Optimizing Sensing Matrices for Compressed Sensing Recovery

A preliminary report 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
October 2016

Approval

The preliminary report entitled

Optimizing Sensing Matrices for Compressed Sensing Recovery

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

all the stardust that makes me, mine and all

and, of course, to

all the fish

well, mostly

the prawns with the people sauce

Abstract

There exist several applications in image processing (eg: video compressed sensing [9] and color image demosaicing) which require separation of constituent images given measurements in the form of a coded superposition of those images. Physically practical code patterns in these applications are non-negative and do not obey the nice coherence properties of other patterns such as Gaussian codes, which can adversely affect reconstruction performance. The contribution of this work is to design code patterns for video compressed sensing and demosaicing by minimizing the mutual coherence given a fixed dictionary. Our method explicitly takes into account the special structure of those code patterns as required by these applications: (1) non-negativity, (2) block-diagonal nature, and (3) circular shifting. In particular, the last property enables for accurate patch-wise reconstruction.

We then deviate from the coded source separation scenario and explore sparsity measures other than the l_0 norm that yield quantities upper-bounding reconstruction error that are easier to calculate than the ominous RIC bound, and are tighter than the ubiquitous coherence bound. We aim to optimize these quantities as functions of the sensing matrix to improve performance.

Keywords – video compressed sensing, source separation, sensing matrices, overlapping patch-wise reconstruction, coherence, optimization, gradient descent, sparsity measures

Contents

List of Figures

Chapter 1

Introduction

Compressed sensing has been explored as an alternative (usually, faster) way of sampling continuous-time signals. Its success with still images has inspired efforts to apply it to video.

A system implementing compression across time was presented in [9]. The reconstruction framework here, however, forces on the inherent scene a time-smoothness assumption and hence cannot well-model sharp changes like occlusions or lighting effects. Other techniques like [15] exploit additional structure within the signal, like periodicity, rigid motion or analytical motion models and cannot be used in the general video sensing case. We try relaxing these constraints using a source-separation approach to the problem.

Next, we aim to design such sensing matrices with low mutual coherence, making them ideal for compressed video. Most current approaches to this problem have their limitations: the method in [5], for instance, involves a step that requires a Cholesky-type decomposition of a 'reduced' Gram matrix, and the non-linear reduction process is not guaranteed to keep the Gram matrix positive-semidefinite. Besides, the methods in both [4, 5, 12] optimize objective functions that are some forms of average of normalized dot products of effective dictionary columns, and minimizing averages doesn't guarantee minimizing the maximum (which is coherence in this case) of the quantities forming this average. Other than these, some authors have taken an information-theoretic route to this problem [3, 13, 16]. 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. Computing this mutual information first requires

estimation of the probability density function of X and 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. Besides, the literature cited so far does not account for the special (positive diagonal) structure of the sensing matrices used for video compressed sensing as in [9] or for demosaicing, a framework which this paper expressly deals with.

There are obvious applications for this in the fields of fast video sensing and in improving multi-spectral imaging and image demosaicing. Besides, this will find applications in the general problem of coded source separation where inputs are coded linear combinations of images sparse in some domain and need to be solved for in a source-separation framework.

Further, we go on to find measures of the 'goodness' of a sensing matrix 'tighter' than coherence, and try optimizing general sensing matrices in the framework constructed by these measures. As we will see, some of these measures change the basic meaning, from the l_0 norm of the sparsity of a signal, to one that follows some 'axioms' one might expect a sparsity measure to follow. We speculate that using one of such measures, error bounds different in nature to and more easily computable than the best current (and hard-to-compute) bounds might be obtained.

The rest of this report is organized as follows:

1. **Preliminaries** reviews the basics of compressed sensing, from the qualitative idea to theoretical guarantees and introduces the source separation problem.

2. **Coded Source Separation for Compressed Video** reviews some past work in using compressed sensing for video data and motivates and describes our proposed framework.

3. **Sensing Matrix Optimization for Compressed Video** introduces our methods for optimizing sensing matrices, specialized for our coded source separation framework.

4. **Optimizing General Matrices** moves on to describe some techniques to optimize sensing matrices based on new sparsity measure definitions.

5. **Experiments and Results** summarizes results from our reconstruction and optimization schemes.

6. **Conclusion and Future Work** briefs upon the take-aways from this work and on future work in this area.

Chapter 2

Preliminaries

This chapter introduces some basic concepts from compressed sensing and source separation that will crop up throughout this work. I have tried to keep the material as self-contained as possible.

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 band-limited (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 band-limited signal, it can be shown that we can't do any better.

Natural signals, however, have more structure than band-limitedness. Natural im-

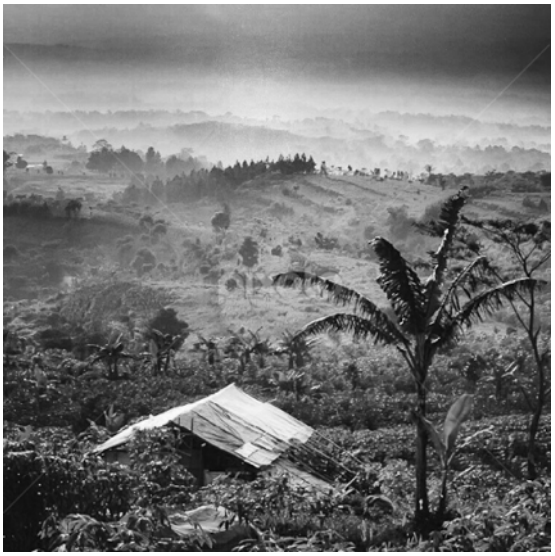

Figure 2.1: Example image for the sparsity analysis in Fig. 2.2

ages, 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 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 – we have a compressive measurement. Now, if the (general)

