2010.

[141] M. B. Wakin, B. M. Sanandaji, and T. L. Vincent. On the observability of linear systems from random, compressive measurements. *IEEE Conf. Decision and Control CDC)*, 2010.

[142] Y. Wang, J. Ostermann, and Y. Q. Zhang. *Video processing and communications*. Prentice Hall, 2002.

[143] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):560–576, July 2003.

[144] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. Inf. Theory*, 22:1–10, 1976.

[145] H. L. Yap, A. Eftekhari, M. B. Wakin, and C. J. Rozell. The restricted isometry property for block diagonal matrices. In *Proc. Conf. Inf. Sci. and Syst. (CISS)*, March 2011.

[146] H. L. Yap and C. J. Rozell. On the relation between block diagonal matrices and compressive Toeplitz matrices. Technical Report, 2011.

[147] Z. Yu, S. Hoyos, and B. M. Sadler. Mixed-signal parallel compressed sensing and reception for cognitive radio, 2008.

[148] B. Yuequan, J. L Beck, and L. Hui. Compressive sampling for accelerometer signals in structural health monitoring. *Structural Health Monitoring*, 10(3):235–246, 2011.

[149] R. E. Ziemer and W. H. Tranter. *Principles of Communication: Systems, Modulation and Noise*. Wiley, 2009.