Signal Recovery Under Perturbations

MTP Stage I Report

submitted in partial fulfillment of the requirements
for the award of the degree of

Master of Technology
in
Computer Science and Engineering

by

Eeshan Malhotra
14305R001

under the guidance of

Prof. Ajit Rajwade

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

October 2016

Contents

1 Introduction **1**

1.1 Perturbed Sensing Matrix 2

1.2 Tomographic Recovery with Unknown View Angles 2

1.3 Recovery Bounds for Composite Signals 2

2 Perturbed Sensing Matrix **3**

2.1 Background 3

2.2 Problem Formulation 4

2.3 Outline of Recovery Framework 4

2.4 Recovery Algorithm 5

2.5 Experimental Results 6

2.6 Extension to Two Dimensions 8

2.7 Conclusions 8

3 Tomographic Recovery with Unknown View Angles **10**

3.1 Background and Existing Work 10

3.2 Problem Formulation 11

3.3 Recovery Algorithm 12

3.3.1 Denoising 12

3.3.2 Coordinate Descent 13

3.4 Experimental Results 14

3.5 Conclusions 16

4 Recovery Bounds for Composite Signals **18**

4.1 Background 18

4.2 Two-Way Dictionary Splitting 20

4.2.1 Algorithm 20

4.2.2 Running time complexity 21

4.2.3 Proof of Optimality 22

4.3 Higher Order Recursive Algorithm 23

4.3.1 Algorithm 24

4.4 Empirical Results 25

5 Conclusion and Future Work **27**

5.1 Perturbed Sensing Matrix 27

5.2 Tomographic Recovery with Unknown View Angles 28

5.3 Recovery Bounds for Composite Signals 28

Chapter 1

Introduction

Compressive sensing allows recovery of sparse signals, even with fewer measurements than required by the Nyquist-Shannon sampling theorem ([1], [2], [3]). Consider the forward model:

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{\Phi x} \\ &= \boldsymbol{\Phi \Psi \theta} \end{aligned}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is the signal that is sparse in the basis $\boldsymbol{\Psi} \in \mathbb{R}^{N \times N}$. $\boldsymbol{\theta} \in \mathbb{R}^N$ is this sparse representation, $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ is the measurement or sensing matrix, and $\boldsymbol{y} \in \mathbb{R}^M$ is the compressed measurement, with $M < N$.

If $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are perfectly known, $\boldsymbol{\theta}$ (and therefore $\boldsymbol{x}$) can be recovered from $\boldsymbol{y}$ ([1], [3]). However, in practice, compressive sensing is often plagued by multiple sources of incomplete, inaccurate, or even inconsistent information. The sources for such inaccuracies are many and varied - including incorrectly calibrated instruments, interference among multiple signals, relative motion between sample and measuring apparatus.

In this work, we tackle some of these, by providing empirically proven algorithms to negate or reduce the effect of these inaccuracies. Existing theoretical bounds are also improved upon, to provide better estimation guarantees.

Specifically, we consider the following scenarios:

1. Perturbation in acquisition matrix for compressed sensing of sparse signals

2. Tomographic reconstruction with unknown or inaccurately known angles

3. Interference among two sparse signals

Although the motivation for these comes from reconstruction under perturbation, a number of results are more general in applicability, as is discussed in the respective sections.

1.1 Perturbed Sensing Matrix

The first part considers inaccuracies in the sensing matrix itself. The task involves concurrent recovery of the signal, as well as the perturbation in the sensing matrix. Motivation for this comes primarily from calibration issues in instruments. Since the Fourier matrix is commonly used - such as in Magnetic Resonance Imaging (MRI) and Computed Tomography (CT), the section derives extensive results specifically for this case. However, a more generalized framework is developed and tested.

1.2 Tomographic Recovery with Unknown View Angles

While incorrectly known angles can be handled as a special case of a sensing matrix perturbations, chapter 3 proposes a much more effective and robust framework for tomographic recovery, that requires no input about the view angles, and makes minimal assumptions on the signal. The essential technique in the algorithm relies on moment based relationships betwwen the image and its projections

1.3 Recovery Bounds for Composite Signals

Lastly, we consider composite signals that have an inherent internal structure, in that they are composed of two interfering signals. Here, no new recovery algorithm is proposed, but the existing recovery guarantees are improved upon, utilising the structure in the signal and the sensing matrix. Finally, the system is generalized to be applicable to arbitrary sparse signals and sensing matrices, even if the structure is not naturally discernible.

Chapter 2

Perturbed Sensing Matrix

2.1 Background

Often, in compressed measurement of sparse signals, the or sensing matrix, D is inaccurately known. For MRI, this may correspond to the frequencies of the Fourier transform deviating from the presumed frequencies the machine is calibrated to ([4],[5],[6],[7]); for parallel beam or fan beam tomography, it may correspond to the projection angles being imperfectly known. These inaccuracies may arise due to errors in calibrating the instrument, relative motion between instrument and sample, or other systematic error sources.

The sensing matrix could, therefore, be considered to be composed of a large ’core’ part which is accurately known, and a small additive perturbation, of which, only basic structural details are known. A concrete framework for such a scenario is described in the following section, followed by a method for simultaneous recovery of the original signal, and the perturbation in the sensing matrix.

Although the general concept of such perturbations has been tackled in existing work ([8]), the perturbation structure proposed here is induced much more naturally from mathematical motivations. The primary results shown here pertain to one-dimensional signals, an extension to two-dimensions (discussed in section 2.6) provides further motivation for the set-up, including use cases arising from calibration error and patient motion in MRI and CT.

The related problem of tomographic reconstruction with angles completely unknown is tackled in chapter 3.

2.2 Problem Formulation

Consider a compressed measurement $\boldsymbol{y} \in \mathbb{R}^M$ of a sparse signal $\boldsymbol{x} \in \mathbb{R}^N$, taken using the sensing matrix $\boldsymbol{\Phi} \in \mathbb{C}^{M \times N}$. That is,

$$\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{\epsilon} \tag{2.1}$$

where $\epsilon \in \mathbb{R}^M$ is the noise in measurement.

If Φ is perfectly known, the recovery can be done simply, using a variety of ways, such as *l1*-norm minimization. That is, solving the optimization problem:

$$\min_{\boldsymbol{x}} ||\boldsymbol{y} - \boldsymbol{\Phi x}||_2^2 + \lambda ||\boldsymbol{x}||_1. \tag{2.2}$$

The minimization of *l1*-norm induces sparsity in the recovered signal.

We consider the case where the discrete Fourier matrix is used for measurement. However, the frequencies at which the Fourier transform is taken are known erroneously, due to sources mentioned in the previous section. So, instead of measuring

$$\boldsymbol{y}_{true} = \boldsymbol{Fx} + \boldsymbol{\epsilon} \tag{2.3}$$

we measure

$$\boldsymbol{y} = \tilde{\boldsymbol{F}}\boldsymbol{x} + \boldsymbol{\epsilon} \tag{2.4}$$

where $\boldsymbol{F}$ is the Fourier matrix at the presumed frequencies, $\boldsymbol{u}$, and $\tilde{\boldsymbol{F}}$ is the Fourier matrix at the perturbed frequencies, $\tilde{\boldsymbol{u}}$.

We can write $\tilde{\boldsymbol{u}} = \boldsymbol{u} + \delta \boldsymbol{u}$. We assume in this chapter that the perturbation, $\delta \boldsymbol{u}$, is small. The problem statement, then, is to recover the signal $\boldsymbol{x}$ and the perturbed frequencies $\tilde{u}$, given the measurement $\boldsymbol{y}$ and the approximate frequencies, $\boldsymbol{u}$.

As we will see in the next section, the framework we developed for this problem has more general applications.

2.3 Outline of Recovery Framework

We use Taylor's series approximation to estimate the matrix $\tilde{\boldsymbol{F}}$. Using the first order approximation,

$$\tilde{\boldsymbol{F}} = \boldsymbol{F} + \boldsymbol{\Delta} \frac{\partial \boldsymbol{F}}{\partial u} + \boldsymbol{R} \tag{2.5}$$

where $\boldsymbol{\Delta}$ is a diagonal matrix, with $\boldsymbol{\Delta}_{i,i} = \boldsymbol{\delta u}$ and $\boldsymbol{R}$ is the second order modeling error due to truncation of the Taylor series.

$$
\begin{aligned}
& \boldsymbol{F_{p,q}}(\boldsymbol{u}) && = e^{-2\pi j u_i \frac{q}{N}} && (2.6)\\
\implies \quad & \left(\frac{\partial \boldsymbol{F}}{\partial u}\right)_{p,q} && = e^{-2\pi j u_i \frac{q}{N}} \left(-2\pi j \frac{q}{N}\right) && (2.7)\\
\implies \quad & \frac{\partial \boldsymbol{F}}{\partial u} && = \boldsymbol{F} \times \boldsymbol{X} && (2.8)
\end{aligned}
$$

where $\boldsymbol{X}$ is a diagonal matrix, with $\boldsymbol{X_{i,i}} = \left(-2\pi j \frac{i}{N}\right)$.

