

OPTIMIZING SENSING MATRICES FOR COMPRESSED SENSING

A thesis submitted in partial fulfillment of
the requirements for the degree of

Dual Degree: Bachelor and Master of Technology
Electrical Engineering (Communication and Signal Processing)

by

Kotwal Alankar Shashikant
(Roll No. 12D070010)

Under the guidance of
Prof. Ajit Rajwade, CSE, IITB
and
Prof. Rajbabu Velmurugan, EE, IITB



Department of Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
June 2017

Approval

The thesis entitled

OPTIMIZING SENSING MATRICES FOR COMPRESSED SENSING

by

Kotwal Alankar Shashikant

(Roll No. 12D070010)

is approved for the degree of

Dual Degree: Bachelor and Master of Technology

Electrical Engineering (Communication and Signal Processing)

Examiner

Examiner

Guide

Co Guide

Chairman

Date: _____

Place: _____

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Kotwal Alankar Shashikant
12D070010

Date: _____

Place: _____

Dedicated to
the prawns with the people sauce

Acknowledgments

This is not a piece of work by just me. This is a piece of work by everyone who made this happen, and everyone who made me happen.

In this, I owe a debt of gratitude to my parents and my sister for having raised me in an environment fostering creativity and resourcefulness. To my advisers, Prof. Ajit Rajwade and Prof. Rajbabu Velmurugan, I express my sincere thanks for insightful discussions and for always challenging me on a level that allowed for sharpening my understanding of essential concepts.

I would like to thank Prof. Suyash Awate, Prof. Mayank Vahia and Dr. Aniket Sule for giving me my first real taste of academic research and constantly inspiring me to pursue it. I also acknowledge tremendous moral support from my friends for being invaluable companions in moments of doubt. I am especially grateful to Bhumesh Kumar and Sharad Mirani for discussions invaluable towards the progress of this thesis.

And finally, I must thank all the stars in the early universe that exploded, traveled forward in space and time, mixed and mingled, and finally enabled my existence.

October 2016

Kotwal Alankar Shashikant

Abstract

THE elegant mathematical formalism behind signal recovery using compressed sensing principles provides theoretical guarantees on the error between recovered and ground truth signals. These guarantees are provided in terms of conditions involving signal sparsity and some property of the sensing matrix. An attempt, therefore, can be made to find ‘good’ sensing matrices by optimizing upper bounds on recovery error. This thesis is a story of such attempts, spanning general matrices and matrices tailored to very specific applications.

We start off by pointing out that there exist several applications in image processing (like video compressed sensing in the compressive camera described in [14] and color image demosaicing) which require separation of constituent images given measurements in the form of a coded superposition of those images. This is a compressive measurement. Physically practical code patterns in these measurements are non-negative and do not obey the nice coherence properties of other patterns such as Gaussian codes, which can adversely affect reconstruction performance. This is inspiration for us to optimize such patterns for coherence while balancing design complexity and optimality with techniques like patchwise reconstruction and circular shifts. This design rule gets us matrices that perform better than random at high compressions and sparsity.

Inspired by the improvement in reconstruction quality in the coded source separation case, we proceed to designing matrices for another camera. This Coded Aperture Compressive Temporal Imaging system [18, 17], or CACTI, generates compressive measurements by mechanical translation of a coded aperture. We optimize coherence, again, for the sensing matrix induced by this coded aperture. This time, however, our design does not perform better than random.

This leads us to explore the empirical nature and ‘looseness’ of compressed sensing

bounds by calculating, step by step, the difference induced by the chain of inequalities in proofs of these bounds in these two situations, and in general. The results are surprising, and reveal that much needs to be done for the conventional compressed sensing bounds to be useful for matrix design in general.

We then deviate from the video compressed sensing scenario and explore another bound on the reconstruction error that is easier to calculate than bounds based on the RIC (restricted isometry constant) whose computation is known to be NP-hard, and tighter than the ubiquitous coherence bound, with an aim to exploit it for optimization. However, we will see that this bound, as well, has its own quirks that make its use in practical compressed sensing matrix optimization intractable.

Moreover, we also perform a bound looseness analysis for the RIC as in [5]. The conclusion of these efforts is that coherence optimization is problematic not only because of the coherence bound on the RIC, but also the RIC bound itself. These negative results imply that despite the success of previous work in designing sensing matrices based on optimization of a matrix quality factor, one needs to exercise caution in using them for practical sensing matrix design.

We then introduce an alternative paradigm for optimizing sensing matrices that overcomes the looseness of compressed sensing upper bounds using an average case error approach. We show a proof-of-concept design using this paradigm that performs convincingly better than coherence-based design in not only the CACTI case, but also for general matrices.

To present a complete story of the design process for these cameras, we note that errors are induced by in coded apertures by the manufacturing process. We show how, by taking into account these errors, we can recover input signals and true coded aperture values.

Keywords – video compressed sensing, source separation, sensing matrices, coherence, optimization, error bounds, sparsity measures, alternative error bounds, restricted isometry, coded aperture, circular shifts, temporal imaging, CACTI, looseness of bound, matrix perturbation, quantization, support prior

Contents

Abstract	iii
List of Figures	ix
1 Introduction	1
2 Preliminaries	5
2.1 Compressed sensing	5
2.1.1 Motivation	5
2.1.2 General framework	6
2.1.3 Reconstruction methods	7
2.1.4 Theoretical guarantees	9
2.2 Source separation	10
2.2.1 The framework	10
2.2.2 Theoretical guarantees	11
2.3 Optimizing compressed sensing – previous work	11
2.3.1 Coherence minimization via the Gram matrix	11
2.3.2 Coherence Minimization via rank-1 approximation	13
2.3.3 Information-theoretic methods	13
2.3.4 Other methods and applications	14
3 Optimizing Coded Source Separation	15
3.1 Sensing framework	15
3.2 Our approach to reconstruction	17
3.3 Optimizing codes for source separation	17
3.3.1 Track I: Direct coherence minimization	17

3.3.2	Track II: Including circular shifts	19
3.3.3	Track III: Optimizing bounds tighter than coherence	21
3.4	Experiments and results	22
3.4.1	Validating our framework	22
3.4.2	Demosaicing	25
3.4.3	Coherence minimization	27
4	Optimizing for the CACTI Camera	35
4.1	Introducing the CACTI camera	35
4.2	Optimizing codes	36
4.3	Experiments and results	37
4.3.1	Simulated data	37
5	Limitations of Coherence-based Bounds	43
5.1	The bound	43
5.2	Empirical analysis of the bound	45
5.2.1	General sensing matrices	46
5.2.2	In coded source separation	47
5.2.3	In the CACTI camera	48
5.3	Discussion	51
6	Alternate Compressed Sensing Bounds	59
6.1	A new bound	59
6.1.1	An l_∞ error-based sparsity criterion	59
6.2	Is this bound feasible to optimize on?	60
7	Et tu, RIC?	65
8	The Average Case: A Proof of Concept	69
8.1	General sensing matrices	70
8.2	In the CACTI camera	72
9	Practical Constraints in Compressed Sensing Design	75
9.1	Quantization errors	75
9.1.1	Effects on coded source separation	76

9.1.2	Effects on the CACTI camera	79
9.2	Calibration	79
9.2.1	Online calibration	83
9.2.2	Offline calibration	84
9.2.3	Calibration results	84
10	Conclusion and Future Work	87
10.1	Takeaways	87
10.2	Future work	88
Appendices		91
A	Derivation of coherence expressions	93
B	Derivation of coherence derivatives	95

List of Figures

2.1	Example image for the sparsity analysis in Fig. 2.2	6
2.2	Plot of DCT coefficients for the image in Fig. 2.1	7
2.3	Shrinking function in [9]	12
3.1	Measurement Model	16
3.2	Motivation behind circularly-shifted optimization	20
3.3	Synthetic image results. Left: input images, Right: reconstructions . . .	23
3.4	Real images, Gaussian matrices. Left: input images, right: reconstructions	23
3.5	Real images, uniform matrices. Left: input images, right: reconstructions .	24
3.6	Separating three images, uniform matrices. Up: input images, down: reconstructions	24
3.7	Sudden change, uniform matrices. Left: input images, right: reconstructions	25
3.8	Average relative root mean square errors in our scheme as a function of s and T with positive random matrices	25
3.9	Demosaicing. Left: inputs, middle, right: reconstructions with {random, non-circularly designed} matrices	26
3.10	Demosaicing close-ups, examples {1, 2, 3}. Clockwise: inputs, reconstructions with {random, {circularly, non-circularly} designed} matrices . . .	27
3.11	Distribution of coherences for 8×8 random positive codes as a function of T	28
3.12	Typical coherence decrease profile	28
3.13	Optimized output from combining six not necessarily close images. Left: inputs, middle, right: reconstructions with {random, non-circularly optimized} matrices	30
3.14	Optimized output from combining two close images. Left: inputs, middle, right: reconstructions with {random, non-circularly optimized} matrices .	31

3.15	Error map for optimized codes as a function of s and T	31
3.16	Average RRMSE using random codes minus average RRMSE using optimized codes, as a function of s and T	32
3.17	Left to right: Circularly-shifted coherence histograms for {random, non-circularly optimized, circularly optimized} matrices	32
3.18	Circularly optimized output from combining two close images. Left to right: reconstructions with {random, circularly optimized} matrices	32
3.19	Close-ups showing subtle texture preservation with optimized matrices, example 1. Left to right: inputs, reconstructions with {random, non-circularly optimized, circularly optimized} matrices	33
3.20	Close-ups showing subtle texture preservation with optimized matrices, example 2. Left to right: inputs, reconstructions with {random, non-circularly optimized, circularly optimized} matrices	33
4.1	Average RRMSE as a function of sparsity for 8×8 signals, sparse in 2D DCT, combined with $T = 2$. Permutations: [5, 3; 6, 8]. Errors for low sparsity are near-zero.	38
4.2	Average RRMSE as a function of sparsity for 8×8 signals, sparse in 2D DCT, combined with $T = 4$. Permutations: [7, 8; 2, 8; 6, 1; 3, 5]	38
4.3	Average RRMSE as a function of sparsity for 8×8 signals, sparse in 2D DCT, combined with $T = 6$. Permutations: [6, 7; 3, 6; 6, 2; 1, 4; 8, 3; 5, 2]	39
4.4	Absolute maximum restricted eigenvalue boxplot as a function of sparsity for 8×8 signals, sparse in 2D DCT, with $T = 2$. Permutations: [5, 3; 6, 8]	40
4.5	Absolute maximum restricted eigenvalue boxplot as a function of sparsity for 8×8 signals, sparse in 2D DCT, with $T = 4$. Permutations: [7, 8; 2, 8; 6, 1; 3, 5]	40
4.6	Absolute maximum restricted eigenvalue boxplot as a function of sparsity for 8×8 signals, sparse in 2D DCT, with $T = 6$. Permutations: [6, 7; 3, 6; 6, 2; 1, 4; 8, 3; 5, 2]	41
5.1	Relative difference between inequalities in the error bound above for a 499×500 matrix drawn from a standard normal distribution	47

5.2	Relative difference between inequalities in the error bound above for a 250×500 matrix drawn from a standard normal distribution	48
5.3	Relative difference between inequalities in the error bound above for a 167×500 matrix drawn from a standard normal distribution	48
5.4	Relative difference between inequalities in the error bound above for a 125×500 matrix drawn from a standard normal distribution	49
5.5	Relative difference between inequalities in the error bound above for a 50×500 matrix drawn from a standard normal distribution	49
5.6	Relative difference between inequalities in the error bound above for 8×8 random positive codes in the coded source separation framework for $T = 2$	50
5.7	Relative difference between inequalities in the error bound above for 8×8 random positive codes in the coded source separation framework for $T = 3$	50
5.8	Relative difference between inequalities in the error bound above for 8×8 random positive codes in the coded source separation framework for $T = 4$	51
5.9	Relative difference between inequalities in the error bound above for 8×8 random positive codes in the coded source separation framework for $T = 5$	51
5.10	Relative difference between inequalities in the error bound above for 8×8 random positive codes in the coded source separation framework for $T = 6$	52
5.11	Relative difference between inequalities in the error bound above for 8×8 random positive codes in the CACTI camera for $T = 2$. Permutations: [3, 1; 5, 5]	52
5.12	Relative difference between inequalities in the error bound above for 8×8 random positive codes in the CACTI camera for $T = 3$. Permutations: [7, 7; 4, 4; 7, 3]	53
5.13	Relative difference between inequalities in the error bound above for 8×8 random positive codes in the CACTI camera for $T = 4$. Permutations: [1, 2; 6, 6; 7, 3; 2, 7]	53
5.14	Relative difference between inequalities in the error bound above for 8×8 random positive codes in the CACTI camera for $T = 5$. Permutations: [7, 6; 4, 8; 4, 2; 6, 5; 2, 3]	54

5.15	Relative difference between inequalities in the error bound above for 8×8 random positive codes in the CACTI camera for $T = 6$. Permutations: [1, 8; 8, 7; 6, 1; 4, 6; 7, 8; 5, 3]	54
5.16	Relative difference between inequalities in the error bound above for 8×8 designed positive codes in the CACTI camera for $T = 2$. Permutations: [5, 3; 6, 8]	55
5.17	Relative difference between inequalities in the error bound above for 8×8 designed positive codes in the CACTI camera for $T = 4$. Permutations: [7, 8; 2, 8; 6, 1; 3, 5]	55
5.18	Relative difference between inequalities in the error bound above for 8×8 designed positive codes in the CACTI camera for $T = 6$. Permutations: [6, 7; 3, 6; 6, 2; 1, 4; 8, 3; 5, 2]	56
5.19	Relative difference between inequalities in the error bound above for 8×8 average-designed positive codes in the CACTI camera for $T = 2$. Permutations: [8, 4; 7, 2]	56
5.20	Relative difference between inequalities in the error bound above for 8×8 average-designed positive codes in the CACTI camera for $T = 4$. Permutations: [3, 1; 1, 7; 6, 3; 8, 1]	57
5.21	Relative difference between inequalities in the error bound above for 8×8 average-designed positive codes in the CACTI camera for $T = 6$. Permutations: [7, 3; 5, 6; 8, 8; 5, 2; 2, 3; 7, 3]	57
6.1	Relative difference between reconstruction error and error bound as a function of sparsity for sparse 50×1 signals, sensed with a 10×50 Gaussian random matrix	61
6.2	Relative difference between reconstruction error and error bound as a function of sparsity for sparse 50×1 signals, sensed with a 25×50 Gaussian random matrix	62
6.3	Relative difference between reconstruction error and error bound as a function of sparsity for sparse 50×1 signals, sensed with a 40×50 Gaussian random matrix	62

6.4	Relative difference between reconstruction error and error bound as a function of sparsity for sparse 50×1 signals, sensed with a 55×50 Gaussian random matrix	63
6.5	Relative difference between reconstruction error and error bound as a function of sparsity for sparse 50×1 signals, sensed with a 70×50 Gaussian random matrix	63
6.6	Relative difference between reconstruction error and error bound as a function of sparsity for sparse 50×1 signals, sensed with a 85×50 Gaussian random matrix	64
7.1	Relative difference between reconstruction error and the error bound in Eq. 7.1 for sparse 550×1 signals with $k = 2$, sensed with a 275×550 Gaussian random matrix	66
7.2	Relative difference between reconstruction error and the error bound in Eq. 7.1 for sparse 550×1 signals with $k = 2$, sensed with a 367×550 Gaussian random matrix	66
7.3	Relative difference between reconstruction error and the error bound in Eq. 7.1 for sparse 550×1 signals with $k = 2$, sensed with a 412×550 Gaussian random matrix	67
7.4	Relative difference between reconstruction error and the error bound in Eq. 7.1 for sparse 550×1 signals with $k = 2$, sensed with a 549×550 Gaussian random matrix	67
7.5	Relative difference between reconstruction error and the error bound in Eq. 7.1 for sparse 550×1 signals with $k = 3$, sensed with a 549×550 Gaussian random matrix	68
8.1	Comparison of sum of squared errors on a dataset of 250 sparse vectors, between 8×50 general sensing matrices optimized with mean square error, optimized with coherence and a general random matrix, targeted at sparsities of 0.08	71

8.2	Comparison of sum of squared errors on a dataset of 250 sparse vectors, between 12×50 general sensing matrices optimized with mean square error, optimized with coherence and a general random matrix, targeted at sparsities of 0.12	71
8.3	Comparison of sum of squared errors on a dataset of 250 sparse vectors, between 25×50 general sensing matrices optimized with mean square error, optimized with coherence and a general random matrix, targeted at sparsities of 0.2	72
8.4	Comparison of sum of squared errors on a dataset of 250 sparse vectors, using 8×8 positive codes with $T = 2$, optimized using the mean square error, optimized with coherence and a positive random code, targeted at sparsities of 0.2	73
8.5	Comparison of sum of squared errors on a dataset of 250 sparse vectors, using 8×8 positive codes with $T = 4$, optimized using the mean square error, optimized with coherence and a positive random code, targeted at sparsities of 0.12	73
8.6	Comparison of sum of squared errors on a dataset of 250 sparse vectors, using 8×8 positive codes with $T = 6$, optimized using the mean square error, optimized with coherence and a positive random code, targeted at sparsities of 0.08	74
9.1	Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 2$ in the coded source separation scenario. Code elements rounded off to the nearest 0.2	76
9.2	Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 4$ in the coded source separation scenario. Code elements rounded off to the nearest 0.2	77
9.3	Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 6$ in the coded source separation scenario. Code elements rounded off to the nearest 0.2	77

9.4	Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 2$ in the coded source separation scenario. Code elements rounded off to binary	78
9.5	Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 4$ in the coded source separation scenario. Code elements rounded off to binary	78
9.6	Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 6$ in the coded source separation scenario. Code elements rounded off to binary	79
9.7	Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 2$. Code elements rounded off to the nearest 0.2	80
9.8	Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 4$. Code elements rounded off to the nearest 0.2	80
9.9	Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 6$. Code elements rounded off to the nearest 0.2	81
9.10	Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 2$. Code elements rounded off to binary	81
9.11	Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 4$. Code elements rounded off to binary	82
9.12	Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 6$. Code elements rounded off to binary	82

9.13	Average RRMSE for calibrated and uncalibrated matrices as a function of matrix noise standard deviation for 8×8 signals, sparse in 2D DCT, combined with $T = 2$. Permutations: [5, 3; 6, 8]	85
9.14	Average RRMSE for calibrated and uncalibrated matrices as a function of matrix noise standard deviation for 8×8 signals, sparse in 2D DCT, combined with $T = 4$. Permutations: [7, 8; 2, 8; 6, 1; 3, 5]	85
9.15	Average RRMSE for calibrated and uncalibrated matrices as a function of matrix noise standard deviation for 8×8 signals, sparse in 2D DCT, combined with $T = 6$. Permutations: [6, 7; 3, 6; 6, 2; 1, 4; 8, 3; 5, 2]	86

Chapter 1

Introduction

RANDOM is the usual paradigm for sensing matrices in compressed sensing. This is because matrices drawn from distributions like the Gaussian or the Bernoulli have excellent properties at reconstructing sparse signals with high probability [10]. The question to be asked next, therefore, is whether we can do better than random. Do there exist principled and mathematically founded ways to find matrices that are ‘optimal’, in some sense, for recovery using compressed sensing methods?

Here is where the strong theoretical foundations beneath sparse recovery come to our aid. They express, in terms of some property of the sensing matrix and input signal sparsity, the ability of recovery algorithms to reconstruct the input signal from compressive measurements. The problem is then reduced to finding matrices which recover best, given some error criterion. The traditional error criterion is the l_2 error between recovered and true vectors, and the traditional sparsity criterion is the l_0 norm.

The bound that arises out of sparse recovery analysis involves a quantity called the restricted isometry constant (RIC) of a matrix, essentially a measure of how well columns of a matrix can be expressed as sparse combinations of each other. This is a notoriously difficult quantity to calculate: the s^{th} RIC requires a listing of s -tuples of columns chosen from n columns. This makes the complexity of calculating the RIC exponential in s . This quantity, therefore, is a difficult one to compute. One, therefore, trades complexity off for tightness of bound by upper-bounding the RIC using the coherence of a matrix. This quantity is an easy one to calculate and optimize, and can be capitalized on for our purposes.

The looseness of the coherence bound, however, is the source of much evil. Therefore, it makes sense to find measures of goodness of a sensing matrix yielding bounds tighter than coherence and to try to optimize sensing matrices in the framework constructed by these measures. As we will see, some of these measures attempt to change the error criterion and sparsity measure. The trade-off between tightness and complexity makes an appearance here as well. It, therefore, makes sense to examine, on simulated data, the tightness of the bound as it propagates across various steps of the inequalities that lead to it. An attempt must, also, be made to compare the performance of the coherence bound and that of the alternative bound. A realization that the coherence is a loose worst case bound then leads us to investigate if the average case error is better as an optimization criterion.

To complete the story of sensing matrix design, we need to take into account deviations of the mask used for sensing from the optimal, designed mask: these include quantization and manufacturing defects.

The rest of this thesis deals with these issues as follows:

1. **Preliminaries** reviews basic concepts in compressed sensing, from the qualitative idea to theoretical guarantees and introduces the source separation problem. It then introduces and critiques current methods for sensing matrix optimization.
2. **Optimizing Coded Source Separation** reviews some past work in using compressed sensing for video data, and covers our approach to optimizing it. It then presents some results demonstrating the superiority of designed matrices over random in this case.
3. **Optimizing for the CACTI Camera** introduces the sensing scheme used by the coded aperture compressive temporal imaging system introduced in [18, 17], and presents results on (and the failures of) extending the optimization scheme for coded source separation to CACTI.
4. **Limitations of Coherence-based Bounds** presents a discussion on the failures of the method in the previous chapter, and an empirical analysis of a compressed sensing bound.

5. **Alternate Compressed Sensing Bounds** summarizes an effort to optimize matrices using a different bound, in an effort to mitigate the limitations of coherence, and discovers instead some limitations of the new bound.
6. **Et tu, RIC?** presents an analysis of the RIC bound in [5].
7. **The average case: A proof of concept** provides an alternate paradigm for designing sensing matrices more efficiently, focussing on the conclusions of the previous chapters, and shows a proof of concept design using this paradigm.
8. **Practical Constraints in Matrix Design** details a calibration method to get around manufacturing and quantization defects.
9. **Conclusion and Future Work** briefs upon the take-aways from this work and on scope for future work in this area.

Chapter 2

Preliminaries

2.1 Compressed sensing

2.1.1 Motivation

One of the fundamental and preliminary problems in the typical signal processing pipeline is the discrete representation of continuous-time signals. A general continuous-time signal has infinite degrees of freedom – at each point on the domain of the function, we are free to choose any value in the range of the function. A discrete representation, therefore, does not preserve all the information in the signal. However, we cannot use continuous information – such a representation would take up infinite space and computation time.

However, most signals we find in everyday life aren't completely random. There is often an underlying structure to them, and we don't need all the infinite degrees to represent the signal. For instance, a fundamental result, the Nyquist-Shannon sampling theorem, says that if the signal is bandlimited (limited in frequency in the spectral domain), a discrete representation spaced at half the minimum period in the spectrum of the signal uniquely determines the signal. For a general bandlimited signal, it can be shown that we can't do any better.

Natural signals, however, have more structure than bandlimitedness. Natural images, for instance, are known to be sparse in spectral domains like the discrete Fourier and cosine transforms. Among the (bounded) set of frequencies in these signals, only few have any significant contribution to the signal energy. The image in Fig. 2.1, for instance, has a DCT spectrum shown in Fig. 2.2. Note that among the 4×10^4 coefficients plotted, only



Figure 2.1: Example image for the sparsity analysis in Fig. 2.2

a few are non-zero.

This seems to suggest that we can get away by sensing only those components that contribute any significant energy and still achieve a good representation of these signals and thus, beat the sampling theorem by exploiting structure.

2.1.2 General framework

We, then, must equip ourselves to sample only some coefficients of that spectral domain that sparsifies a given signal. Since the spectral transforms are linear functions of the input signal, it is sufficient to consider linear combinations of signal elements.

Thus, our general sensing framework, for obtaining the measurement y from the inherent signal x , given a sensing matrix (that dictates the above linear combinations) ϕ is

$$y = \phi x \tag{2.1}$$

The ϕ here is a short, fat matrix because the number of elements in y is less than the number of elements in x due to the compressive nature of the measurement. Now, if the (general) basis ψ sparsifies the signal x , we write

$$y = \phi\psi c = Ac \tag{2.2}$$

where c is the vector of coefficients of the signal x in the basis ψ .

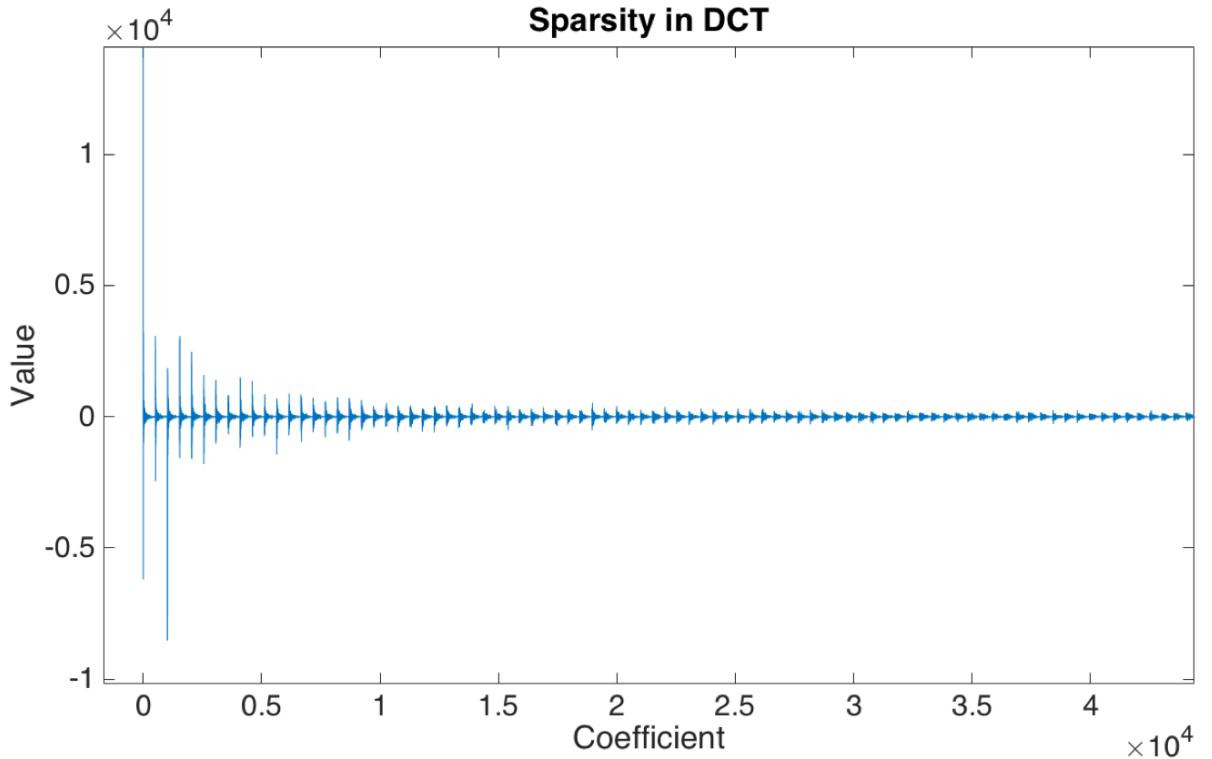


Figure 2.2: Plot of DCT coefficients for the image in Fig. 2.1

2.1.3 Reconstruction methods

The goal, then, is to reconstruct the signal x (equivalently, c) from the compressive measurement y . We formulate the problem as follows: we want the ‘sparsest’ (in ψ) x that satisfies the measurement equation. The definition of sparsity in this context is usually taken to be the l_0 norm of the vector x .

Thus the optimization problem that faces us is

$$\min_c \|c\|_0 \text{ such that } y = Ac \quad (2.3)$$

This optimization problem, however, can be shown [10] to be combinatorial in c – there’s no polynomial time solution to this problem. However, greedy methods can be used to select the support of the vector c and then estimate the coefficients in the support. Examples of such methods are matching pursuit [Alg. 1] and orthogonal matching pursuit [Alg. 2] [4]. These algorithms are summarized in brief for reference:

Often, the recovery problem is often relaxed to an l_p norm optimization problem:

$$\min_c \|c\|_p \text{ such that } y = Ac \quad (2.4)$$

Data: Signal: $\mathcal{Y}(x)$, dictionary \mathcal{D}

Result: List of coefficients: $(a_n, f_n(x))$.

Initialization

$$R_1(x) \leftarrow \mathcal{Y}(x)$$

$$n \leftarrow 1;$$

while $\|R_n(x)\| < \text{threshold}$ **do**

$$f_n(x) \leftarrow \arg \max_{f_i(x) \in \mathcal{D}} \|R_n(x) - f_i(x)\|$$

$$a_n \leftarrow \|R_n(x) - f_n(x)\|$$

$$R_{n+1}(x) \leftarrow R_n(x) - a_n f_n(x)$$

$$n \leftarrow n + 1$$

end

Algorithm 1: Matching Pursuit

Data: Signal: $\mathcal{Y}(x)$, dictionary \mathcal{D}

Result: List of coefficients: $(a_n, f_n(x))$.

Initialization

$$R_1(x) \leftarrow \mathcal{Y}(x)$$

$$n \leftarrow 1;$$

$$\mathcal{S} \leftarrow \emptyset$$

while $\|R_n(x)\| < \text{threshold}$ **do**

$$f_n(x) \leftarrow \arg \max_{f_i(x) \in \mathcal{D}} \|R_n(x) - f_i(x)\| \quad \mathcal{S} = \mathcal{S} \cup f_n(x)$$

$$\mathbf{a} \leftarrow \arg \min_{w \in \mathbb{R}^k} \|\mathcal{Y}(x) - \sum_{f_i(x) \in \mathcal{S}} w_i f_i(x)\|$$

$$R_{n+1}(x) \leftarrow \mathcal{Y}(x) - \sum a_n f_n(x)$$

$$n \leftarrow n + 1$$

end

Algorithm 2: Orthogonal Matching Pursuit

A common choice for p in the above is 1, because that convexifies the problem while still promoting sparsity. The optimization problem with $p = 1$ is known as basis pursuit.

The noisy case can be handled in a similar manner, by changing the constraint:

$$\min_c \|c\|_p \text{ such that } \|y - Ac\|_2 \leq \epsilon \quad (2.5)$$

2.1.4 Theoretical guarantees

l_0 optimization

Suppose we found some method of performing the minimization in Eq. 2.3. Under what conditions would an s -sparse vector c be accurately recovered by solving Eq. 2.3?

To answer this, assume that $y = Ac_1$. Now, for any c_2 that is s -sparse, $c_1 - c_2$ is $2s$ -sparse. Therefore, if $y = Ac_2$, we must have $A(c_1 - c_2) = 0$. If c_1 is to be the unique solution to Eq. 2.3, we must have $c_1 = c_2$, and therefore, cannot have any linearly-dependent subset of $2s$ columns in A . This can be extended to the noisy case [10].

l_1 optimization

A number of properties of the sensing matrix have been used [10] to derive reconstruction error bounds on the matrix A . We mention a couple of these that will be useful further.

Let us assume, for the purposes of this section, that the $k \times N$ matrix A has l_2 -normalized columns. Then, the coherence μ of the matrix A is defined as

$$\mu = \min_{1 \leq i \neq j \leq N} \langle a_i, a_j \rangle \quad (2.6)$$

Further, the s^{th} restricted isometry constant δ_s of the matrix A is defined as

$$\delta_s(A) = \max_{S \in \{1, \dots, N\}, \text{card}(S) \leq s} \lambda_{\max}(A_S^T A_S - I) \quad (2.7)$$

where A_S is the restriction of the columns of the matrix A to the subset S of the set $[N]$ of numbers from 1 to N .

It can be shown [10] that if the $2s^{\text{th}}$ restricted isometry constant $\delta_{2s} \leq 4/\sqrt{41}$, then the solution c^* of 2.4 with $p = 1$ approximates the inherent, nearly s -sparse c within an error bound determined by δ_{2s} :

$$\|c - c^*\|_1 \leq Lc^\# + M\sqrt{s}\epsilon \quad (2.8)$$

$$\|c - c^*\|_2 \leq \frac{L}{\sqrt{s}} c^\# + M\epsilon \quad (2.9)$$

where $c^\#$ is the restriction of c to the largest (in magnitude) s entries of c . L and M are increasing functions of the RIC. This points to the fact that one way of minimizing the reconstruction error for s -sparse signals is to minimize the $2s^{\text{th}}$ RIC. The RIC calculation, however, involves a combinatorial optimization over the subset S of the set $[N]$, and cannot be calculated in polynomial time – and is therefore difficult to optimize.

However, it can be shown that

$$\delta_{2s}(A) \leq (s-1)\mu(A) \quad (2.10)$$

and therefore, a looser, but easier way to reduce errors is to minimize the coherence μ of A . We will find applications of this later.

2.2 Source separation

Source separation is a classical problem in signal processing. It comes in two flavors: one in which both the nature of the signals and the mixing process is unknown (also referred to as blind source separation), and the easier case where the signals are still unknown but the mixing model is known. In the compressed sensing, we control the sensing framework – so when (if) we use the source separation approach, the relevant paradigm is the second, easier one.

2.2.1 The framework

We consider the case in which two sources are combined in some (known) model, with the possible addition of bounded noise. In this case, the measurement model [25] is

$$z = Ax + Be + n \quad (2.11)$$

where A and B are general deterministic dictionaries. For convenience, we assume that they are l_2 -normalized in their columns. The vectors x and e are assumed to be sparse (we have a bit of leeway here: the source x can be approximately sparse as well). The noise n needs no constraint other than $\|n\| \leq \epsilon$, allowing arbitrary bounded noise models.

2.2.2 Theoretical guarantees

Under the assumptions of Eq. 2.11, [25] proves the following about recovery of the vector $w = [x^T e^T]$: if $\|n\|_2 \leq \epsilon$, $\mu_b < \mu_a$ and

$$\|w\|_0 = \|x\|_0 + \|e\|_0 < \max \left\{ \frac{2(1 + \mu_a)}{\mu_a + 2\mu_d + \sqrt{\mu_a^2 + \mu_m^2}}, \frac{1 + \mu_d}{2\mu_d} \right\} \quad (2.12)$$

where μ_a and μ_b are the coherences of the individual dictionaries A and B , μ_m is the cross-coherence of A and B by taking pairs of columns, one from A and one from B , and μ_d is the coherence of the joint dictionary $[A \ B]$, then the solution w^* to the basis pursuit problem formed by this measurement model satisfies, in relation to the true w ,

$$\|w - w^*\|_2 \leq C(\epsilon + \eta) + D\|w - w_{\mathcal{W}}\|_1 \quad (2.13)$$

where $w_{\mathcal{W}}$ is w restricted to the top $\|w\|_0$ elements and C and D are non-negative constants depending on the matrix coherence and signal sparsity. It is this theoretical guarantee that our sensing and recovery framework rely upon for coded source separation.