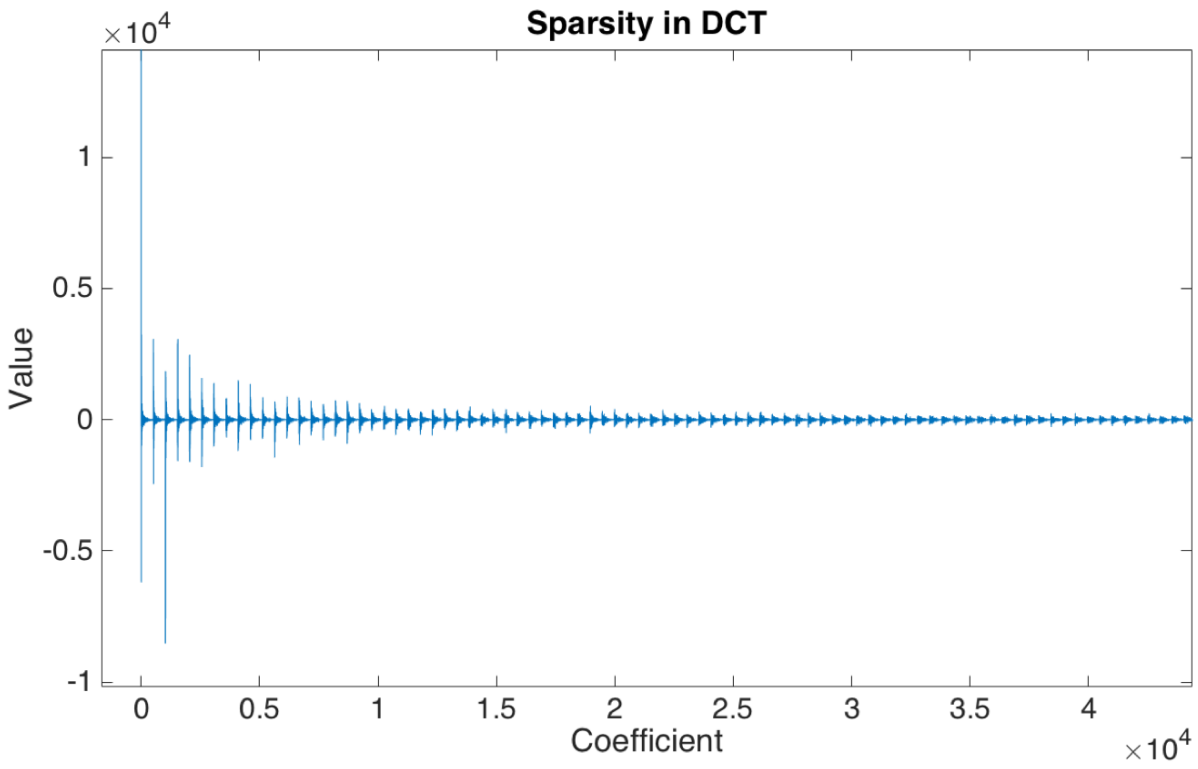

Figure 2.2: Plot of DCT coefficients for the image in Fig. 2.1

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.

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

$$\min_c \|c\|_0 \text{ such that } y = Ac \tag{2.3}$$

This optimization problem, however, can be shown [6] 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] [1] These algorithms are summarized in brief for reference:

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 $\|R_n(x)\| <$ threshold **do**

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

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

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

$n \leftarrow n + 1$

end

Algorithm 1: Matching Pursuit

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

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

Initialization

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

$n \leftarrow 1$;

$\mathcal{S} \leftarrow \Phi$

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

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

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

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

$n \leftarrow n + 1$

end

Algorithm 2: Orthogonal Matching Pursuit

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

$$\min_c \|c\|_p \text{ such that } y = Ac \tag{2.4}$$

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:

$$\min_{c} \|c\|_p \text{ such that } \|y - Ac\|_2 \leq \epsilon \tag{2.5}$$

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 [6].

l_1 optimization

A number of properties of the sensing matrix have been used [6] 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

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

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

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

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 [6] 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}:

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

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

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

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

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 classic 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 precisely control the sensing framework – so when (if) we use source separation in compressed sensing, 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 [14] is

$$z = Ax + Be + n \tag{2.11}$$

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

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

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,

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

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

Chapter 3

Coded Source Separation for Compressed Video

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 Previous work

Linear coded combinations of input frames were exploited in implementation in [9], 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

$$Y = \sum_{i=1}^{T} \phi_i X_i \tag{3.1}$$

Since the output of this operation is a coded combination, this sensing framework constrains the ϕ_1 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

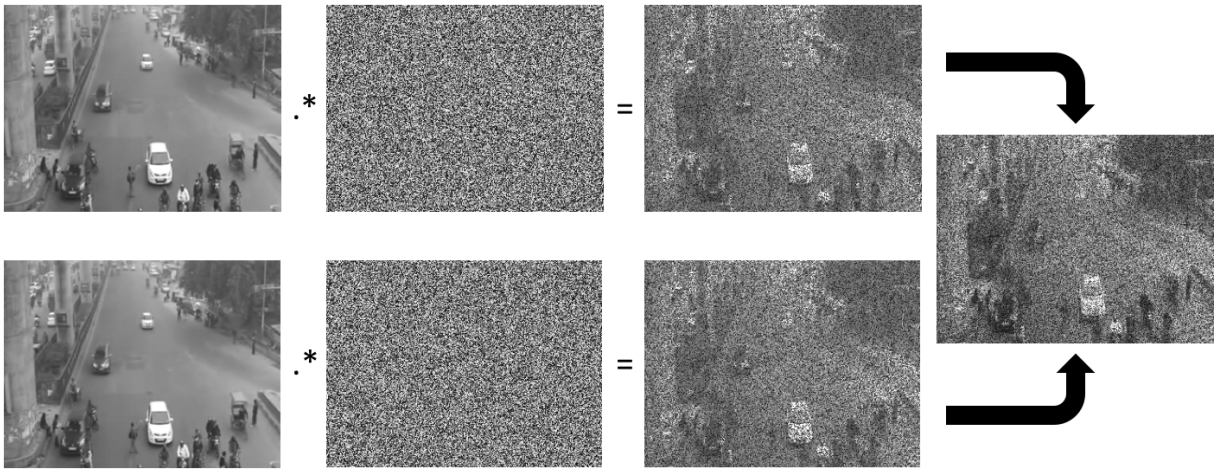

Figure 3.1: Measurement Model

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

$$X_i = \sum_{j=1}^{K} D_{ji}\alpha_j \tag{3.2}$$

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:

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

This problem can be approximately solved with sparse recovery techniques like orthogonal matching pursuit [1].

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. Other techniques like [15] 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

We try relaxing these constraints using a source-separation approach [14], where precise error bounds on the recovery of the images have been derived, with possible improvement

using the techniques in [2]. Each of the coded snapshots is treated as a mixture of sources, each sparse in some basis. 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 [14].