Replacing this in equation 2.5 and equation 2.4, ignoring the remainder term, we get

$$
\begin{aligned}
& \tilde{\boldsymbol{F}} && \approx \boldsymbol{F} + \boldsymbol{\Delta} \times (\boldsymbol{FX}) && (2.9)\\
\implies \quad & \boldsymbol{y} && \approx \Big(\boldsymbol{F} + \boldsymbol{\Delta}\,(\boldsymbol{FX})\Big)\boldsymbol{x}. && (2.10)
\end{aligned}
$$

This equation forms the basis for the recovery algorithm proposed in the following section. It was mentioned earlier that the work has more general application, and as such, the algorithm proposed applies equally well to any system following the general framework:

$$\boldsymbol{y} \approx (\boldsymbol{A} + \boldsymbol{\Delta B})\,\boldsymbol{x}. \tag{2.11}$$

That is, any system where the perturbation in the sensing matrix takes the form of the sum of two known matrices, but with each row of the second matrix amplified by a different, unknown value.

Existing work ([9]) considers a related, but different problem, where $\boldsymbol{y} \approx \boldsymbol{\Delta B x}$. Work by Yang et al ([10]) considers the problem where $\boldsymbol{y} \approx (\boldsymbol{A} + \boldsymbol{B\Delta})\,\boldsymbol{x} + \boldsymbol{\epsilon}$, and work by Nichols et al ([11]) considers the issue of the basis being imperfectly known. Though these appear superficially similar to the framework proposed here, but because of the post-multiplication of a perturbation matrix (Δ, here), lead to entirely different classes of methods and results.

2.4 Recovery Algorithm

The recovery algorithm essentially splits the problem of estimating the signal, $\boldsymbol{x}$, and the perturbations, $\boldsymbol{\delta u}$, into two optimization problems, each of which is convex, and can be solved using standard convex optimization methods, such as gradient descent.
The two optimization problems involved are:

- **Step I**
  Assuming $\boldsymbol{\Delta}$ is estimated correctly, solve for best $\boldsymbol{x}$ as:

$$\boldsymbol{x} = \underset{\hat{\boldsymbol{x}}}{\text{argmin}}\, ||\boldsymbol{y} - (\boldsymbol{F} + \boldsymbol{\Delta F X})\hat{\boldsymbol{x}})||_2^2 + \lambda||\hat{\boldsymbol{x}}||_1 \tag{2.12}$$

- **Step II**
  Assuming x is estimated correctly, solve for best *diagonal* $\boldsymbol{\Delta}$ as:

$$\begin{aligned} \boldsymbol{\Delta} = \text{argmin}_{\hat{\boldsymbol{\Delta}}} \quad & ||\boldsymbol{y} - (\boldsymbol{F} + \hat{\boldsymbol{\Delta}}\boldsymbol{F X})\boldsymbol{x})||_2^2 \\ \text{s.t.} \quad & (\delta \boldsymbol{u})^2 \leq \boldsymbol{E}^2 \end{aligned} \tag{2.13}$$

where $\delta\boldsymbol{u}$ is the set of diagonal entries of Δ, and vector $\boldsymbol{E} \in \mathbb{R}^M$ is an assumed hard upper bound on the magnitude of the perturbations - a constant vector, containing the same value E_{max} in each element. The value of the parameter E_{max} must be guessed in practice, or alternatively, a trial-and-error method used.

Multiple iterations are performed, alternating between the two steps. The results of each subproblem help estimate the parameters for the other more accurately. This alternating algorithm is performed till convergence is achieved.

The complete algorithm can be written in pseudo-code as:

Algorithm 1 Recovery with perturbed sensing matrix

```
Initialize $\boldsymbol{\Delta} \leftarrow \mathbf{0}^{M \times M}$
converged ← False
while not converged do
    $\boldsymbol{x}$ ← Best estimate using equation 2.12
    $\boldsymbol{\Delta}$ ← Best estimate using equation 2.13
    if change in both estimates is less than threshold then
        converged ← True
Output $\boldsymbol{x}$, $\boldsymbol{\Delta}$
```

The above alternating minimization algorithm is guaranteed to converge to a minimum. However, it is susceptible to finding a local minimum. In experiments, though, the convergence was almost always extremely close to the true values.

2.5 Experimental Results

The algorithm was tested using simulated data, generating a large number of random signals, with varying parameters.

The recovery error was measured as

$$relative\_error = \frac{||\boldsymbol{x}_{true} - \boldsymbol{x}_{recovered}||_2}{||\boldsymbol{x}_{true}||_2} \tag{2.14}$$

A number of parameters were varied in the experiments, to test the robustness of the algorithm. The effect of changing the length of the signal and $||x||_2$ was negligible. However, results varied significantly with changing the sparsity of the signal, the number of measurements taken (dimensions of sensing matrix), and the maximum magnitude of each perturbation in δu, E_{max}. Figure 2.1 shows the impact of varying these parameters.

In the typical working range of $E_{max} \approx 0.1$, it was typically possible for the alternating minimization algorithm to achieve a relative error of $\leq 5\%$, while a naive approach of simply ignoring the perturbations incurred an error that was many times larger, often upwards of 20%.

A practical issue is the tuning of the parameter λ used for optimization. In the experiments above, the value was tuned clairvoyantly. The rate of convergence was also found to vary with change in parameter E_{max}. While lower E_{max} values clearly improves the accuracy of the algorithm, the number of iterations required to reach this low error grows quickly.

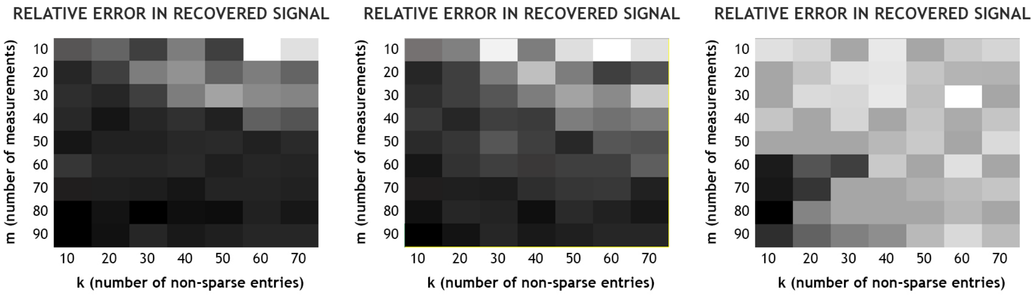

Figure 2.1: *Relative error in signal with varying sparsity and number of measurements*
White represents the highest relative error: $\sim 95\%$*; Black indicates lowest relative error:* $\sim 2.5\%$
Left to Right: $E_{max} = 0.01, 0.1, 0.5$

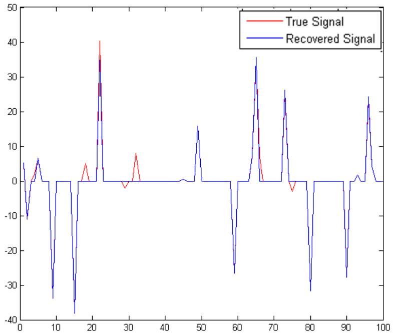

Figure 2.2: *Typical signal recovery:* $N = 100, M = 70, k = 20, E_{max} = 0.1$

2.6 Extension to Two Dimensions

A direct extension of the algorithm to two dimensions can be achieved by vectorizing a two dimensional signal. However, an interesting case arises when we consider the case of 2-D radial MRI or tomographic reconstruction, with uncertainty in frequencies arising only due to erroneously know *angles*. This is a realistic representation of errors arising due to mis-calibration.

In such a case, the errors in each frequency pair (u_i, v_i) are not independent. All frequencies lying along a radial line should have the same inaccuracy in their angle estimates. This reduces the degrees of freedom of the system significantly, and hence, is a more constrained problem.

This case is solved using polar coordinates. Consider the Fourier transform along a radial pattern at frequencies described by the vectors $\boldsymbol{r}$ and $\boldsymbol{\theta}$. $\boldsymbol{r}$ values are assumed to be accurately known, but $\boldsymbol{\theta}$ values are known imprecisely.