2.3 Optimizing compressed sensing – previous work

It can be verified that the constants on the right hand sides of the error bounds in Eq. 2.9 and 2.13 are increasing functions of the RIC. Upper bounds on these constants in terms of the coherence, therefore, are increasing functions of the coherence. A straightforward strategy to optimize matrices, therefore, is to reduce an upper bound on the right hand sides by reducing coherence.

Given a sparsifying basis ψ , then, it is necessary to construct an ‘optimal’ sensing matrix ϕ such that the coherence of the effective dictionary $A = \phi\psi$ is low. Most previous work and our first method do this in terms of $\mu(D)$.

2.3.1 Coherence minimization via the Gram matrix

One way to look at the coherence is [9] to look at the absolute maximum non-diagonal element of $G = D^T D$. The goal is to reduce the magnitudes of the non-diagonal elements. [9] tries to minimize the following function, with a parameter t :

$$\mu_t(D) = \frac{\sum_{i \neq j} (|g_{ij}| > t) |g_{ij}|}{\sum_{i \neq j} (|g_{ij}| > t)} \quad (2.14)$$

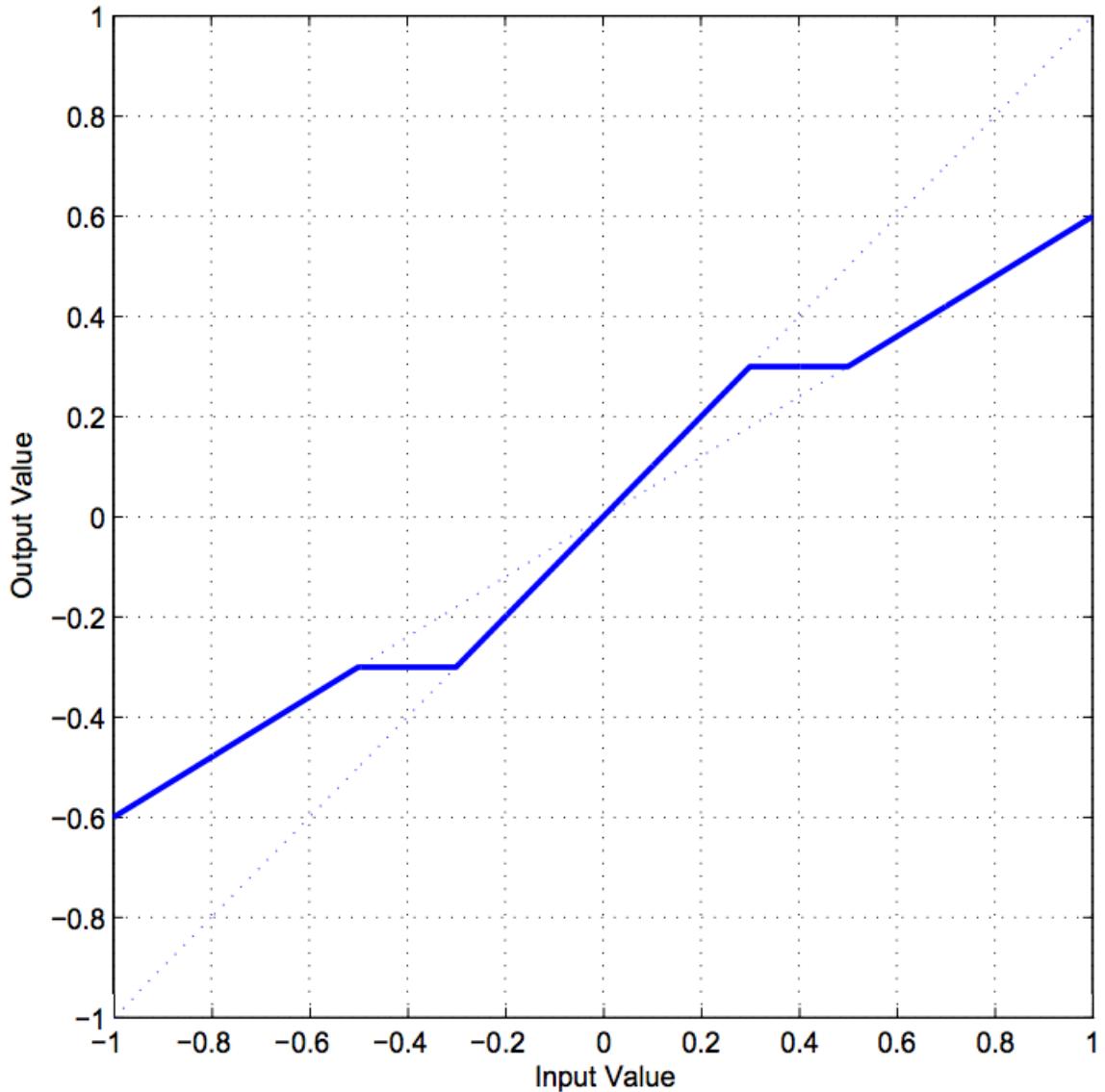


Figure 2.3: Shrinking function in [9]

This is an absolute average of off-diagonal Gram matrix entries above t . To achieve this, [9] processes the entries of the Gram matrix with a ‘shrinking’ function Fig. 2.3, forces the shrunk Gram matrix to be low-rank to get a ‘new’ Gram matrix, and builds the square root of this matrix to obtain the updated dictionary.

However, this method gives no guarantees on whether the actual maximum value decreases or not (notice the method minimizes the *average* value of off-diagonal elements above t). Also, the square-root step involves an assumption that the input matrix is positive semi-definite, which is not always the case. When it is not, one needs to force the offending eigenvalues to zero. Guarantees on whether coherence decreases across these

iterations don't exist.

2.3.2 Coherence Minimization via rank-1 approximation

An equivalent way to look at the problem is making the columns of D as ‘orthogonal’ to each other as possible. This implies that the Gram matrix G should be as close to the identity matrix as possible. [8] solves the problem of estimating ϕ given ψ this way ([8] also solves the problem of estimating both jointly from sample signals, but that is not applicable in the general video scenario). Knowing that we need $G = \psi^T \phi^T \phi \psi \approx I$, $\psi \psi^T \phi^T \phi \psi \psi^T \approx \psi \psi^T$. With $\psi \psi^T = V \Lambda V^T$ and $\phi V = \Gamma$, we need $\Lambda \Gamma^T \Gamma \Lambda \approx \Lambda$. So we solve

$$\min_{\Gamma} \|\Lambda \Gamma^T \Gamma \Lambda - \Lambda\|_F \quad (2.15)$$

This can be written as

$$\min_{\Gamma} \left\| \Lambda - \sum_i \nu_i \nu_i^T \right\|_F = \min_{\Gamma} \left\| \Lambda - \sum_{i,i \neq j} \nu_i \nu_i^T - \nu_j \nu_j^T \right\|_F \quad (2.16)$$

where ν_i is the i^{th} column of $\Lambda \Gamma^T$. This, however, is a rank-1 approximation problem which can be solved non-iteratively with the singular value decomposition of $\Lambda - \sum_{i,i \neq j} \nu_i \nu_i^T$. We do this by initializing $\Lambda \Gamma^T$ to a random matrix and successively optimizing for all j . This in turn yields Γ , and therefore ϕ .

The paper however does not proceed with the SVD of E , which is the natural thing to do as per the Eckart-Young theorem. Instead it proceeds with eigenvalue decomposition, which makes the technique less rigorous since it is no longer guaranteed to be positive semi-definite, and this requires the negative eigenvalues to be set to 0. Also, this method minimizes some appropriate average of the Gram matrix elements and therefore isn't guaranteed to minimize the maximum of off-diagonal entries.

2.3.3 Information-theoretic methods

Some authors have taken an information-theoretic route to the problem [6, 24, 30]. These papers design sensing matrices ϕ such that the mutual information between a set of small patches $\{x_i\}_{i=1}^n$ and their corresponding projections $\{y_i\}_{i=1}^n$ where $Y_i = \phi x_i$, is maximized. The minimum mean square error (MMSE), it can be shown, is lower bounded by a quantity decreasing in this mutual information. Maximizing this mutual information,

therefore, gives the MMSE more leeway to decrease. However, this gives no upper bound on MMSE and nothing prevents it from being much larger. Besides, computing this mutual information first requires estimation of the probability density function of X and hence Y using Gaussian mixture models, for instance. This can be expensive and is an iterative process. Moreover these learned GMMs for a class of patches may not be general enough to ably represent patches of other classes.

2.3.4 Other methods and applications

There have been an array of other efforts for compressed sensing design recently, all using the coherence as a goodness criterion for sensing matrices. For instance, [20] designs an optimal energy-preserving sensing matrix for Poisson compressed sensing, where the optimizing criterion is the coherence directly. [1] uses a method similar to [8] for optimizing general sensing matrices for coherence with gradient descent. In [23] is a method to design sensing matrices maximally incoherent with the sparsifying orthogonal basis. [22] applies coherence minimization to design structured matrices for the Coded Aperture Snapshot Spectral Imaging (CASSI) system [11, 29]. [3] and [21] apply coherence-based design to environmental sounds and electromagnetic compressed sensing applications respectively.

Chapter 3

Optimizing Coded Source Separation

We now look at how compressed sensing principles may be used for video data. In practical situations, it is easier to combine video frames across time than to combine frames across both space and time, which would have been superior. However, we find that the right linear combination of video frames gives good reconstruction results. Our final aim is to develop a framework and a set of codes that provide for optimal reconstruction on video data compressed across time.

3.1 Sensing framework

Linear coded combinations of input frames were exploited in implementation in [14], where T vectorized input frames $\{x_i\}_{i=1}^T$ are sensed so that the vectorized output y appears as a coded combination (dictated by the ‘sensing matrices’ ϕ_i) of the inputs. The sensing framework (depicted in Fig. 3.1) is

$$y = \sum_{i=1}^T \phi_i x_i \quad (3.1)$$

Since the output of this operation is a coded combination, this sensing framework constrains the ϕ_i to be a diagonal matrix, with the code elements on the diagonal.

The sparsifying basis here is a 3D dictionary learned on video patches. Given this dictionary, called D , any given signal x , and in particular, its frames $\{x_i\}_{i=1}^T$ can be

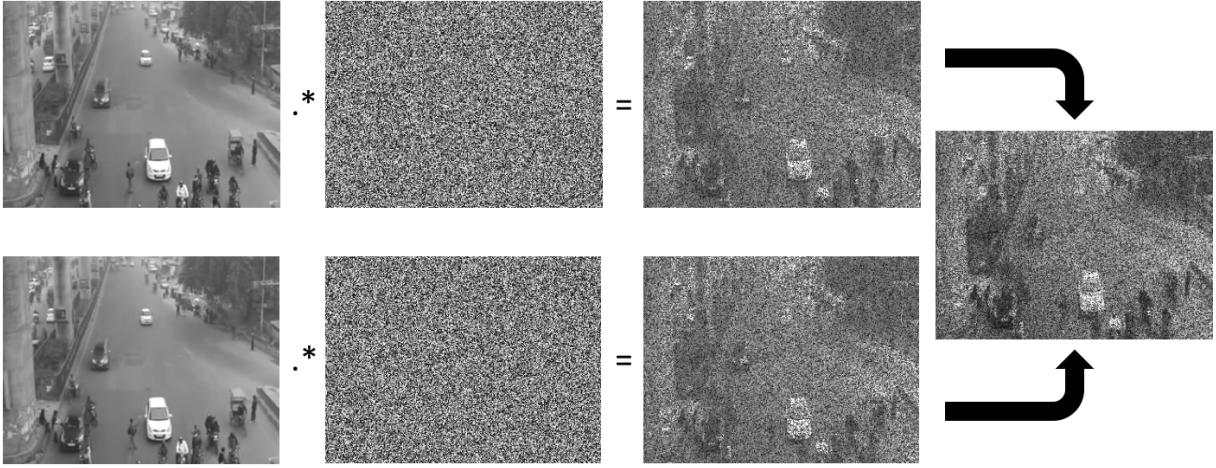


Figure 3.1: Measurement Model

approximately reconstructed as a sum of its projections α_j on the K atoms in D :

$$x_i = \sum_{j=1}^K D_{ji} \alpha_j \quad (3.2)$$

where D_{ji} is the i^{th} frame in the j^{th} 3-D dictionary atom D_{ji} . From the measurements and the dictionary, the input images are recovered solving the following optimization problem:

$$\min_{\alpha} \|\alpha\|_0 \text{ subject to } \left\| y - \sum_{i=1}^T \phi_i \sum_{j=1}^K D_{ji} \alpha_j \right\|_2 \leq \epsilon \quad (3.3)$$

This problem can be approximately solved with sparse recovery techniques like orthogonal matching pursuit [4].

The drawback here, though, is that the 3D dictionary imposes a smoothness assumption on the scene. Since a linear combination of dictionary atoms cannot ‘speed’ an atom up, the typical speeds of objects moving in the video must be roughly the same as the dictionary. Also, because of the nature of the training data, the dictionary fails to sparsely represent sudden scene changes caused by, say, lighting or occlusion. Though it is possible to model a video sequence as a base frame followed by some innovation in the rest of the frames, this representation makes optimizing simpler and performs as well in reconstruction. Also, the remarkable success of sparse recovery makes us safe in modeling consecutive frames independently. Other techniques like [28] exploit additional structure within the signal, like periodicity, rigid motion or analytical motion models and cannot be used in the general video sensing case.

3.2 Our approach to reconstruction

We try relaxing these constraints using a source-separation approach [25], where precise error bounds on the recovery of the images have been derived, with possible improvement using the techniques in [5].

We, therefore, propose to use a recovery method different from the one used in [14], within the same acquisition framework. Thus, our signals are still acquired according to Eq. 3.1. However, the choice of the sparsifying basis is different: we use a DCT basis D to model each frame in the input data. The dictionary Ψ sparsifying the entire video sequence, thus, is a block-diagonal matrix with the $n \times n$ sparsifying basis D on the diagonal. Thus,

$$y = (\phi_1 \ \dots \ \phi_T) (D\alpha_1 \ \dots \ D\alpha_T)^T \quad (3.4)$$

$$= (\phi_1 D \ \dots \ \phi_T D) (\alpha_1 \ \dots \ \alpha_T)^T \quad (3.5)$$

Given a measurement y , we recover the input $\{x_i\}_{i=1}^T$ through the DCT coefficients α by solving the optimization problem

$$\min_{\alpha} \|\alpha\|_1 \text{ subject to } y = \Phi\Psi\alpha, \quad \alpha = (\alpha_1 \ \alpha_2 \ \dots \ \alpha_T)^T \quad (3.6)$$

In our implementation we used the **CVX** [12] solver for solving the convex optimization problem in Eq. 3.6.

We experimented with basis pursuit recovery with Gaussian-random sensing matrices, getting excellent results with no visible ghosting for both similar and radically different images. Unfortunately, the more realizable positive sensing matrices do not have the nice incoherence properties of Gaussian-random matrices, which are sufficient conditions for near-accurate recovery as derived in [25]. At higher compressions and sparsities in input signals, we find, positive matrices do not work as well, making a case for designing good matrices of this kind.

3.3 Optimizing codes for source separation

3.3.1 Track I: Direct coherence minimization

Our aim here is to optimize the sensing matrices ϕ_i directly for minimum coherence with gradient descent. We now calculate gradients of the coherence with respect to the elements

of ϕ_i . As in Eq. 3.5, with an $n \times n$ dictionary D , we have the effective dictionary

$$\Phi\Psi = \begin{pmatrix} \phi_1 D & \phi_2 D & \dots & \phi_T D \end{pmatrix} \quad (3.7)$$

The expression for the coherence of a general dictionary 2.6 contains `max` and `abs` functions that a gradient-based scheme cannot handle. Instead, we soften the `max` and convert the `abs` to a square by using, for large enough θ ,

$$\max_i \{t_i^2\}_{i=1}^n \approx \frac{1}{\theta} \log \sum_{i=1}^n e^{\theta t_i^2} \quad (3.8)$$

The advantage of using the square soft coherence is that lowering β gives us some kind of average coherence, as compared to $\beta = \infty$, where we get the maximum. This enables us to reduce our design method to the methods used by others to minimize some average coherence, as well as gives us the framework to design structured matrices.

We now need to evaluate the coherence of this dictionary as a function of the elements of Φ . We will call the index varying from 1 to T as μ or ν , and the index varying from 1 to n as α , β or γ . The μ^{th} block of Φ is thus ϕ_μ . Let the β^{th} diagonal element of ϕ_μ be $\phi_{\mu\beta}$. Define the α^{th} column of D^T to be d_α . Then, it can be shown [Appendix A] that the normalized dot product between the β^{th} column of the μ^{th} block and the γ^{th} column of the ν^{th} block is

$$M_{\mu\nu}(\beta\gamma) = \frac{\sum_{\alpha=1}^n \phi_{\mu\alpha} \phi_{\nu\alpha} d_\alpha(\beta) d_\alpha(\gamma)}{\sqrt{(\sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta)) (\sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma))}} \quad (3.9)$$

Finally, using the squared soft-max function [Eq. 3.8] to deal with the `max` and the `abs` in the coherence expression, we get the squared soft coherence \mathcal{C} to be

$$\mathcal{C} = \frac{1}{\theta} \log \left[\sum_{\mu=1}^T \sum_{\nu=1}^{\mu-1} \sum_{\beta=1}^n \sum_{\gamma=1}^n e^{\theta M_{\mu\nu}^2(\beta\gamma)} + \sum_{\mu=1}^T \sum_{\beta=1}^n \sum_{\gamma=1}^{\beta-1} e^{\theta M_{\mu\mu}^2(\beta\gamma)} \right] \quad (3.10)$$

In the above, the first term corresponds to all $(\mu > \nu)$ blocks that are ‘below’ the block diagonal. Here, we consider all terms in the given block for the maximum. The second term corresponds to $(\mu = \nu)$ blocks on the block diagonal. Here, we consider only consider $(\beta > \gamma)$ below-diagonal elements for the maximum.

Calculation of coherence derivatives

We note that the \mathcal{C} computed in the section above is a function of Φ . We differentiate \mathcal{C} with respect to $\phi_{\delta\epsilon}$. For this, we define the numerator of the expression for $M_{\mu\nu}(\beta\gamma)$ as

$\chi_{\mu\nu}(\beta\gamma)$ and the denominator as $\xi_{\mu\nu}(\beta\gamma)$. The derivative of the objective function can be found in terms of these quantities. Defining $\uparrow_{\mu\delta}$ to be the Kronecker delta function that is 1 only if $\mu = \delta$, it can be shown [Appendix B]

$$\frac{d\chi_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}} = d_\epsilon(\beta)d_\epsilon(\gamma)(\phi_{\mu\epsilon}\uparrow_{\nu\delta} + \uparrow_{\mu\delta}\phi_{\nu\epsilon}) \quad (3.11)$$

$$\frac{d\xi_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}} = \frac{1}{\xi_{\mu\nu}(\beta\gamma)} \left[\phi_{\mu\epsilon}d_\epsilon^2(\beta)\uparrow_{\mu\delta} \sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma) + \phi_{\nu\epsilon}d_\epsilon^2(\gamma)\uparrow_{\nu\delta} \sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta) \right] \quad (3.12)$$

Using these, we do gradient descent with adaptive step-size and use a multi-start strategy to combat the non-convexity of the problem.

Time complexity and the need for something more

The calculation of coherence for a matrix requires us to evaluate normalized dot products between columns of the matrix. In our case, the size of the matrix is $n \times nT$, and each dot product needs $\mathcal{O}(n)$ operations, warranting the calculation of $\mathcal{O}(n^3T^2)$ quantities. Optimizing this rapidly becomes intractable as n increases. The performance of gradient descent on this non-convex optimization problem also worsens as the dimensionality of the search-space ($\mathcal{O}(nT)$) increases.

Empirically, we observe that it is intractable to design codes that are more than 20×20 in size in any reasonable time. This points to the fact that we need something more to make designing effective codes possible.

3.3.2 Track II: Including circular shifts

The computational intractability of optimizing large codes leads us to designing smaller masks and tiling them to fit the image size we're dealing with. A small coherence for the designed patch guarantees good reconstruction for patches exactly aligned with the code block; however, other patches see a code that is a circular shift of the original code. Fig 3.2 provides a visual explanation. The big outer square denotes the image. On top of the image we show tiled designed codes. Now, the patch in red clearly multiplies with the exact designed code; however the patch in green multiplies with a code shifted in both the coordinates circularly.

This points to designing sensing matrices that have small coherence in all their circular permutations (note that these permutations happen in two dimensions and must

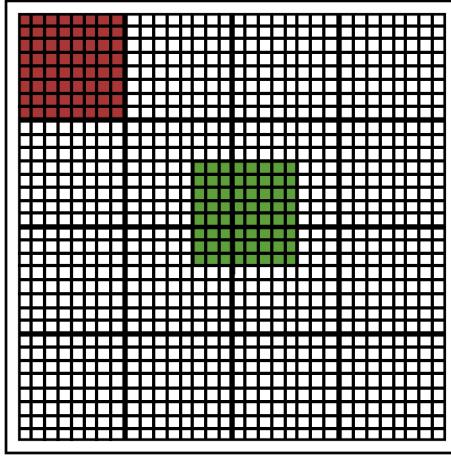


Figure 3.2: Motivation behind circularly-shifted optimization

be handled as such). To this end, we modify the above objective function to minimize the maximum coherence resulting from all circularly-shifted vectorized versions of Φ . We thus have

$$\mathcal{C} = \frac{1}{\theta} \log \left[\sum_{\zeta \in \text{perm}(\Phi)} \left[\sum_{\mu=1}^T \sum_{\nu=1}^{\mu-1} \sum_{\beta=1}^n \sum_{\gamma=1}^n e^{\theta M_{\mu\nu}^{(\zeta)2}(\beta\gamma)} + \sum_{\mu=1}^T \sum_{\beta=1}^n \sum_{\gamma=1}^{\beta-1} e^{\theta M_{\mu\mu}^{(\zeta)2}(\beta\gamma)} \right] \right] \quad (3.13)$$

where $M_{\mu\nu}^{(\zeta)}(\beta\gamma)$ represents the normalized dot product between the β^{th} column of the μ^{th} block and the γ^{th} column of the ν^{th} block, resulting from the instance of the circular permutation ζ of Φ . Derivatives of this expression are found exactly like in Appendix B, except that the μ , ν , β and γ parameters are subjected to the appropriate circular permutation.

The time complexity for determining this maximum coherence among all circular permutations is $\mathcal{O}(n^5 T^2)$, out of which a $\mathcal{O}(n^3 T^2)$ term arises from the calculation of coherence for each circular permutation, and a $\mathcal{O}(n^2)$ arises from the fact that there are n^2 such permutations. The advantage here, though, is that we don't need to optimize masks having very high values of n ; we can do away with keeping n a small constant because the scheme works for any n such that n -sized patches are sparse in the dictionary D . This scheme is, thus, more scalable in terms of the size of the input image. Therefore the effective dimension of the optimization problem in such a scheme is, in terms of the variables that matter, $\mathcal{O}(T^2)$.

It is worth mentioning that this simple idea has been largely ignored in literature concerning sensing matrix optimization. As mentioned in the introduction, previous at-

tempts mostly use an average coherence minimization technique [8, 9, 20] for full-sized sensing matrices, and are not as scalable as ours is for large images because they involve optimization problems in variables whose dimensions are at least of the order of image size. Sensing matrices can be designed at the patch level as well, for instance using information theoretic techniques as in [6, 24, 30], but the methods therein are not designed to account for the issue of overlapping reconstruction. To the best of our knowledge, ours is the first piece of work to handle this important issue in a principled manner.

3.3.3 Track III: Optimizing bounds tighter than coherence

The coherence bound mentioned in Eq. 2.10 is a very pessimistic bound: it arises from applying Geršgorin disk theorem – that bounds the eigenvalues of a matrix in terms of their distance from diagonal elements – to the definition of the RIC as in Eq. 2.7 and approximating the maximum column sum as $(s - 1)$ times the maximum element constituting the sum [10].

Geršgorin radii

Instead, we can try to minimize the maximum Geršgorin radius, achieving a tighter bound than coherence on the RIC. In our framework, then, do the following: given a particular s -cardinality subset S of indices from 1 to nT , we want to evaluate dot products of (normalized) columns of $\Phi\Psi$. Let us call the sensing matrix with normalized columns A . Restricting this to the columns specified by S reduces us to A_S . Note that

$$\begin{aligned} [A_S^T A_S - I]_{ij} &= [A^T A - I]_{S_i S_j} \\ &= M_{\mu\nu}(\beta\gamma) - \mathbf{1}_{S_i = S_j} \end{aligned}$$

where we calculate the μ, ν, β, γ arguments for the M by the appropriate column number: $\mu_i^S = \text{floor}(S_i/n)$ and $\beta_i^S = S_i \bmod n$. Call $M_{\mu_i^S \mu_j^S}(\beta_i^S \beta_j^S)$ as ω_{ij}^S . This is symmetric in the arguments i and j .

We now want to calculate row absolute sums for the matrix $A_S^T A_S - I$. Since by definition $M_{\mu\mu}(\beta\beta) = 1$,

$$\sum_j |\omega_{ij}^S - \mathbf{1}_{S_i = S_j}| = \sum_{j \neq i} |\omega_{ij}^S|$$

Finally, using the square soft-max function, we get the maximum row absolute sum, the Geršgorin radius and our objective function \mathcal{C} to be

$$\mathcal{C}(\Phi) = \frac{1}{\theta} \log \left[\sum_S \sum_i \exp \left\{ \theta \sum_{j \neq i} |\omega_{ij}^S| \right\} \right]$$

Derivatives of this quantity are calculated in a similar way to the coherence function derivatives.

Brauer ellipse bounds

A similar bound to the Geršgorin bound is the Brauer ellipse bound, which bounds the eigenvalue in an ellipse around diagonal elements, instead of circles. This is provably better than the Geršgorin bound, and so can be used to get a tighter bound on the coherence.

However, these optimization problems are combinatorial in the size of the matrices involved and the sparsity one needs to optimize for. These are presented here only as attempts to see if they are feasible. It turns out they aren't.

3.4 Experiments and results

3.4.1 Validating our framework

We start with testing the proposed framework visually. In all such results in this chapter, we show successive frames top-to-bottom, and different types of reconstruction left-to-right. Here, for the sake of saving time, all reconstructions are done in a non-overlapping way. We first use two synthetic images that are known to have very low sparsity. These are 20×20 images, with only 3 out of the 400 DCT coefficients set to non-zero values. The results, with relative root mean errors of the order of 10^{-5} , for these are shown in Fig. 3.3. The results are similar for Gaussian sensing matrices and positive random matrices.

Next, we test on two video frames that are very similar, with Gaussian random matrices. The relative root mean square errors are around 0.0019 for each image. The results are shown in Fig. 3.4.

Next, with positive random diagonal matrices, the relative root mean square errors are around 0.0036 for each image. The results are shown in Fig. 3.5. Looking at these

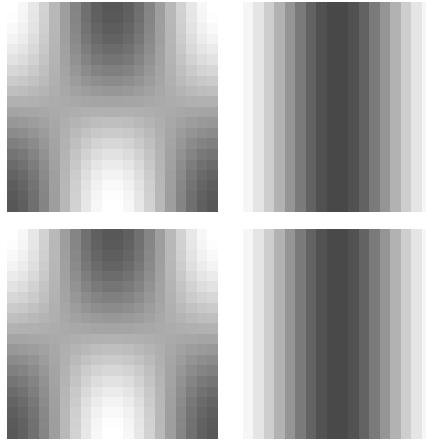


Figure 3.3: Synthetic image results. Left: input images, Right: reconstructions



Figure 3.4: Real images, Gaussian matrices. Left: input images, right: reconstructions

results, one notices that there is very little to no ghosting, that is, appearance of features from one image into the other, in the output images even when the images are very close to each other. This is a very desirable property in any algorithm that separates images from compressed video.

To evaluate how this works for multiple images, we try separating three images with uniform matrices. See Fig. 3.6. Here, we notice ghosting happening in the third frame. However, with better-designed sensing matrices, one can think of getting rid of this effect. The relative root mean square errors here are worse, around 0.005 for each image.

To simulate sudden changes, we run the optimization with two very different input images. We can separate these well, as is shown in Fig. 3.7.



Figure 3.5: Real images, uniform matrices. Left: input images, right: reconstructions



Figure 3.6: Separating three images, uniform matrices. Up: input images, down: reconstructions

We do a numerical comparison between our designed codes and random codes for various values of $s = \|x\|_0/n$ and T . We randomly generate T s -sparse (in 2D DCT) 8×8 signals $\{x_i\}_{i=1}^T$, combine them using random matrices to get y . Average relative root mean square errors on recovering the input signals from y as a function of s and T are shown in Figs. 3.8 and 3.15, and a difference map is shown in Fig. 3.16. Errors are near-zero in the region where both T and s are small, and one can expect reasonable quality reconstructions till $T = 4$ from random matrices. To increase T further, we would



Figure 3.7: Sudden change, uniform matrices. Left: input images, right: reconstructions

need to optimize our sensing matrix appropriately, as is shown further in this paper.

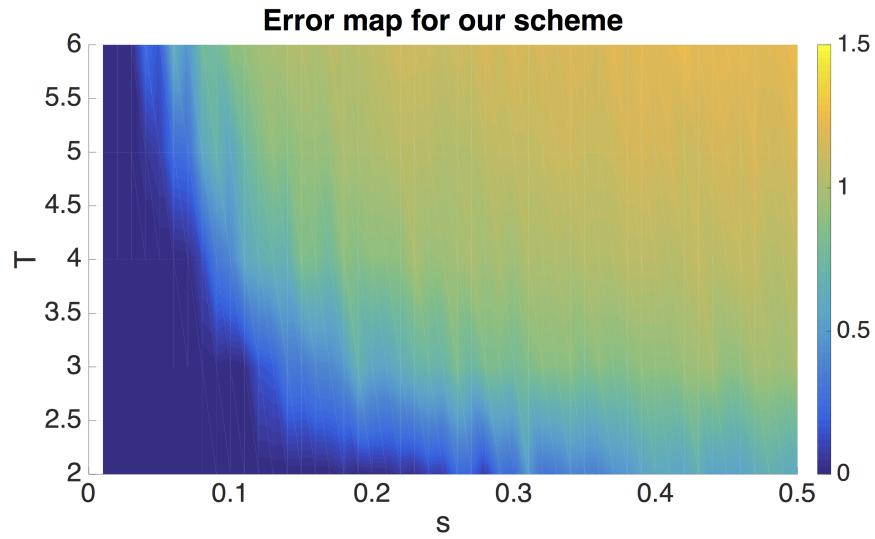


Figure 3.8: Average relative root mean square errors in our scheme as a function of s and T with positive random matrices

3.4.2 Demosaicing

To demonstrate the utility of this scheme, we show results on demosaicing RGB images. The general demosaicing problem involves addressing the difficulty that on a camera sensor, a single pixel can sense only one of the three R, G and B channels. Therefore, raw

camera data needs to be interpolated to recover all the three channels. Traditional approaches to demosaicing involve the use of the Bayer pattern, which tiles a fixed [B, G; G, R] pattern over the image and use variants of algorithms like edge-directed interpolation which are tuned to the Bayer pattern. The Matlab `demosaic` function, for instance, uses [19], which takes a gradient-corrected bilinear interpolated approach. However, recently a case has been made for panchromatic demosaicing [13], where we sense a linear combination of the three channels and use techniques from compressive recovery to reconstruct. However, it turns out that the Bayer pattern has very high mutual coherence, so it is unsuitable for compressive recovery. Here, we propose to design the mosaic patterns by minimizing coherence.

We design 8×8 codes for linearly combining the three channels using our method and visually compare overlapping reconstructions. As Figs. 3.9 and 3.10 show, results from



Figure 3.9: Demosaicing. Left: inputs, middle, right: reconstructions with $\{\text{random, non-circularly designed}\}$ matrices

the designed case are more faithful to the ground-truth than the random reconstructions are. The random reconstructions show (more) color artifacts, especially in areas where the input image varies a lot (car headlights in the top image, around parrot eyes in the bottom). Our designed codes do not show as many color artifacts. The relative root mean square errors don't differ much for these two cases, but subtle details of color are better preserved by our matrices. In Fig. 3.10, notice in the first case the green artifacts near car headlights and the leftmost cyclist in the random reconstruction that is, while that area is better-reconstructed with our matrices. The car headlight area on the car at the right

is also better-reconstructed by our matrices. In the bottom, notice less color artifacts in the densely-varying area near the eye and on the bottom part of the beak.