We propose to use a recovery method different from the one used in [9], 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,

$$Y = \begin{pmatrix} \phi_1 & \dots & \phi_T \end{pmatrix} \begin{pmatrix} D\alpha_1 & \dots & D\alpha_T \end{pmatrix}^T \tag{3.4}$$

$$= \begin{pmatrix} \phi_1 D & \dots & \phi_T D \end{pmatrix} \begin{pmatrix} \alpha_1 & \dots & \alpha_T \end{pmatrix}^T \tag{3.5}$$

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

$$\min_{\alpha} \|\alpha\|_1 \text{ subject to } Y = \Phi\Psi\alpha,\ \alpha = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_T \end{pmatrix}^T \tag{3.6}$$

In our implementation we used the `CVX` [7] solver for solving the convex optimization problem in Eq. 3.6.

Chapter 4

Sensing Matrix Optimization for Compressed Video

As seen in Chapter 2, (tractable) compressed sensing recovery using the l_1 norm succeeds only under certain conditions on the matrix ϕ and the sparsifying basis ψ. It has been shown [14] that if the sparsity of a signal in a basis, given by the l_0 norm of its coefficient vector α in the dictionary $D = \phi\psi$ with coherence $\mu(D)$ satisfies

$$\|\alpha\|_0 \leq \frac{1}{2}\left(1 + \frac{1}{\mu(D)}\right) \tag{4.1}$$

and the compressed measurement yields Y, then the optimization problem

$$\min_{\alpha} \|\alpha\|_1 \text{ subject to } Y = \phi\psi\alpha \tag{4.2}$$

necessarily yields the true coefficient vector α.

Clearly, the guarantee on recovery would apply to 'more' signals (greater allowed values of $\|\alpha\|_0$, so less sparse signals are allowed) if the value of $\mu(D)$ is small. Most approaches to sensing matrix optimization, thus, focus on finding a sensing matrix (and sometimes, jointly finding a sensing matrix and sparsifying basis) such that $\mu(D)$ is minimized.

Given a sparsifying basis ψ, then, it is necessary to construct an 'optimal' sensing matrix. Most previous work and our first method do this in terms of $\mu(D)$.

4.1 Previous work

4.1.1 Minimization via the Gram matrix

One way to look at the coherence is [5] 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. [5] tries to minimize the following function, with a parameter t:

$$\mu_t(D) = \frac{\sum_{i \neq j} (|g_{ij}| > t) \, |g_{ij}|}{\sum_{i \neq j} (|g_{ij}| > t)} \tag{4.3}$$

This is an absolute average of off–diagonal Gram matrix entries above t. To achieve this, [5] processes the entries of the Gram matrix by a 'shrinking' function Fig. 4.1, forces the shrunk Gram matrix to be low–rank to get a 'new' Gram matrix, and builds the square root of the 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.

4.1.2 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. [4] solves the problem of estimating ϕ given ψ this way ([4] 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

$$\min_{\Gamma} \left\| \Lambda \Gamma^T \Gamma \Lambda - \Lambda \right\|_F \tag{4.4}$$

This can be written as

$$\min_{\Gamma} \left\| \Lambda - \sum_i \nu_i \nu_i^T \right\|_F = \min_{\Gamma} \left\| \Lambda - \sum_{i, i \neq j} \nu_i \nu_i^T - \nu_j \nu_j^T \right\|_F \tag{4.5}$$

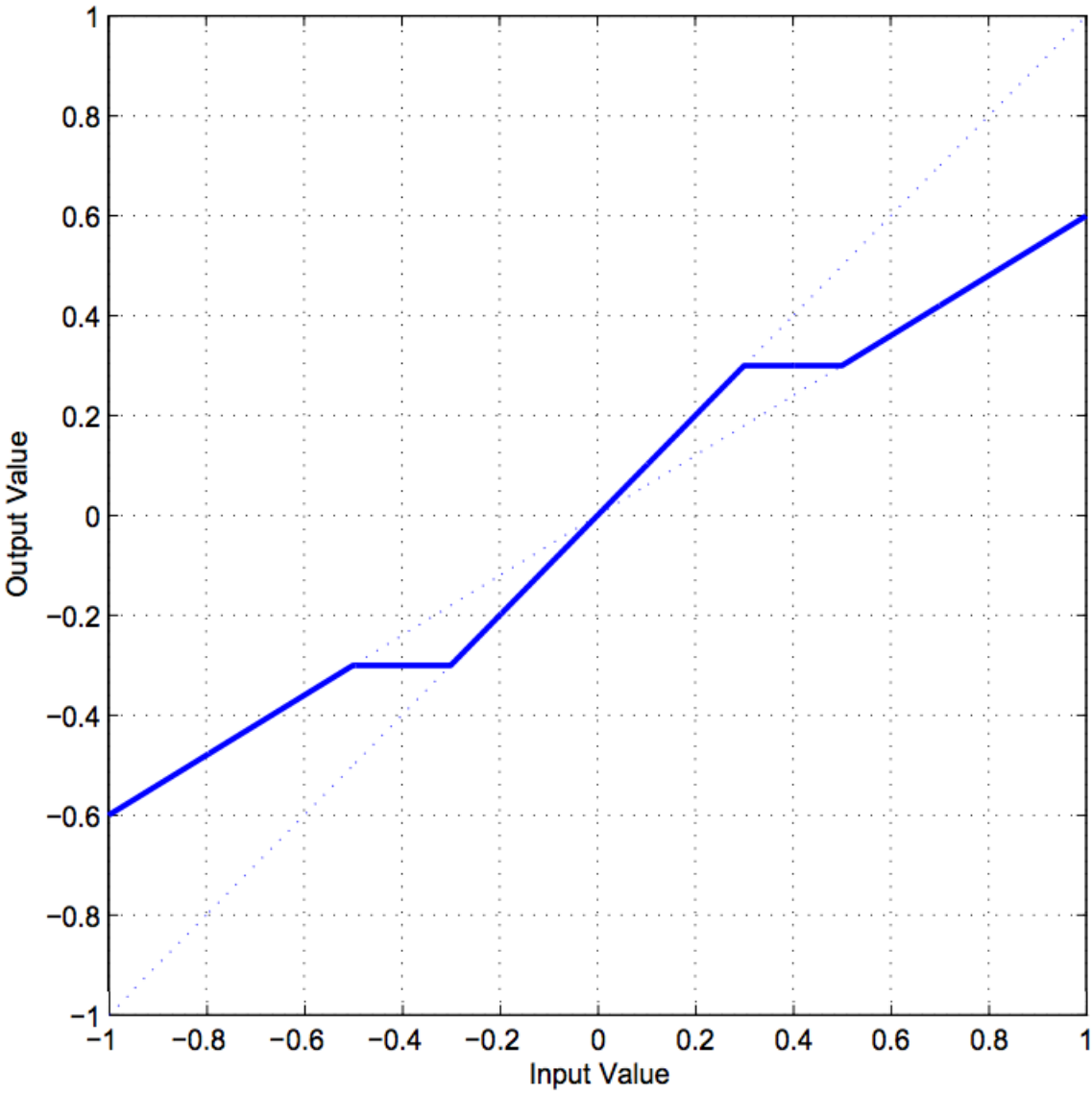

Figure 4.1: Shrinking function in [5]

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.