Akin to the 1-D case, we define $\boldsymbol{\theta}$ to be the presumed angles, $\tilde{\boldsymbol{\theta}}$ to be the perturbed angles, and $\delta\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}$. The Fourier transform of an image, or or general 2-D signal I, of size $N \times N$ at frequencies defined by $\boldsymbol{r}$ and $\tilde{\boldsymbol{\theta}}$ is given by

$$\tilde{\boldsymbol{F}}(\boldsymbol{r}, \tilde{\boldsymbol{\theta}})\boldsymbol{I} = \sum_{x,y} \left(e^{-2\pi jr\left(\frac{x\cos\tilde{\theta}}{N} + \frac{y\sin\tilde{\theta}}{N}\right)} \times I(x,y) \right) \tag{2.15}$$

Once again, using the Taylor series approximation, this can be estimated as $\tilde{\boldsymbol{F}} \approx \boldsymbol{F} + \boldsymbol{\Delta}\frac{\partial \boldsymbol{F}}{\partial \theta} + \boldsymbol{R}$

Using this in equation 2.15, we get

$$\tilde{\boldsymbol{F}}(\boldsymbol{r}, \tilde{\boldsymbol{\theta}})\boldsymbol{I} \approx \sum_{x,y} \left[\left(e^{-2\pi jr\left(\frac{x\cos\theta}{N} + \frac{y\sin\theta}{N}\right)} + \Delta \times e^{-2\pi jr\left(\frac{x\cos\theta}{N} + \frac{y\sin\theta}{N}\right)} \times 2\pi jr \left(\frac{x\sin\theta}{N} - \frac{y\cos\theta}{N} \right) \right) \times I(x,y) \right] \tag{2.16}$$

This is of the form $\boldsymbol{y} \approx (\boldsymbol{A} + \boldsymbol{\Delta B})\,\boldsymbol{x}$, and hence fits into the proposed framework. The only consideration is that we assumed the signal to be sparse, but natural images are not sparse in the canonical basis. This is easily rectified - since images are likely to be sparse in a basis such as DCT, we simply absorb the DCT matrix, $\boldsymbol{U}$, into $\boldsymbol{A}$ and $\boldsymbol{B}$. $\boldsymbol{A'} \leftarrow \boldsymbol{AU}$ and $\boldsymbol{B'} \leftarrow \boldsymbol{BU}$ suffice.

One caveat to the above extension is that our algorithm assumes the error on the inaccurately known variable to be small. While such a constraint on $\boldsymbol{\theta}$ is not unreasonable, this may still translate into a large error in the frequency values. Experimentally, this was found to be true, and accurate recovery was possible only with extremely small values of perturbation in $\boldsymbol{\theta}$.

2.7 Conclusions

In this chapter a recovery algorithm was designed for compressed sensing of sparse signals under the supposition that the measurement matrix had unknown perturbations. The method addresses a very practical problem of imperfectly calibrated sensing instruments. An extension, for the case of 2-D images was developed, although more work is required to perfect this framework.

Future work on this problem involves utilizing multiple signals, measured under the same perturbation of the sensing matrix. This provides more data points for estimating the perturbation. The number of unknowns - the signals to be recovered - also increase, but this outweighs the gain in the number of known quantities.

Chapter 3

Tomographic Recovery with Unknown View Angles

3.1 Background and Existing Work

The previous chapter tackled the problem where the the sensing matrix was imperfectly known. An extreme case of this scenario is when the view angles are complete unknown. One such setting is cryo-electron microscopy (cryoEM), where tomographic techniques are used to determine the structure of a molecule or cell [12]. Multiple repetitions of the procedure are performed on identical specimens, each of which is arbitrarily oriented. With no way to control the relative orientation of the replica specimens, the tomographic angles are essentially unknown. Another use case of such a technique is in insect tomography [13], or tomography of objects performing unknown rigid motion, which is equivalent to performing a tomographic reconstruction on a fixed object, with the view angles of the projects being unknown. Uncertainty in view angles, though to a lower degree, may also occur due to patient motion in medical imaging [14].

In such a scenario, the method discussed in chapter 2 will not be applicable, since it counts on the error in the sensing matrix being small. Here, we propose a method to reconstruct images from parallel beam tomographic projections, with the view-angles completely unknown, utilizing the relationships in the moments of the image and its projections.

It has been shown in [15] that under certain modest conditions, the view angles can be uniquely determined from the tomographic projections, and that the estimated angles are in principle stable under noise [16] in almost all cases. However the associated algorithms rely on some strong assumptions.

In principle, most existing work on this problem relies on projections at nearby angles (or moments thereof) are being similar, and utilizes this assumption to formulate an embedding of projections onto a lower-dimensional space using dimensionality reduction techniques. For instance, [17] uses Spherical Locally Linear Embedding (sLLE) to embed projections on a circular, 2-D space; [18] utilizes spherical multi-dimensional scaling to reduce dimensionality; [19] uses a Laplacian graph-based manifold learning method to embed projections from various view angles on the circle. Others such as [16] have utilized simple heuristics like a nearest neighbour search to find angle ordering.

These algorithms provide only an ordering on the angles, and one must depend on knowledge of the underlying probability distribution of the angles to recover the actual angle values. Further, the use of such methods necessitates the availability of projections from a large number of angles, to have reasonably small variance [19]. While in certain scenarios, obtaining a large number of measurements may be feasible, it is always advantageous to be able to reconstruct with a fewer view angles.

The method proposed below is mathematically motivated, general in applicability, robust to a fair degree of noise, and viable even with projections from a few angles being available.

3.2 Problem Formulation

Consider the set of angles, $\boldsymbol{\theta} \triangleq \{\theta_1, \theta_2, ..., \theta_P\}$ and the corresponding projections of a 2-D image from these angles, $\boldsymbol{P_\theta} \triangleq \{P_{\theta_i}\}_{i=1}^{P}$. Projection, P_{θ_i}, is a vector of line integrals of the image, as viewed from angle θ_i (from a fixed 0 angle). That is, the s^{th} element of the vector P_{θ_i}, $P_{\theta_i}(s)$, is the line integral along a ray at a distance s from the origin, and inclined at an angle θ_i to the y-axis. Given infinitely many projections (and corresponding angles), it is possible to uniquely reconstruct the original image. Practically, using well-known techniques such as filtered back projection (FBP) or or moment-based methods such as [20] the image can be reasonably reconstructed from a fewer view angles.

Our aim is to recover the angles $\boldsymbol{\theta}$ (and hence the original image), given only the set of tomographic projections $\boldsymbol{P_\theta}$.

There is an inherent ambiguity associated with recovering the view-angles: Rotating the initial image by a fixed angle ϕ would produce the exact same set of projections, but at an angular shift of ϕ. Further, reflecting the original image also allows for the same set of projections to be produced. Therefore, we may have an inherent ambiguity as

$$\hat{\theta}_i = \sigma\theta_i + 2n_i\pi + \phi$$

where θ_i is the original angle from where the i^{th} projection has been acquired, $\hat{\theta}_i$ is the estimated value of this angle, $\sigma \in \{+1, -1\}$, $n_i \in \mathbb{Z}$, and ϕ is the fixed rotational ambiguity.

This is not a limitation as such, and we will evaluate the correctness of results allowing for such an ambiguity.

3.3 Recovery Algorithm

The algorithm consists of three stages:

1. Denoising noisy tomographic projections
2. Angle estimation through iterative coordinate descent
3. Image reconstruction using the estimated angles

Existing algorithms for step 3 perform quite well. For testing, FBP is used for its simplicity, though better algorithms inspired from the compressed sensing literature may be used [21]. Algorithms for steps 1 and 2 are proposed here.

3.3.1 Denoising

A patch-based PCA denoising method was used to reduce the noise in the projections. This algorithm is adapted from a similar algorithm for 2-D images, as described in [22] (a popular method, widely used for denoising).

We consider patches of size $d \times 1$ from a moving window across each projection. For each patch, we find its L nearest patches (in the L^2-norm sense). We perform PCA on this set of L vectors and project each one along the principal directions to produce eigencoefficients. To denoise the patch, we manipulate these coefficients using Wiener-like updates of the form

$$\hat{x_{il}} = y_{il}\left(\frac{\sigma_l^2}{\sigma_l^2 + \sigma^2}\right) \tag{3.1}$$

where $\hat{x_{il}}$ is an estimate of the l^{th} denoised coefficient for patch i ($1 \leq l \leq d$), y_{il} is the corresponding noisy coefficient for patch i, and σ_l^2 (the mean squared value of the l^{th} coefficient across all L patches) is estimated as: $\hat{\sigma_l^2} = \max\left(0, \frac{1}{L}\sum_{i=1}^{L} y_i^2 - \sigma^2\right)$.