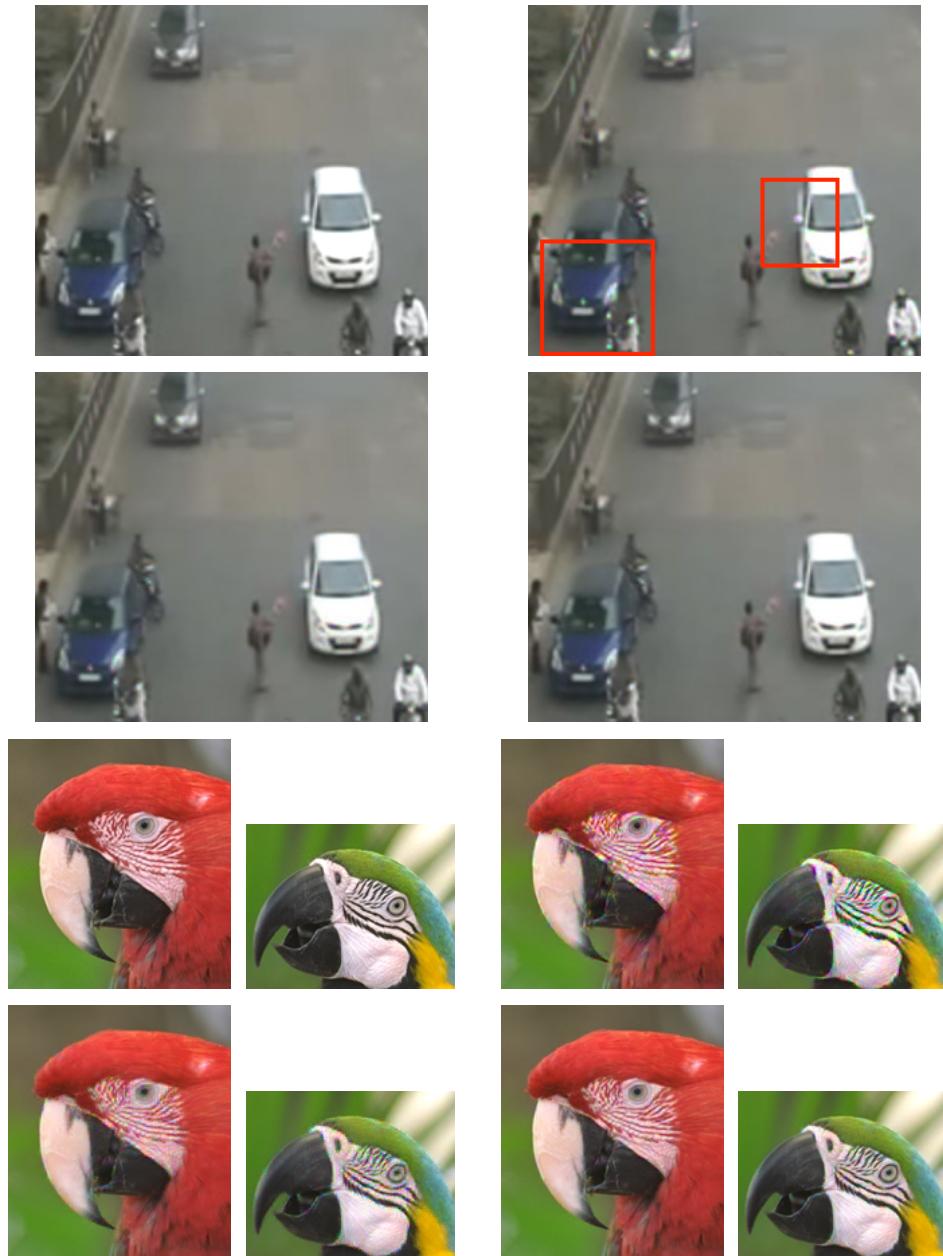


Figure 3.10: Demosaicing close-ups, examples {1, 2, 3}. Clockwise: inputs, reconstructions with {random, {circularly, non-circularly} designed} matrices

3.4.3 Coherence minimization

The coherence of a uniform random matrix of the type we're interested in has a typical value around 0.8 for 8×8 codes. The distribution of these values is shown in the boxplot

in Fig 3.11. The typical profile of descent on coherence from a random initialization is

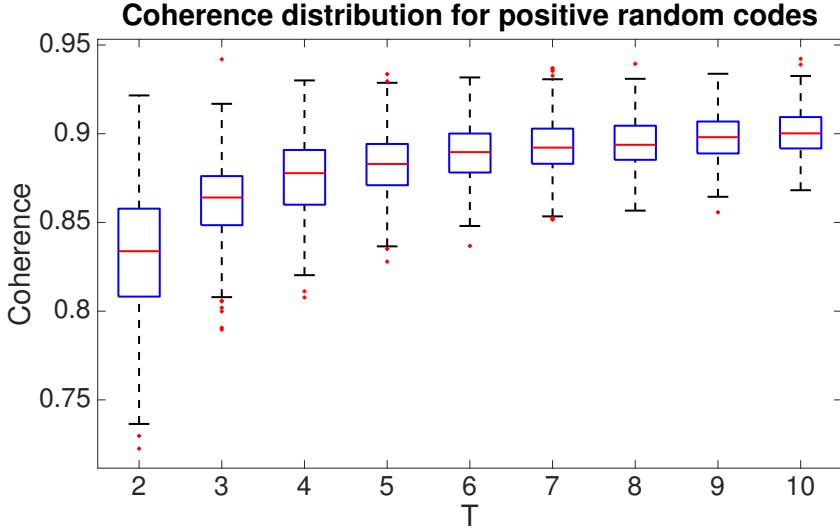


Figure 3.11: Distribution of coherences for 8×8 random positive codes as a function of T

shown in Fig. 3.12.

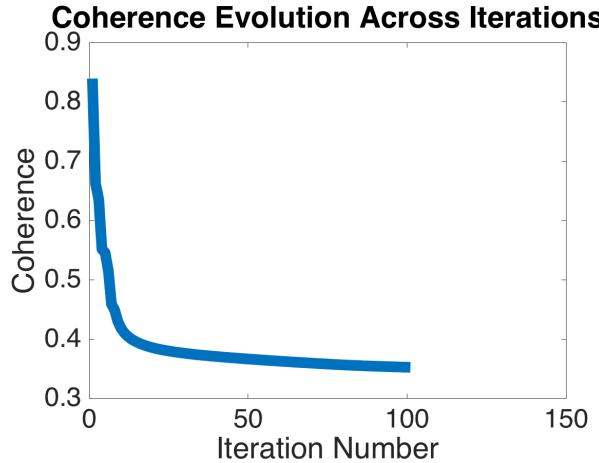


Figure 3.12: Typical coherence decrease profile

The minimum coherence we have been able to achieve in this scheme has been around 0.27 (for $T = 2$). It is interesting to note that all initialization instances lead to coherences (for $T = 2$) of at the most 0.35, and hence empirically yield nearly as good matrices.

We first visually validate that our matrices perform better than positive random matrices. We design 8×8 codes and tile them, reconstructing patchwise with overlapping patches. An example of this running on six not necessarily close frames in a video is shown in Fig. 3.13 (Fig. 3.14 shows an example for $T = 2$). Ghosting artifacts marked out in Fig. 3.13 in white boxes in the random matrix reconstructions are absent or lower in

the designed matrix reconstructions. These outputs show that on the large scale, we do as well as random matrices for low T and better for high T . For a small scale comparison, see Figs. 3.19 and 3.20.

Finally, we do a numerical comparison similar to the one in Fig. 3.8. The resulting error map is shown in Fig. 3.15. On an average, we see that we perform better than the random case. To characterize this, we compute the difference between these two error maps (random minus optimized). This is not very significant numerically, though it does produce significant changes in subtle texture as seen in Figs. 3.19 and 3.20.

Note that though random matrices beat our designed matrices at some low values of s and T , our motivation for optimization was to achieve better performance than random at high s and T , which we comprehensively achieve.

Circularly-symmetric coherence minimization

Again, we design 8×8 codes for $T = 2$. To show coherence improvement between positive random codes, and codes designed with and without circular permutations, we plot the distribution of coherences of $\Phi^{(\zeta)} D$ in Fig. 3.17 for all circular permutations ζ . Note that even though the coherences of non-circularly designed matrices are much lower than positive random matrices, the maximum coherence among all permutations is quite large. The circularly-designed matrices, however, have permuted coherences clustered around a low value. We then expect good reconstruction with all circular permutations, yielding good expected reconstructions for images.

Similar to the above section, we validate our matrices visually. Following the same conventions, here is an output for the $T = 2$ case [Fig. 3.18].

We now look at reconstructions from random and both classes of our designed matrices on a small scale. As a first example, we show a close-up from the car video sequence shown earlier [Fig. 3.19]. Note, to start off, that the reconstruction of the numberplate and headlight area is much clearer in our case than the random matrix case. Further, notice the presence of major ghosting in the random case, especially near the rear-view mirrors, bonnet (marked by arrows) and headlights (marked by boxes), while our reconstructions remain free of these artifacts. Adding circular optimization to the picture further improves image quality especially in the bonnet area, where the non-circular reconstruction is slightly splotchy. Next, in Fig. 3.20, which is a smaller part of the same image, the

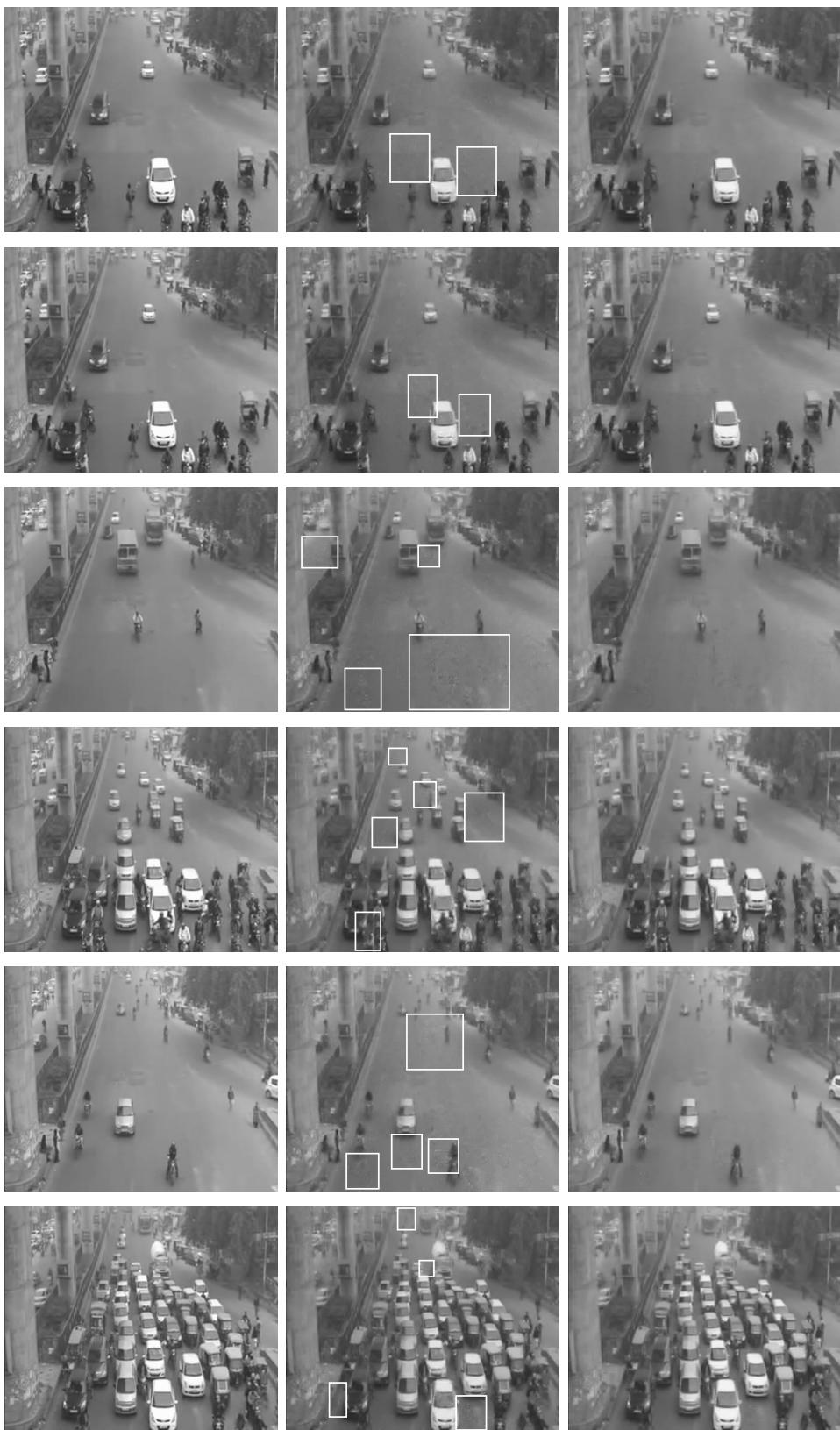


Figure 3.13: Optimized output from combining six not necessarily close images. Left: inputs, middle, right: reconstructions with $\{\text{random, non-circularly optimized}\}$ matrices



Figure 3.14: Optimized output from combining two close images. Left: inputs, middle, right: reconstructions with {random, non-circularly optimized} matrices

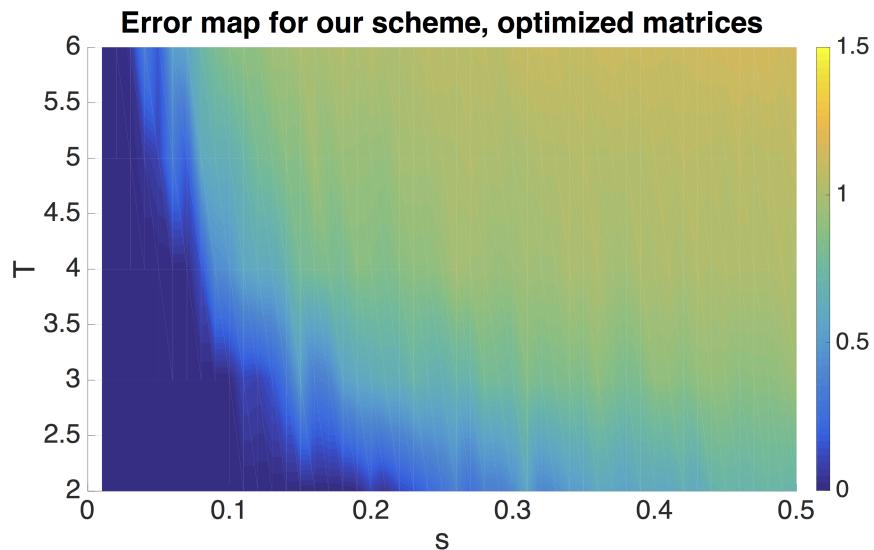


Figure 3.15: Error map for optimized codes as a function of s and T

superiority of our reconstruction is clearer, with the circular optimization smoothing out blotchier parts of the bonnet.

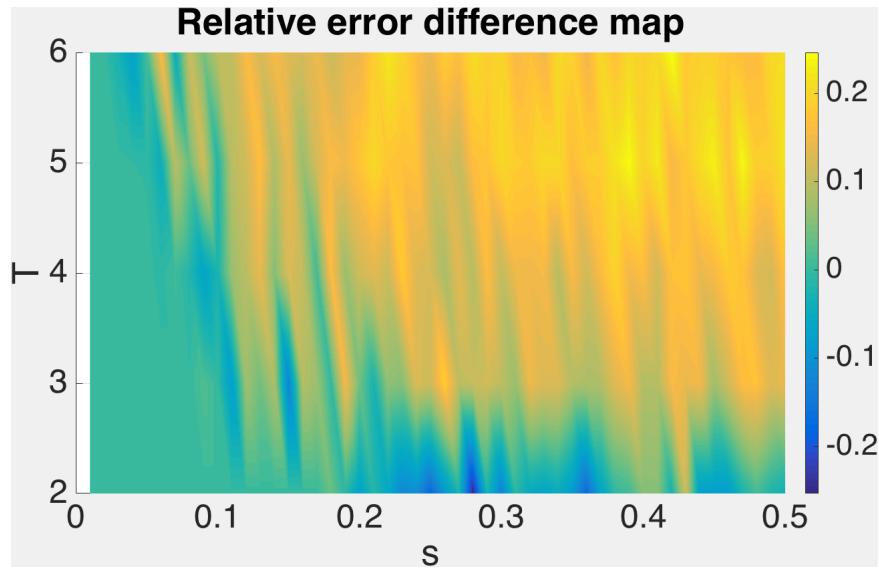


Figure 3.16: Average RRMSE using random codes minus average RRMSE using optimized codes, as a function of s and T

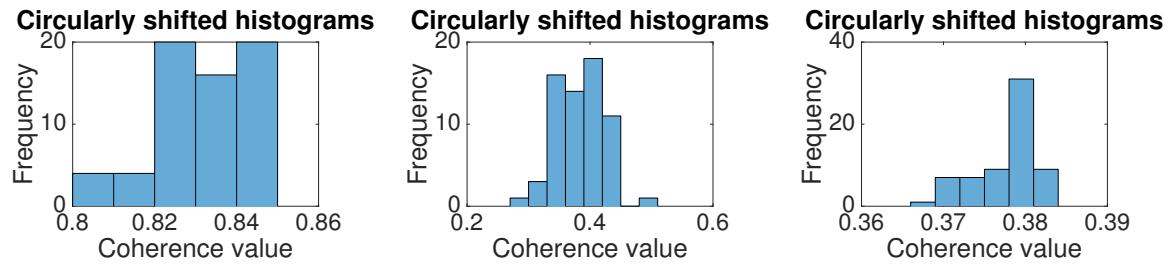


Figure 3.17: Left to right: Circularly-shifted coherence histograms for {random, non-circularly optimized, circularly optimized} matrices



Figure 3.18: Circularly optimized output from combining two close images. Left to right: reconstructions with {random, circularly optimized} matrices

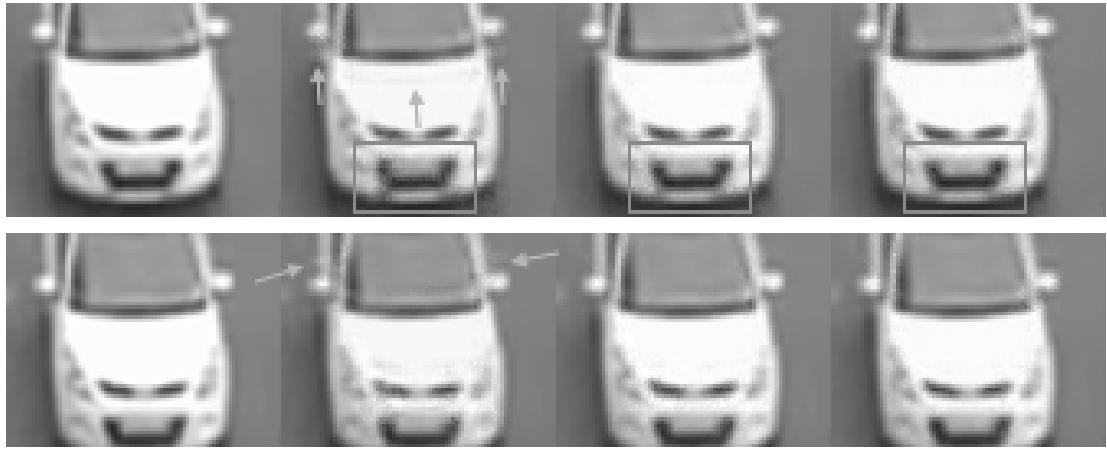


Figure 3.19: Close-ups showing subtle texture preservation with optimized matrices, example 1. Left to right: inputs, reconstructions with {random, non-circularly optimized, circularly optimized} matrices

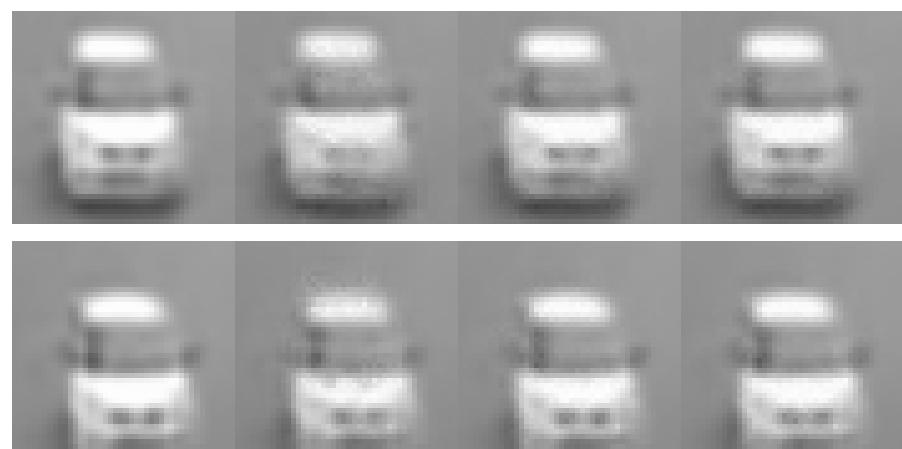


Figure 3.20: Close-ups showing subtle texture preservation with optimized matrices, example 2. Left to right: inputs, reconstructions with {random, non-circularly optimized, circularly optimized} matrices

Chapter 4

Optimizing for the CACTI Camera

INSPIRED by the events of the previous chapter, we turn to another compressive camera, the coded aperture compressive temporal imaging system introduced in [18, 17], to attempt to design masks better than random. The results of this study constitute the beginning of an important lesson in compressed sensing.

4.1 Introducing the CACTI camera

The principal idea behind the design of the CACTI camera, in the words of the authors of [18] is using "mechanical translation of a coded aperture for code division multiple access (CDMA) compression of video". The setup is shown in Fig. 1 of [18].

Linear coded combinations appear here as well, allowing us to use our previous framework in reconstructing and in optimizing codes. To recapitulate, T vectorized input frames $\{x_i\}_{i=1}^T$ are combined so that the vectorized output y appears as a coded combination (dictated by the ‘sensing matrices’ ϕ_i) of the inputs. However, this time, the sensing matrices ϕ_i are not independent across i : the mechanical translation amounts to a fixed circular shift in the elements of ϕ_i , dictated by the set mechanical translations. The sensing framework here is, with ϕ_i being the i^{th} circular shift,

$$y = \sum_{i=1}^T \phi_i x_i \tag{4.1}$$

Again, the sensing framework constrains the ϕ_i to be a diagonal matrix, with the code elements on the diagonal. To recover the input $\{x_i\}_{i=1}^T$ through the DCT coefficients

α given a measurement y , as before, we solve the optimization problem

$$\min_{\alpha} \|\alpha\|_1 \text{ subject to } y = \Phi\Psi\alpha, \quad (4.2)$$

$$\alpha = (\alpha_1 \ \alpha_2 \ \dots \ \alpha_T)^T, \quad (4.3)$$

$$\Phi = \begin{pmatrix} \phi_{11} & 0 & \dots & 0 & \dots & \dots & \phi_{T1} & 0 & \dots & 0 \\ 0 & \phi_{12} & \dots & 0 & \dots & \dots & 0 & \phi_{T2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \phi_{1n} & \dots & \dots & 0 & 0 & \dots & \phi_{Tn} \end{pmatrix}, \quad (4.4)$$

$$\Psi = \begin{pmatrix} D & 0 & \dots & 0 \\ 0 & D & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & D \end{pmatrix} \quad (4.5)$$

Again, we use the CVX [12] solver for solving the convex optimization problem in Eq. 4.5.

4.2 Optimizing codes

We follow a policy similar to the one in the previous chapter for optimizing codes. To minimize coherence, we write down the expression for the coherence of the joint dictionary $\Phi\Psi$:

$$\Phi\Psi = (\phi_1 D \ \phi_2 D \ \dots \ \phi_T D) \quad (4.6)$$

It is important, here, to note where each ϕ_{ij} came from. For this purpose, let us define a vector ϕ of code elements with elements ϕ^k for k going from 1 to n . The i^{th} permutation, therefore, takes ϕ to ϕ_i . Clearly, if we apply the permutation i to the vector $(1 \ 2 \ 3 \ \dots \ n)^T$ and call the resultant vector p_i , the j^{th} element of p_i will denote which element of ϕ got circularly shifted to ϕ_{ij} . With this definition, therefore, we have $\phi_{ij} = \phi^{p_{ij}}$.

In a similar way to the previous chapter, following the same definition, we write the normalized dot product between the β^{th} column of the μ^{th} block and the γ^{th} column of

the ν^{th} block as

$$M_{\mu\nu}(\beta\gamma) = \frac{\sum_{\alpha=1}^n \phi_{\mu\alpha}\phi_{\nu\alpha}d_\alpha(\beta)d_\alpha(\gamma)}{\sqrt{(\sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta)) (\sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma))}} \quad (4.7)$$

$$= \frac{\sum_{\alpha=1}^n \phi^{p_{\mu\alpha}}\phi^{p_{\nu\alpha}}d_\alpha(\beta)d_\alpha(\gamma)}{\sqrt{(\sum_{\alpha=1}^n \phi^{p_{\mu\alpha}2} d_\alpha^2(\beta)) (\sum_{\tau=1}^n \phi^{p_{\nu\tau}2} d_\tau^2(\gamma))}} \quad (4.8)$$

With the numerator of the above expression renamed to $\chi_{\mu\nu}(\beta\gamma)$ and the denominator renamed to $xi_{\mu\nu}(\beta\gamma)$, we write,

$$\frac{d\chi_{\mu\nu}(\beta\gamma)}{d\phi^{p_{\delta\epsilon}}} = d_\epsilon(\beta)d_\epsilon(\gamma)(\phi_{\mu\epsilon} \uparrow_{\nu\delta} + \uparrow_{\mu\delta} \phi_{\nu\epsilon}) \quad (4.9)$$

$$= d_\epsilon(\beta)d_\epsilon(\gamma)(\phi^{p_{\mu\epsilon}} \uparrow_{\nu\delta} + \uparrow_{\mu\delta} \phi^{p_{\nu\epsilon}}) \quad (4.10)$$

$$\frac{d\xi_{\mu\nu}(\beta\gamma)}{d\phi^{p_{\delta\epsilon}}} = \frac{1}{\xi_{\mu\nu}(\beta\gamma)} \left[\phi_{\mu\epsilon}d_\epsilon^2(\beta) \uparrow_{\mu\delta} \sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma) + \phi_{\nu\epsilon}d_\epsilon^2(\gamma) \uparrow_{\nu\delta} \sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta) \right] \quad (4.11)$$

$$= \frac{1}{\xi_{\mu\nu}(\beta\gamma)} \left[\phi^{p_{\mu\epsilon}}d_\epsilon^2(\beta) \uparrow_{\mu\delta} \sum_{\tau=1}^n \phi^{p_{\nu\tau}2} d_\tau^2(\gamma) + \phi^{p_{\nu\epsilon}}d_\epsilon^2(\gamma) \uparrow_{\nu\delta} \sum_{\alpha=1}^n \phi^{p_{\mu\alpha}2} d_\alpha^2(\beta) \right] \quad (4.12)$$

As before, we do gradient descent with adaptive step-size and use a multi-start strategy to combat the non-convexity of the problem.

4.3 Experiments and results

4.3.1 Simulated data

Here, we experiment with toy data where we can precisely control the sparsity of the input signals. Specifically, assuming a set of mechanical translations, we randomly generate T s -sparse (in 2D DCT) 8×8 signals $\{x_i\}_{i=1}^T$, combine them using random and designed matrices and add noise bounded in norm to $\epsilon = 10^{-5}$ to get y . Average RRMSE errors on doing this over a set of 100 vectors, as a function of signal sparsity and compression level T are shown in Figs. 4.1, 4.2, and 4.3.

These figures don't tell a very happy story: the optimization technique completely fails to produce any statistically significant improvement in the error over random matrices. The coherence decreases are significant: in the best case, the coherence for $T = 2$ decreases from 0.7911 to 0.3462, for $T = 4$ decreases from 0.7921 to 0.4952, and for $T = 6$ decreases from 0.9156 to 0.5430.

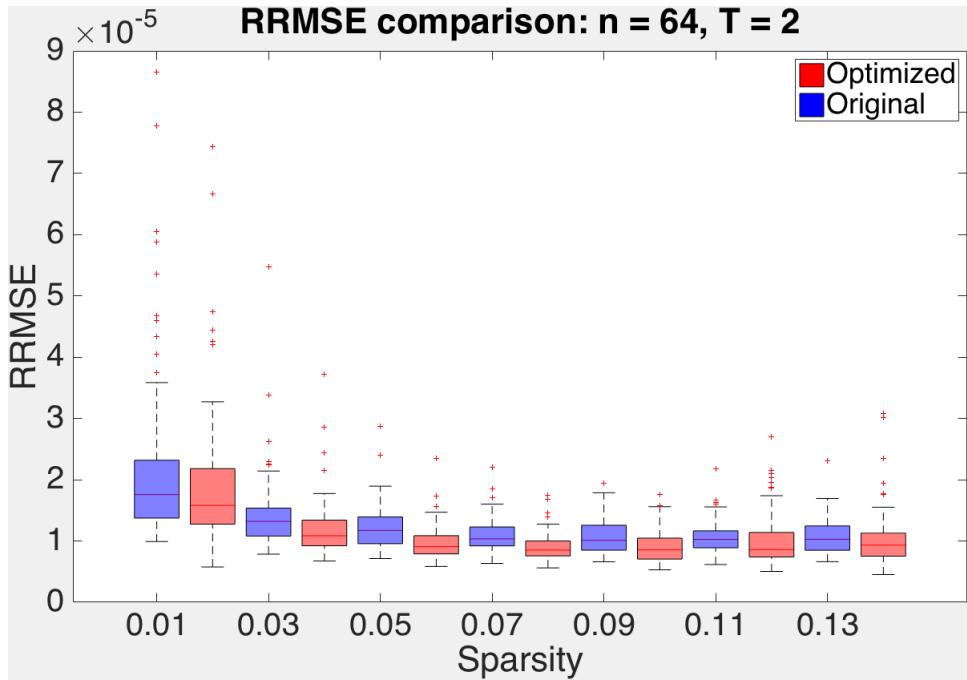


Figure 4.1: Average RRMSE as a function of sparsity for 8×8 signals, sparse in 2D DCT, combined with $T = 2$. Permutations: [5, 3; 6, 8]. Errors for low sparsity are near-zero.

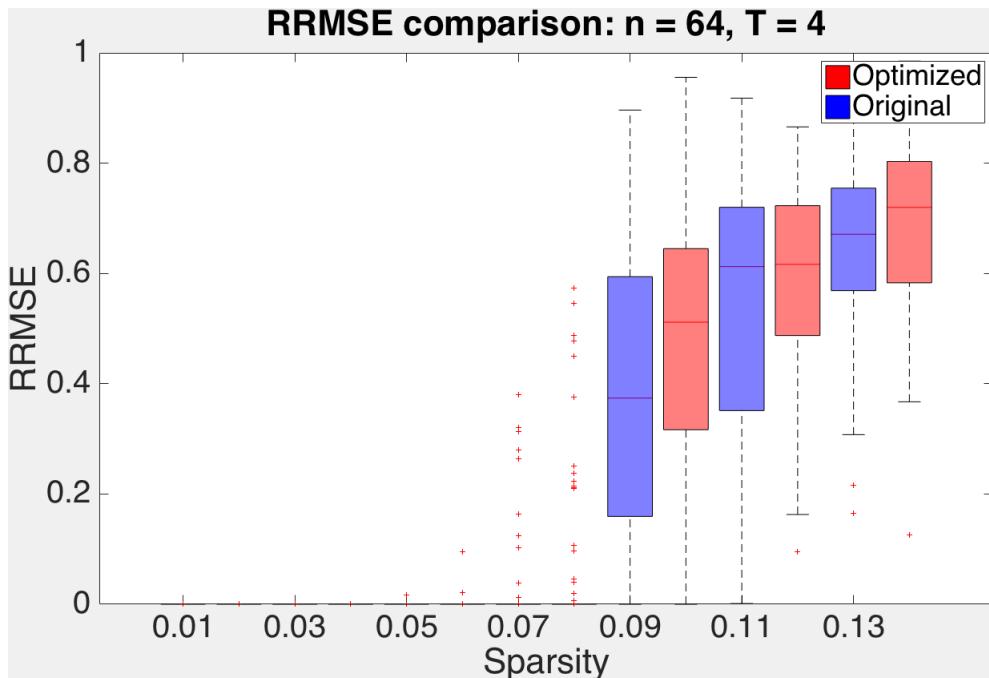


Figure 4.2: Average RRMSE as a function of sparsity for 8×8 signals, sparse in 2D DCT, combined with $T = 4$. Permutations: [7, 8; 2, 8; 6, 1; 3, 5]

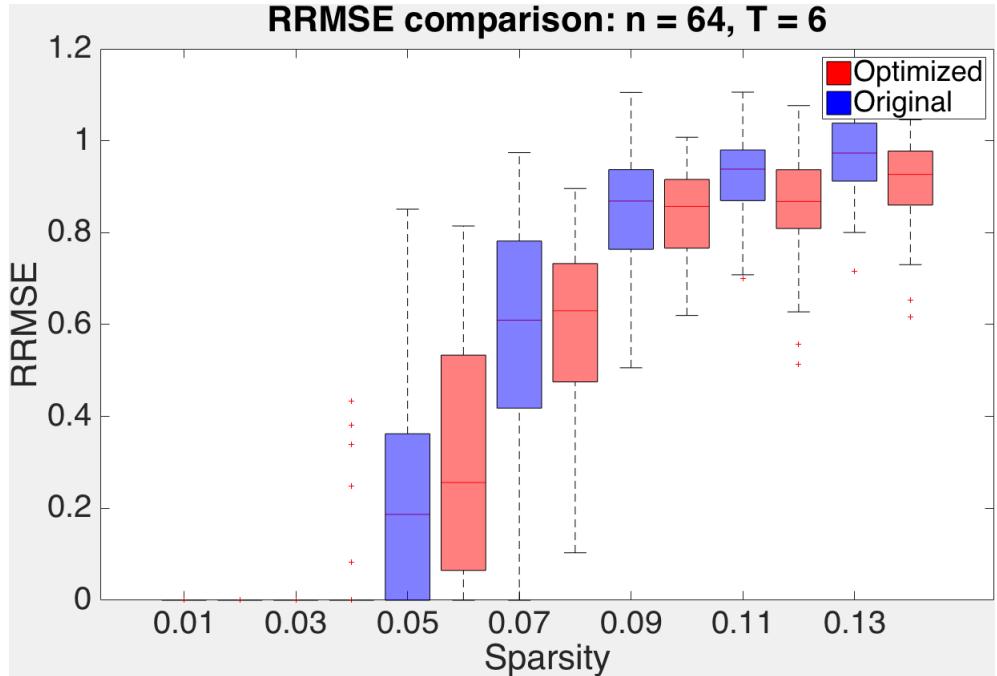


Figure 4.3: Average RRMSE as a function of sparsity for 8×8 signals, sparse in 2D DCT, combined with $T = 6$. Permutations: [6, 7; 3, 6; 6, 2; 1, 4; 8, 3; 5, 2]

What causes the algorithm to fail? The underlying assumption in this method is that the bound that RIC establishes on the maximum recovery error surface plotted against the space of sensing matrices behaves close to the actual maximum error surface (coherence further loosens up the bound). However, this might not be the case: the looser the bound gets, the more freedom the error surface has to deviate from the behavior of the bound. Then, minimizing the maximum in the bound may not correspond to minimizing the maximum in the actual error surface.

To quantify this concept, note that the s^{th} RIC of a matrix is the following:

$$\delta_s = \max_{\mathcal{S} \in \{1 \dots n\}, k} |\lambda_k(A_{\mathcal{S}}^T A_{\mathcal{S}} - I)| \quad (4.13)$$

We therefore plot, for the entire dataset of vectors we used to make the RRMSE plot, the absolute maximum eigenvalue of $A_{\mathcal{S}}^T A_{\mathcal{S}} - I$, where \mathcal{S} is the support of the vector, for both random A of the form imposed by CACTI, and designed A . The error in reconstructing this particular vector is bounded tighter than coherence by this absolute maximum eigenvalue. This gives us a handle on how well minimizing coherence minimizes this eigenvalue across supports, and therefore how much we lose by relaxing the RIC to coherence.