Again, however, this method minimizes some appropriate average of the Gram matrix elements and therefore isn't guarenteed to minimize the maximum of off-diagonal entries.

4.2 Our approach for video compressed sensing

4.2.1 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

$$\Phi\Psi = \begin{pmatrix} \phi_1 D & \phi_2 D & \dots & \phi_T D \end{pmatrix} \tag{4.6}$$

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,

$$\max_i \{t_i^2\}_{i=1}^n \approx \frac{1}{\theta} \log \sum_{i=1}^n e^{\theta t_i^2} \tag{4.7}$$

We 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

$$M_{\mu\nu}(\beta\gamma) = \frac{\sum_{\alpha=1}^n \phi_{\mu\alpha}\phi_{\nu\alpha} d_\alpha(\beta) d_\alpha(\gamma)}{\sqrt{\left(\sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta)\right)\left(\sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma)\right)}} \tag{4.8}$$

Finally, using the squared soft-max function [Eq. 4.7] to deal with the `max` and the `abs` in the coherence expression, we get the squared soft coherence $\mathcal{C}$ to be

$$\mathcal{C} = \frac{1}{\theta} \log \left[\sum_{\mu=1}^{T} \sum_{\nu=1}^{\mu-1} \sum_{\beta=1}^{n} \sum_{\gamma=1}^{n} e^{\theta M_{\mu\nu}^2(\beta\gamma)} + \sum_{\mu=1}^{T} \sum_{\beta=1}^{n} \sum_{\gamma=1}^{\beta-1} e^{\theta M_{\mu\mu}^2(\beta\gamma)} \right] \tag{4.9}$$

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]

$$\frac{d\chi_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}} = d_\epsilon(\beta)d_\epsilon(\gamma)\left(\phi_{\mu\epsilon} \uparrow_{\nu\delta} + \uparrow_{\mu\delta} \phi_{\nu\epsilon}\right) \tag{4.10}$$

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

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.

4.2.2 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

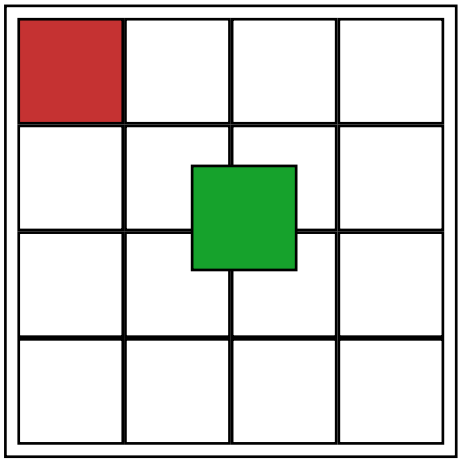

Figure 4.2: Motivation behind circularly-shifted optimization

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 4.2 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 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

$$\mathcal{C} = \frac{1}{\theta} \log \left[\sum_{\zeta \in \text{perm}(\Phi)} \left[\sum_{\mu=1}^{T} \sum_{\nu=1}^{\mu-1} \sum_{\beta=1}^{n} \sum_{\gamma=1}^{n} e^{\theta M_{\mu\nu}^{(\zeta)2}(\beta\gamma)} + \sum_{\mu=1}^{T} \sum_{\beta=1}^{n} \sum_{\gamma=1}^{\beta-1} e^{\theta M_{\mu\mu}^{(\zeta)2}(\beta\gamma)} \right] \right] \tag{4.12}$$

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 $\mathcal{O}(n^5T^2)$, out of which a $\mathcal{O}(n^3T^2)$ 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 attempts mostly use an average coherence minimization technique [4, 5, 12] 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 [3, 13, 16], 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.

4.2.3 Track III: Optimizing bounds tighter than coherence

The coherence bound mentioned in Eq. 2.10 is a very pessimistic bound: it arises from applying Gershgorin's circle 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 [6].

Gershgorin radii

Instead, we can try to minimize the maximum Gershgorin radius, achieving a tighter bound than coherence on the RIC. F weor 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

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

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 $M_{\mu_i^S \mu_j^S}\left(\beta_i^S \beta_j^S\right)$ 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 Gershgorin radius and our objective function $\mathcal{C}$ to be

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

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

Brauer ellipse bounds

A similar bound to the Gershgorin 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 Gershgorin bound, and so can be used to get a tighter bound on the coherence.

However, these optimizations 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.

Chapter 5

Optimizing General Sensing Matrices

COMPRESSED

5.1 An l_1-based error criterion

5.2 An l_∞-based error criterion

5.3 Measures of sparsity

Chapter 6

Experiments and Results

We now present results from the proposed framework for reconstruction and optimization of sensing matrices. We first show that our reconstruction framework performs well on real, sparse images in a variety of situations, and then move on to how our optimization procedure helps improve this performance.

6.1 Validating our framework

We start with testing the proposed framework visually. In all such results in this paper, 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. 6.1. 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. 6.2.

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. 6.3. Looking at these 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

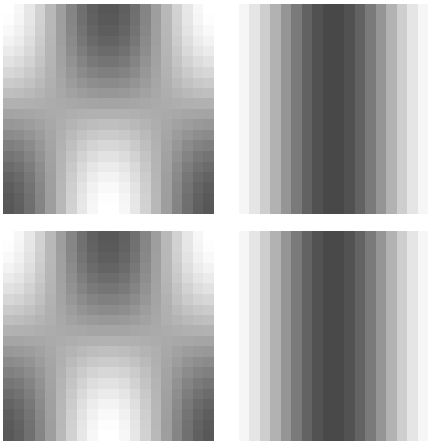

Figure 6.1: Synthetic image results. Left: input images, Right: reconstructions

Figure 6.2: Real images, Gaussian matrices. Left: input images, right: reconstructions

from compressed video.

To evaluate how this works for multiple images, we try separating three images with uniform matrices. See Fig. 6.4. 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. 6.5.