The patch-based approach proposed above has two distinct advantages: First, the use of patches (with size appropriately tuned) allows for similarity among different parts of the same projection as well; Second, this method works even when the total number of projections is considerably lower. When finding similarity between entire projections in a small set, it is unlikely that we find very similar projections, but we are much more likely to find short matching patches.

3.3.2 Coordinate Descent

This is the key part of the algorithm. We utilize the Helgasson-Ludwig Consistency Conditions (HLCC) [23], which describe the relationship between the geometric moments of the underlying image $f(x,y)$ and its projections from a given angle. For a particular angle θ_i, the n^{th} order moment of P_{θ_i} is calculated as follows:

$$m_{\theta_i}^{(n)} = \int_{-\infty}^{\infty} P_{\theta_i}(s) s^n ds \tag{3.2}$$

Now, there exist $n+1$ image moments of the n^{th} order. For natural numbers p, q, such that $p+q=n$, the n^{th} order image moments can be calculated as [24]:

$$v_{p,q} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) x^p y^q dxdy \tag{3.3}$$

The HLCC describe a relationship between the n^{th} order projection moments and the n^{th} order image moments:

$$m_{\theta_i}^{(n)} = \sum_{j=0}^{n} \binom{n}{j} (\cos\theta_i)^{n-j} (\sin\theta_i)^j \, v_{n-j,j} \tag{3.4}$$

Using projection moments from multiple angles, (3.4) can be written in matrix form, $\boldsymbol{m}^{(\boldsymbol{n})} = \boldsymbol{A}^{(\boldsymbol{n})}\boldsymbol{v}^{(\boldsymbol{n})}$ where, for the n^{th} order equation, $\boldsymbol{A}^{(\boldsymbol{n})}$ is the $P \times (n+1)$ matrix defined by $\boldsymbol{A}_{ij}^{(\boldsymbol{n})} = \binom{n}{j}(\cos\theta_i)^{n-j}(\sin\theta_i)^j$, and $\boldsymbol{v}^{(\boldsymbol{n})} \triangleq \{v_{p,q} | p+q=n, p,q \in \mathbb{Z}\}$, and $\boldsymbol{m}^{(\boldsymbol{n})}$ is a vector containing the projection moments of order n at P different angles. This equation can be used to determine $\boldsymbol{v}^{(\boldsymbol{n})}$, and for this, we need $P \geq n+1$.

Since, in practice, the projections are noisy, despite denoising, equation 3.4 will not be satisfied exactly. Further, uncertainty in the value of θ_is will translate into more approximation errors. In our problem setting, we do not actually know the values of θ_i. We will, instead, use *hypothesized* values of θ_i. For hypothesized values of $\boldsymbol{\theta}$ and the set of image moments, $\boldsymbol{v} = \{\boldsymbol{v}^{(\boldsymbol{n})}\}_{n=0}^{k}$, we can define our energy function, E as:

$$E(\boldsymbol{\theta}, \boldsymbol{v}) = \sum_{n=0}^{k}\sum_{i=1}^{p} \left(m_{\theta_i}^{(n)} - \sum_{j=0}^{n} \boldsymbol{A}_{i,j}^{(\boldsymbol{n})} \boldsymbol{v}_{n-j,j} \right)^2 \tag{3.5}$$

This equation is the crux of the algorithm. $E(\boldsymbol{\theta}, \boldsymbol{v})$ is the objective we must minimize by appropriately selecting values of the parameters $\boldsymbol{\theta}$ and $\boldsymbol{v}$. This is achieved through an iterative approach, where estimates for each θ_i are improved one at a time, and the values of $\boldsymbol{v}$ recalculated using a pseudo-inverse. We can now describe the iterative algorithm using coordinate descent.

For each θ_i, the algorithm performs a 1-D brute-force search for the value that minimizes E. The image moments are then re-estimated, and the value of E updated At each step, exactly one θ_i is updated. Since each update always reduces E, and E is a non-negative quantity, it is guaranteed to converge eventually. In practice, it is seen that it takes a small number of iterations to converge.

Algorithm 2 Coordinate Descent Algorithm

```
Randomly initialize $\boldsymbol{\theta}$ estimates, by picking each $\theta_i$ uniformly from $-\pi$ to $\pi$
Calculate projection moments, $\boldsymbol{m}_{\boldsymbol{\theta}_i}^{(j)}$ for $1 \leq j \leq k$ (First k orders).
Estimate image moments of the first $k$ orders, $\boldsymbol{v}^{(i)}$, $1 \leq i \leq k$.
Calculate $E$ using 3.5.
Initialize $\Delta E \leftarrow \infty$
while $\Delta E > \epsilon$ do
    for each $\theta_i$ do
        for $\theta_i$ in $-\pi$ to $\pi$, with appropriate step size do
            Recalculate image moments using new $\theta$
            Calculate $E$ again, using updated values of $\theta_i$ and image moments
            if $E <$ previous-best-estimate then
                update the best estimate for $\theta_i$
                $\Delta E \leftarrow$ Old value of $E -$ new value of $E$
                Update the value of $E$
```

Further, the highest order moment to consider, k is a tunable parameter. When we increase k by 1, we increase the number of knowns by p - one known projection moment of order k for each of the p angles - and the unknowns by only k - image moments of the k^{th} order, introducing more redundancy in the system. So, in theory, it is feasible to use a very large k. However, in experiments, it was seen that while angle recovery generally improves with increasing the value of k, going beyond order 6 or 7, the gain was marginal, but increased computation time significantly.

The search space is not convex, and so, the convergence point using coordinate descent may be sensitive to initialization values of θ_is. To counteract this to a certain extent, we used a multi-start strategy, where the above algorithm is run multiple times, each time with a random initialization for $\boldsymbol{\theta}$. Finally, the iteration with the least value of E at convergence is chosen as the optimal set of assignments for $\boldsymbol{\theta}$. Using this multi-start strategy, we almost always obtained accurate angle estimates.

3.4 Experimental Results

Experiments were conducted for multiple square images, with a size of 200x200 pixels, under zero mean, i.i.d. Gaussian noise with standard deviation varying from 1% to 10% of the standard deviation of the tomographic projections. It was observed that the final angles recovered were consistently close (almost entirely within $\pm 1^\circ$, occasional outliers reaching $\pm 5^\circ$) to the actual angles used to compute the projections. In each case, ten uniformly random initialization values were used for angle estimates, and the search was performed in 1° steps, with a value of $k = 5$. To test the robustness of the algorithm to different angle distributions, experiments were conducted on

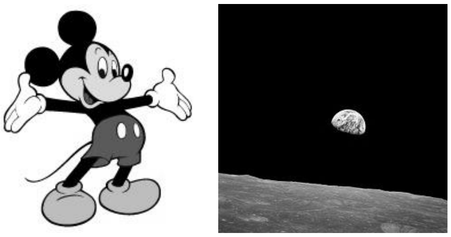

Figure 3.1: *Original images used. Reconstruction results in Fig 3.2*

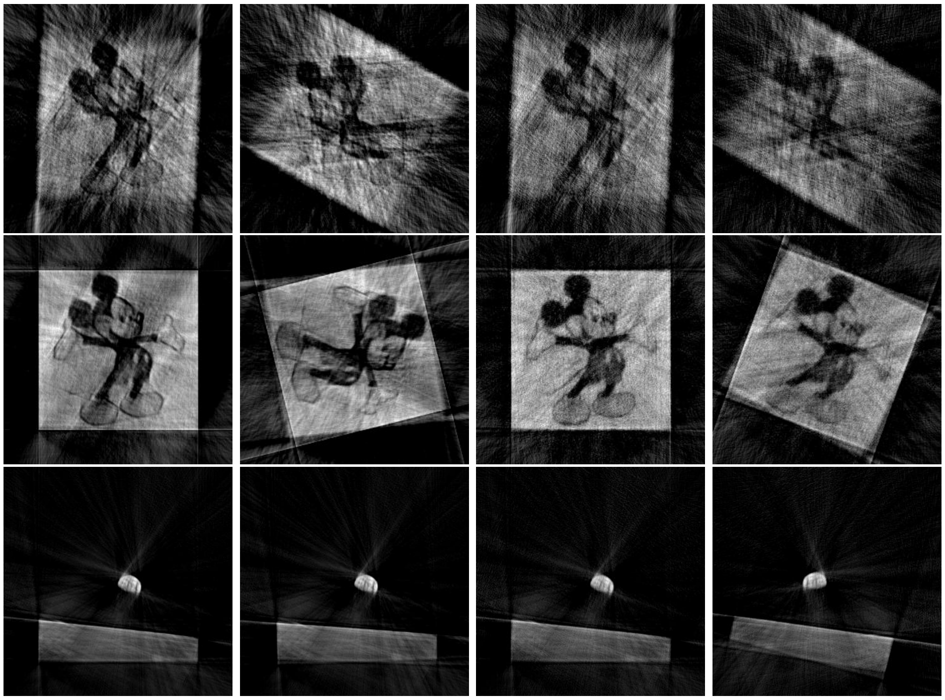

Figure 3.2: *FBP reconstruction using a non-uniform distribution of angles. Left to right: Using (a) 5% noise & actual angles, (b) 5% noise & estimated angles, (c) 10% noise & actual angles (d) 10% noise & estimated angles Top row: Reconstructions done with 30 angles, Middle row: Reconstructions done with 100 angles Bottom row: Reconstructions done with 100 angles Image canvas sizes have been expanded to allow for images to fit rotated reconstructions*

non-integer angles sampled from (a) a $(0, 180)$ uniform distribution, (b) a non-uniform distribution, picking angles from each successive 30° interval in the ratio 0.2:0.3:0.12:0.3:0.35 and (c) an extremely peaky distribution with 10 angle values sampled close to each of 10 randomly chosen, spread out peaks.