The $T = 2$ case is surprising: decreasing coherence over random seems to increase the values of the absolute maximum eigenvalues, which goes directly against the assumption

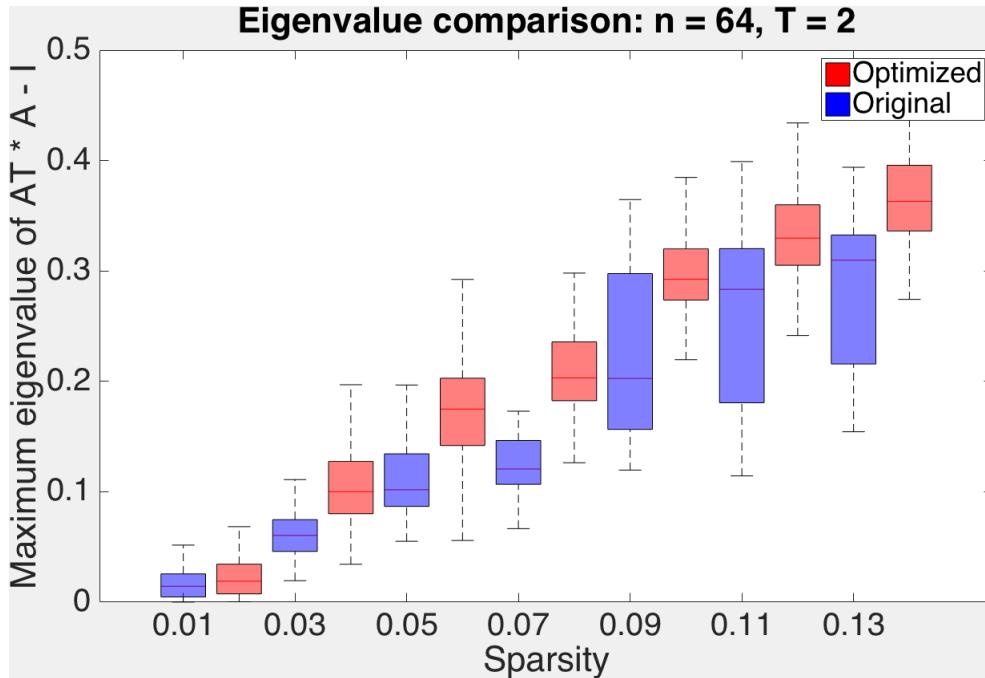


Figure 4.4: Absolute maximum restricted eigenvalue boxplot as a function of sparsity for 8×8 signals, sparse in 2D DCT, with $T = 2$. Permutations: [5, 3; 6, 8]

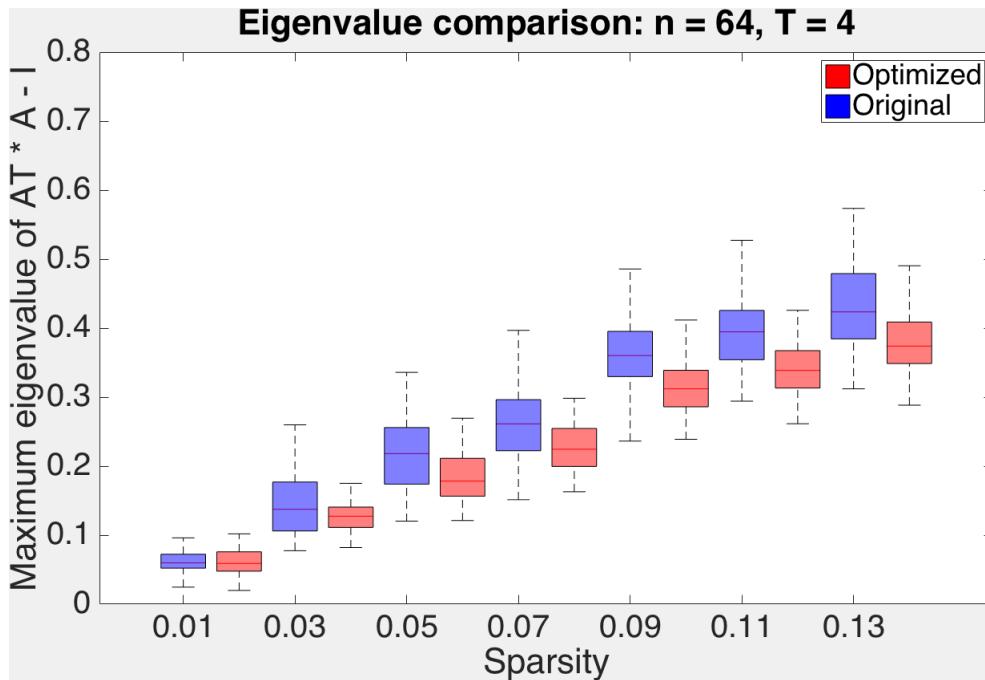


Figure 4.5: Absolute maximum restricted eigenvalue boxplot as a function of sparsity for 8×8 signals, sparse in 2D DCT, with $T = 4$. Permutations: [7, 8; 2, 8; 6, 1; 3, 5]

involved in minimizing coherence. The $T = 4$ and $T = 6$ cases behave better in terms of eigenvalues, though their performance in terms of RRMSE error isn't very good. These

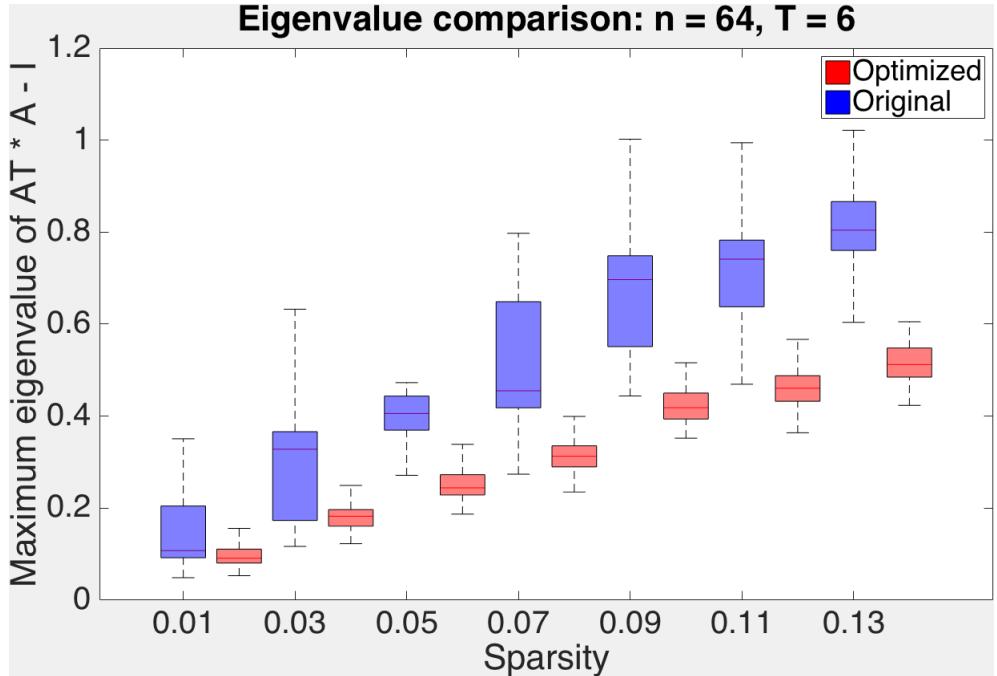


Figure 4.6: Absolute maximum restricted eigenvalue boxplot as a function of sparsity for 8×8 signals, sparse in 2D DCT, with $T = 6$. Permutations: [6, 7; 3, 6; 6, 2; 1, 4; 8, 3; 5, 2]

findings point to the fact that the problem lies not only in the relaxation of the RIC to the coherence, but also in the RIC bound itself.

A tempting thought, at this juncture, is to maximize some average of the off-diagonal dot products of columns in the effective dictionary, instead of the coherence, which is the maximum of these. This is because the coherence bound on the RIC relaxes the sum of $k - 1$ off-diagonal elements to $k - 1$ times the maximum, which is the coherence. Designing matrices optimizing square of these off-diagonal elements and performing simulated data experiments similar to the above produce similar RRMSE behavior: the matrices designed this way are no better than matrices designed using just coherence, and certainly no better than random matrices.

This warrants a more detailed empirical understanding of error bounds in compressed sensing, focusing on how bounds evolve across inequalities that give rise to them. This will be the subject of the next chapter.

Chapter 5

Limitations of Coherence-based Bounds

THE surprising failure of coherence minimization in improving recovery performance for the kind of matrices needed by the CACTI camera challenges the coherence as a basic element in understanding compressed sensing recovery. The badness, we saw in the last chapter, however, stems not only from coherence but also from the steps in which the bound is derived. It makes sense, therefore, to examine a recovery guarantee proof step by step and analyze, empirically, how looseness propagates across chains of inequalities leading to the bound.

5.1 The bound

The bound we choose to examine is the one proved in for recovery of nearly sparse vectors using basis pursuit denoising. Let the compressed sensing scenario be $z = Ax + n$, with an overcomplete $m \times n$ -sized A , $n \times 1$ -sized x and $m \times 1$ -sized z . Let μ denote the coherence of A . Suppose we recover \hat{x} by solving the basis pursuit denoising problem.

$$\hat{x} = \arg \min_{\tilde{x}} \|\tilde{x}\|_1 \text{ such that } \|z - A\tilde{x}\|_2 < \epsilon \quad (5.1)$$

Given a particular $n_x < n$, define \mathcal{X} as the set of indices of the n_x absolute greatest entries of x . Define the best n_x -sparse approximation to x , $x_{\mathcal{X}}$, by setting the x values at indices not in \mathcal{X} to zero. Then, if

$$n_x < \frac{1}{2} \left(1 + \frac{1}{\mu} \right)$$

and $\|n\|_2 < \eta$, we have the upper bound

$$\|x - \hat{x}\|_2 \leq C_0(\epsilon + \eta) + C_1\|x - x_{\mathcal{X}}\|_1 \quad (5.2)$$

We will quote some relevant steps from the proof of the bound, following [25] here. Let $h = \hat{x} - x$. Construct h_0 by setting elements of h at indices not in \mathcal{X} to zero. Let $e_0 = 2\|x - x_{\mathcal{X}}\|_1$. Then, using the above definitions,

$$\|x\|_1 \geq \|\hat{x}\|_1 = \|\hat{x}_{\mathcal{X}}\|_1 + \|\hat{x}_{\mathcal{X}^C}\|_1 = \|x_{\mathcal{X}} + h_0\|_1 + \|h - h_0 + x_{\mathcal{X}^C}\|_1 \quad (5.3)$$

$$\geq \|x_{\mathcal{X}}\|_1 - \|h_0\|_1 + \|h - h_0\|_1 - \|x_{\mathcal{X}^C}\|_1 \quad (5.4)$$

$$\implies \|h - h_0\|_1 \leq 2\|x_{\mathcal{X}^C}\|_1 + \|h_0\|_1 \quad (5.5)$$

$$\implies \|h - h_0\|_1 \leq \|h_0\|_1 + e_0 \quad (5.6)$$

$$\implies \|h\|_1 \leq 2\|h_0\|_1 + e_0 \quad (5.7)$$

where the last step follows from the reverse triangle inequality. Furthermore,

$$\|Ah\|_2 = \|A\hat{x} - y - (Ax - y)\|_2 \quad (5.8)$$

$$\leq \|A\hat{x} - y\|_2 + \|Ax - y\|_2 \quad (5.9)$$

$$\leq \eta + \epsilon \quad (5.10)$$

An application of the Geršgorin disk theorem to $\|Ah_0\|^2$ gives, since h_0 is perfectly sparse

$$(1 - \mu(n_x - 1))\|h_0\|_2^2 \leq \|Ah_0\|_2^2 \leq (1 + \mu(n_x - 1))\|h_0\|_2^2 \quad (5.11)$$

Next,

$$|h^T A^T A h_0| \geq |h_0^T A^T A h_0| - |(h - h_0)^T A^T A h_0| \quad (5.12)$$

$$\geq (1 - \mu(n_x - 1))\|h_0\|_2^2 - \left| \sum_{k \in \mathcal{X}} \sum_{l \in \mathcal{X}^C} [h_0^T]_k a_k^T a_l [h]_l \right| \quad (5.13)$$

$$\geq (1 - \mu(n_x - 1))\|h_0\|_2^2 - \mu\|h_0\|_1\|h - h_0\|_1 \quad (5.14)$$

$$\geq (1 - \mu(n_x - 1))\|h_0\|_2^2 - \mu\|h_0\|_1(\|h_0\|_1 + e_0) \quad (5.15)$$

$$\geq (1 - \mu(n_x - 1))\|h_0\|_2^2 - \mu n_x\|h_0\|_2^2 - \mu\sqrt{n_x}\|h_0\|_2 e_0 \quad (5.16)$$

$$= (1 - \mu(2n_x - 1))\|h_0\|_2^2 - \mu\sqrt{n_x}\|h_0\|_2 e_0, \quad (5.17)$$

Eq. 5.13 is a result of the Geršgorin bound in Eq. 5.11, Eq. 5.14 arises from $|a_k^T a_l| \leq \mu$, $\forall k \neq l$ and Eq. 5.15 comes from the condition in Eq. 5.6. Eq. 5.16 is an application of

the Cauchy-Schwarz inequality. Next,

$$\|h_0\|_2 \leq \frac{|h^T A^T A h_0| + \mu \sqrt{n_x} \|h_0\|_2 e_0}{(1 - \mu(2n_x - 1)) \|h_0\|_2} \quad (5.18)$$

$$\leq \frac{\|Ah\|_2 \|Ah_0\|_2 + \mu \sqrt{n_x} \|h_0\|_2 e_0}{(1 - \mu(2n_x - 1)) \|h_0\|_2} \quad (5.19)$$

$$\leq \frac{(\epsilon + \eta) \sqrt{1 + \mu(n_x - 1)} \|h_0\|_2 + \mu \sqrt{n_x} \|h_0\|_2 e_0}{(1 - \mu(2n_x - 1)) \|h_0\|_2} \quad (5.20)$$

$$= \frac{(\epsilon + \eta) \sqrt{1 + \mu(n_x - 1)} + \mu \sqrt{n_x} e_0}{1 - \mu(2n_x - 1)} \quad (5.21)$$

The proof further goes on to bound $\|h\|_2$. The rest of the proof, however, is an application of the bound yet derived, and we will study the looseness of just this part of the proof. For completeness, we quote the bound here.

$$\begin{aligned} \|h\|_2 &\leq \frac{1 - \mu(2n_x - 1) + \sqrt{\mu n_x} \sqrt{1 + \mu(n_x - 1)}}{\sqrt{1 + \mu(2n_x - 1)}} (\epsilon + \eta) + \\ &\quad \frac{2\sqrt{\mu + \mu^2}}{1 - \mu(2n_x - 1)} \|x - x_{\mathcal{X}}\|_1 \end{aligned} \quad (5.22)$$

5.2 Empirical analysis of the bound

We will concentrate on the simplest case, where the vector x is exactly sparse, and set $n_x = \|x\|_0$. $\|x\|_0$ is set as the maximum l_0 norm that the bound allows, which is the greatest integer below $0.5(1 + 1/\mu)$. The definitions then reduce e_0 to 0. We generate a set of sparse vectors of a size suitable to be sensed with a selected sensing matrix, add noise bounded in norm by $\eta = \epsilon = 10^{-5}$, and plot a boxplot of the relative difference between the left and right hand sides in selected inequalities in the proof above.

The relative differences we choose to show are due to, in order,

1. The triangle inequality in Eq. 5.12, with respect to the left hand side of Eq. 5.12
2. The Geršgorin bound and triangle inequality in Eq. 5.13, with respect to the left hand side of Eq. 5.12
3. Replacing dot products with their maximum, coherence, in Eq. 5.14, with respect to the left hand side of Eq. 5.12
4. The application of Eq. 5.6 in Eq. 5.15, with respect to the left hand side of Eq. 5.12

5. The bound relating the l_1 and l_2 norms in Eq. 5.17, with respect to the left hand side of Eq. 5.12
6. The rearrangement of Eq. 5.18, with respect to the left hand side of Eq. 5.18
7. The Cauchy-Swartz inequality in Eq. 5.19, with respect to the left hand side of Eq. 5.17
8. The application of Eq. 5.6 in Eq. 5.21, with respect to the left hand side of Eq. 5.17
9. The leftmost side of the Geršgorin bound in Eq. 5.11, with respect to $\|Ah_0\|_2^2$
10. The rightmost side of the Geršgorin bound in Eq. 5.11, with respect to $\|Ah_0\|_2^2$
11. The actual RRMSE error for the simulated vector in question, which is the left hand side of Eq. 5.22, and the bound predicted by Eq. 5.22, with respect to the actual RRMSE.

5.2.1 General sensing matrices

Here, to allow a high $\|x\|_0$, we first choose a random 499×500 matrix drawn from a Gaussian distribution. Also as a benchmark for a compressive scenario, we test with 250×500 , 167×500 , 125×500 and 50×500 matrices corresponding to 50%, 33%, 25% and 10% sparsity. All this is done at a reconstruction tolerance of $\epsilon = 10^{-6}$.

Figs. 5.2, 5.3, 5.4 and 5.5 show the relative differences for these scenarios. As is immediately noticed, the differences in the first two cases are shockingly high: the bound is off by two orders of magnitude compared to the actual error surface. The other three are off by a relative error well above 1.

Consistent among all these figures is, however, the presence of jumps between relative differences across steps. There appear jumps in the transition from step 3 to step 4, and from step 7 to step 8 (the baseline for comparison changes between step 5 and step 6, so the jump here isn't significant. Also, these two steps are rearrangements of each other, so there's no loss happening in between). The fact that the Geršgorin disk theorem does not cause much trouble is an effect of the fact that the matrix was drawn from a Gaussian

random distribution, because Gaussian matrices are known to have low RIC and coherence values [10].

We will contrast this behavior to what happens with matrices of the kind we encounter in practical signal processing scenarios further.

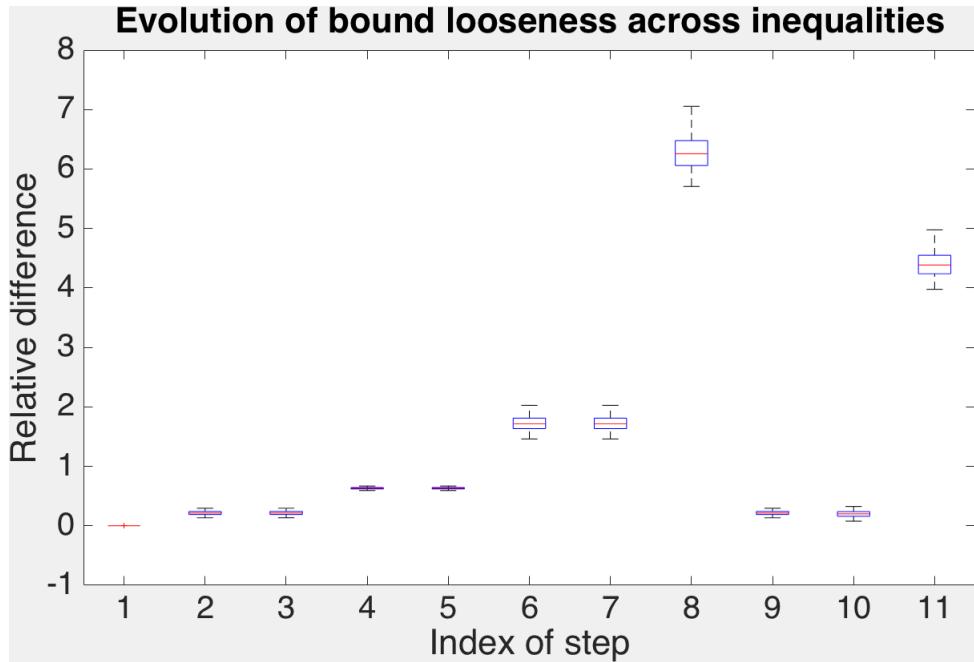


Figure 5.1: Relative difference between inequalities in the error bound above for a 499×500 matrix drawn from a standard normal distribution

5.2.2 In coded source separation

The maximum sparsity coded source separation affords, in most cases, is 1, because of the $n \times nT$ size of the matrix, and because typical matrices where the mask values are drawn from positive uniform distributions have coherence values around 0.8 to 0.9. Nevertheless, we repeat the same process as above, expecting the bound to work a little better in light of this low sparsity.

In contrast to expectations, the same steps show significant jumps in all the steps the general sensing matrices show in. The bound does not deprecate as badly as in the general sensing matrix case, just by one order of magnitude, but offers enough room for failure nevertheless.

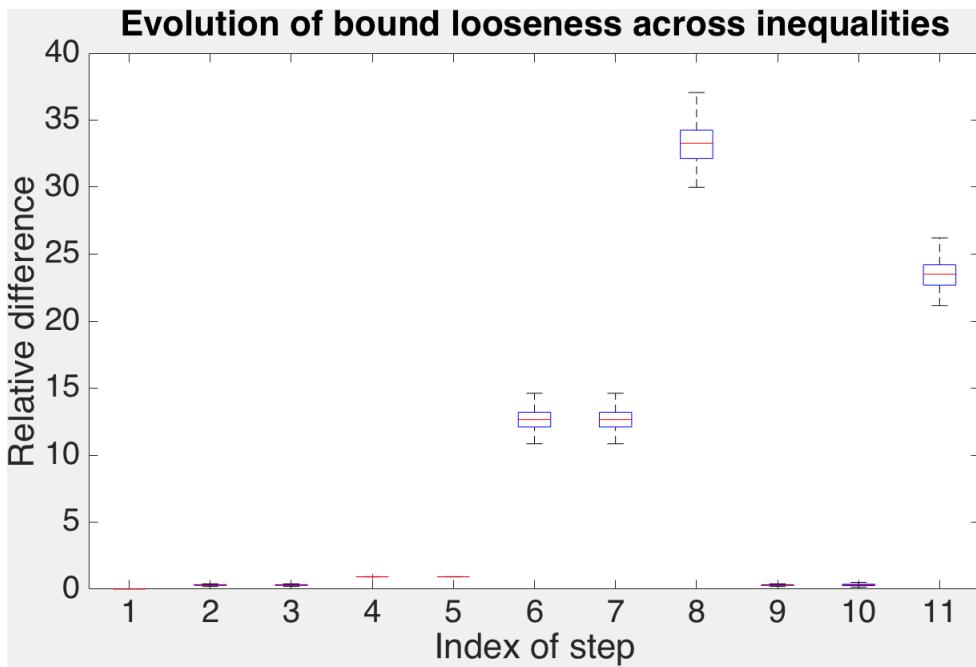


Figure 5.2: Relative difference between inequalities in the error bound above for a 250×500 matrix drawn from a standard normal distribution

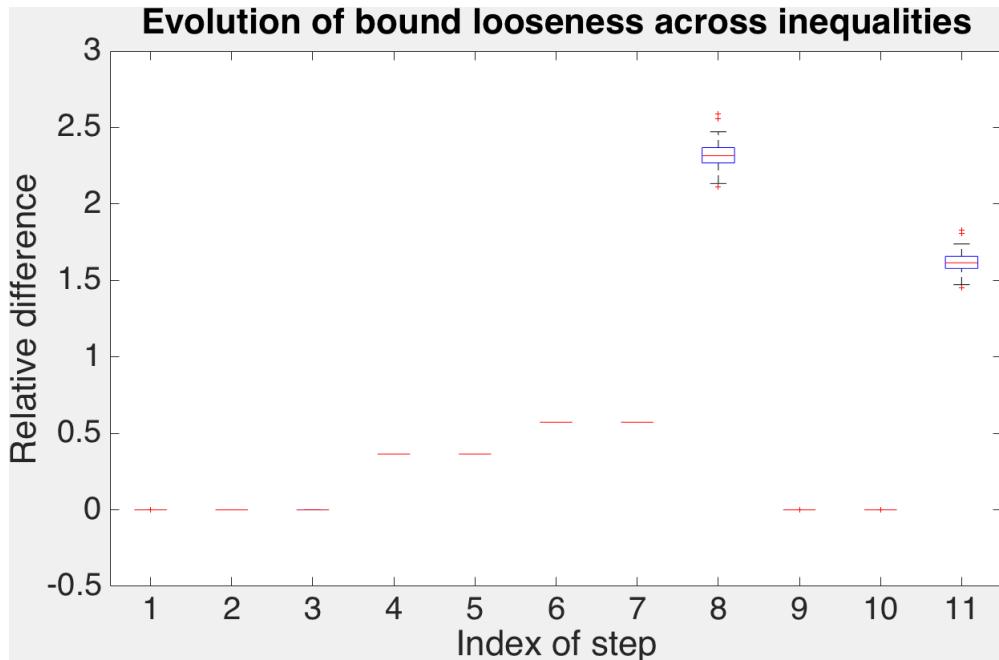


Figure 5.3: Relative difference between inequalities in the error bound above for a 167×500 matrix drawn from a standard normal distribution

5.2.3 In the CACTI camera

Similarly, we test the matrices we use in the CACTI camera to for looseness in bound. Similar behavior as in the coded source separation case is observed, with a similar (more

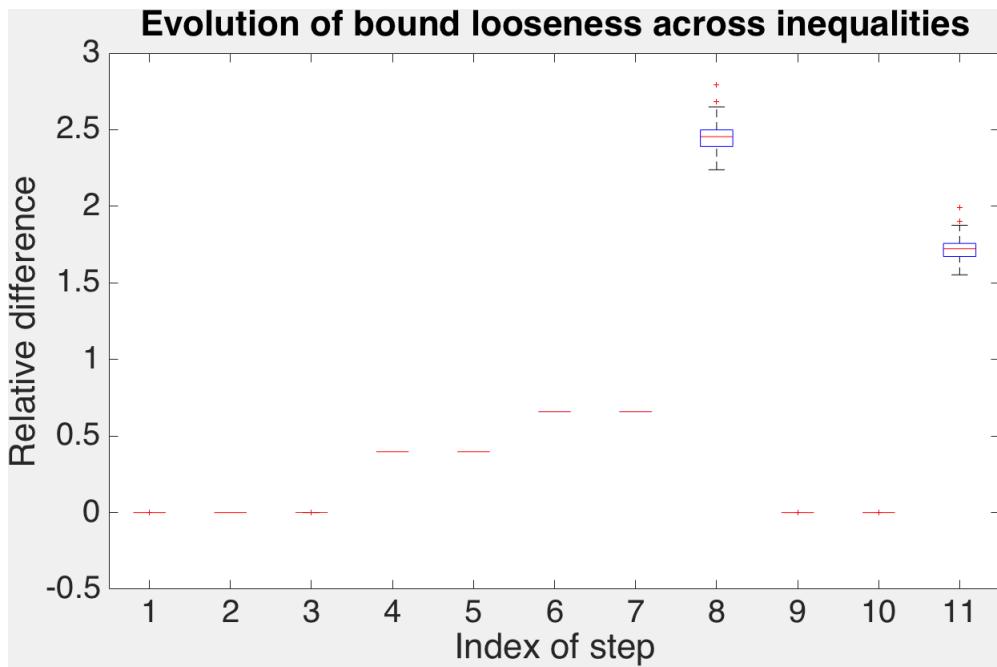


Figure 5.4: Relative difference between inequalities in the error bound above for a 125×500 matrix drawn from a standard normal distribution

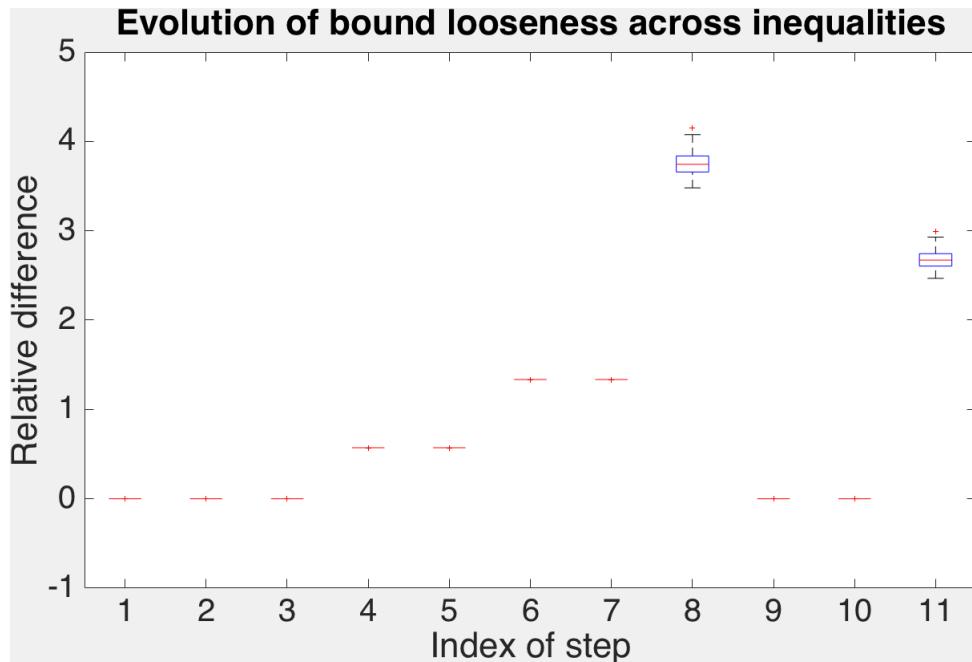


Figure 5.5: Relative difference between inequalities in the error bound above for a 50×500 matrix drawn from a standard normal distribution

or less) depreciation. On the same lines, we plot looseness evolution for matrices designed for the CACTI camera in Chapter 4, both using coherence and using the sum of squares of off-diagonal dot products of columns of the effective dictionary.