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

Figure 6.3: Real images, uniform matrices. Left: input images, right: reconstructions

Figure 6.4: Separating three images, uniform matrices. Up: input images, down: reconstructions

square errors on recovering the input signals from y as a function of s and T are shown in Figs. 6.6 and 6.13. 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 need to optimize our sensing matrix appropriately, as is shown further in this paper.

Figure 6.5: Sudden change, uniform matrices. Left: input images, right: reconstructions

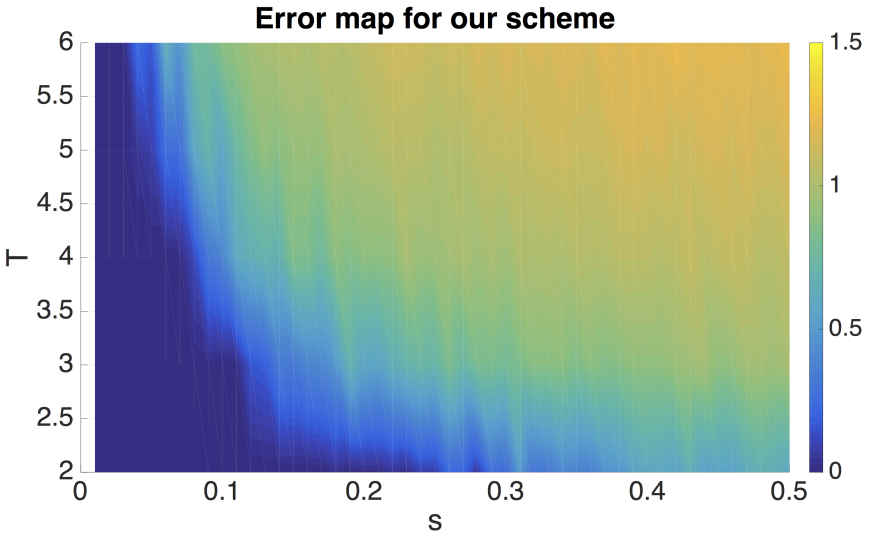

Figure 6.6: Average relative root mean square errors in our scheme as a function of s and T with positive random matrices

6.2 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 [11], which takes a gradient-corrected bilinear interpolated approach. However, recently a case has been made for panchromatic demosaicing [8], 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. 6.7 and 6.8 show, results from

Figure 6.7: Demosaicing. Left: inputs, middle, right: reconstructions with {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. 6.8, notice in the first case the green artifacts near car headlights and the leftmost cyclist in the random reconstruction that is, while that ares 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.

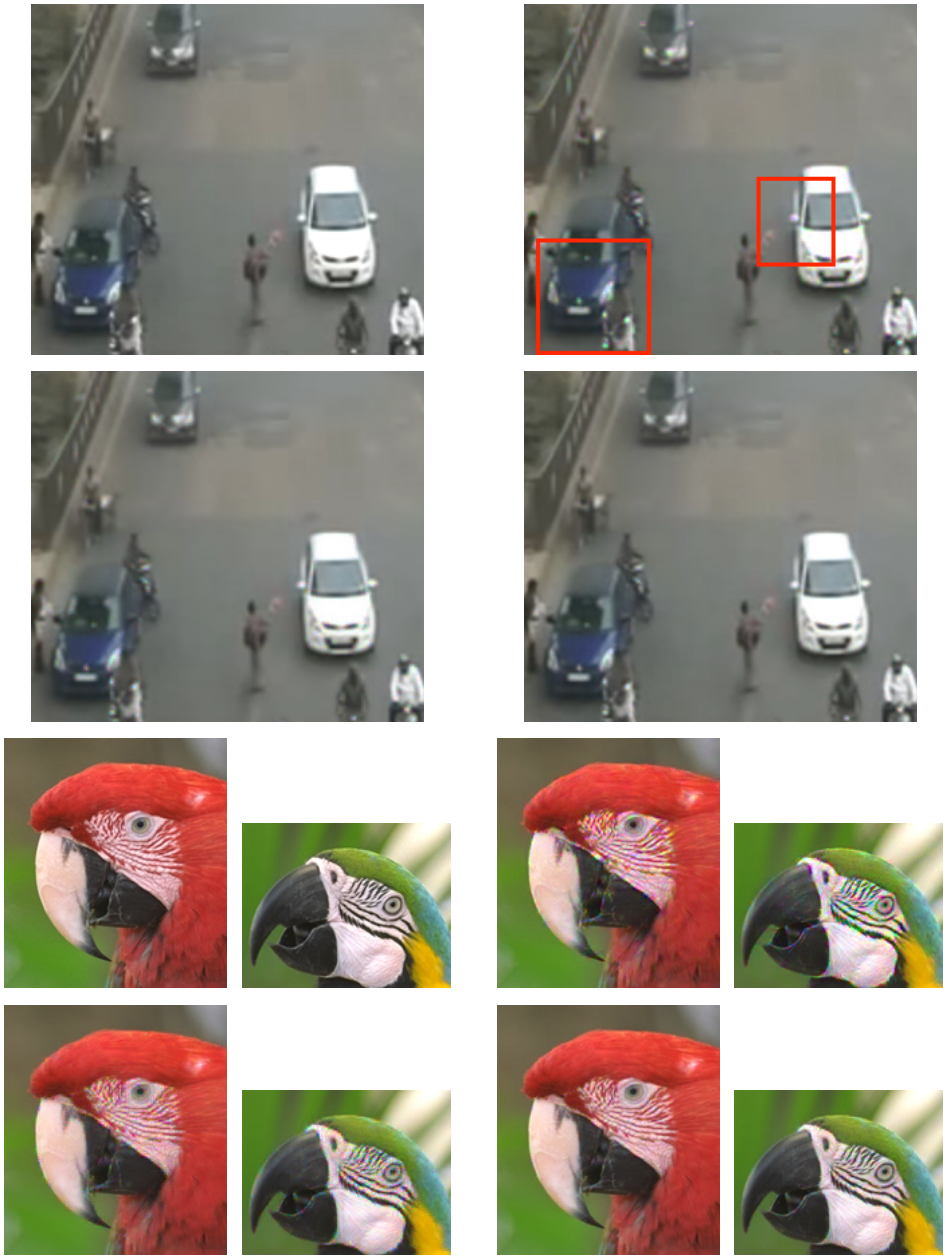

Figure 6.8: Demosaicing close-ups, examples {1, 2, 3}. Clockwise: inputs, reconstructions with {random, {circularly, non-circularly} designed} matrices

6.3 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 6.9. The typical profile of descent on coherence from a random initialization is shown in Fig. 6.10.