Reconstruction results using filtered back-projection and with a set of 30 angles as input for two images are included in Figure 3.2 along with the angle estimation results in the top row of Figure 3.3. Although, reconstructed images are also shown for visual interpretation, it should be noted that the reconstruction suffers from significant noise because of the limitations of FBP, and should be considered only in comparison to the baseline provided alongside. The quality of angle recovery can be seen in Figure 3.3 for different images and angle distributions at a noise level of 10%.

Table 3.1: *Statistics of errors in angle recovery, 5% noise*

Error	Earthrise	Earthrise	Mickey	Mickey
	30 angles	100 angles	30 angles	100 angles
$\leq 1°$	13	94	20	66
$\leq 3°$	29	99	29	96
$\leq 5°$	30	100	29	100
$> 5°$	0	0	1	0

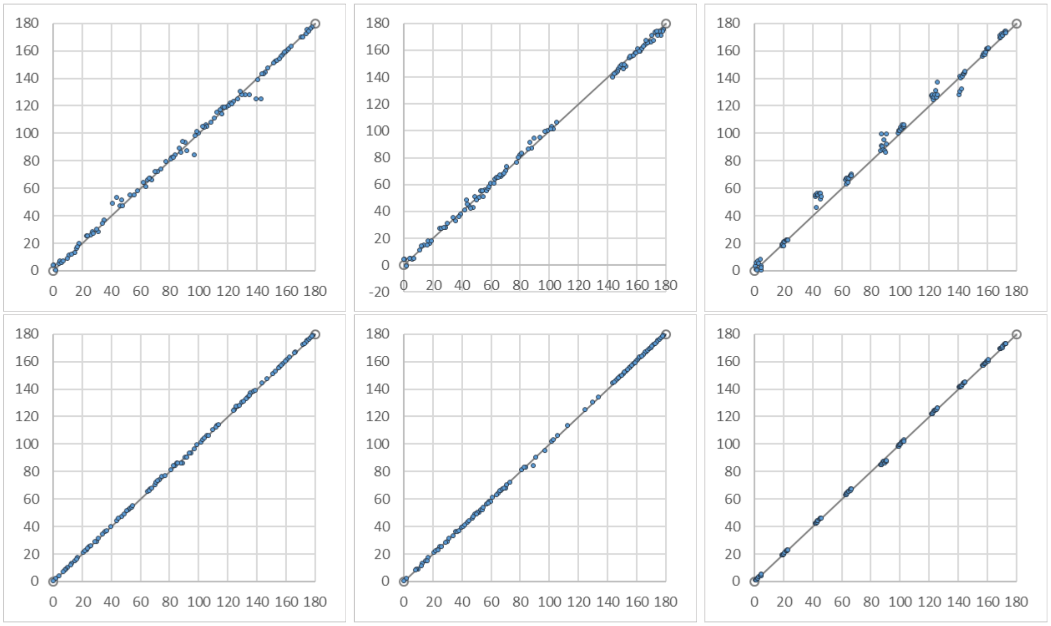

Figure 3.3: *In each figure, X and Y axes show true and estimated angles respectively, under 10% noise. Top row: image 'mickey'; Bottom row: image 'earthrise'. Angles drawn from - Left: uniform distribution; Centre: non-uniform distribution; Right: very peaky distribution. For visualization, the offset ϕ in the estimated angles has been removed manually*

3.5 Conclusions

In this chapter, a general method for image reconstruction from projections from unknown view angles was proposed, and its efficiency empirically demonstrated in a wide variety of scenarios - with varying number of view angles, with different distributions for generation of the angles, and at multiple noise levels. The key idea is to iteratively improve angle estimates to reduce HLCC residuals, using a coordinate descent strategy.

On experimenting with values of the maximum order of image moments to be considered, it was discovered that there is a clear trade-off between accuracy and computational time, but only up to a (fairly low) threshold, after which, increasing the order causes no noticeable improvement in the recovered angles. However, this remains a parameter which must be tuned by the user.

While the algorithm presented here works even in the extreme scenario where the angles are completely known, it could be used in the case where these are imperfectly known, using the approximate knowledge to initialize the angle estimates. This use case is similar to one presented in chapter 2. However, for such a scenario, the method in chapter 2 is recommended, since the perturbed sensing algorithm is much more sensitive to precise with smaller errors. Further, the algorithm presented here uses filtered back projection for reconstruction, which often does not yield results as good as with traditional compressive sensing related optimizers.

Chapter 4

Recovery Bounds for Composite Signals

4.1 Background

Consider the setting familiar in compressive sensing ([3], [2]). We have measurements $\boldsymbol{y} \in \mathbb{R}^M$ from the k-sparse signal $\boldsymbol{x} \in \mathbb{R}^N$ generated using the model

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{\eta} \tag{4.1}$$

where $\boldsymbol{D} \in \mathbb{R}^{M \times N}$ is the measurement matrix or dictionary with unit-normalized columns, and $\boldsymbol{\eta} \in \mathbb{R}^M$ is the measurement noise, with $||\boldsymbol{\eta}||_2 \leq \epsilon$. The signal $\boldsymbol{x}$ can be recovered using basis pursuit ([25]), and a coherence-based recovery guarantee is well known ([26]).

We consider the case where the signal $\boldsymbol{x}$ is a composite of two individually sparse signals $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$. That is

$$\boldsymbol{D} = [\boldsymbol{A}\ \boldsymbol{B}]\ ,\ \ \boldsymbol{x}^T = [\boldsymbol{x}_a^T\ \boldsymbol{x}_b^T]$$

where $\boldsymbol{A} \in \mathbb{R}^{M \times N_a}$, $\boldsymbol{B} \in \mathbb{R}^{M \times N_b}$,
$\boldsymbol{x}_a \in \mathbb{R}_a^N$, $\boldsymbol{x}_b \in \mathbb{R}_b^N$, $N_a + N_b = N$.

Our aim is to recover both the individual interfering signals $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$. A naive bound can be obtained on the recovery, disregarding the internal structure of $\boldsymbol{x}$. However, in such a case, the recovery guarantee can be improved. [27] and [28] both explore such interfering signals. We specifically use results from [28], which show that the guarantee can be expressed in terms of the coherence values of the individual sub-dictionaries $\boldsymbol{A}$ and $\boldsymbol{B}$.