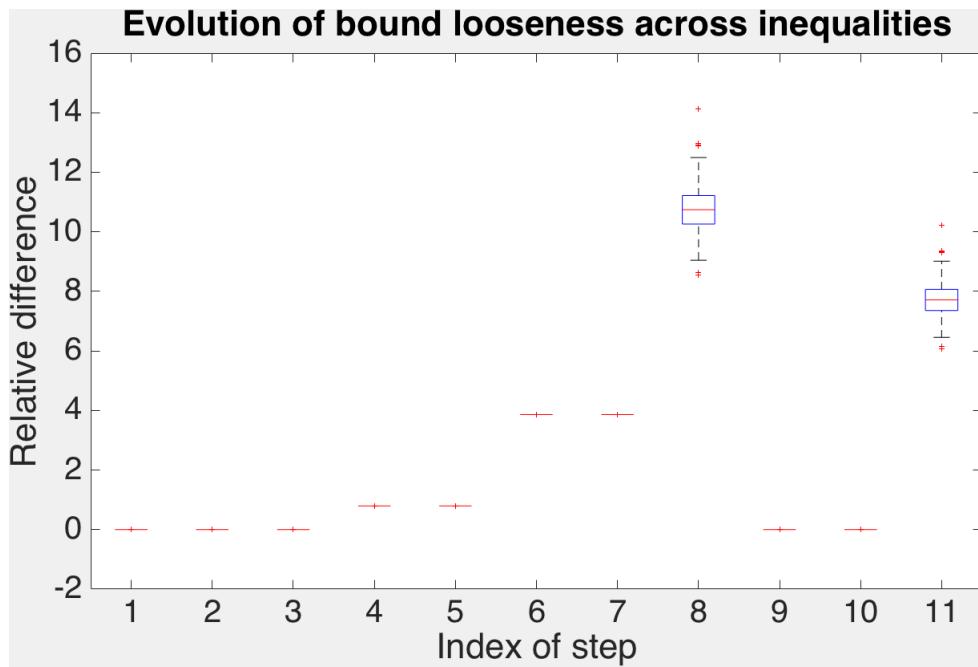


Figure 5.6: Relative difference between inequalities in the error bound above for 8×8 random positive codes in the coded source separation framework for $T = 2$

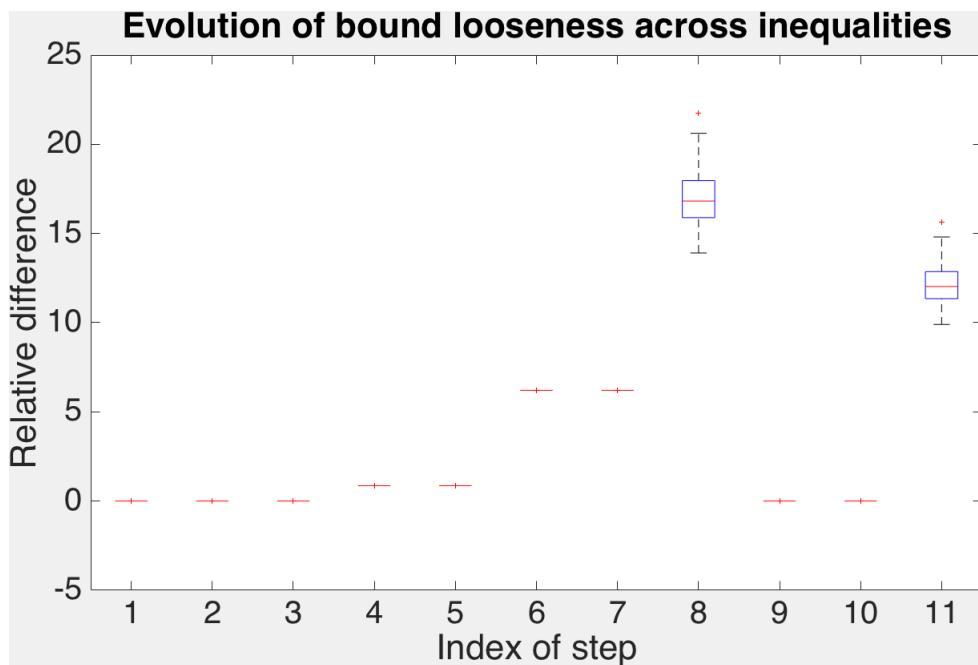


Figure 5.7: Relative difference between inequalities in the error bound above for 8×8 random positive codes in the coded source separation framework for $T = 3$

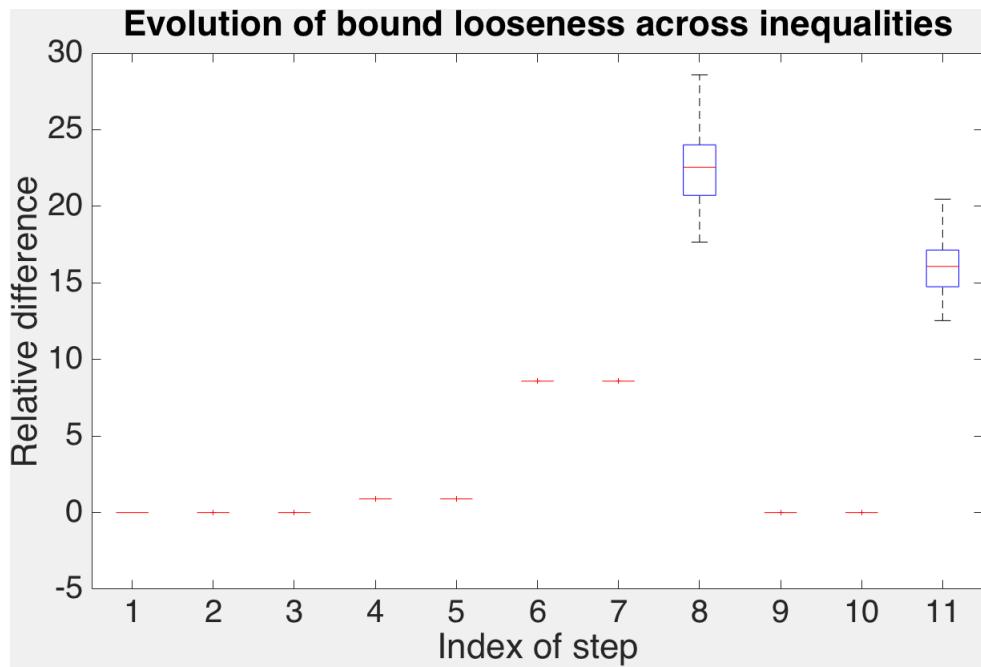


Figure 5.8: Relative difference between inequalities in the error bound above for 8×8 random positive codes in the coded source separation framework for $T = 4$

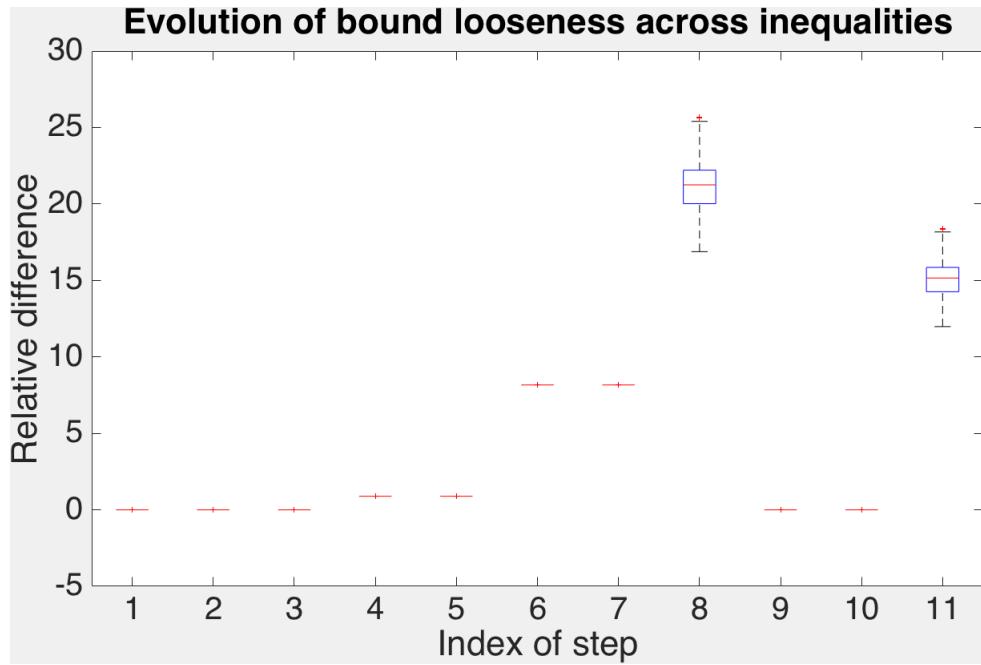


Figure 5.9: Relative difference between inequalities in the error bound above for 8×8 random positive codes in the coded source separation framework for $T = 5$

5.3 Discussion

While the figures above do not explain why coherence succeeds in the coded source separation case but doesn't in the CACTI case, the fact that there are common steps that

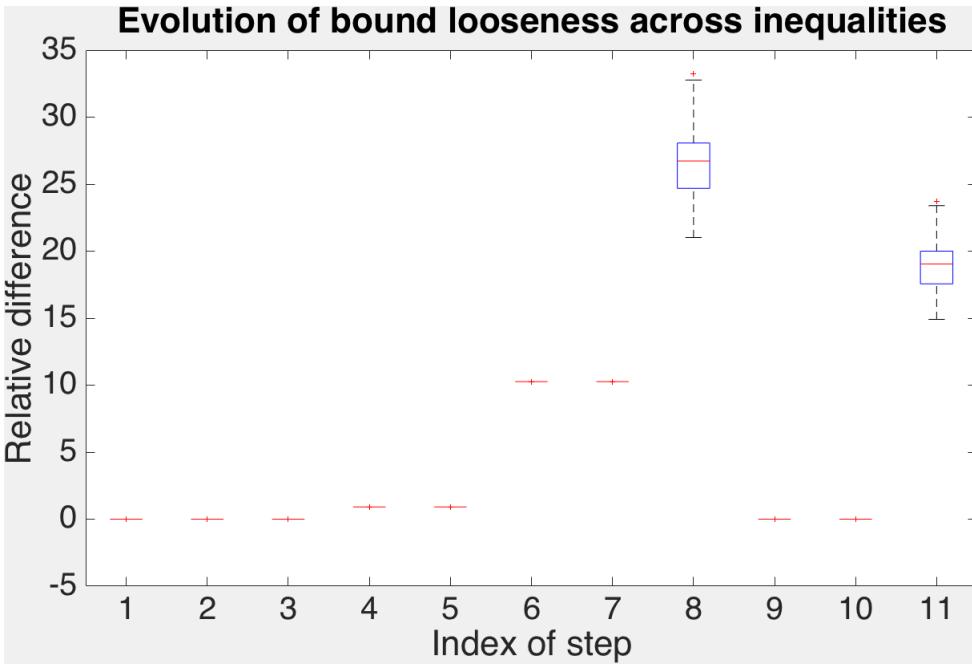


Figure 5.10: Relative difference between inequalities in the error bound above for 8×8 random positive codes in the coded source separation framework for $T = 6$

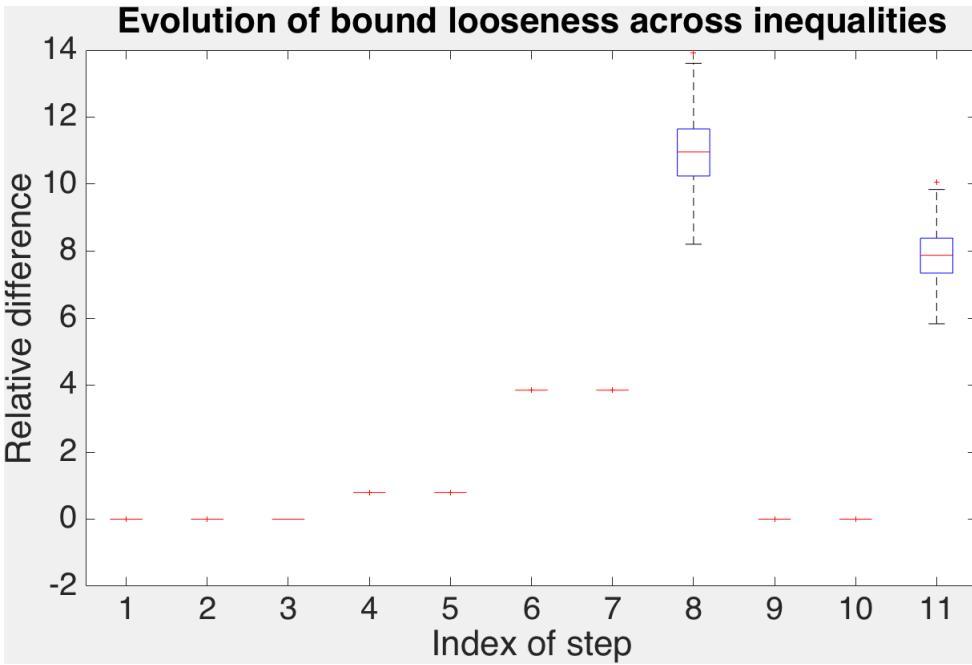


Figure 5.11: Relative difference between inequalities in the error bound above for 8×8 random positive codes in the CACTI camera for $T = 2$. Permutations: [3, 1; 5, 5]

cause significant looseness of bound is a big takeaway. These steps can be isolated and the precise inequalities causing problems can be pointed to. In our case, these are the ones leading to step 4 from step 3, and from step 7 to step 8. We will examine these

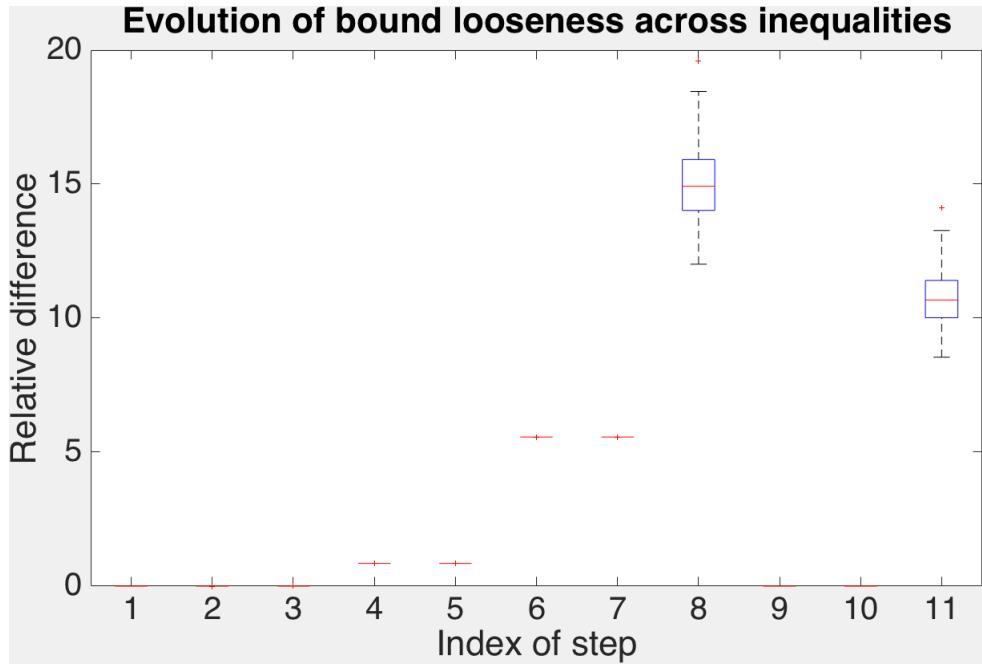


Figure 5.12: Relative difference between inequalities in the error bound above for 8×8 random positive codes in the CACTI camera for $T = 3$. Permutations: [7, 7; 4, 4; 7, 3]

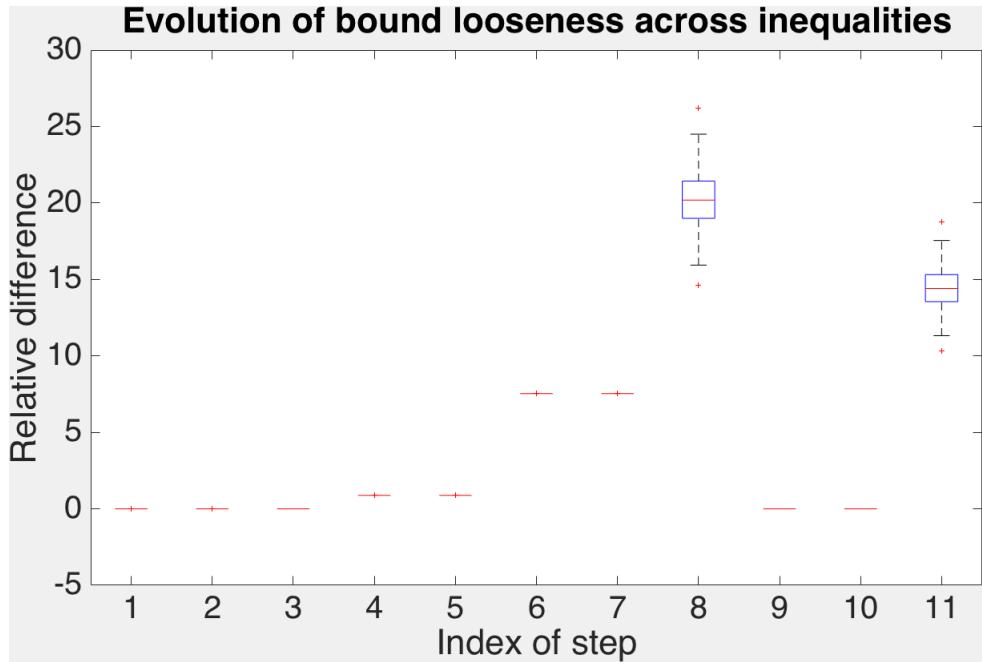


Figure 5.13: Relative difference between inequalities in the error bound above for 8×8 random positive codes in the CACTI camera for $T = 4$. Permutations: [1, 2; 6, 6; 7, 3; 2, 7]

inequalities in some detail now.

The transition from step 3 to step 4 involves the use of the constraint in Eq. 5.6.

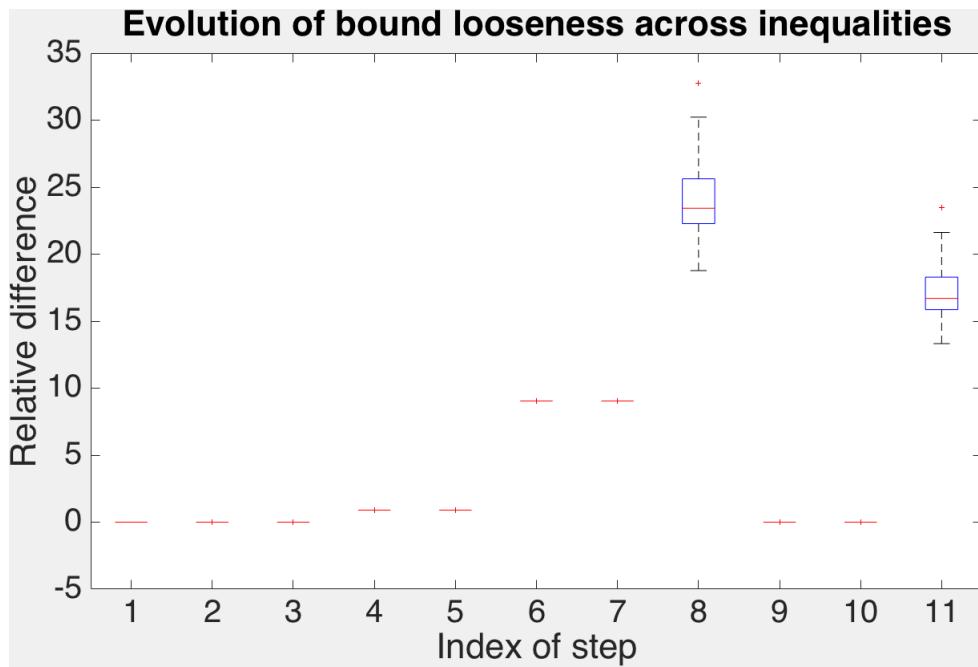


Figure 5.14: Relative difference between inequalities in the error bound above for 8×8 random positive codes in the CACTI camera for $T = 5$. Permutations: [7, 6; 4, 8; 4, 2; 6, 5; 2, 3]

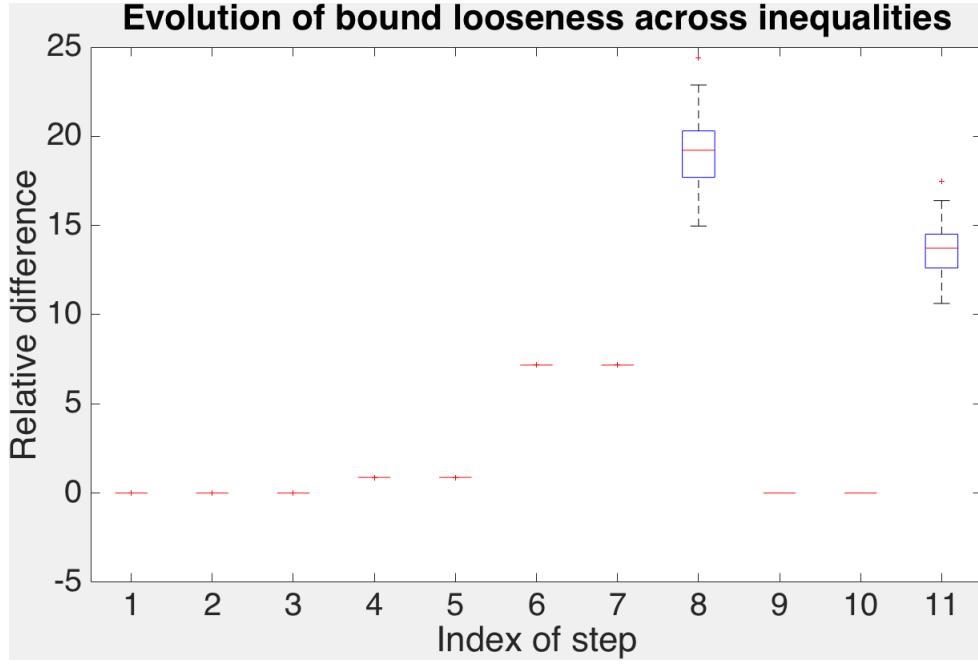


Figure 5.15: Relative difference between inequalities in the error bound above for 8×8 random positive codes in the CACTI camera for $T = 6$. Permutations: [1, 8; 8, 7; 6, 1; 4, 6; 7, 8; 5, 3]

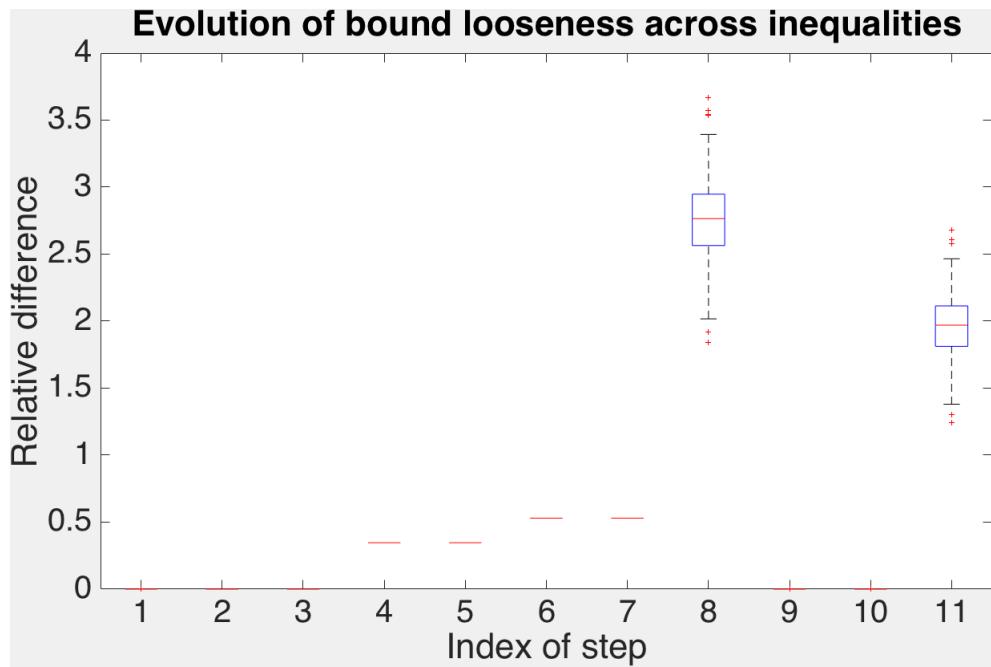


Figure 5.16: Relative difference between inequalities in the error bound above for 8×8 designed positive codes in the CACTI camera for $T = 2$. Permutations: [5, 3; 6, 8]

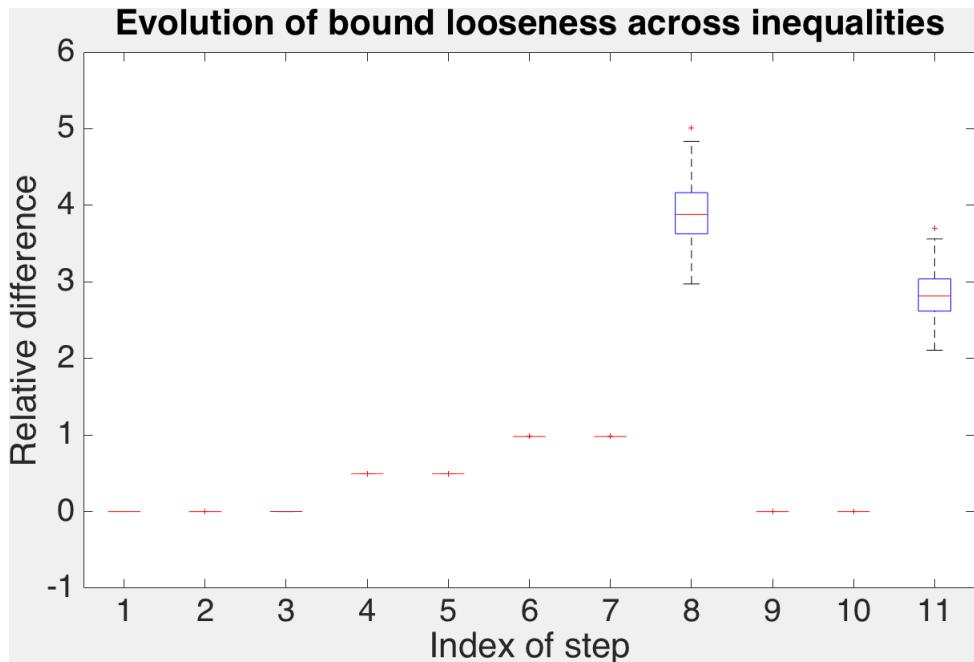


Figure 5.17: Relative difference between inequalities in the error bound above for 8×8 designed positive codes in the CACTI camera for $T = 4$. Permutations: [7, 8; 2, 8; 6, 1; 3, 5]

Further, going from step 7 to step 8 involves the constraint in Eq. 5.10, and the right side of the Geršgorin inequality in Eq. 5.11. The culprits here, therefore, are Eq. 5.10,

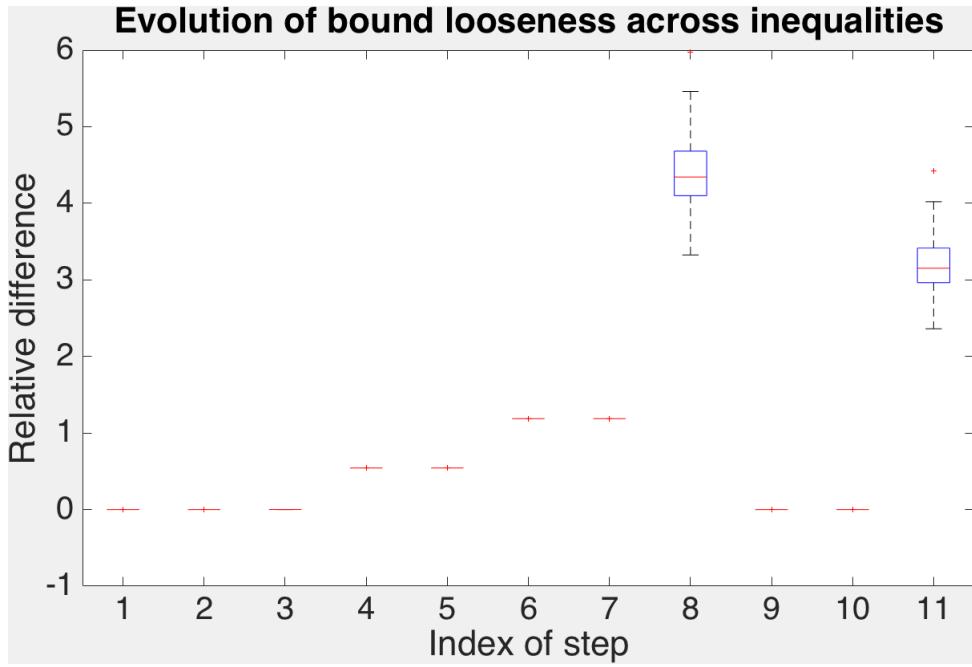


Figure 5.18: Relative difference between inequalities in the error bound above for 8×8 designed positive codes in the CACTI camera for $T = 6$. Permutations: [6, 7; 3, 6; 6, 2; 1, 4; 8, 3; 5, 2]

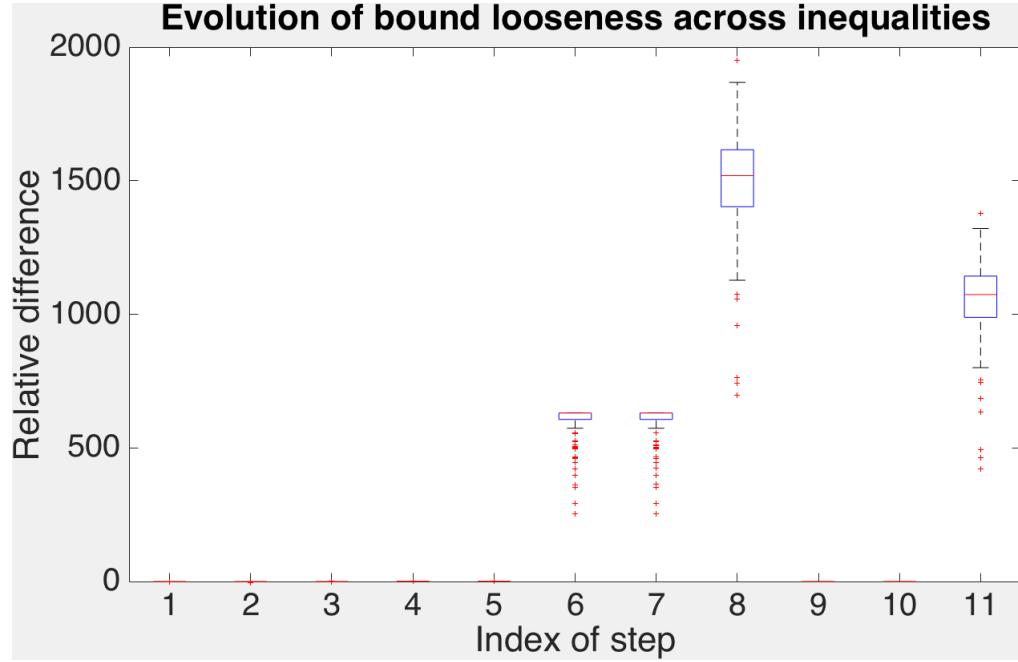


Figure 5.19: Relative difference between inequalities in the error bound above for 8×8 average-designed positive codes in the CACTI camera for $T = 2$. Permutations: [8, 4; 7, 2]

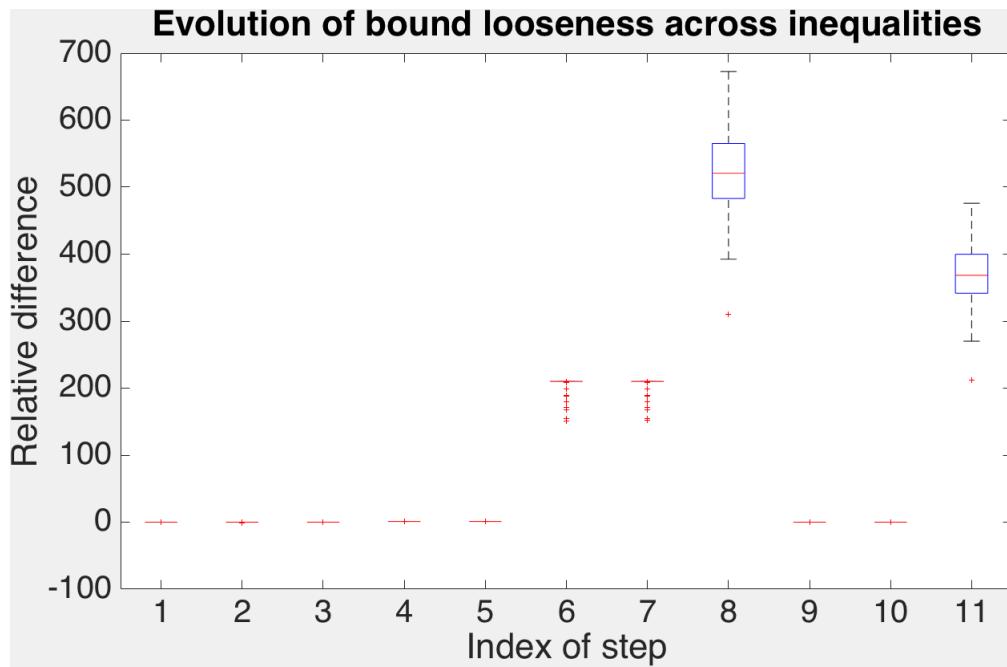


Figure 5.20: Relative difference between inequalities in the error bound above for 8×8 average-designed positive codes in the CACTI camera for $T = 4$. Permutations: [3, 1; 1, 7; 6, 3; 8, 1]

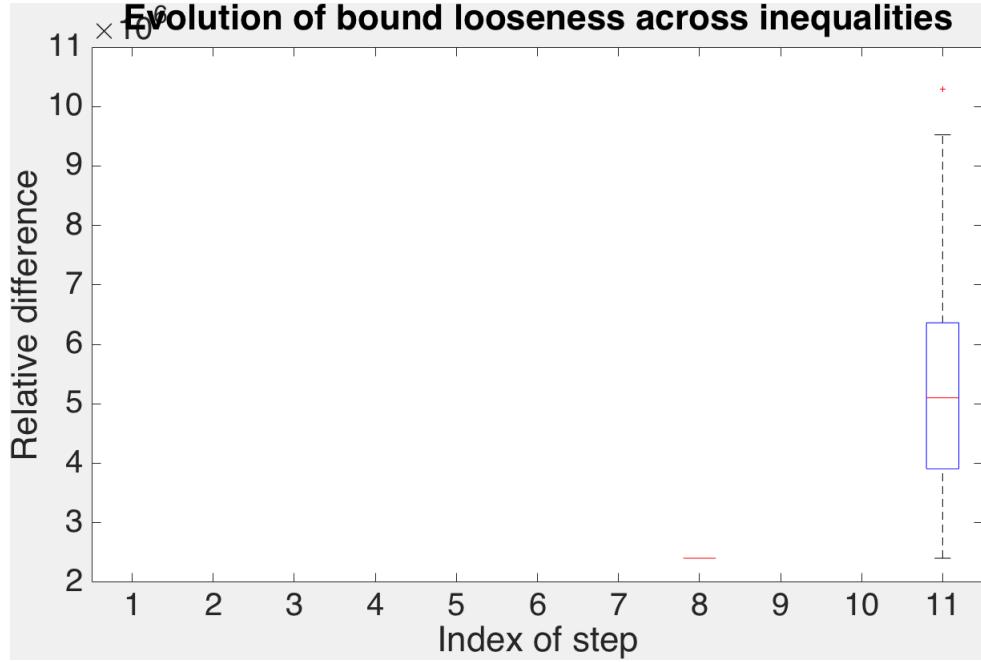


Figure 5.21: Relative difference between inequalities in the error bound above for 8×8 average-designed positive codes in the CACTI camera for $T = 6$. Permutations: [7, 3; 5, 6; 8, 8; 5, 2; 2, 3; 7, 3]

Eq. 5.6 and the right side of Eq. 5.11.

However, these are fundamental constraints that come from the problem itself. For instance, the constraint in Eq. 5.10 comes from the nature of the noise. However, as we saw empirically in our situation, there are hardly any (in our dataset, none) vectors that meet this bound. It can be inferred, therefore, that at the cost of accommodating a set of rare worst case vectors, optimizing worst case bounds does not do well on the average case signal.

At this point, it is worthwhile to stop to consider the array of design schemes based on coherence. These methods ([9, 8, 20, 21, 3, 1, 23, 22]) attempt to design matrices with an extremely loose bound. The sheer number of these methods very emphatically states the popularity of the coherence as a measure of matrix goodness. Why these methods are successful is unclear at this moment, given the results in this chapter. Better, tighter bounds on sparse recovery will possibly make these methods more effective at optimizing matrices in their particular applications.

A full analysis of average case error bounds, however, requires estimating a posterior on the space of input vectors as well as assuming a specific statistical model for noise, and involves an intractable quantity to calculate, as stated in [6]. Such an analysis is beyond the scope of this thesis. A sampling-based approach seems to be a direction in which to proceed for any tractable sensing matrix design considering the average case.

Chapter 6

Alternate Compressed Sensing Bounds

THE looseness of coherence-based bound, as we saw very emphatically in the previous chapter, is a cause of major problems in matrix design. This leads us to ask whether there are bounds on reconstruction error tighter than coherence that are easier to compute and optimize. We now look at such a bound on reconstruction error.

6.1 A new bound

6.1.1 An l_∞ error-based sparsity criterion

[27] investigates the performance bounds on sparse recovery by bounding the reconstruction error in the l_∞ norm. This bound relaxes the definition of sparsity to $s(x) = \|x\|_1/\|x\|_\infty$. This relaxation leads to a computable sufficient condition for accurate sparse recovery.

Define $\omega_\diamond(Q, s)$ as follows:

$$\omega_\diamond(Q, s) = \min_{s(z) \leq s} \frac{\|Qz\|_\diamond}{\|z\|_\infty} \quad (6.1)$$

then it can be shown that if $\|w\|_\diamond \leq \epsilon$ and if x is k -sparse, then the basis pursuit solver (substituting $p = 1$ in Eq. 2.4) yields an \hat{x} that satisfies

$$\|\hat{x} - x\|_\infty \leq \frac{2\epsilon}{\omega_\diamond(A, 2k)} \quad (6.2)$$

The bound on the l_2 error given by the l_∞ error gives us

$$\|\hat{x} - x\|_2 \leq \frac{2\epsilon\sqrt{2k}}{\omega_\diamond(A, 2k)} \quad (6.3)$$

Therefore, one clear way to optimize A is to maximize the quantity $\omega_\diamond(A, 2k)$. Fortunately, it turns out that this quantity can be written as a minimum of n convex optimization problems [27]

$$\omega_\diamond(Q, s) = \min_{i \in 1..n} \min_{\lambda \in \mathbb{R}^{n-1}} \|q_i - Q(:, \sim i)\lambda\|_\diamond \text{ subject to } \|\lambda\|_1 \leq s - 1 \quad (6.4)$$

where Q_i is the i^{th} column of Q and $Q(:, \sim i)$ represents the matrix Q with the i^{th} column removed. This break-up into multiple optimization problems can be exploited for optimization, setting $\diamond = 2$. The gradient of the overall objective function is the gradient of that inner objective function whose value is the highest. Simple matrix differentiation then leads to a gradient descent scheme.