The minimum coherence we have been able to achieve in this scheme has been around

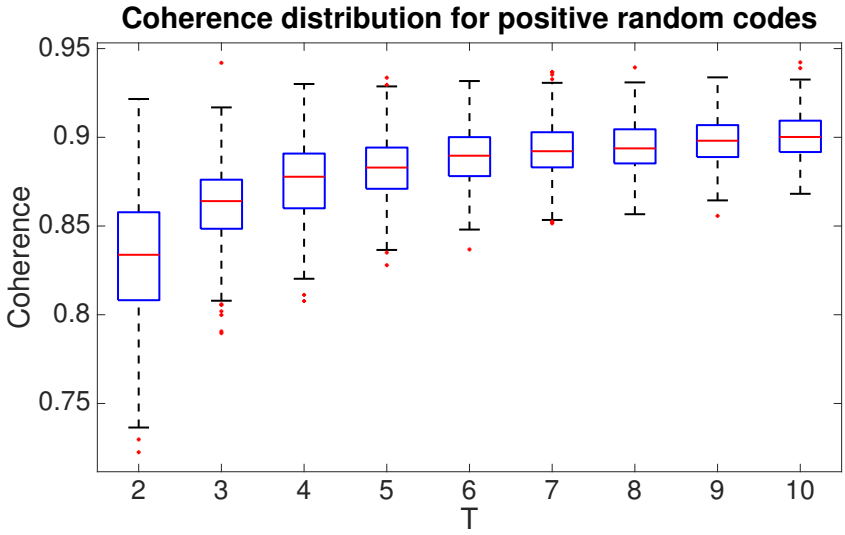

Figure 6.9: Distribution of coherences for 8×8 random positive codes as a function of T

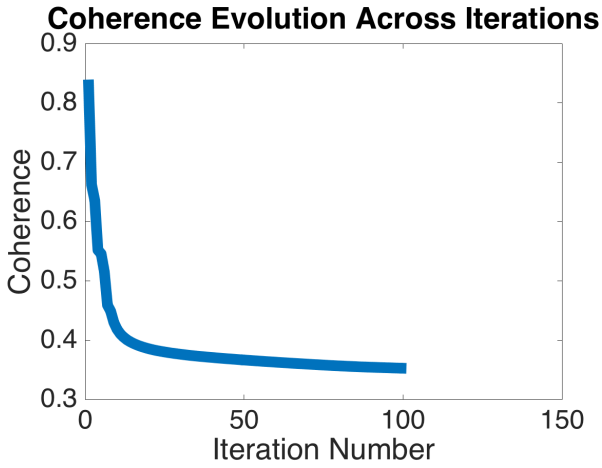

Figure 6.10: Typical coherence decrease profile

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. 6.11 (Fig. 6.12 shows an example for $T = 2$). Ghosting artifacts marked out in Fig. 6.11 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. 6.16 and 6.17.

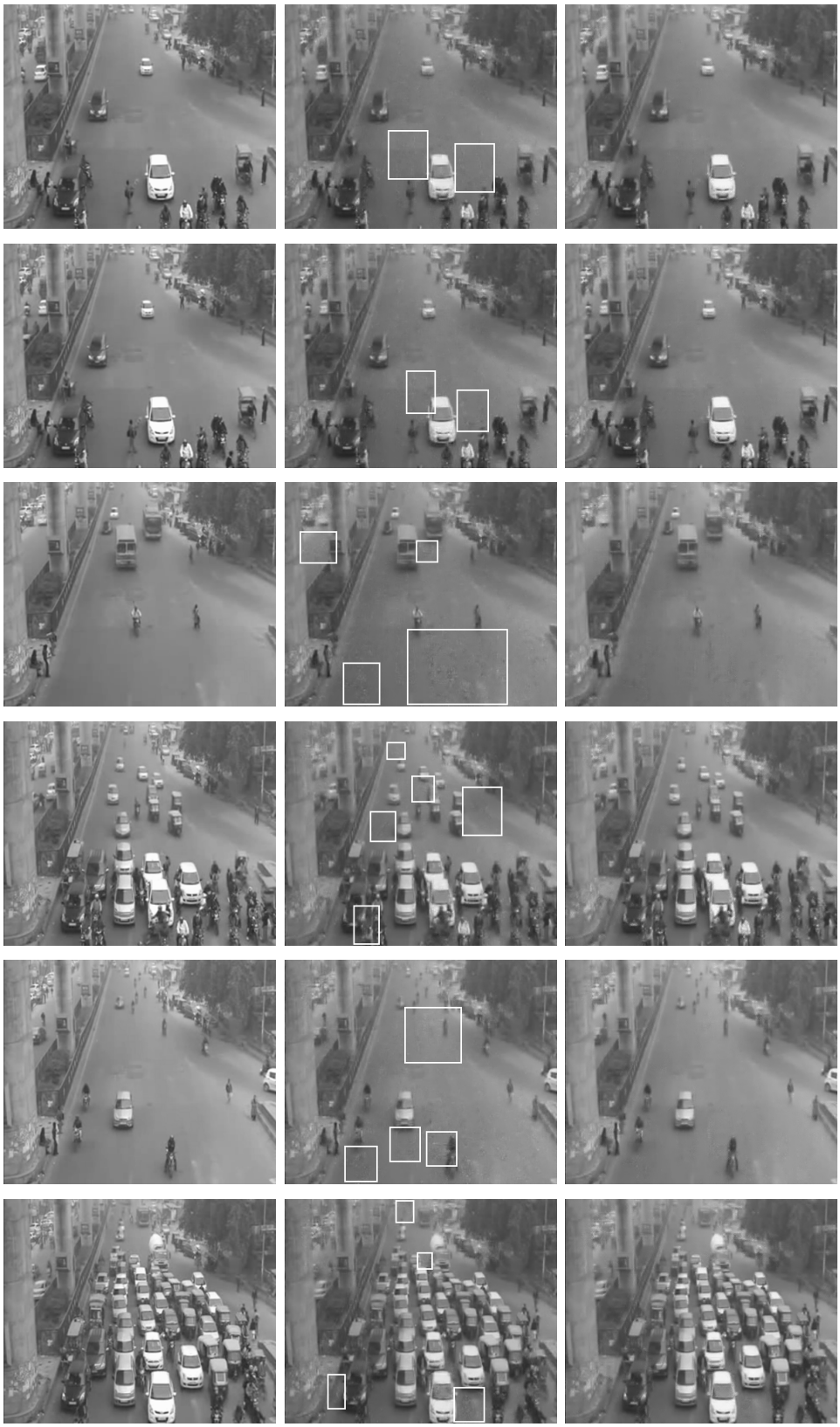

Figure 6.11: Optimized output from combining six not necessarily close images. Left: inputs, middle, right: reconstructions with {random, non-circularly optimized} matrices

Figure 6.12: Optimized output from combining two close images. Left: inputs, middle, right: reconstructions with {random, non-circularly optimized} matrices

Finally, we do a numerical comparison similar to the one in Fig. 6.6. The resulting error map is shown in Fig. 6.13. On an average, we see that we perform better than the random case (note the colorbar scale changes). To characterize this, we compute the difference between these two error maps (random minus optimized). The differences add up to a positive quantity (4.5119 in this case), and thus on an average, here, we're better by an relative root mean square factor of 0.018. This is not very significant, though it does produce significant changes in subtle texture as seen in Figs. 6.16 and 6.17.