$$\mu_a := \max_{i,j,i \neq j} |\boldsymbol{A}_i^T \boldsymbol{A}_j|, \qquad \mu_b := \max_{i,j,i \neq j} |\boldsymbol{B}_i^T \boldsymbol{B}_j|,$$

and their mutual coherence, defined as

$$\mu_m := \max_{i,j} |\boldsymbol{A}_i^T \boldsymbol{B}_j|.$$

Consequently, the coherence for dictionary $\boldsymbol{D}$ can be expressed as

$$\mu_d := \max_{i,j,i\neq j} |\boldsymbol{D}_i^T \boldsymbol{D}_j| = \max\{\mu_a, \mu_b, \mu_m\}$$

where $\boldsymbol{X}_i$ indicates the i^{th} column of matrix $\boldsymbol{X}$.
Assuming without loss of generality that $\mu_b \leq \mu_a$, the basis pursuit bound, then, says (Theorem 4 in [28]):
If

$$k < \max\left\{\frac{2(1+\mu_a)}{\mu_a + 2\mu_d + \sqrt{\mu_a^2 + \mu_m^2}}, \frac{1+\mu_d}{2\mu_d}\right\} \tag{4.2}$$

then the solution $\hat{\boldsymbol{x}}$ of the basis pursuit problem

$$\begin{aligned} &\text{minimize } ||\tilde{\boldsymbol{x}}||_1 \\ &\text{subject to } ||\boldsymbol{y} - \boldsymbol{D}\tilde{\boldsymbol{x}}||_2 \leq \eta \end{aligned}$$

$$\text{satisfies } ||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2 \leq C(\epsilon + \eta) + C'(||\boldsymbol{x} - \boldsymbol{x}_s||_1) \tag{4.3}$$

where k is the number of non-zero entries in $\boldsymbol{x}$, $\boldsymbol{x}_s$ is $\boldsymbol{x}$ after setting all but the k largest components to 0, and $C, C' \geq 0$ are constants. Further, it holds that (Appendix D, [28])

$$(1-\hat{\delta})||\boldsymbol{x}||_2^2 \leq ||\boldsymbol{D}\boldsymbol{x}||_2^2 \leq (1+\hat{\delta})||\boldsymbol{x}||_2^2 \tag{4.4}$$

with

$$\hat{\delta} = \min\left\{\frac{1}{2}\left(\mu_a(k-2) + k\sqrt{\mu_a^2 + \mu_m^2}\right), \mu_d(k-1)\right\} \tag{4.5}$$

The requirement in equation (4.2) improves on the bound for a general $\boldsymbol{D}$ (requiring $k \leq \frac{1+\mu_d}{2\mu_d}$).

This is the key result that this chapter builds on and extends. While the work of Studer and Baraniuk [28] utilises the underlying interfering signals, we propose a partition of the composite signal into two fresh signals, such that the contents of the original component signals are maintained, while improving the recovery guarantee. The induced partition is such that it optimizes the requirement on the sparsity of the signals.

While the recovery algorithm itself is unchanged, our proposed algorithm produces a tighter coherence-based bound for recovery using basis pursuit, for a given dictionary $\boldsymbol{D}$.

It is worth noting that although the motivation for this comes from the compressed measurement of a signal composed of two interfering signals, the results are also applicable to any arbitrary signal $\boldsymbol{x}$ and corresponding dictionary $\boldsymbol{D}$, with no inherent structure.

4.2 Two-Way Dictionary Splitting

First, it is abundantly clear that swapping any two columns $\boldsymbol{D}_i$ and $\boldsymbol{D}_j$ of a dictionary, while swapping the corresponding elements $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ has no impact on the product $\boldsymbol{Dx}$. In fact, if $\boldsymbol{D}'$ is a permutation of the columns of $\boldsymbol{D}$, and $\boldsymbol{x}'$ the corresponding transformation of $\boldsymbol{x}$, then $\boldsymbol{Dx} = \boldsymbol{D}'\boldsymbol{x}'$. Moreover, $\boldsymbol{x}$ and $\boldsymbol{x}'$ clearly have the same sparsity, and if the order of permutation is known, $\boldsymbol{x}'$ determines $\boldsymbol{x}$ exactly. Therefore, the problem of recovering the signal $\boldsymbol{x}$, given $\boldsymbol{D}$, and measurements $\boldsymbol{y} = \boldsymbol{Dx} + \boldsymbol{\eta}$ is equivalent to recovering $\boldsymbol{x}'$, given $\boldsymbol{D}$, $\boldsymbol{D}'$ and $\boldsymbol{y}$.

Secondly, for a given $\boldsymbol{D}$, the quantity in the RHS of equation (4.2) can be maximised directly by maximising the quantity

$$F = \frac{2(1+\mu_a)}{\mu_a + 2\mu_d + \sqrt{\mu_a^2 + \mu_m^2}} \tag{4.6}$$

where $\mu_a \geq \mu_b$.

We can now define the problem of dictionary split induction as: Given a dictionary $\boldsymbol{D} \in \mathbb{R}^{M \times N}$, devise matrices $\boldsymbol{A}$, $\boldsymbol{B}$, such that the columns of $\boldsymbol{D}'$ can be permuted to give $\boldsymbol{D}$, where $\boldsymbol{D}' = [\boldsymbol{A}\ \boldsymbol{B}]$, and F is maximised.

4.2.1 Algorithm

We solve the problem of dictionary splitting by mapping it to an equivalent graph problem. The transformation is as follows:

Let $G = (V, E)$ be a complete, undirected, weighted graph, with vertices V and edges E, such that each vertex corresponds to a column of the matrix $\boldsymbol{D}$, and the weight of edge uv equals the dot product of the columns corresponding to vertices u and v. The problem, then, is to define a cut through G that partitions it into two disjoint components, $A(V_a, E_a)$ and $B(V_b, E_b)$. Let E_x be all the edges crossing the cut, and V_x be the set of vertices connected by edges in E_x. Note that

(a) $V_a \subseteq V$, $V_b \subseteq V$, $V_a \cup V_b = V$

(b) $E_a \subseteq E$, $E_b \subseteq E$, $E_x \subseteq E$, $E_a \cup E_b \cup E_x = E$.

Let us also define the following quantities on the graph G:

$$\mu_a := \max_{e \in E_a} w(e) \quad ; \quad \mu_b := \max_{e \in E_b} w(e)$$

$$\mu_m := \max_{e \in E_x} w(e)$$

$$\mu_d := \max_{e \in E} w(e) = \max\{\mu_a, \mu_b, \mu_m\}$$

where $w(e)$ denotes the weight of edge e.

There is a one-to-one correspondence between the terms μ_a, μ_b, μ_m, μ_d defined on the graph G and the equivalent coherence values defined on the dictionary $\boldsymbol{D}$. The matrix $\boldsymbol{A}$ can be constructed by selecting the columns from $\boldsymbol{D}$ corresponding to sets V_a and concatenating them in an arbitrary order. Similarly, the matrix $\boldsymbol{B}$ can be constructed from V_b.

This is essentially a greedy algorithm. Intuitively, we try to process edges starting from the heaviest, and to add each to the set E_x of cut edges, as long as doing so does not create a contradiction in the colouring of vertices. After each step, the edges in consideration are those that are adjacent to edges already included in E_x. In the following iteration, the heaviest of these is selected as the candidate for inclusion.

Initially, each vertex is unassigned (i.e. $colour(v) = 0$), and the Q, the queue of edges yet to be processed, is empty ($Q = \phi$). As vertices are processed, they are assigned to set $\mathbf{A}(colour(v) = 1)$ or $\mathbf{B}(colour(v) = 2)$, and the edges emanating from the vertex are added to the queue Q.

This algorithm provides us with a two-way split in the dictionary that allows for the tightest recovery guarantee. The process for recovering the signal (i.e. basis pursuit) is unaltered.

4.2.2 Running time complexity

Let the number of vertices in graph G be n (This corresponds to the number of columns in dictionary $\boldsymbol{D}$). Steps 1 through 7 involve n sorting operations, each consuming time $\mathcal{O}(n \log n)$, for a total of $\mathcal{O}(n^2 \log n)$ operations. Steps 8 through 12 consist of one time assignments, and hence, can be performed in $\mathcal{O}(1)$ time.

The while loop in steps 13 through 23 runs for at most $\mathcal{O}(n^2)$ iterations, since each iteration processes one edge. All operations inside the loop are $\mathcal{O}(1)$ complexity, except the merge operation (step 18) and the pop operation (step 22). Although, with a naïve implementation, a merge might take time $\mathcal{O}(n^2)$, with the use of an appropriate data structure, such as a Fibonacci heap ([29]) for representing Q and L, we can get the amortised time complexity of a single merge operation down to $\mathcal{O}(1)$, while still getting a time complexity of a single pop operation of $\mathcal{O}(\log n^2) = \mathcal{O}(\log n)$. Since a merge operation only happens when we encounter an unprocessed vertex (inside the *if* clause), there are $\mathcal{O}(n)$ merges. A pop operation occurs in every iteration of the loop, so there are $\mathcal{O}(n^2)$ pops. Thus, all iterations collectively consume time $\mathcal{O}(n^2 \log n)$.
Hence, overall, the algorithm consumes time $\mathcal{O}(n^2 \log n)$.