6.2 Is this bound feasible to optimize on?

The constraint that the coherence of a matrix should be low implies that no column must be written as a multiple of the other, and the pairwise projections of columns on each other should be small. In other words, low coherence implies that no column should be written as a sparse linear combination of other columns only with $k = 1$. A careful look at the objective function in Eq. 6.4 reveals that it generalizes coherence by penalizing not only expressions of one column in terms of another, but also expressions of one column as sparse combinations of other columns, where the sparsity is encouraged by the l_1 constraint on the linear combination.

[27] claims, therefore, qualitatively, that this consideration of multiple columns, instead of pairs of columns as in the coherence, allows for a tighter bound than the one given by coherence. It is indeed true that the bound holds for all sparsity levels, in contrast to the coherence bound which holds only if $k \leq 0.5(1 + 1/\mu)$. Given the computational tractability and apparent tightness, therefore, it is a tempting thought to use this bound for matrix optimization.

So, we go ahead and assess the feasibility of this bound on optimizing matrices. The first order of business here is to check, in order to avoid falling into the same trap as we did with coherence, the looseness of the bound. To this end, we proceed similar as we did with coherence. We generate 50×1 sparse vectors, varying the sparsity, and reconstruct them from noisy compressive measurements generated by a $m \times 50$ Gaussian random

matrix for $m = 10, 25, 40, 55, 70, 85$ and uniform noise bounded in norm by $\epsilon = 10^{-5}$, and compare the reconstruction error with the bound.

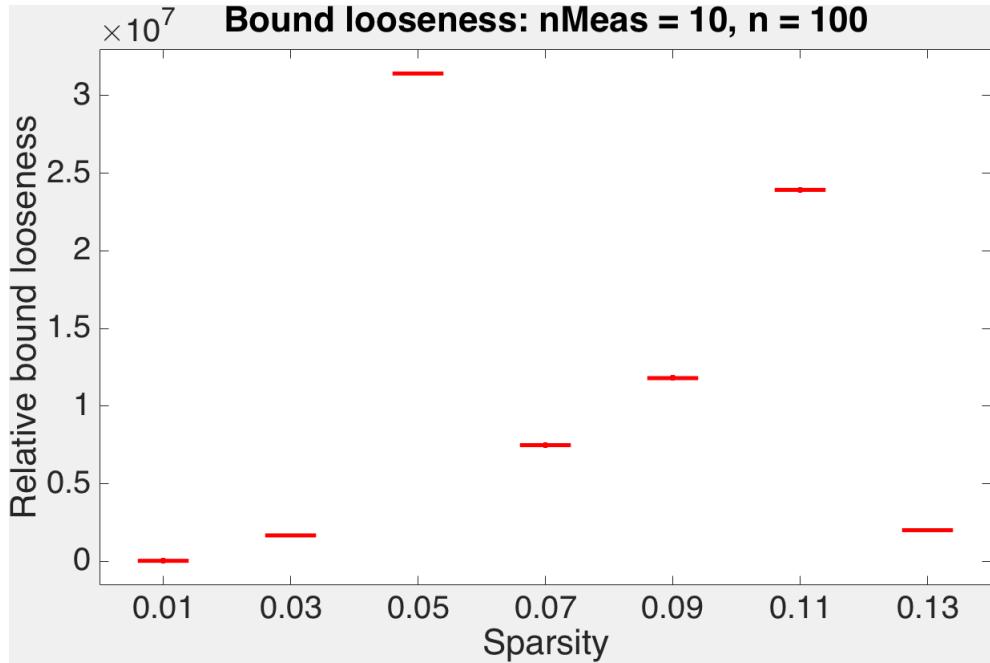


Figure 6.1: Relative difference between reconstruction error and error bound as a function of sparsity for sparse 50×1 signals, sensed with a 10×50 Gaussian random matrix

This, again, is not a very happy situation. While the bound shows promise, it works for very low sparsity levels. The coherence bound for all the Gaussian matrices we used here works for vectors that are 1-sparse, or when $s = 0.01$. One of the selling points of the bound, which was its applicability at sparsities the coherence cannot be applied at, now seems in doubt: the bound is practically unusable above $s = 0.03$, or 3 entries out of 100, given the large y -axis scale in the above plots.

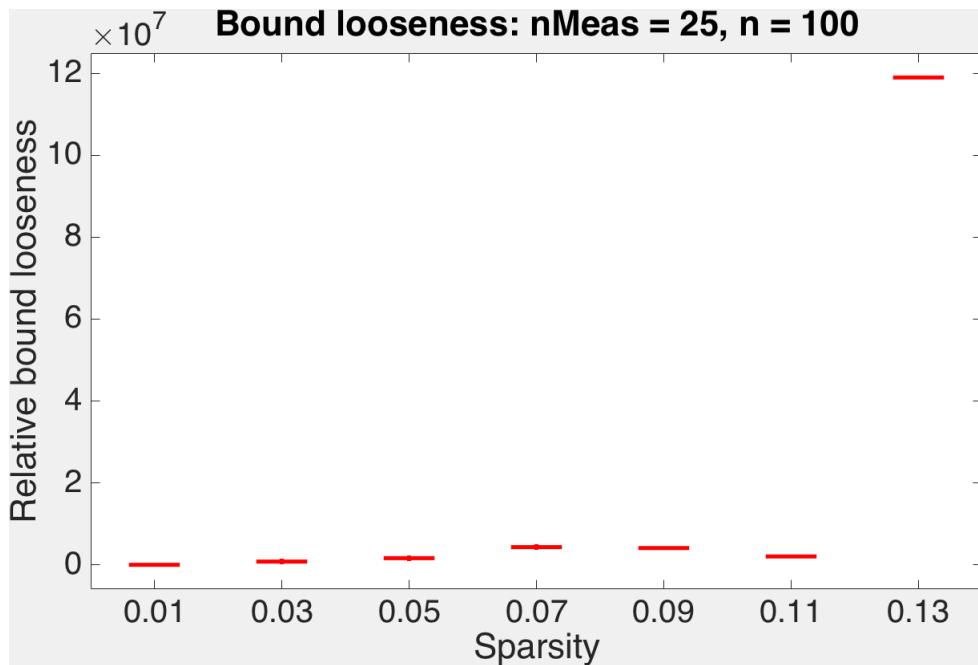


Figure 6.2: Relative difference between reconstruction error and error bound as a function of sparsity for sparse 50×1 signals, sensed with a 25×50 Gaussian random matrix

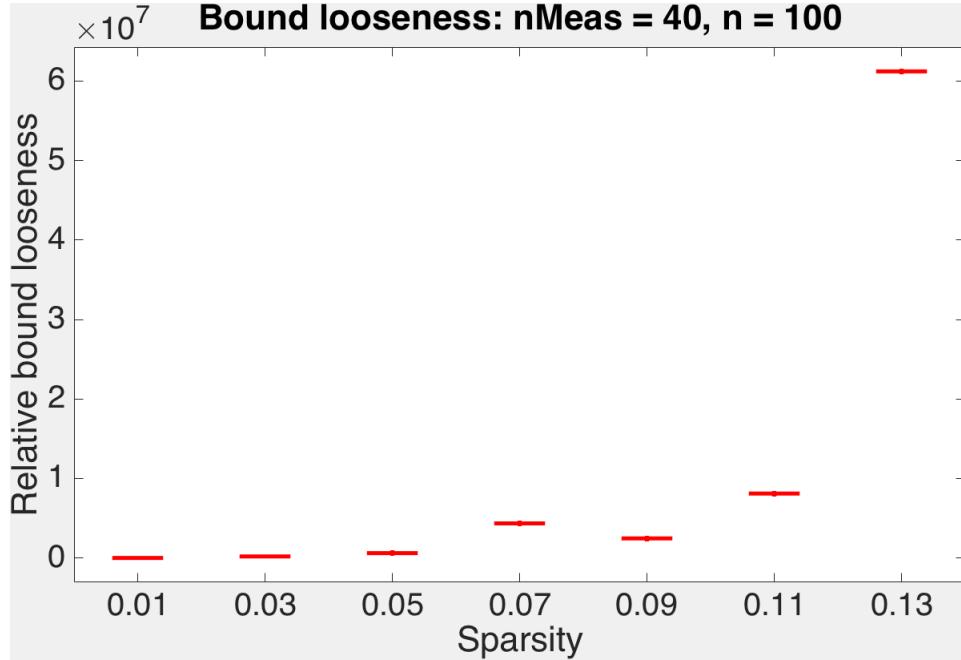


Figure 6.3: Relative difference between reconstruction error and error bound as a function of sparsity for sparse 50×1 signals, sensed with a 40×50 Gaussian random matrix

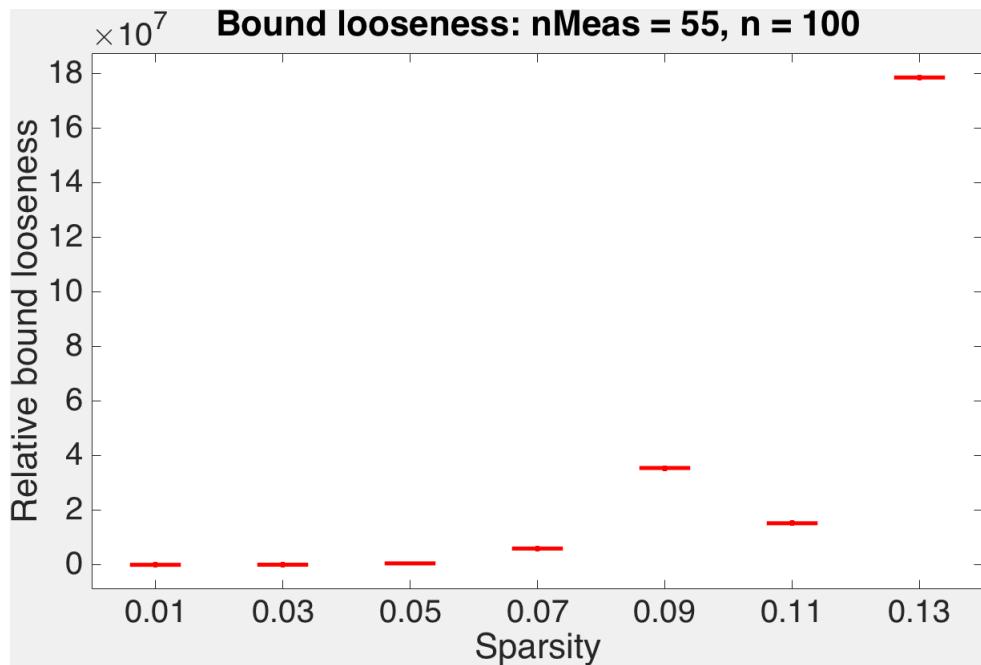


Figure 6.4: Relative difference between reconstruction error and error bound as a function of sparsity for sparse 50×1 signals, sensed with a 55×50 Gaussian random matrix

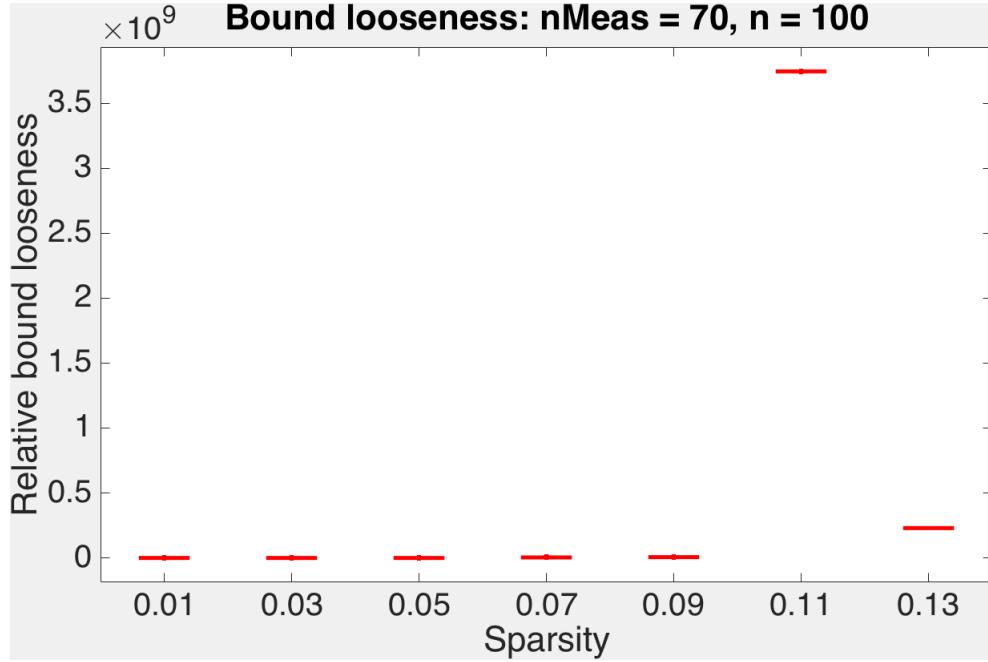


Figure 6.5: Relative difference between reconstruction error and error bound as a function of sparsity for sparse 50×1 signals, sensed with a 70×50 Gaussian random matrix

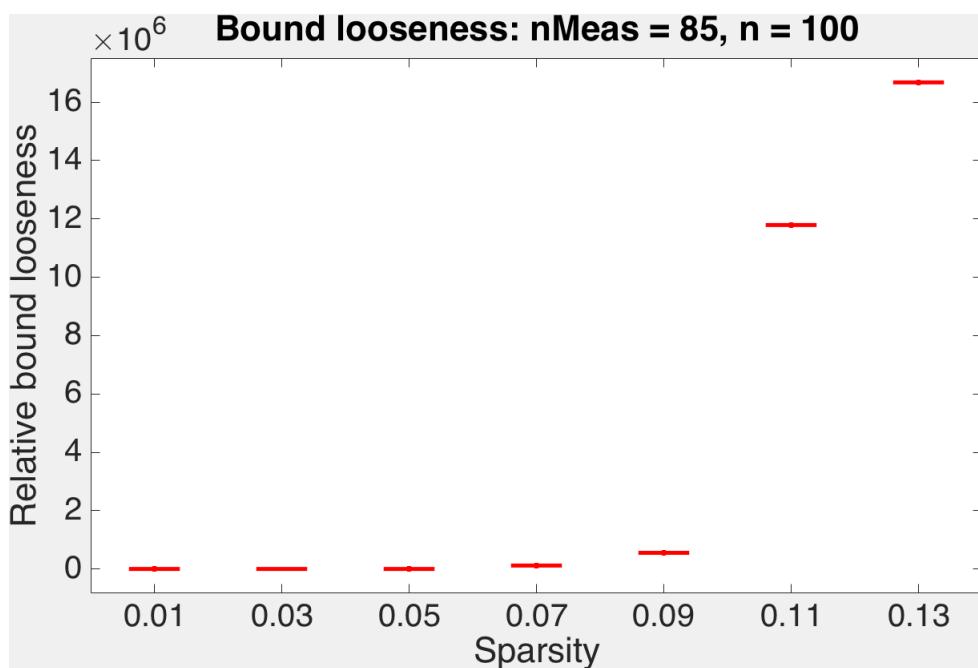


Figure 6.6: Relative difference between reconstruction error and error bound as a function of sparsity for sparse 50×1 signals, sensed with a 85×50 Gaussian random matrix

Chapter 7

Et tu, RIC?

The error bound introduced by [27], quoted in Eq. 6.3 has been interpreted as a compromise between coherence and the RIC. Coherence penalizes the dot products of normalized column pairs, and therefore the projections of columns on each other (see Eq. 2.6). This qualitatively expresses the ability of one column to approximate the other, or in other words, how well a 1-sparse combination of $n - 1$ of columns from A can represent the remaining column. Looking at Eq. 6.4, one realizes that the linear combination being penalized here is not just 1-sparse: all linear combinations of $n - 1$ columns with a coefficient vector λ are penalized for how well they can represent the n^{th} column, as long as $\|\lambda\|_1 \leq s$. While no theoretical claims can be made about which bound is better, a looseness analysis of the RIC bound seems to be a fitting final section of this paper.

The RIC-based bound error bound in [5] states that if $\delta_k < 0.307$

$$\|\hat{x} - x\|_2 \leq \frac{\epsilon}{0.307 - \delta_k} \quad (7.1)$$

Though the RIC is intractable to compute, it is computable for small sparsity levels for reasonably sized matrices. We, therefore, calculate the relative difference between the left and right hand sides of the bound in Eq. 7.1 with respect to the left hand side, which is the actual error between the actual vector and the reconstruction. We randomly generate $m \times 550$ matrices for $m = 275, 367, 412$ and 549 (Figs. 7.1, 7.2, 7.3 and 7.4 respectively), and random positive $k = 2$ -sparse 550×1 vectors. These numbers are selected so that the RIC condition $\delta_k \leq 0.307$ is satisfied. Then, reconstructing using the basis pursuit solver in Eq. 5.1, we calculate the l_2 error between the original and reconstructed vectors. We also perform the same analysis on a $k = 3$ -sparse vector for a 549×500 matrix (Fig. 7.5),

since the matrix instance we chose permits the RIC condition to hold.

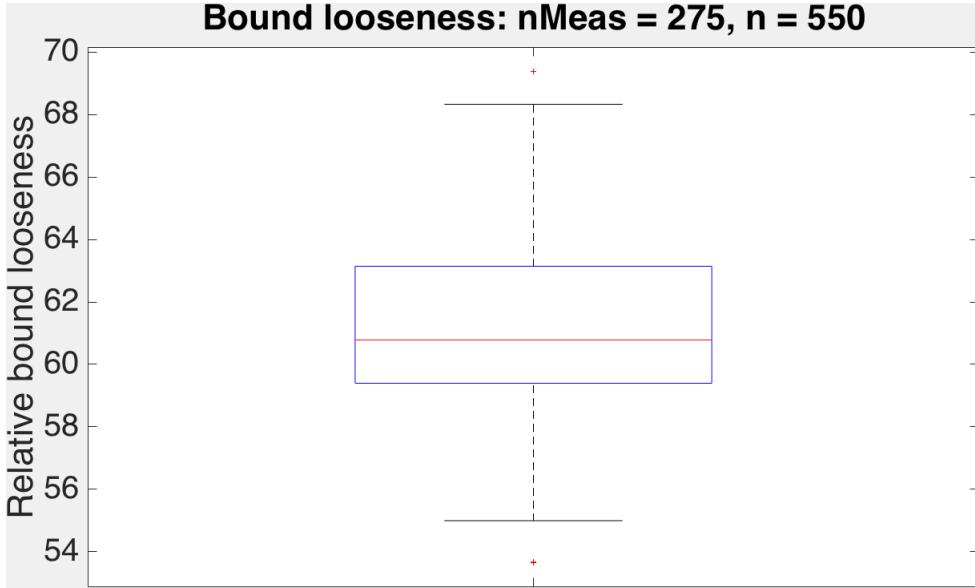


Figure 7.1: Relative difference between reconstruction error and the error bound in Eq. 7.1 for sparse 550×1 signals with $k = 2$, sensed with a 275×550 Gaussian random matrix

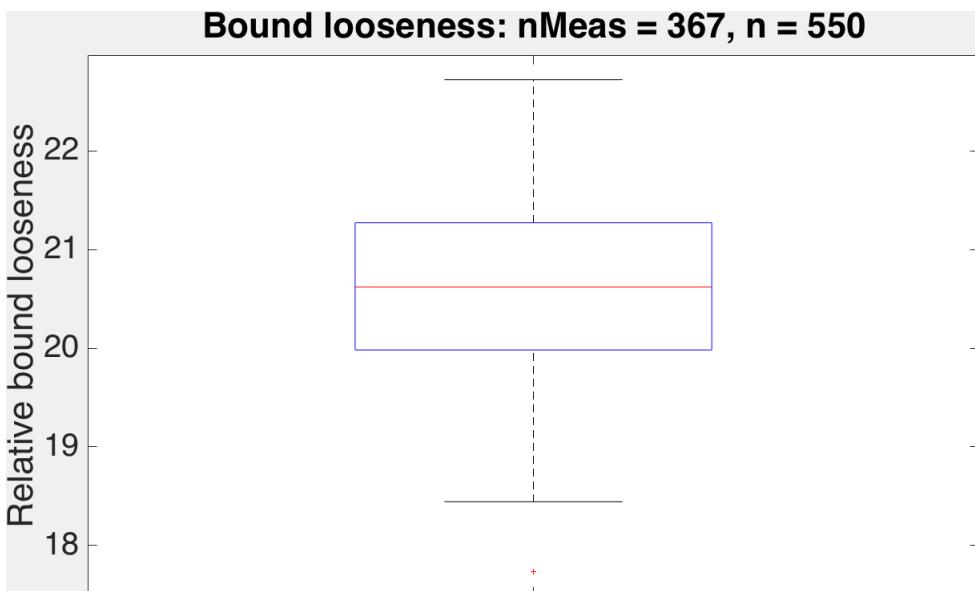


Figure 7.2: Relative difference between reconstruction error and the error bound in Eq. 7.1 for sparse 550×1 signals with $k = 2$, sensed with a 367×550 Gaussian random matrix

The values of looseness of bound in the RIC case are not close to zero either. The RIC, therefore, does not establish a tight bound on the recovery error. The problem optimizing

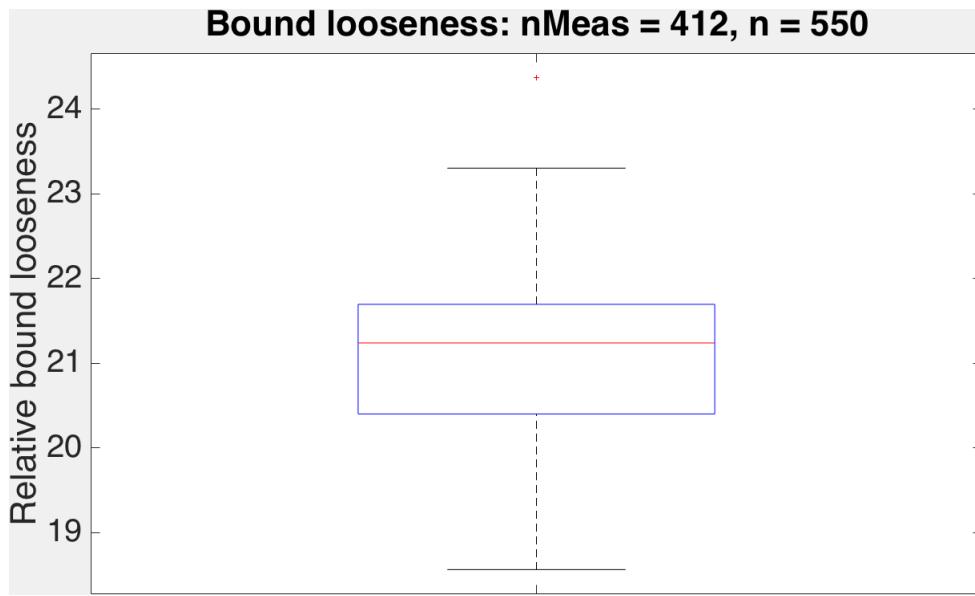


Figure 7.3: Relative difference between reconstruction error and the error bound in Eq. 7.1 for sparse 550×1 signals with $k = 2$, sensed with a 412×550 Gaussian random matrix

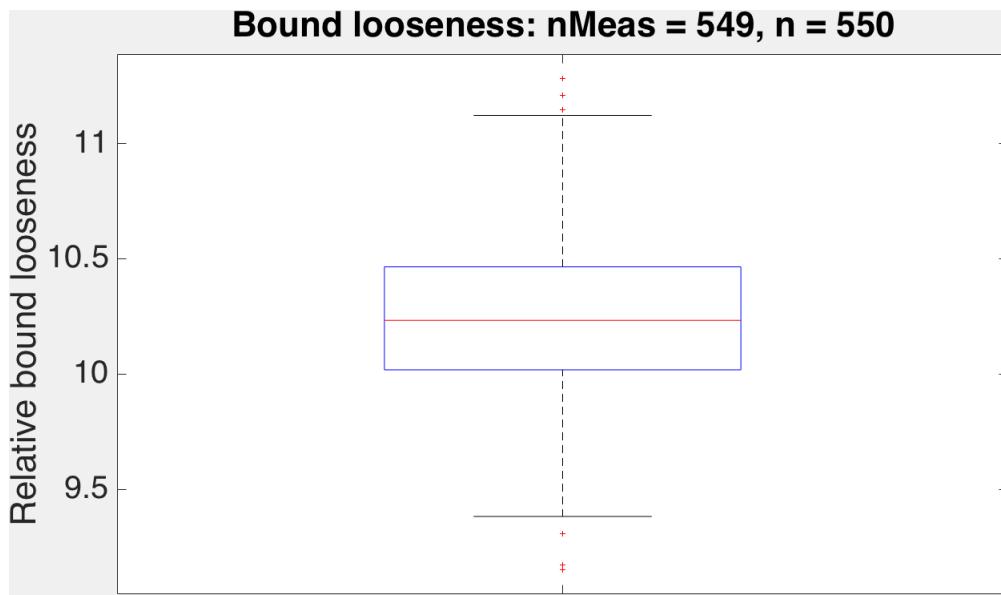


Figure 7.4: Relative difference between reconstruction error and the error bound in Eq. 7.1 sparse 550×1 signals with $k = 2$, sensed with a 549×550 Gaussian random matrix

with coherence now, we establish empirically, is twofold: the coherence establishes a loose bound on the RIC, and the RIC establishes a loose bound on the recovery error.

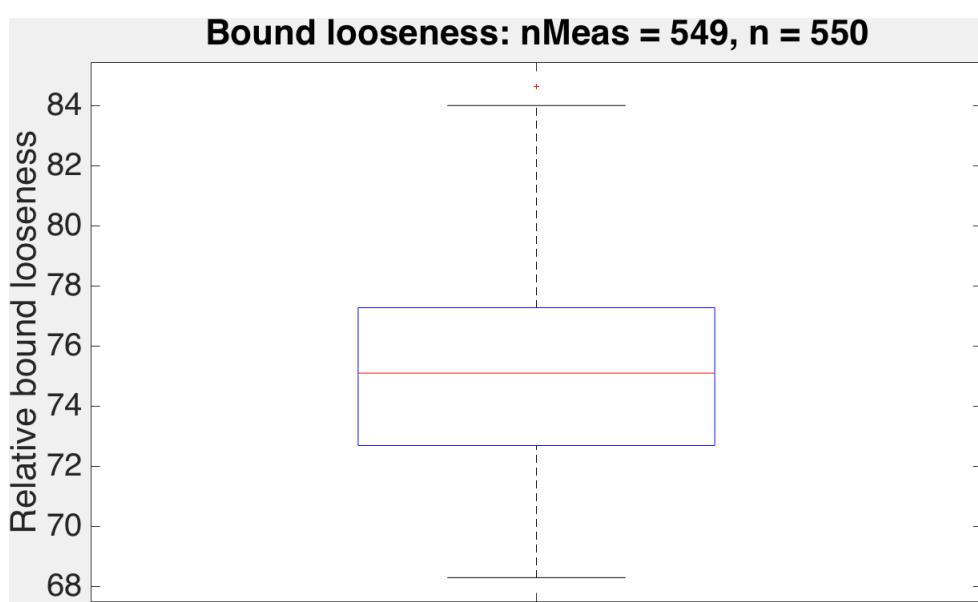


Figure 7.5: Relative difference between reconstruction error and the error bound in Eq. 7.1 for sparse 550×1 signals with $k = 3$, sensed with a 549×550 Gaussian random matrix

Chapter 8

The Average Case: A Proof of Concept

Investigating the source of looseness in compressed sensing bounds led us to a dead end: the looseness comes from steps that are at the core of the problem themselves. We saw in Chapter 5, for instance, how bounding the measurement noise was a major source of error, but a major step in the proof as well. The conclusion, then, was that while the error bound takes into consideration all vectors in the input space, the fact that the error bound is not met in a large dataset of vectors we simulated makes the consideration seem unnecessary. The bound is loose at the expense of taking into account a low probability set of vectors.

We now note that any compressed sensing bound, being universal to all vectors, characterizes recovery error in terms of sensing matrix properties and signal sparsity. No other property of the input signal is used. Therefore, in general, any approach targeting worst case errors in terms of these quantities will encounter the same worst case low probability vector pitfall as coherence. A possible direction of future work, therefore, is to circumvent these vectors by considering an average case error analysis.

The average case minimum mean square error (MMSE), introduced in [6] as

$$\mathbb{E}_{\mathbf{X}} \left[\text{tr} \left\{ (\mathbf{x} - \mathbb{E}_{\mathbf{X}|\mathbf{Y}}[\mathbf{x}|\mathbf{y}]) (\mathbf{x} - \mathbb{E}_{\mathbf{X}|\mathbf{Y}}[\mathbf{x}|\mathbf{y}])^T \right\} \right] \quad (8.1)$$

is successfully lower-bounded in the same paper by a function of the mutual information between \mathbf{x} and $\mathbf{y} = \Phi\mathbf{x}$. The aim is to design Φ so as to optimize the mutual information between \mathbf{x} and $\mathbf{y} = \Phi\mathbf{x}$. The mutual information along with an entropy term is proved to be a lower bound, but that certainly may not necessarily decrease the average recovery error.

A full analysis of average case error, however, requires estimating a posterior on the space of input vectors as well as assuming a specific statistical model for noise, and involves the MMSE in Eq. 8.1, an intractable quantity to calculate for many commonly occurring priors and likelihoods. Such an analysis is beyond the scope of this paper. A sampling-based approach with an appropriate prior over the space of input signals seems to be a direction in which to proceed for any tractable sensing matrix design considering the average case. This method, however, will require coming up with a prior distribution such that the posterior $\mathbb{E}[\mathbf{x}|\mathbf{y}]$ can be sampled from. The parameterization can then be used to optimize sensing matrices with respect to the expectation.

For the purposes of providing a proof-of-concept, we circumvent the calculation of the MMSE as follows: we take a dataset of random vectors from the prior distribution we are interested in, use basis pursuit for signal recovery instead of the full Bayesian $\mathbb{E}[\mathbf{x}|\mathbf{y}]$. The MMSE is then approximated as the mean squared error on this dataset. With this, we perform a random search on the sensing matrix (A). In the two cases we have considered in this paper – the CACTI camera and general matrices – we show superior recovery performance with matrices designed with the average case.

8.1 General sensing matrices

We design $m \times 50$ matrices for $m = 25, 12$ and 8 (Figs. 8.1, 8.2 and 8.2 respectively) targeted at perfectly sparse vectors with sparsities of $s = 0.2, 0.12$ and 0.08 respectively. We take a set $\{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_z\}$ of z randomly generated support sets, and draw random vectors having supports from this set (this is done so that we do not have to explore the entire set of sparse vectors at a given sparsity, limiting the number of sparse recovery problems we have to solve per objective function evaluation, and hence reducing computational complexity). A random search is done on the elements of the matrix to minimize the squared error on this dataset. We then compare, on an independently generated test set of vectors having supports from these sets, the performance of a matrix designed to minimize coherence and the random matrix we started from with our matrix. It is clear from these boxplots that our method does no worse than coherence-based design in the general sensing matrix case.