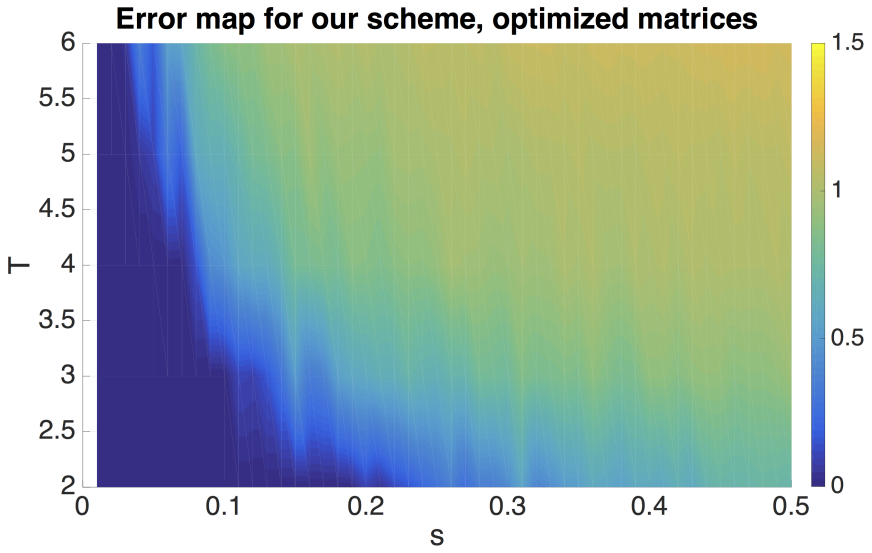

Figure 6.13: Error map for optimized codes as a function of s and T

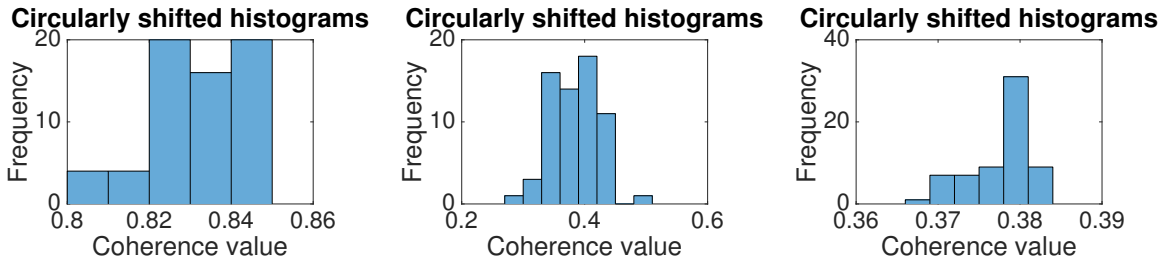

Figure 6.14: Left to right: Circularly-shifted coherence histograms for {random, non-circularly optimized, circularly optimized} matrices

6.3.1 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. 6.14 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. 6.15].

Figure 6.15: Circularly optimized output from combining two close images. Left to right: reconstructions with {random, circularly optimized} matrices

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. 6.16]. 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. 6.17, which is a smaller part of the same image, the superiority of our reconstruction is clearer, with the circular optimization smoothing out blotchier parts of the bonnet.

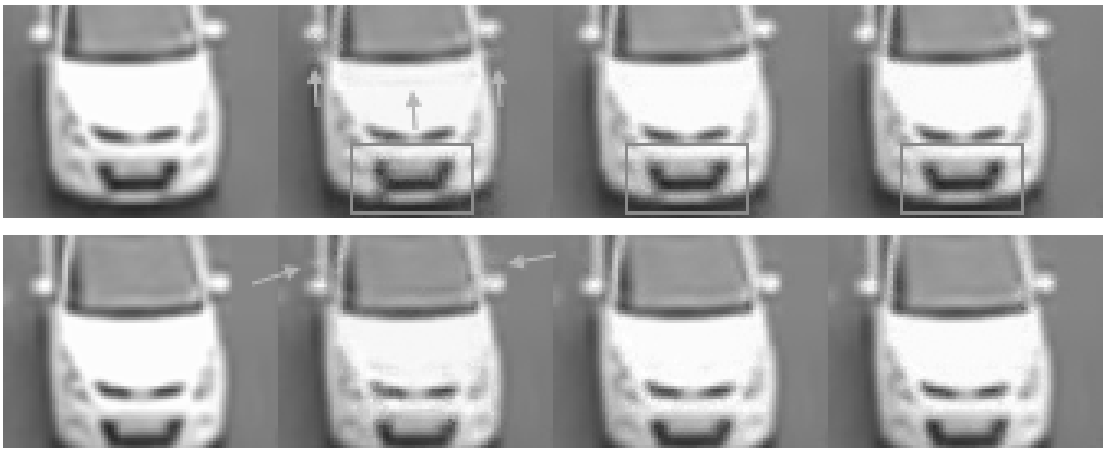

Figure 6.16: Close-ups showing subtle texture preservation with optimized matrices, example 1. Left to right: inputs, reconstructions with {random, non-circularly optimized, circularly optimized} matrices