Algorithm 3 Two-way Splitting Algorithm

```
for $v \in V$ do
    $colour(v) \leftarrow 0$
    $L(v) \leftarrow$ edges from $v$, in decreasing order of weight
end for
$E_x$ = cut edges $\leftarrow \phi$
$Q$ = list of edges to process $\leftarrow \phi$
$processed \leftarrow \phi$

$e = uv \leftarrow$ heaviest edge in G
$colour(u) \leftarrow 1$
$L(u) \leftarrow L(u) \setminus \{e\}$
$Q \leftarrow L(u)$
$processed \leftarrow \{u\}$

while size($processed$) $< n$ do
    if one of $\{u, v\}$ (say $v$) is s.t. $colour(v) == 0$ then
        if $colour(u) == 1$ then $colour(v) \leftarrow 2$
        if $colour(u) == 2$ then $colour(v) \leftarrow 1$
        $L(v) \leftarrow L(v) \setminus \{e\}$
        Merge $L(v)$ into $Q$
        $processed \leftarrow processed \cup \{v\}$
    end if
    if $colour(u)! = colour(v)$ then $E_x \leftarrow E_x \cup \{e\}$
    $e = uv \leftarrow$ Pop heaviest edge from $Q$
end while

$\mathbf{V}_a \leftarrow \{v \mid colour(v) == 1\}$
$\mathbf{V}_b \leftarrow \{v \mid colour(v) == 2\}$
Output $\mathbf{V}_a$, $\mathbf{V}_b$
```

4.2.3 Proof of Optimality

Consider the heaviest edge in G, say $e^* = uv$. At the end of any possible assignment, either $\max\{\mu_a, \mu_b\} = w(e^*)$, or $\mu_m = w(e^*)$. With this consideration in mind, it is clear from equation (4.6) that the latter case is always advantageous i.e. to maximize F, $e^* \in E_x$. By the same logic, the same reasoning can be applied to the next heaviest edge, the ones after, and so on, until it is impossible to add an edge to E_x - because doing so will create an inconsistent assignment of the vertices. In other words, adding edge e to E_x will create an odd length cycle, which cannot possibly occur in a graph cut.

But this addition is precisely the kind that is not allowed by our algorithm. $E_x \leftarrow E_x \cup \{e\}$ only if one of the vertices is unassigned ($colour(v) == 0$). It can also be verified by an exchange argument that the edge e that is not included in E_x must be the lightest edge in the cycle.

Thus, the assignment returned by Algorithm 3 must be optimal i.e., the tightest bound can be provided by exploiting the dictionary structure as suggested by Algorithm 3.

4.3 Higher Order Recursive Algorithm

Algorithm 3 gives us an optimal two-way split for $\boldsymbol{D}$. The split is optimal in the sense of maximising F in equation (4.6), and minimizing $\hat{\delta}$ in equation (4.5). The process, in theory, can be generalized to splitting $\boldsymbol{D}$ into any number of splits to improve the bound. However, to consider each possible split naively would require exponential time, and would be prohibitively expensive, for any reasonable size dictionary. We now describe a recursive, polynomial-time heuristic to create a multi-way split in $\boldsymbol{D}$, to further tighten the bounds.

Consider equation (4.5). When the value of $\hat{\delta}$ is equal to the first term in the minimization, we can imagine the dictionary $\boldsymbol{D}$ being equivalent to a hypothetical dictionary $\tilde{\boldsymbol{D}}$, with

$$\tilde{\mu_d}(k-1) = \frac{1}{2}\left(\mu_a\,(k-2) + k\sqrt{\mu_a^2 + \mu_m^2}\right). \tag{4.7}$$

For this bound to be better than that obtained by considering no structure in the dictionary, $\tilde{\mu_d}$ must be less than the second term in the minimization in equation (4.5). That is, $\tilde{\mu_d} < \mu_d$. By construction, Algorithm 3 produces $\mu_m = \mu_d$ (Since the heaviest edge in the graph is first placed in the set of cut edges). Therefore, this condition can be expressed as

$$\frac{1}{2}\left(\mu_a\,(k-2) + k\sqrt{\mu_a^2 + \mu_m^2}\right) < \mu_m(k-1)$$

$$\implies \frac{\mu_a}{\mu_m} < 1 - \frac{k}{2}\left(1 - \sqrt{\frac{k-2}{k-1}}\right). \tag{4.8}$$

As k increases, $\frac{\mu_a}{\mu_m}$ asymptotically converges to a value of $\frac{3}{4}$, as show in the graph below.

In fact, for large k, equation 4.7 reduces to:

$$\tilde{\mu_d} = \frac{1}{2}\left(\mu_a + \sqrt{\mu_a^2 + \mu_m^2}\right). \tag{4.9}$$

That is, the role of coherence of the hypothetical dictionary is played by the quantity

$$\frac{1}{2}\left(\mu_a + \sqrt{\mu_a^2 + \mu_m^2}\right)$$

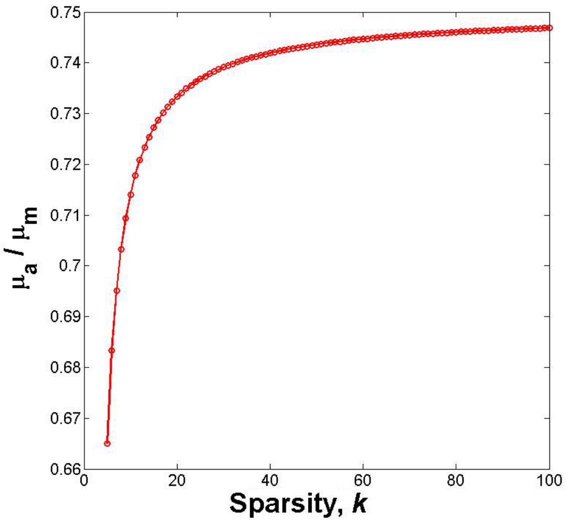

Figure 4.1: μ_a/μ_m *vs* k

It also follows that $\tilde{\mu_d} \leq \mu_m = \mu_d$ when $\mu_a \leq \frac{3}{4}\mu_m$ (for large k), resulting in tighter bounds for recovery.

These results use the coherence values of sub-dictionaries **A** and **B** as μ_a, and μ_b respectively. However, we can recursively apply the dictionary splitting algorithm to each of these, to improve the *effective* coherence values ($\tilde{\mu_d}$ in equation 4.9), $\tilde{\mu_a}$, and $\tilde{\mu_b}$.

4.3.1 Algorithm

The algorithm presented below uses a greedy approach, using the optimal 2-way split at each level, leveraging equation 4.9 at each step to determine the equivalent coherence of the split pair. The algorithm uses a call to **Split_2Way**($\boldsymbol{D}$), a function designed as presented in Algorithm 3, to return the optimal two-way split for dictionary $\boldsymbol{D}$. The multi-way split algorithm executed to depth l, achieves a 2^l-way split.

Unfortunately, unlike Algorithm 3, this algorithm is not guaranteed to produce the optimal split from the set of all possible 2^l-way splits, and counter-examples can be constructed to prove its sub-optimality. However, it is guaranteed to produce an effective-coherence that is at least as good as the split in Algorithm 3, since this case is subsumed in Algorithm 4.

Algorithm 4 Depth l multi-way Split Algorithm

```
function EFFECTIVE_COHERENCE($\boldsymbol{D}, l$)
    $\boldsymbol{A}, \boldsymbol{B}$ = Split_2Way($\boldsymbol{D}$)
    $\mu_a \leftarrow$ Effective_Coherence $(\boldsymbol{A}, (l-1))$
    $\mu_b \leftarrow$ Effective_Coherence $(\boldsymbol{B}, (l-1))$
    $\mu_m \leftarrow \max_{i,j} |\boldsymbol{A}_i^T \boldsymbol{B}_j|$
    $\mu_d \leftarrow \max \{\mu_a, \mu_b, \mu_m\}$
    if l==1 then
        return $\mu_d$
    else
        return min $\left\{\mu_d, \frac{1}{2}\left(\mu_a + \sqrt{\mu_a^2 + \mu_m^2}\right)\right\}$
end function

$l \leftarrow$ maximum depth to explore
Output Effective_Coherence $(\boldsymbol{D}, l)$
```

4.4 Empirical Results

For empirical tests, each dictionary, $\boldsymbol{D}$, was designed by shuffling together the columns of two orthogonal matrices. $\boldsymbol{D}$ will have a low value of coherence. However, using the two-way splitting algorithm described in Section 4.2, we were able to reduce the effective coherence even further - perfectly splitting the two orthogonal matrices each time. The improvement in the bound on k and $\hat{\delta}$ vary with the size of the matrix. The details are summarized in the following table.

Unsplit dictionary

N	200	500	1000
μ_d	0.141	0.0894	0.0632
Upper bound on k	4.036	6.090	8.406
$\hat{\delta}$	6.930	31.216	44.209

Split dictionary

μ_a	0	0	0
μ_b	0	0	0
μ_m	0.141	0.0894	0.0632
μ_d	0.141	0.0894	0.0632
Effective coherence $\tilde{\mu}_d$	0.720	0.0448	0.0317
Upper bound on k	7.521	11.648	16.289
$\hat{\delta}$	3.536	15.652	22.136

For each experiment, the orthogonal matrices were perfectly recovered, giving us μ_a and μ_b

of 0. $\mu_m = \mu_d$, by construction, as specified in Algorithm 3. The bound on k and $\hat{\delta}$ improved, typically by a factor of approximately 2 in each case.