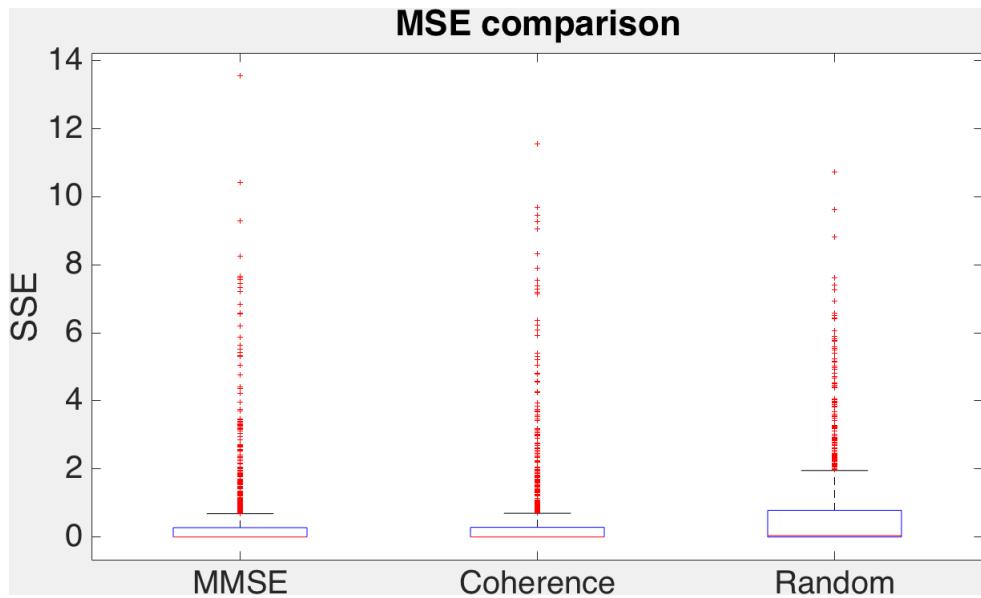


Figure 8.1: Comparison of sum of squared errors on a dataset of 250 sparse vectors, between 8×50 general sensing matrices optimized with mean square error, optimized with coherence and a general random matrix, targeted at sparsities of 0.08

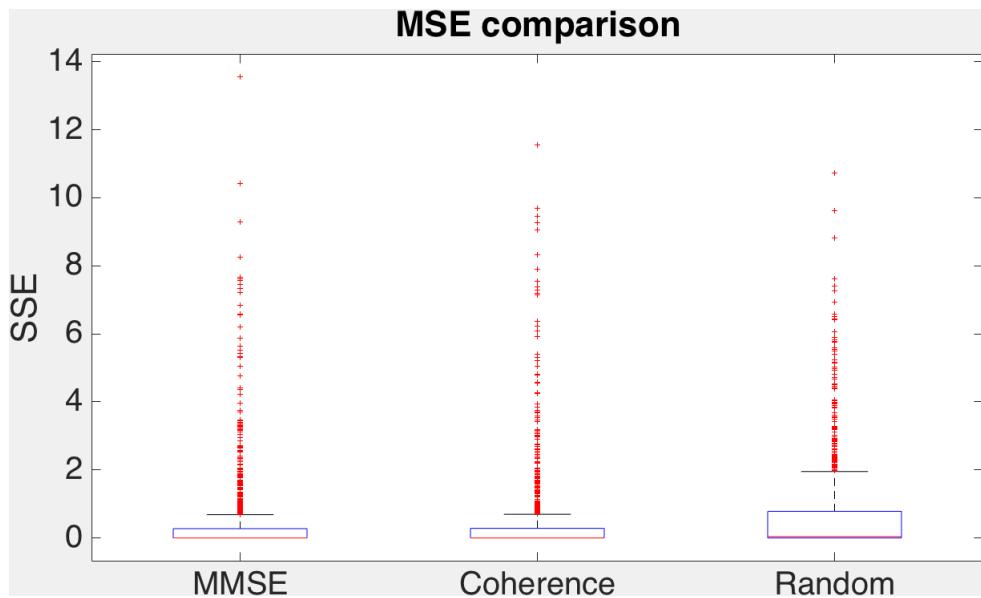


Figure 8.2: Comparison of sum of squared errors on a dataset of 250 sparse vectors, between 12×50 general sensing matrices optimized with mean square error, optimized with coherence and a general random matrix, targeted at sparsities of 0.12

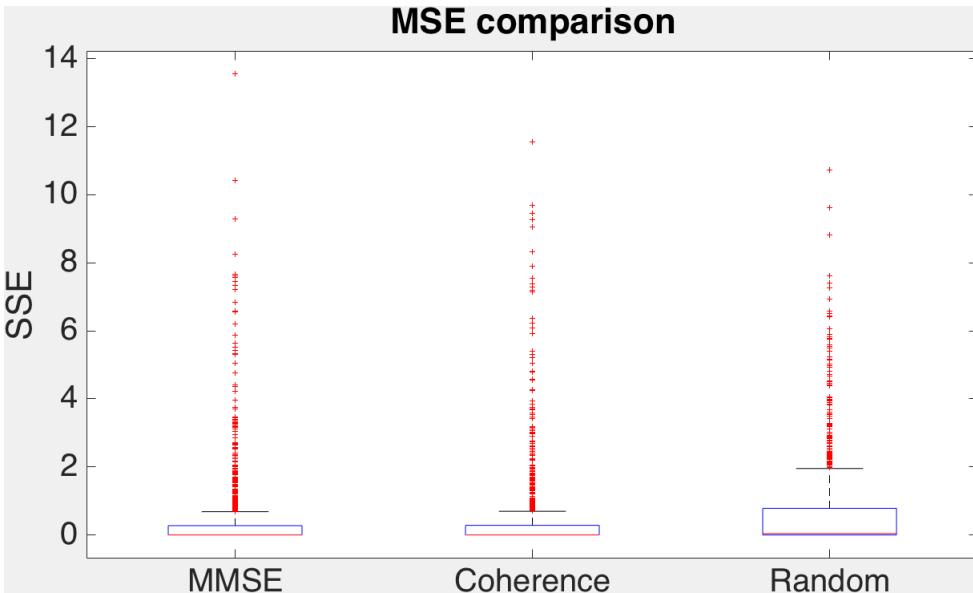


Figure 8.3: Comparison of sum of squared errors on a dataset of 250 sparse vectors, between 25×50 general sensing matrices optimized with mean square error, optimized with coherence and a general random matrix, targeted at sparsities of 0.2

8.2 In the CACTI camera

Similarly, we design 8×8 codes for $T = 2, 4$ and 6 (Figs. 8.4, 8.5 and 8.6 respectively) targeted at perfectly sparse vectors with sparsities of $s = 0.2, 0.12$ and 0.08 respectively. Optimizing the code using a random dataset of vectors frown from similarly generated set of support sets, we compare the performance of the designed matrix on an independent test dataset of vectors having these supports.

The fact that mean squared error design does better than coherence design and random codes, where coherence design did worse than random codes, demonstrates clearly the potency of the MSE-based design over coherence-based design. It turns out, interestingly, that if we design matrices without the support set constraint with the same number of train vectors, MSE-based design still beats coherence and random by the same margin as above.

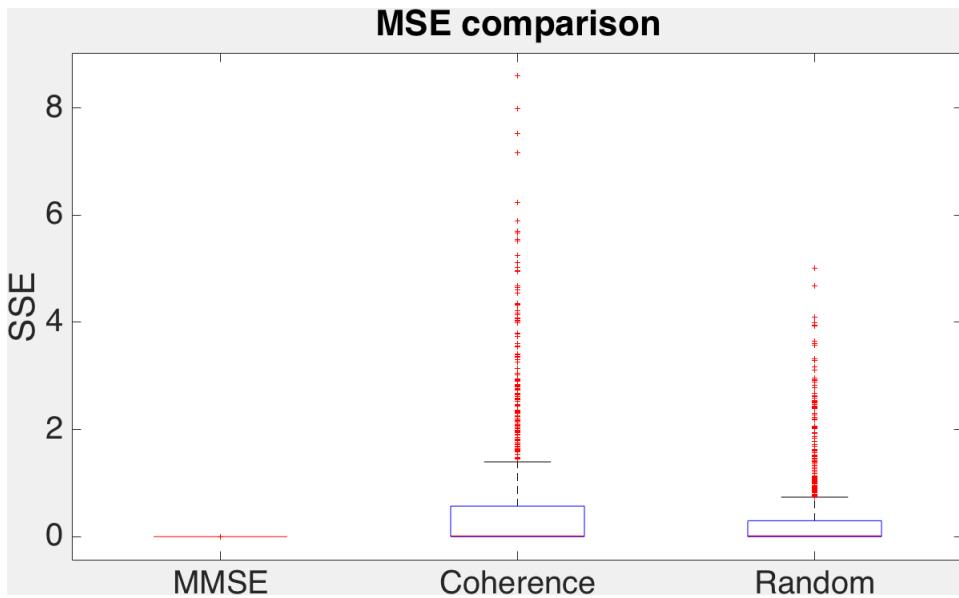


Figure 8.4: Comparison of sum of squared errors on a dataset of 250 sparse vectors, using 8×8 positive codes with $T = 2$, optimized using the mean square error, optimized with coherence and a positive random code, targeted at sparsities of 0.2

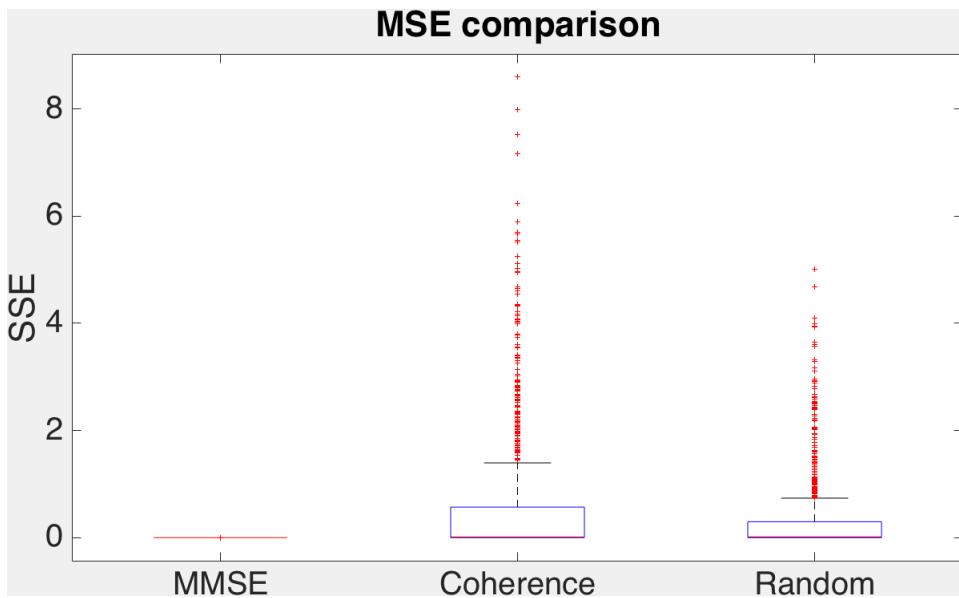


Figure 8.5: Comparison of sum of squared errors on a dataset of 250 sparse vectors, using 8×8 positive codes with $T = 4$, optimized using the mean square error, optimized with coherence and a positive random code, targeted at sparsities of 0.12

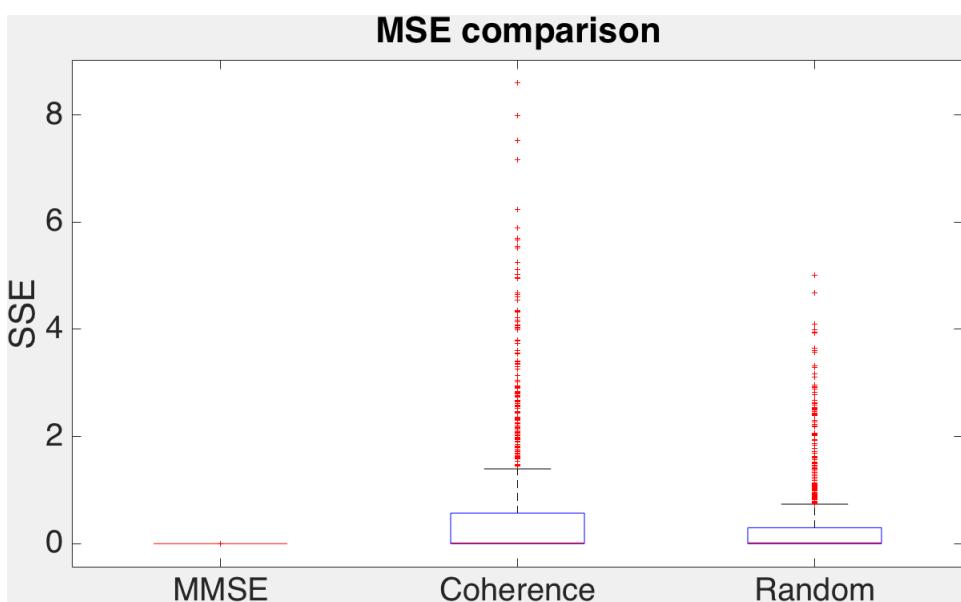


Figure 8.6: Comparison of sum of squared errors on a dataset of 250 sparse vectors, using 8×8 positive codes with $T = 6$, optimized using the mean square error, optimized with coherence and a positive random code, targeted at sparsities of 0.08

Chapter 9

Practical Constraints in Compressed Sensing Design

OPTIMIZATION methods for matrices, whether successful or not, must consider the important issue of perturbations in sensing matrices. These arise from designed matrices being floating point numbers of high precision, while only a finite set of measurement gains might be allowed by hardware. These kinds of effects are best modeled by quantization. Physical manufacturing and aging can cause continuously varying noise in the quantized mask. We present here an analysis of effects of these perturbations. Though admittedly simple, it must be stressed that evaluating designed matrices for stability under perturbations is an important step.

9.1 Quantization errors

Quantization is a major effect in both the cameras we considered. In the camera in [14], the quantization arises from the fact that exposure times, which are used to generate the codes that linearly combine input frames, can take values that are integer multiples of the time resolution of the camera shutter. In the CACTI camera [18], the mask is realized as a transparency with space-varying transmission. Manufacturing such transmission is, again, limited to the precision of the instrument that manufactures the mask. In other systems like the Rice single pixel camera [7], compressive sampling is realized by turning on and off mirrors, which correspond to a binary encoding. We test the sensitivity of matrices to

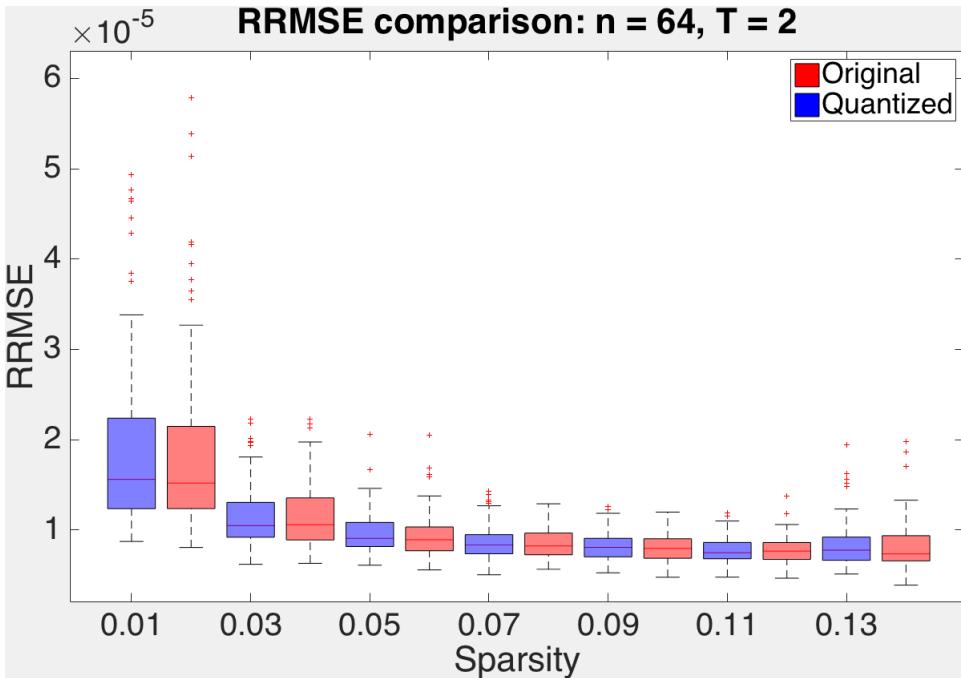


Figure 9.1: Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 2$ in the coded source separation scenario. Code elements rounded off to the nearest 0.2

perturbation by quantizing the mask values and comparing reconstruction performance (RRMSE error on a toy dataset) to what we achieve with exact values. Conditional on the matrix performing similarly under quantization, we can use offline calibration as outlined below to mitigate quantization well along with noise.

9.1.1 Effects on coded source separation

We generate 8×8 random vectors sparse in the 2D DCT domain and pass them through both the quantized and original sensing matrices. We then solve basis pursuit to obtain reconstructions using the respective matrices, from which we obtain relative errors. A comparison of errors given by a matrix and its quantized version is shown in the figures below, for quantization levels of 0.2 and 1. Noise bounded in norm at $\epsilon = 10^{-5}$ was added.

The fact that RRMSEs don't deviate much from designed matrix RRMSE values is an encouraging result. It means that under quantization noise, matrices behave similar to what the original matrices used to behave like, and hence offline calibration can be used to correct for and obtain exact values of code elements.

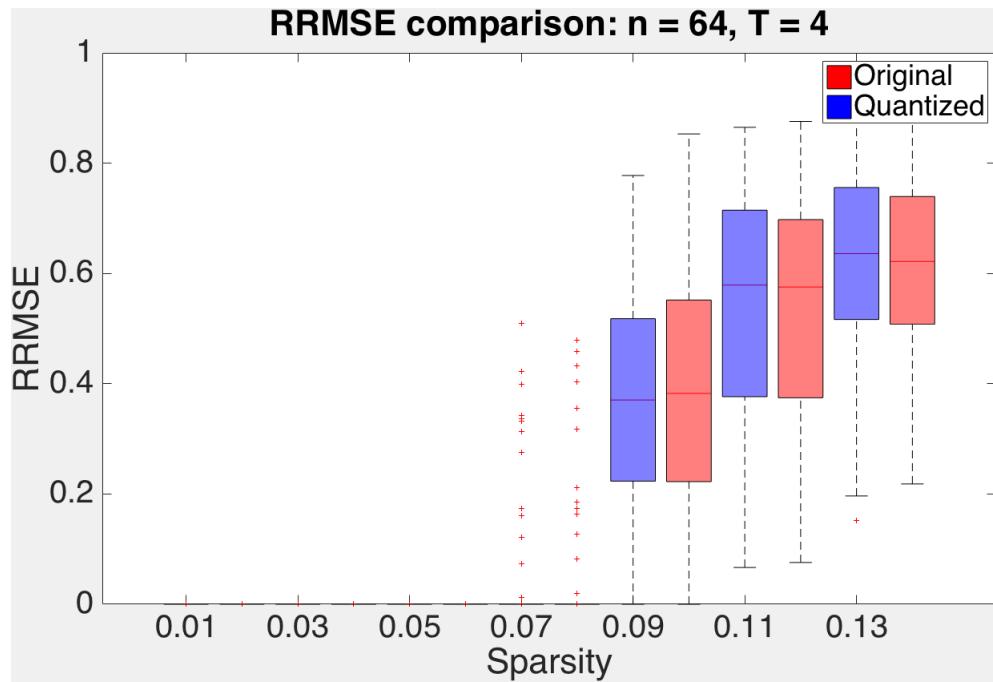


Figure 9.2: Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 4$ in the coded source separation scenario. Code elements rounded off to the nearest 0.2

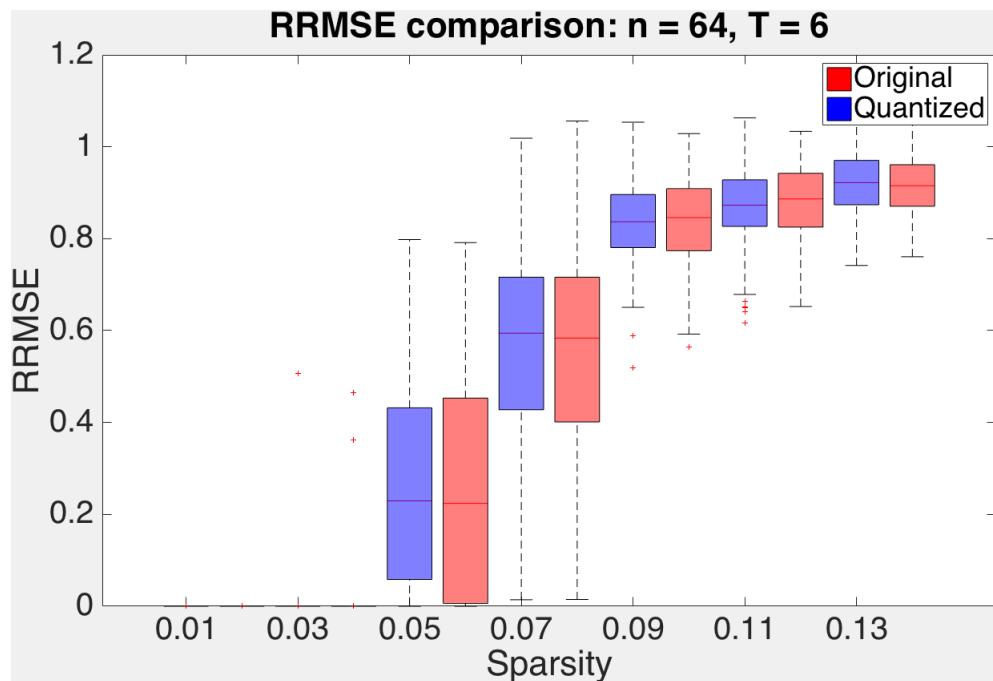


Figure 9.3: Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 6$ in the coded source separation scenario. Code elements rounded off to the nearest 0.2

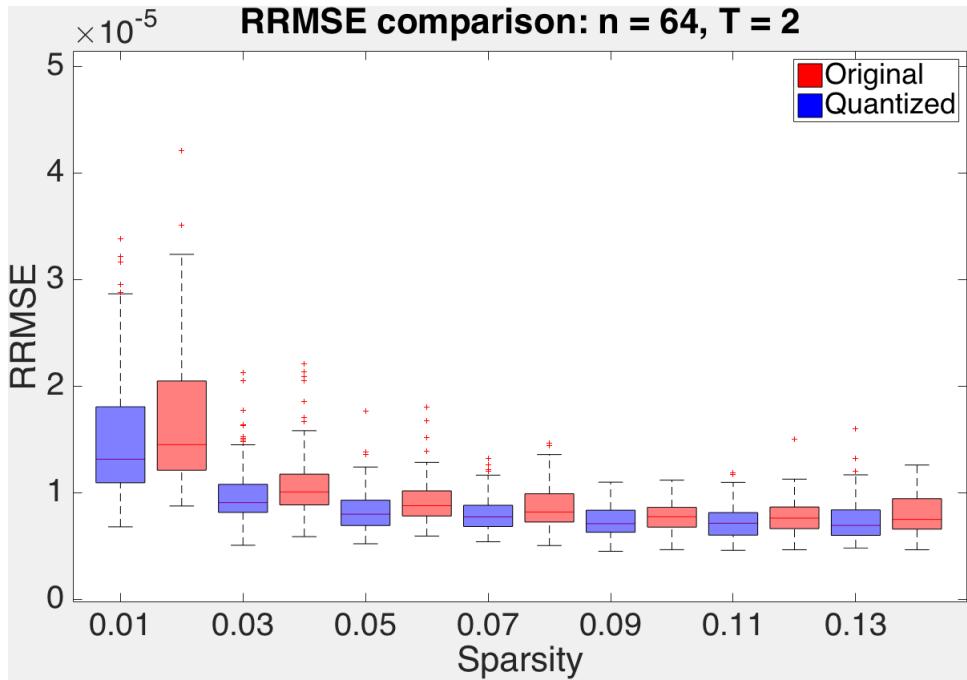


Figure 9.4: Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 2$ in the coded source separation scenario. Code elements rounded off to binary

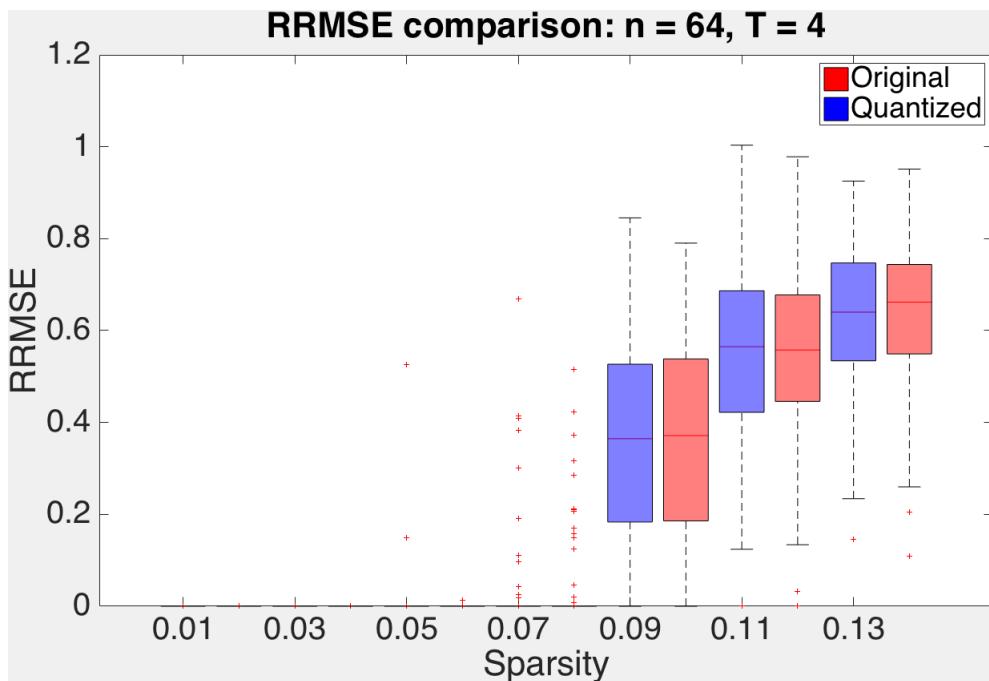


Figure 9.5: Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 4$ in the coded source separation scenario. Code elements rounded off to binary

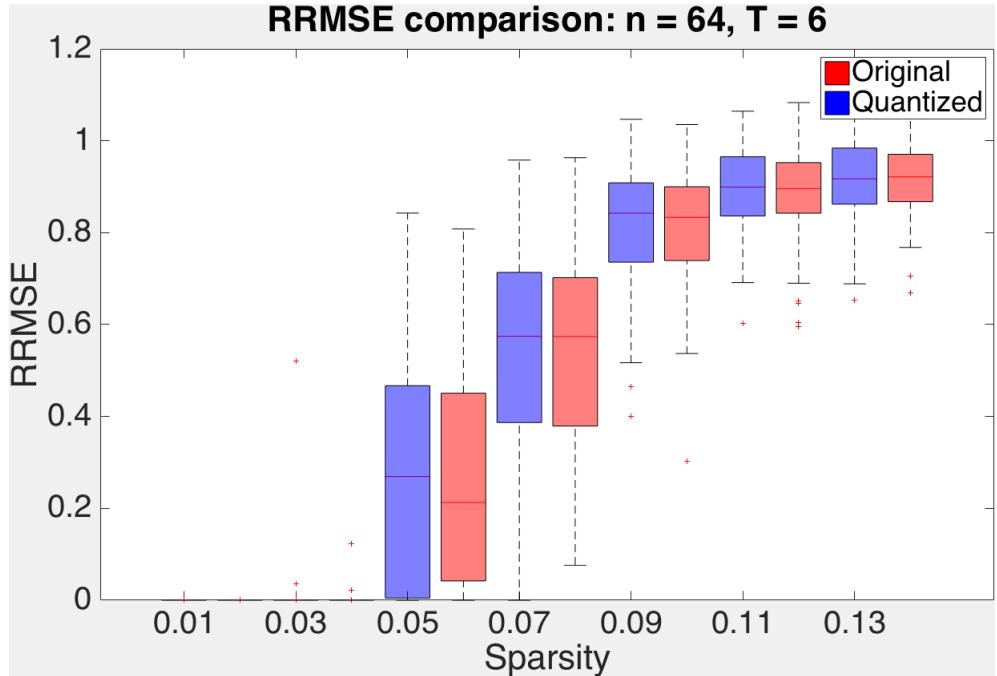


Figure 9.6: Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 6$ in the coded source separation scenario. Code elements rounded off to binary

9.1.2 Effects on the CACTI camera

Repeating the experiment on the CACTI camera, we replicate the results obtained in the section above. Offline calibration can, therefore, also be used in the CACTI camera for accurate recovery of input signals.

9.2 Calibration

Continuously varying noise introduced in the sensing matrix needs to be corrected for to achieve accurate reconstruction. We consider calibrating the sensing matrices for the CACTI camera. This can be attempted either online or offline, as described in the sections below.

To introduce some notation, we consider the general source separation model, with y being a compressive measurement, and $\{x_i\}_T^{i=1}$ being input frames. Let ϕ be the original codes, which on degradation, go to $\tilde{\phi}$. Consistent with our original definitions, we let the j^{th} element of ϕ be ϕ^j . Let the noise introduced by the degradation be $\delta^i = \tilde{\phi}^i - \phi^i$. Given

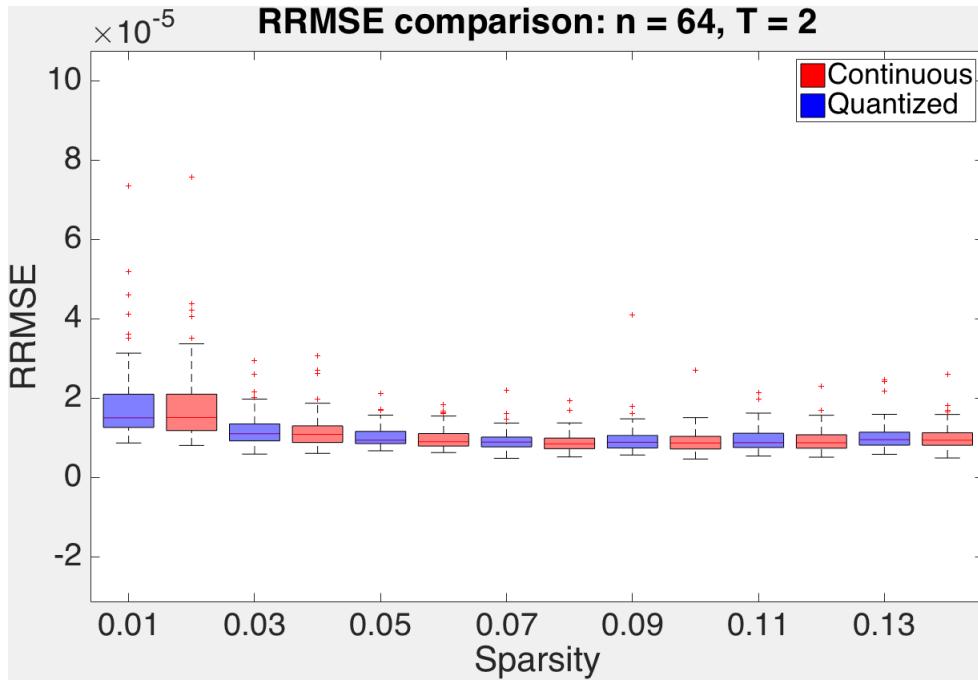


Figure 9.7: Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 2$. Code elements rounded off to the nearest 0.2

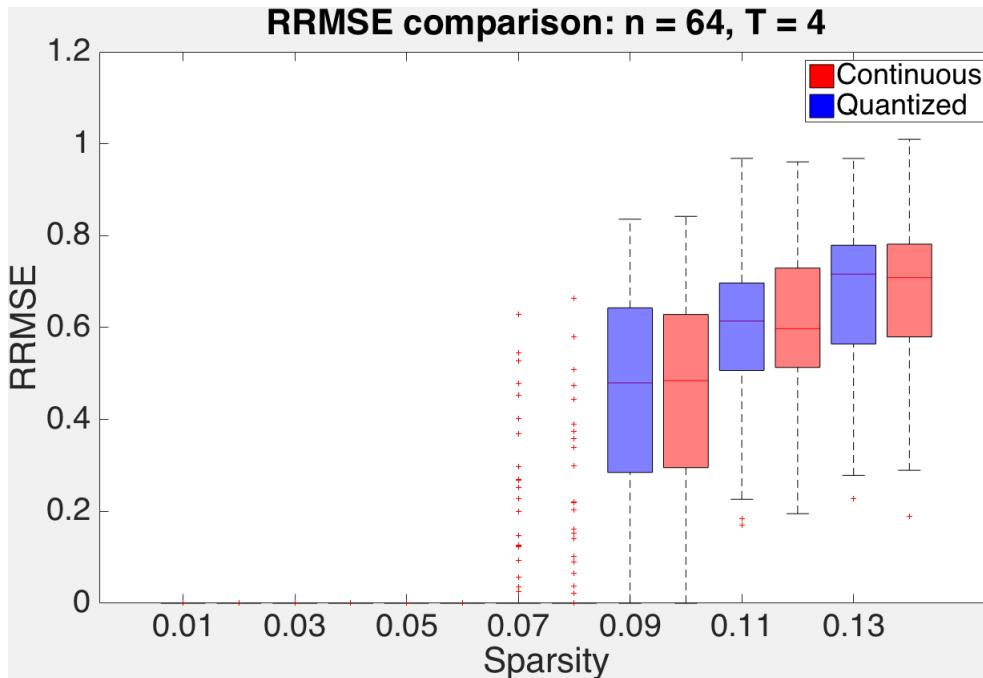


Figure 9.8: Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 4$. Code elements rounded off to the nearest 0.2

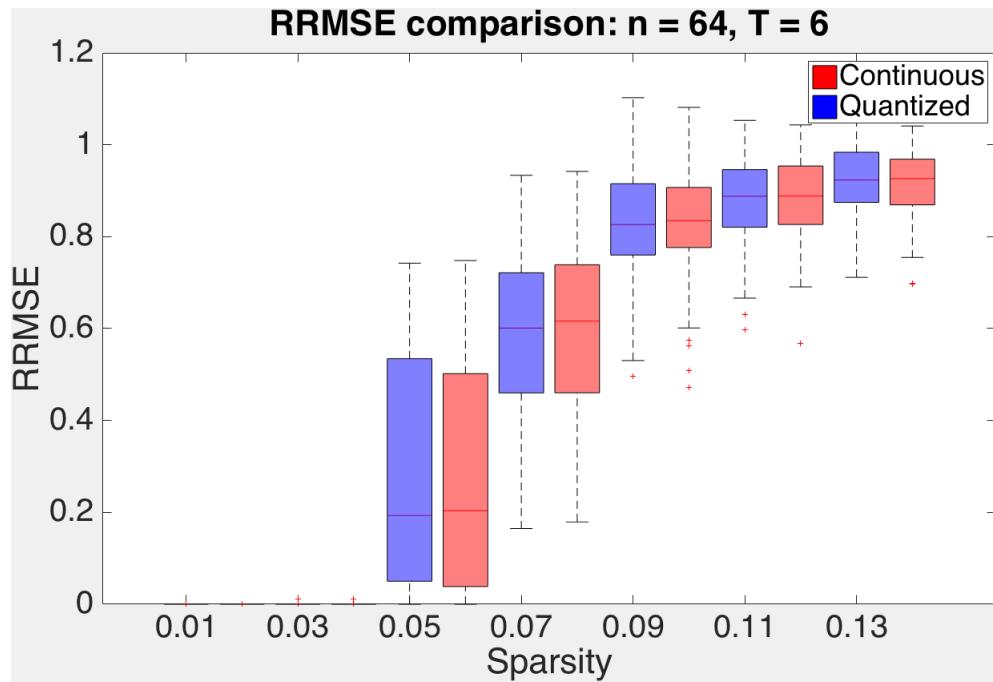


Figure 9.9: Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 6$. Code elements rounded off to the nearest 0.2

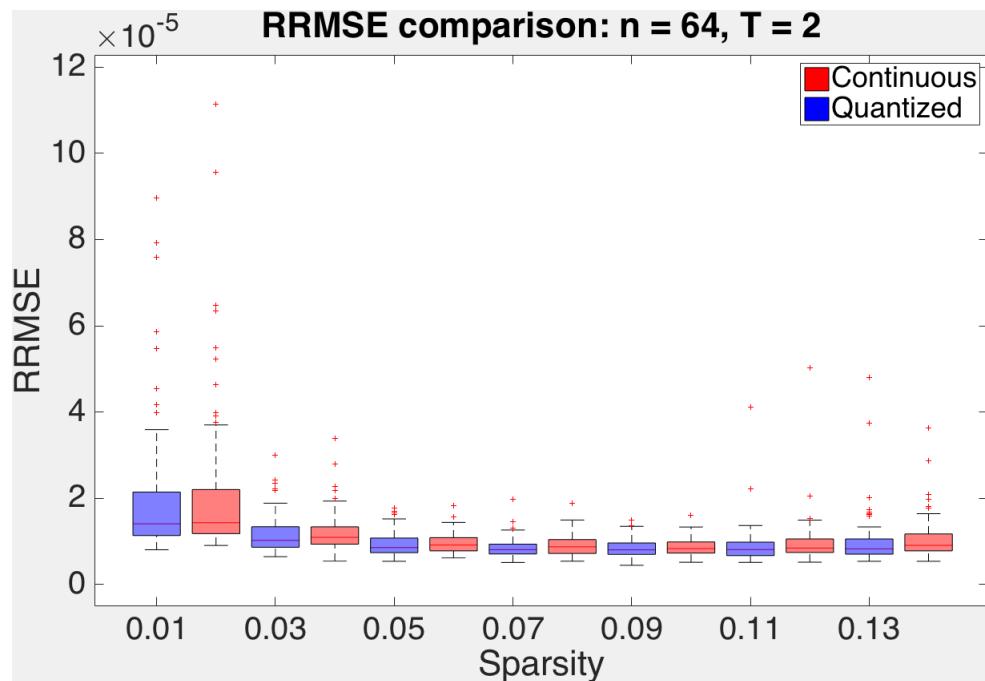


Figure 9.10: Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 2$. Code elements rounded off to binary