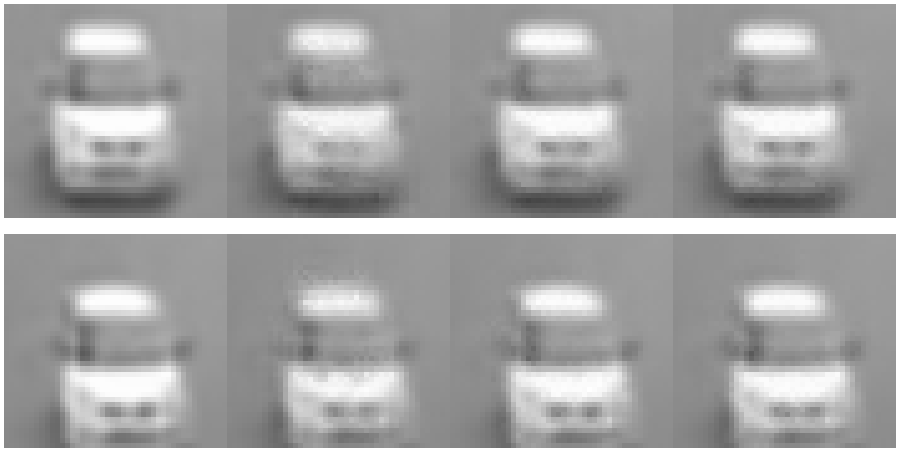

Figure 6.17: Close-ups showing subtle texture preservation with optimized matrices, example 2. Left to right: inputs, reconstructions with {random, non-circularly optimized, circularly optimized} matrices

Chapter 7

Conclusion and Future Work

In the purview of this report, we dealt with problems arising in physically interesting compressed sensing situations: the positivity of sensing matrices and difficulty in obtaining compressive measurements across space and time. We then tried to optimize, within and without this framework, sensing matrices for accurate recovery.

7.1 Take-aways

We cast the video compressed sensing problem as one of separation of coded linear combinations of signals sparse in a given basis. We evaluated this scheme and found it works well for low sparsity levels, and yields reasonably visually good reconstructions. However, especially at high T, we found that random matrices aren't good enough; we need something more.

We then provided an analytical expression for the coherence of the sensing matrix in the coded source separation scheme and optimized for coherence using these. Results showed better quality and less ghosting visually, and less error numerically. However as image size increased, the optimization problem became rapidly intractable, so we settled for optimizing small masks such that they have small coherence in all circular permutations, so they can be tiled for overlapping patchwise reconstruction. We attempted optimizing some tighter lower bounds arising in the derivation of the coherence bound, but these optimizations turned out to be infeasible.

We then moved on to optimizing general sensing matrices with reconstruction error metrics other than the l_2 norm and sparsity measures other than the l_0 norm. Results

from preliminary testing of these were encouraging.

7.2 Future work

The immediate road ahead leads to finding methods (algorithms) better than finite differencing to optimize general sensing matrices. Rigorous testing on synthetic data and real images will be done to validate the hypothesis that this optimization method is better than the state of the art.

The long-term goal is to explore how changing the definition of sparsity and bending it to follow the four basic axioms seen before affects reconstruction error guarantees. It remains to be seen if such a scheme produces quantities amenable to evaluation and optimization in polynomial time. The ultimate goal is to apply this to the general compressed sensing scenario and achieve better results than the state of the art on real image performance.

Most code used in generating results and optimizing sensing matrices in this report lives in the Bitbucket repository at `alankarkotwal/coded-sourcesep` [10]. Gradient descent lives in the `src/descent` folder, circularly-symmetric gradient descent in `src/descent-circular` and reconstruction code in `src/circular`.

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

$$\begin{aligned}
\tilde{M}_{\mu\nu} &= D^T\phi_\mu^T\phi_\nu D \\
&= D^T\phi_\mu\phi_\nu D \\
&= \begin{pmatrix} d_1 & d_2 & \dots & d_n \end{pmatrix} \begin{pmatrix} \phi_{\mu1}\phi_{\nu1} & 0 & \dots & 0 \\ 0 & \phi_{\mu2}\phi_{\nu2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \phi_{\mu n}\phi_{\nu n} \end{pmatrix} \begin{pmatrix} d_1^T \\ d_2^T \\ \vdots \\ d_n^T \end{pmatrix} \\
&= \begin{pmatrix} d_1 & d_2 & \dots & d_n \end{pmatrix} \begin{pmatrix} \phi_{\mu1}\phi_{\nu1}d_1^T \\ \phi_{\mu2}\phi_{\nu2}d_2^T \\ \vdots \\ \phi_{\mu n}\phi_{\nu n}d_n^T \end{pmatrix} \\
&= \sum_{\alpha=1}^{n} \phi_{\mu\alpha}\phi_{\nu\alpha}d_\alpha d_\alpha^T
\end{aligned}$$

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

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

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 $\xi^2_{\mu\nu}(\beta\gamma)$)

$$\xi^2_{\mu\nu}(\beta\gamma) = \left(\sum_{\alpha=1}^{n} \phi^2_{\mu\alpha} d^2_\alpha(\beta)\right)\left(\sum_{\tau=1}^{n} \phi^2_{\nu\tau} d^2_\tau(\gamma)\right) \tag{A.2}$$

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

$$M_{\mu\nu}(\beta\gamma) = \frac{\sum_{\alpha=1}^{n} \phi_{\mu\alpha}\phi_{\nu\alpha} d_\alpha(\beta) d_\alpha(\gamma)}{\sqrt{\left(\sum_{\alpha=1}^{n} \phi^2_{\mu\alpha} d^2_\alpha(\beta)\right)\left(\sum_{\tau=1}^{n} \phi^2_{\nu\tau} d^2_\tau(\gamma)\right)}} = \frac{\chi_{\mu\nu}(\beta\gamma)}{\xi_{\mu\nu}(\beta\gamma)} \tag{A.3}$$

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

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

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

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

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,

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

Next,

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

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

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

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,

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

Next,

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

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,

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

This completes the calculation of derivatives.

Bibliography

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] M. Elad. Optimized projections for compressed sensing. *IEEE Transactions on Signal Processing*, 55(12):5695–5702, Dec 2007.

[6] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhüser Basel, 2013.

[7] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[8] 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.

[9] 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.

[10] Alankar Kotwal. Implementation. `http://bitbucket.org/alankarkotwal/coded-sourcesep/`.

[11] 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.

[12] 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.

[13] 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.

[14] Christoph Studer and Richard G. Baraniuk. Stable restoration and separation of approximately sparse signals. *Applied and Computational Harmonic Analysis*, 37(1):12 – 35, 2014.

[15] 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.

[16] 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.

Conference Papers

- Alankar Kotwal and Ajit Rajwade. Optimizing Codes for Source Separation in Compressed Video Recovery and Color Image Demosaicing, *submitted*, 42^{th} International Conference on Acoustics, Speech and Signal Processing 2017. Paper here.

Acknowledgements

This is not just a piece of work by 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 advisors, 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 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.

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