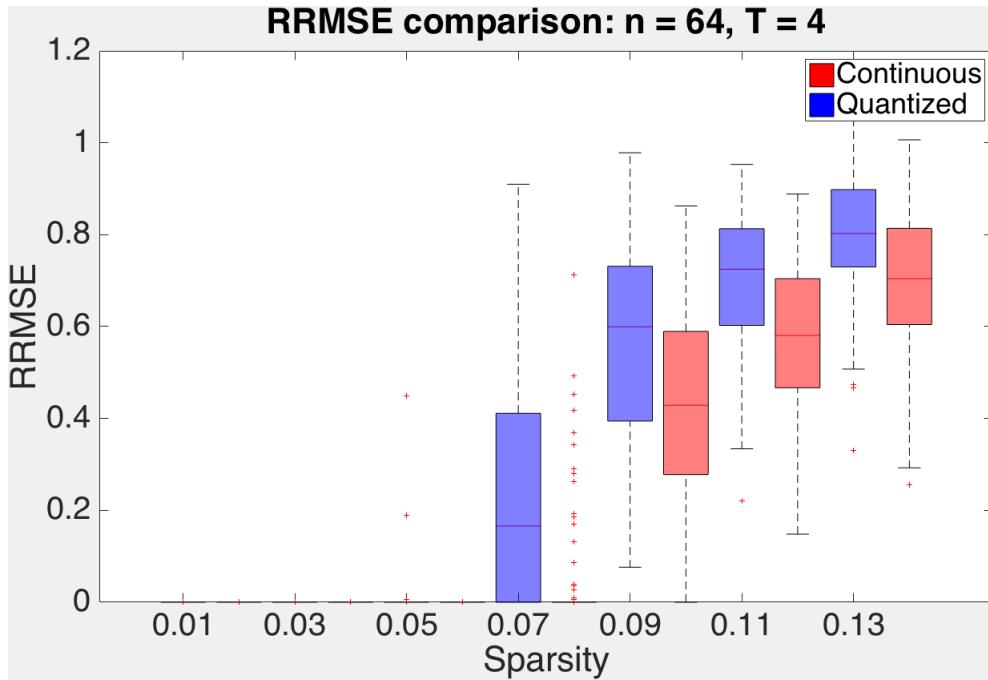


Figure 9.11: Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 4$. Code elements rounded off to binary

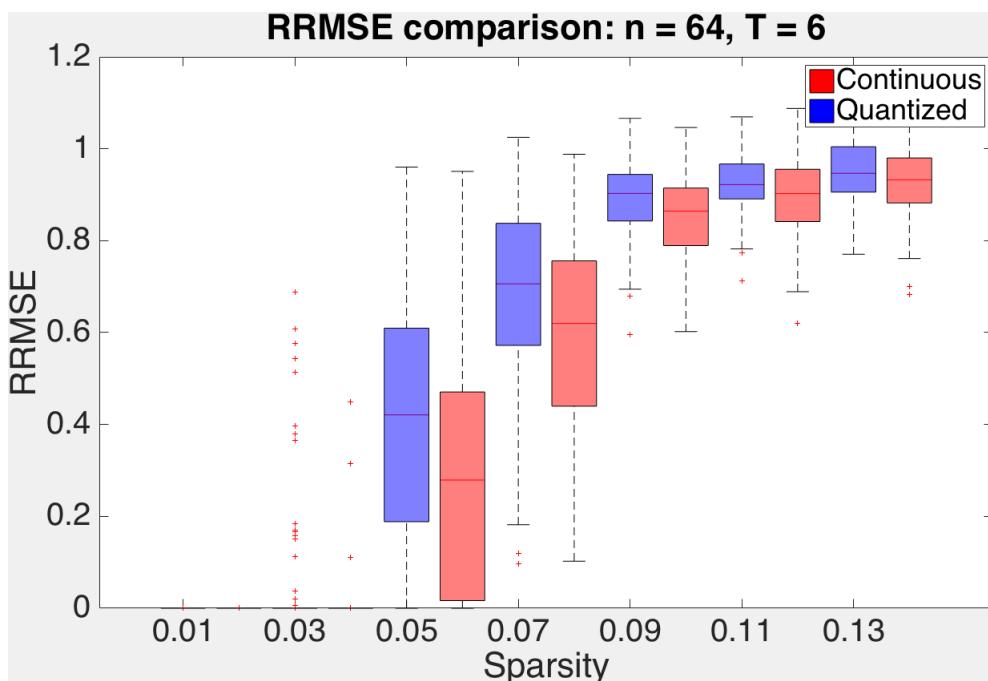


Figure 9.12: Comparison between RRMSEs with original and quantized matrices as a function of sparsity on 8×8 vectors sparse in 2D DCT, combined with $T = 6$. Code elements rounded off to binary

this, the sensing model is

$$y = \begin{pmatrix} \tilde{\phi}_1 & \tilde{\phi}_2 & \dots & \tilde{\phi}_T \end{pmatrix} \begin{pmatrix} x_1 & x_2 & \dots & x_T \end{pmatrix}^T \quad (9.1)$$

$$= \begin{pmatrix} \tilde{\phi}_1 & \tilde{\phi}_2 & \dots & \tilde{\phi}_T \end{pmatrix} \begin{pmatrix} D\alpha_1 & D\alpha_2 & \dots & D\alpha_T \end{pmatrix} \quad (9.2)$$

However, in contrast to the earlier case, we do not know $\tilde{\phi}$, just ϕ . The reconstruction technique we would have used without considering noise effects would have been

$$\min_{\alpha} \|\alpha\|_1 \text{ subject to } y = \Phi\Psi\alpha, \quad \alpha = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_T \end{pmatrix}^T \quad (9.3)$$

Considering noise effects, with an estimate δ , the improved reconstruction technique is

$$\min_{\alpha} \|\alpha\|_1 \text{ subject to } y = (\Phi + \Delta)\Psi\alpha, \quad \alpha = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_T \end{pmatrix}^T \quad (9.4)$$

where Δ is defined appropriately.

9.2.1 Online calibration

Online calibration entails a joint estimation of δ and $\{x_i\}_{i=1}^T$ on the go as measurements come in. Multiple sets $\{x_i^k\}_{i=1, k=1}^{i=T, k=K}$ can also be used to achieve better estimation of δ . A tempting method to try out is an alternating minimization. Setting an initial value for δ , one can solve Eq. 9.4 individually for all K sets to get an estimate for the input frames. Refining the estimate for δ , then, reduces to finding those δ consistent with the input frames. This is an overcomplete linear system in δ , yielding n equations for each set of measurements. There are n variables involved, and one can do a pseudoinverse solution. The magnitudes of the elements of δ are typically small, and can be clipped at appropriate values. The linear system is detailed in Eq. 9.9.

The problem with this method is that because the effective dictionary is overcomplete, a fit is always possible for any given measurement. The alternating optimization provides no mechanism to ensure that the resulting x s are the sparsest among the possibilities as well. This points to selecting the sparsest input signals as well as the smallest error deviations. One can, therefore, solve the joint optimization problem

$$\left(\{x_i^k\}_{i=1, k=1}^{i=T, k=K}, \delta \right) = \arg \min_{\{\tilde{x}_i^k\}_{i=1, k=1}^{i=T, k=K}, \tilde{\delta}} \sum_{k=1}^K [(y^k - \Phi\Psi\alpha^k) + \lambda_1 \|\alpha^k\|_1] + \lambda_2 \|\delta\|_2 \quad (9.5)$$

This joint optimization framework, however, needs a tuning of the parameters λ_1 and λ_2 . We do not attempt the joint optimization in this work.

9.2.2 Offline calibration

Offline calibration entails estimation of δ from a given set of ground truth signals that are propagated through the forward model of the system to generate measurements. The method is exactly the one used to refine the estimate of δ in the second step of the alternating optimization above. To detail, we have in terms of the elementwise product ·

$$y^k = \sum_{t=1}^T \tilde{\phi}_t \cdot x_t^k \quad (9.6)$$

$$= \sum_{t=1}^T \tilde{\phi}^{p_t} \cdot x_t^k \quad (9.7)$$

$$= \sum_{t=1}^T (\phi^{p_t} + \delta^{p_t}) \cdot x_t^k \quad (9.8)$$

$$\implies y^k - \phi^{p_t} \cdot x_t^k = \delta^{p_t} \cdot x_t^k \quad (9.9)$$

where δ_t^p and ϕ_t^p are the appropriate circularly shifted versions of δ and ϕ . Eq. 9.9 constitute n equations – one for each element of y – in n variables – the entries of δ . Stacking these equations over all k , one gets nK equations in n variables, which can be solved using a pseudoinverse.

9.2.3 Calibration results

We do not attempt the joint optimization in this work. Instead, we perform offline calibration for the CACTI camera to demonstrate that the simple method above produces results much better than the ones we get without considering the effects of noise.

We generate $K = 100$ random input signals x^k for $T = 2, 4, 6$, and generate the compressive measurement y^k for each vector using Eq. 9.2. These are then used to calibrate by finding the values of elements of δ . Finally, a validation set of 1000 vectors is used to test the improvement in quality due to calibration, by finding their recovery error using both calibrated and uncalibrated matrices. This is repeated across various amounts of noise added to ϕ .

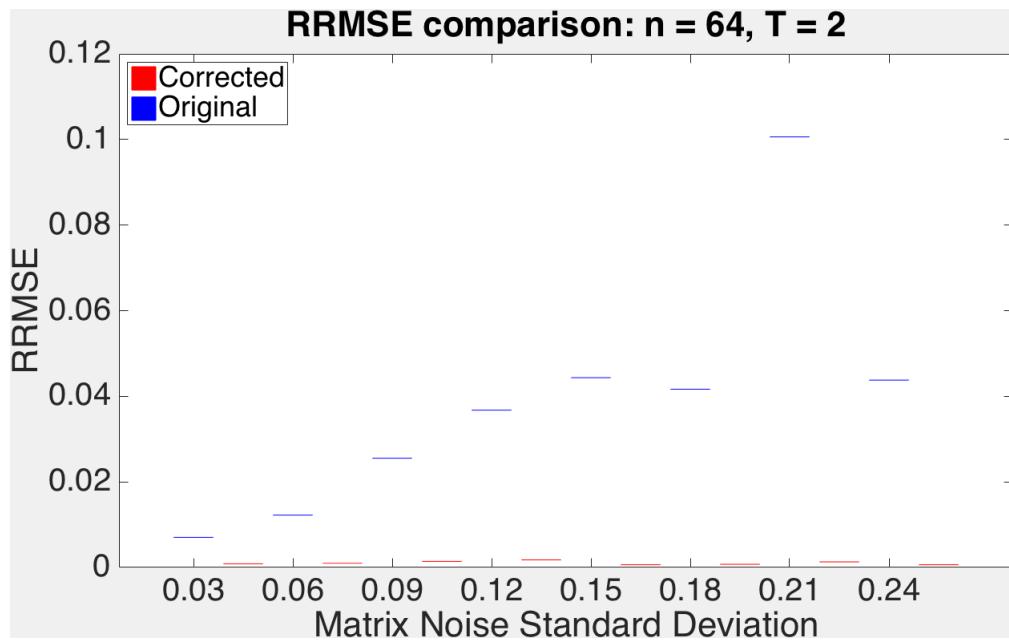


Figure 9.13: Average RRMSE for calibrated and uncalibrated matrices as a function of matrix noise standard deviation for 8×8 signals, sparse in 2D DCT, combined with $T = 2$. Permutations: [5, 3; 6, 8]

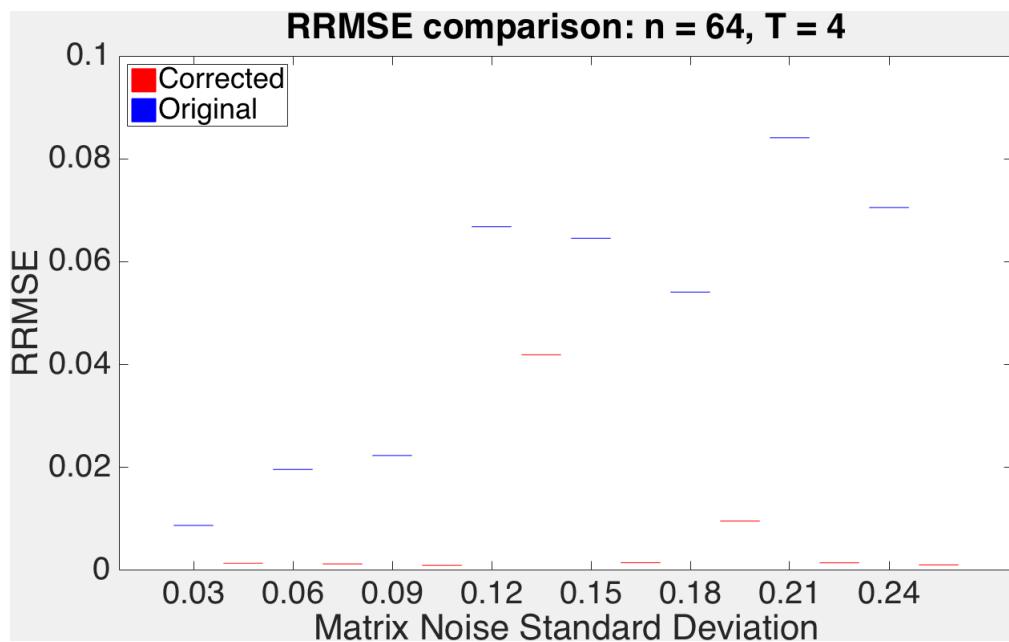


Figure 9.14: Average RRMSE for calibrated and uncalibrated matrices as a function of matrix noise standard deviation for 8×8 signals, sparse in 2D DCT, combined with $T = 4$. Permutations: [7, 8; 2, 8; 6, 1; 3, 5]

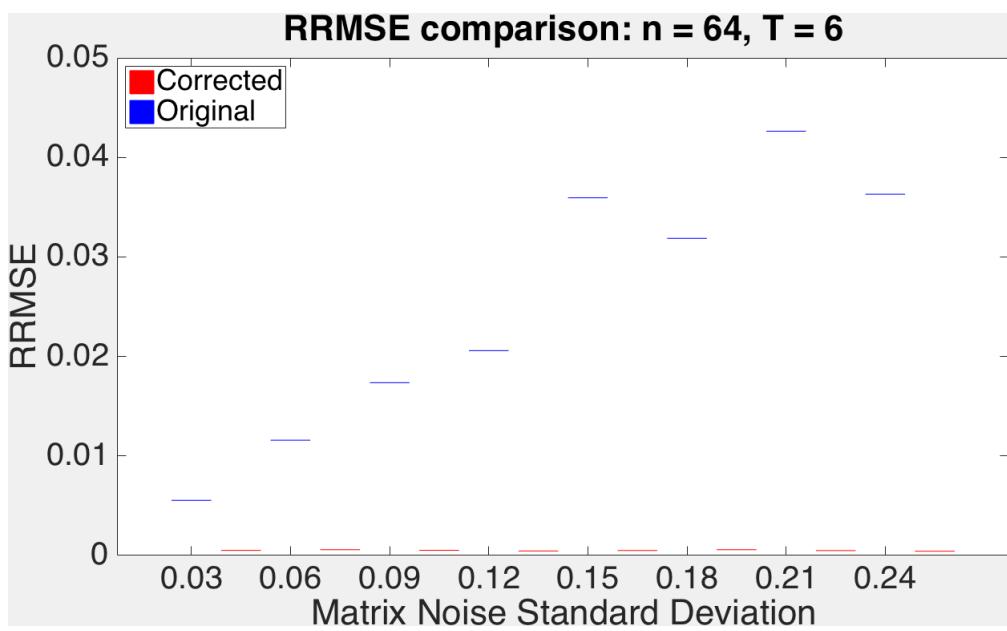


Figure 9.15: Average RRMSE for calibrated and uncalibrated matrices as a function of matrix noise standard deviation for 8×8 signals, sparse in 2D DCT, combined with $T = 6$. Permutations: [6, 7; 3, 6; 6, 2; 1, 4; 8, 3; 5, 2]

Chapter 10

Conclusion and Future Work

In the purview of this report, we dealt with problems arising in physically interesting compressed sensing situations: the positivity of codes in the cameras in [14] and [18, 17]. We tried to optimize, within this framework, sensing matrices for accurate recovery. We assessed successes and difficulties that we encountered on the way, and followed them up with a set of experiments determining just how well compressed sensing bounds work in practise. The result was a valuable lesson about the practical applicability of theoretical bounds, and a possible alternative to current sparse recovery analysis.

All code used in generating results and optimizing sensing matrices in this report lives in the Bitbucket repository at [alankarkotwal/coded-sourcesep](https://bitbucket.org/alankarkotwal/coded-sourcesep) [16].

10.1 Takeaways

The initial success of coherence in optimizing coded source separation, as well as success of previous work using coherence to optimize sensing matrices served as a push for us to use coherence in the CACTI camera as well. This usage acquainted us with a pitfall in compressed sensing design using existing bounds. The recovery error seems to be so loosely bounded (as confirmed by experiments calculating bound looseness for the bound in [25]), in this case, by both the RIC and the coherence, that optimizing the error bound in terms of the sensing matrix seems to do little towards optimizing the actual error.

We then investigated if there was a computable bound that circumvented this difficulty. An l_∞ error-based criterion for perfectly sparse vectors exists and looks promising

at the outset. We, however, discovered that the bound produced by this criterion, while technically applicable to a broader set of signals than coherence is, does not do too well in that regime and does worse than the coherence bound a majority of the time when both are applicable. We then resorted to calculating the RIC error bound in [5] for reasonably-sized matrices at low sparsity levels, and compared the actual error over a dataset of sparse vectors to the bound. This lead us to the conclusion that even the RIC does not establish a tight bound on the recovery error.

A bit of thought revealed that all worst case bounds have a fundamental flaw: they take into consideration all vectors, and possibly a small, low-probability set of vectors loosen the bound up for all other vectors. We therefore proposed an average case analysis and presented a proof-of-concept design using this criterion.

Then, deviating from the bounds, we show that quantization induced by manufacturing and continuous noise by degradation can be tackled in a very systematic way using offline calibration, and pointed out a possible way towards online calibration by a joint optimization of mask elements and reconstructed input signals.

10.2 Future work

The compressed sensing bound, being universal to all vectors, characterizes recovery error in terms of sensing matrix properties and signal sparsity. No other property of the input signal is used. Therefore, in general, any approach targeting worst case errors in terms of these quantities will fall into the same worst case low probability vector trap as coherence. A major area of future work, therefore, is to circumvent these vectors by considering an average case error analysis, and finding suitable priors for natural images for whom the quantity in Eq. 8.1 can be efficiently optimized.

There is also some work to be done in the area of reexamining the sparsity measure used in compressed sensing theory. [15] takes an important step in that direction by identifying some axioms one would expect a sparsity measure to obey, and cataloging which popular sparsity measures follow these axioms. The advantage of softening the sparsity measure to one of these ‘good’ measures is that it blurs the difference between exactly and approximately sparse signals, and that it gets rid of the combinatorial nature of the compressed sensing problem. A characterization of recovery error in terms of a

sparsity measure like this one may be advantageous towards compressed sensing design.

Appendices

Appendix A

Derivation of coherence expressions

Recalling our definitions, we call the index varying from 1 to T as μ or ν , and the index varying from 1 to n as α , β or γ . The μ^{th} block of Φ is thus ϕ_μ . Let the β^{th} diagonal element of ϕ_μ be $\phi_{\mu\beta}$. Define the α^{th} column of D^T to be d_α . Thus, the Gram matrix $\tilde{M} = \Psi^T \Phi^T \Phi \Psi$ has the block structure

$$\begin{aligned}
\tilde{M}_{\mu\nu} &= D^T \phi_\mu^T \phi_\nu D \\
&= D^T \phi_\mu \phi_\nu D \\
&= \begin{pmatrix} d_1 & d_2 & \dots & d_n \end{pmatrix} \begin{pmatrix} \phi_{\mu 1} \phi_{\nu 1} & 0 & \dots & 0 \\ 0 & \phi_{\mu 2} \phi_{\nu 2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \phi_{\mu n} \phi_{\nu n} \end{pmatrix} \begin{pmatrix} d_1^T \\ d_2^T \\ \vdots \\ d_n^T \end{pmatrix} \\
&= \begin{pmatrix} d_1 & d_2 & \dots & d_n \end{pmatrix} \begin{pmatrix} \phi_{\mu 1} \phi_{\nu 1} d_1^T \\ \phi_{\mu 2} \phi_{\nu 2} d_2^T \\ \vdots \\ \phi_{\mu n} \phi_{\nu n} d_n^T \end{pmatrix} \\
&= \sum_{\alpha=1}^n \phi_{\mu\alpha} \phi_{\nu\alpha} d_\alpha d_\alpha^T
\end{aligned}$$

The $\beta\gamma^{\text{th}}$ element of $\tilde{M}_{\mu\nu}$, thus, is

$$\tilde{M}_{\mu\nu}(\beta\gamma) = \sum_{\alpha=1}^n \phi_{\mu\alpha} \phi_{\nu\alpha} d_\alpha(\beta) d_\alpha(\gamma) \quad (\text{A.1})$$

Now we need to normalize the columns of $\Phi\Psi$. Squared column norms are diagonal elements of $\tilde{M}_{\mu\nu}$. So the product of the squared norms of the β^{th} column of the μ^{th} block

and the γ^{th} column of the ν^{th} block is (call this $\xi_{\mu\nu}^2(\beta\gamma)$)

$$\xi_{\mu\nu}^2(\beta\gamma) = \left(\sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta) \right) \left(\sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma) \right) \quad (\text{A.2})$$

Let the normalized Gram matrix be M . Thus, following the same conventions as above (define the numerator of the expression to be $\chi_{\mu\nu}(\beta\gamma)$),

$$M_{\mu\nu}(\beta\gamma) = \frac{\sum_{\alpha=1}^n \phi_{\mu\alpha} \phi_{\nu\alpha} d_\alpha(\beta) d_\alpha(\gamma)}{\sqrt{(\sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta)) (\sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma))}} = \frac{\chi_{\mu\nu}(\beta\gamma)}{\xi_{\mu\nu}(\beta\gamma)} \quad (\text{A.3})$$

Finally, using the square soft-max function to deal with the `max` in the coherence expression, we get the squared soft coherence \mathcal{C} to be

$$\mathcal{C} = \frac{1}{\theta} \log \left[\sum_{\mu=1}^T \sum_{\nu=1}^{\mu-1} \sum_{\beta=1}^n \sum_{\gamma=1}^n e^{\theta M_{\mu\nu}^2(\beta\gamma)} + \sum_{\mu=1}^T \sum_{\beta=1}^n \sum_{\gamma=1}^{\beta-1} e^{\theta M_{\mu\mu}^2(\beta\gamma)} \right] \quad (\text{A.4})$$

In the above, the first term corresponds to all $(\mu > \nu)$ blocks that are ‘below’ the block diagonal. Here, we consider all terms in the given block for the maximum. The second term corresponds to $(\mu = \nu)$ blocks on the block diagonal. Here, we consider only consider $(\beta > \gamma)$ below-diagonal elements for the maximum.

Appendix B

Derivation of coherence derivatives

Differentiating the expression for the squared soft coherence above, we get

$$\begin{aligned} \frac{d\mathcal{C}(\Phi)}{d\phi_{\delta\epsilon}} &= \frac{1}{\theta e^{\theta\mathcal{C}(\Phi)}} \left[\sum_{\mu=1}^T \sum_{\nu=1}^n \sum_{\beta=1}^n \sum_{\gamma=1}^n 2\theta e^{\theta M_{\mu\nu}^2(\beta\gamma)} M_{\mu\nu}(\beta\gamma) \frac{dM_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}} \right. \\ &\quad \left. + \sum_{\mu=1}^T \sum_{\beta=1}^n \sum_{\gamma=1}^{\beta-1} 2\theta e^{\theta M_{\mu\mu}^2(\beta\gamma)} M_{\mu\mu}(\beta\gamma) \frac{\theta M_{\mu\mu}(\beta\gamma)}{d\phi_{\delta\epsilon}} \right] \end{aligned} \quad (\text{B.1})$$

Next, we calculate the derivatives in the above equation, $dM_{\mu\nu}(\beta\gamma)/d\phi_{\delta\epsilon}$. Define the numerator of the expression for $M_{\mu\nu}(\beta\gamma)$ as $\chi_{\mu\nu}(\beta\gamma)$, and thus, $M_{\mu\nu}(\beta\gamma) = \chi_{\mu\nu}(\beta\gamma)/\xi_{\mu\nu}(\beta\gamma)$. Clearly,

$$\frac{dM_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}} = \frac{\xi_{\mu\nu}(\beta\gamma) \frac{d\chi_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}} - \chi_{\mu\nu}(\beta\gamma) \frac{d\xi_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}}}{\xi_{\mu\nu}(\beta\gamma)^2} \quad (\text{B.2})$$

Next,

$$\begin{aligned} \frac{d\chi_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}} &= \frac{d}{d\phi_{\delta\epsilon}} \sum_{\alpha=1}^n \phi_{\mu\alpha} \phi_{\nu\alpha} d_{\alpha}(\beta) d_{\alpha}(\gamma) \\ &= \sum_{\alpha=1}^n d_{\alpha}(\beta) d_{\alpha}(\gamma) \frac{d}{d\phi_{\delta\epsilon}} (\phi_{\mu\alpha} \phi_{\nu\alpha}) \end{aligned}$$

Notice that a term in the above summation can be non-zero only if $\alpha = \epsilon$. Thus,

$$\begin{aligned} \frac{d\chi_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}} &= d_{\epsilon}(\beta) d_{\epsilon}(\gamma) \frac{d}{d\phi_{\delta\epsilon}} (\phi_{\mu\epsilon} \phi_{\nu\epsilon}) \\ &= d_{\epsilon}(\beta) d_{\epsilon}(\gamma) \left(\phi_{\mu\epsilon} \frac{d\phi_{\nu\epsilon}}{d\phi_{\delta\epsilon}} + \frac{d\phi_{\mu\epsilon}}{d\phi_{\delta\epsilon}} \phi_{\nu\epsilon} \right) \end{aligned}$$

Now, notice that $d\phi_{\mu\epsilon}/d\phi_{\delta\epsilon}$ is non-zero only if $\mu = \epsilon$. Denote by $\uparrow_{\mu\epsilon}$ the Kronecker delta function, which is 1 only if $\mu = \epsilon$, 0 otherwise. Then,

$$\frac{d\chi_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}} = d_{\epsilon}(\beta) d_{\epsilon}(\gamma) (\phi_{\mu\epsilon} \uparrow_{\nu\delta} + \uparrow_{\mu\delta} \phi_{\nu\epsilon}) \quad (\text{B.3})$$

Next,

$$\begin{aligned}
\frac{d\xi_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}} &= \frac{d}{d\phi_{\delta\epsilon}} \sqrt{\left(\sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta) \right) \left(\sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma) \right)} \\
&= \frac{1}{2\xi_{\mu\nu}(\beta\gamma)} \frac{d}{d\phi_{\delta\epsilon}} \left(\sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta) \sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma) \right) \\
&= \frac{1}{2\xi_{\mu\nu}(\beta\gamma)} \left[\sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta) \frac{d}{d\phi_{\delta\epsilon}} \left(\sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma) \right) \right. \\
&\quad \left. + \sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma) \frac{d}{d\phi_{\delta\epsilon}} \left(\sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta) \right) \right]
\end{aligned}$$

Again, a term in one of the above summations is non-zero only if α or τ is the same as ϵ .

Thus,

$$\frac{d}{d\phi_{\delta\epsilon}} \left(\sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta) \right) = 2\phi_{\mu\epsilon} d_\epsilon^2(\beta) \uparrow_{\mu\delta}$$

Thus,

$$\frac{d\xi_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}} = \frac{1}{\xi_{\mu\nu}(\beta\gamma)} \left[\phi_{\mu\epsilon} d_\epsilon^2(\beta) \uparrow_{\mu\delta} \sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma) + \phi_{\nu\epsilon} d_\epsilon^2(\gamma) \uparrow_{\nu\delta} \sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta) \right] \quad (\text{B.4})$$

This completes the calculation of derivatives.

Bibliography

- [1] V. Abolghasemi, S. Ferdowsi, B. Makkiabadi, and S. Sanei. On optimization of the measurement matrix for compressive sensing. In *2010 18th European Signal Processing Conference*, pages 427–431, Aug 2010.
- [2] H. Arguello and G. R. Arce. Rank minimization code aperture design for spectrally selective compressive imaging. *IEEE Transactions on Image Processing*, 22(3):914–954, March 2013.
- [3] B. Bouchhima, R. Amara, and M. T. H. Alouane. Design of optimal matrices for compressive sensing: Application to environmental sounds. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 130–134, Aug 2015.
- [4] T. T. Cai and L. Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory*, 57(7):4680–4688, July 2011.
- [5] T. T. Cai, L. Wang, and G. Xu. New bounds for restricted isometry constants. *IEEE Transactions on Information Theory*, 56(9):4388–4394, Sept 2010.
- [6] William R. Carson, Minhua Chen, Miguel R. D. Rodrigues, Robert Calderbank, and Lawrence Carin. Communications-inspired projection design with application to compressive sensing. *SIAM Journal on Imaging Sciences*, 5(4):1185–1212, 2012.
- [7] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, March 2008.
- [8] J. M. Duarte-Carvajalino and G. Sapiro. Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization. *IEEE Transactions on Image Processing*, 18(7):1395–1408, July 2009.

- [9] M. Elad. Optimized projections for compressed sensing. *IEEE Transactions on Signal Processing*, 55(12):5695–5702, Dec 2007.
- [10] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel, 2013.
- [11] 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, Oct 2007.
- [12] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, March 2014.
- [13] K. Hirakawa and P. J. Wolfe. Spatio-spectral color filter array design for optimal image recovery. *IEEE Transactions on Image Processing*, 17(10):1876–1890, Oct 2008.
- [14] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar. Video from a single coded exposure photograph using a learned over-complete dictionary. In *2011 International Conference on Computer Vision*, pages 287–294, Nov 2011.
- [15] N. Hurley and S. Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, Oct 2009.
- [16] Alankar Kotwal. Implementation. <http://bitbucket.org/alankarkotwal/coded-sourcesep/>.
- [17] Patrick Llull, Xuejun Liao, Xin Yuan, Jianbo Yang, David Kittle, Lawrence Carin, Guillermo Sapiro, and David Brady. Compressive sensing for video using a passive coding element. In *Imaging and Applied Optics*, page CM1C.3. Optical Society of America, 2013.
- [18] Patrick Llull, Xuejun Liao, Xin Yuan, Jianbo Yang, David Kittle, Lawrence Carin, Guillermo Sapiro, and David J. Brady. Coded aperture compressive temporal imaging. *Opt. Express*, 21(9):10526–10545, May 2013.

- [19] Rico Malvar, Li wei He, and Ross Cutler. High-Quality Linear Interpolation for Demosaicing of Bayer-Patterned Color Images. In *International Conference of Acoustic, Speech and Signal Processing*, May 2004.
- [20] M. Mordechay and Y. Y. Schechner. Matrix optimization for poisson compressed sensing. In *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pages 684–688, Dec 2014.
- [21] R. Obermeier and J. A. Martinez-Lorenzo. Sensing matrix design via mutual coherence minimization for electromagnetic compressive imaging applications. *IEEE Transactions on Computational Imaging*, 3(2):217–229, June 2017.
- [22] A. Parada-Mayorga and G. R. Arce. Colored coded aperture design in compressive spectral imaging via minimum coherence. *IEEE Transactions on Computational Imaging*, 3(2):202–216, June 2017.
- [23] Marcio P. Pereira, Lisandro Lovisolo, Eduardo A.B. da Silva, and Marcello L.R. de Campos. On the design of maximally incoherent sensing matrices for compressed sensing using orthogonal bases and its extension for biorthogonal bases case. *Digital Signal Processing*, 27:12 – 22, 2014.
- [24] F. Renna, M. R. D. Rodrigues, M. Chen, R. Calderbank, and L. Carin. Compressive sensing for incoherent imaging systems with optical constraints. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5484–5488, May 2013.
- [25] Christoph Studer and Richard G. Baraniuk. Stable restoration and separation of approximately sparse signals. *Applied and Computational Harmonic Analysis*, 37(1):12 – 35, 2014.
- [26] G. Tang and A. Nehorai. Performance analysis of sparse recovery based on constrained minimal singular values. *IEEE Transactions on Signal Processing*, 59(12):5734–5745, Dec 2011.
- [27] G. Tang and A. Nehorai. Computable performance bounds on sparse recovery. *IEEE Transactions on Signal Processing*, 63(1):132–141, Jan 2015.

- [28] A. Veeraraghavan, D. Reddy, and R. Raskar. Coded strobing photography: Compressive sensing of high speed periodic videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):671–686, April 2011.
- [29] Ashwin Wagadarikar, Renu John, Rebecca Willett, and David Brady. Single disperser design for coded aperture snapshot spectral imaging. *Appl. Opt.*, 47(10):B44–B51, Apr 2008.
- [30] Yair Weiss, Hyun Sung Chang, and William T. Freeman. Learning Compressed Sensing. In *Allerton Conference on Communication, Control, and Computing*, 2007.

Publications

Preprints

- Alankar Kotwal and Ajit Rajwade. *Optimizing Codes for Source Separation in Compressed Video Recovery and Color Image Demosaicing*: long, arXiv:1609.02135 [cs.CV]. 2016.

Papers

- Alankar Kotwal and Ajit Rajwade. *Optimizing Codes for Source Separation in Compressed Video Recovery and Color Image Demosaicing*. Submitted to Elsevier Signal Processing.
- Alankar Kotwal and Ajit Rajwade. *Optimizing Matrices for Compressed Sensing Using Existing Goodness Measures: Negative Results*. To be submitted to IEEE Transactions on Computational Imaging.