Since our proposed method only presents an approach to improve the recovery bounds (like in Theorem 4, [28]), and not a new algorithm for actual recovery, we refrain from showcasing experiments that empirically compare the recovered signal to the true signal. We employ the well known basis pursuit denoising algorithms for recovery, and the retrieved signal will be *identical* to the scenario where our dictionary splitting technique is not used to calculate the bounds.

Chapter 5

Conclusion and Future Work

Recovery of compressed signals was studied under a variety of different kinds of perturbation. Each scenario prompted the use of a separate class of algorithmic techniques. Perturbations studied in chapters 2 and 3 were tackled by providing better recovery algorithms, whereas in chapter 4, a technique to devise more efficient recovery guarantees was proposed and tested.

5.1 Perturbed Sensing Matrix

In chapter 2 a recovery technique for sparse signals was designed even if the sensing matrix was imperfectly known. The method addresses a very practical problem of imperfectly calibrated sensing instruments. An extension, for the case of 2-D images was developed, although more work is required to perfect this framework.

The algorithm devised works when the errors in the sensing matrix are small. This is in stark contrast to the algorithm discussed in chapter 3, where the angles are completely unknown. A potential pipeline might include using the algorithm for completely unknown view-angles, and determining the approximate value of each, followed by fine-tuning the precise values using the algorithm for perturbed radial-scan sensing matrix.

A prospective line of future work involves a pooling process where multiple signals are measured under the same systemic perturbation, with information from all the measurements pooled together to accurately recover the sensing matrix, followed by the signal recovery in a subsequent step. A second thread for future work includes theoretical results - proving (or contradicting the idea) that the optimization problem similar to one in equation 2.12 leads to a unique solution that is optimal.

5.2 Tomographic Recovery with Unknown View Angles

In chapter 3, a method is proposed for image reconstruction from projections from unknown view angles. Empirically, it was found to be robust to the number and distribution of angles, and to reasonable noise levels. The algorithm essentially utilises and iterative strategy to improve angle estimates and reduce HLCC residuals, using a coordinate descent approach.

In practical applications, besides unknown view angles, different projections could be acquired at different and unknown shifts. This scenario can be handled easily assuming the background intensity as 0, since that would produce projections padded with an appropriate number of zeros.

There exist several open avenues for future research, such as handling projections with impulse noise or missing bins, projections with fan-beam geometry, reconstructions of 3D objects, and exploring the relationship of this problem with tasks such as automated instrument calibration in compressed sensing.

The above results were presented at IEEE International Conference on Image Processing (ICIP 2016)

5.3 Recovery Bounds for Composite Signals

In chapter 4, we extended existing results on recovery guarantees for composite signals with two interfering constituent sparse signals. The results were generalized to arbitrary signals and corresponding dictionary, where structure in the sensing matrix may not be automatically visible. In fact, the first algorithm proposed in the paper discovers (in polynomial time) the optimal partition of the measurement matrix that provides the tightest bound with a two-part dictionary. Additionally, a further extension - the multi-way splitting algorithm is also presented, which provides a heuristic to extend the result by recursively applying the two-way splitting algorithm to each partition, and provides an even stronger bound on the recovery. While the multi-way splitting algorithm improves the bound, it is not guaranteed to return the optimal split (unlike the two-way splitting algorithm) - since the possible number of splits grows exponentially large.

Neither the algorithm for actually recovering the signal nor the objective function, namely $\min ||\boldsymbol{x}||_1$ s.t $||\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}||_2 <= \eta$, is modified. The splitting schemes proposed only improve the theoretical bound on recovery for a given dictionary. Both algorithms are applicable to any generic dictionary. Future work will deal with providing an efficient algorithm for a general order, multi-way split in dictionaries.

Results derived here have been submitted for review to The 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2017)

Acknowledgment

I would like to express my gratitude to Prof. Ajit Rajwade, for introducing me to the world of signal processing, and guiding me at every step of the project. His advice has been critical, and his patience, invaluable.

A huge debt is also owed to Dr. Karthik Gurumoorthy for his guidance and constant support. He helped shape not only the results presented, but also my way of thinking.

Bibliography

[1] Emmanuel J Candes, Justin K Romberg, and Terence Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[2] Emmanuel J Candès, Justin Romberg, and Terence Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[3] Emmanuel J Candès and Michael B Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[4] James OâĂŹCallaghan, Jack Wells, Simon Richardson, Holly Holmes, Yichao Yu, Simon Walker-Samuel, Bernard Siow, and Mark F Lythgoe, "Is your system calibrated? mri gradient system calibration for pre-clinical, high-resolution imaging," *PloS one*, vol. 9, no. 5, pp. e96568, 2014.

[5] Ethan K Brodsky, Alexey A Samsonov, and Walter F Block, "Characterizing and correcting gradient errors in non-cartesian imaging: Are gradient errors linear time-invariant (lti)?," *Magnetic resonance in medicine*, vol. 62, no. 6, pp. 1466–1476, 2009.

[6] Payal S Bhavsar, Nicholas R Zwart, and James G Pipe, "Fast, variable system delay correction for spiral mri," *Magnetic resonance in medicine*, vol. 71, no. 2, pp. 773–782, 2014.

[7] Nii Okai Addy, Holden H Wu, and Dwight G Nishimura, "Simple method for mr gradient system characterization and k-space trajectory estimation," *Magnetic resonance in medicine*, vol. 68, no. 1, pp. 120–129, 2012.

[8] Gongguo Tang, Badri Narayan Bhaskar, Parikshit Shah, and Benjamin Recht, "Compressed sensing off the grid," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7465–7490, 2013.

[9] Benjamin Friedlander and Thomas Strohmer, "Bilinear compressed sensing for array self-calibration," in *2014 48th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2014, pp. 363–367.

[10] Zai Yang, Cishen Zhang, and Lihua Xie, "Robustly stable signal recovery in compressed sensing with structured matrix perturbation," *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4658–4671, 2012.

[11] Jonathan M Nichols, Albert K Oh, and Rebecca M Willett, "Reducing basis mismatch in harmonic signal recovery via alternating convex search," *IEEE Signal Processing Letters*, vol. 21, no. 8, pp. 1007–1011, 2014.

[12] J. Frank, *Three-Dimensional Electron Microscopy of Macromolecular Assemblies*, Elsevier, 1996.

[13] S. Walker et al, "In vivo time- resolved microtomography reveals the mechanics of the blowfly flight motor," *PLOS BIOLOGY*, vol. 12, 2014.

[14] M. Wood, M. Shivji, and P. Stanchev, "Planar-motion correction with use of k-space data acquired in fourier MR imaging," *Journal of Magnetic Resonance Imaging*, vol. 5, no. 1, pp. 57–64, 1995.

[15] S. Basu and Y. Bresler, "Uniqueness of tomography with unknown view angles," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 1094–1106, 2000.

[16] S. Basu and Y. Bresler, "Feasibility of tomography with unknown view angles," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 1107–1122, 2000.

[17] Y. Fang, M. Sun, S. Vishwanathan, and K. Ramani, "sLLE: Spherical locally linear embedding with applications to tomography," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1129–1136.

[18] Y. Fang, S. Murugappan, and K. Ramani, "Estimating view parameters from random projections for tomography using spherical mds," *BMC medical imaging*, vol. 10, no. 1, pp. 12, 2010.

[19] R. Coifman, Y. Shkolnisky, F. Sigworth, and A. Singer, "Graph laplacian tomography from unknown random projections," *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1891–1899, 2008.

[20] T.J. Wang and T.W. Sze, "The image moment method for the limited range CT image reconstruction and pattern recognition," *Pattern Recognition*, vol. 34, no. 11, pp. 2145–2154, 2001.

[21] S.J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An Interior-Point Method for Large-Scale l1-Regularized Least Squares," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 1, no. 4, pp. 606–617, 2008.

[22] D. D. Muresan and T. Parks, "Adaptive principal components and image denoising," in *ICIP (1)*, 2003, pp. 101–104.

[23] F. Natterer, *The Mathematics of Computerized Tomography*, Wiley, 1986.

[24] A. Rosenfeld and A. Kak, *Digital picture processing*, vol. 1, Elsevier, 2014.

[25] Scott Shaobing Chen, David L Donoho, and Michael A Saunders, “Atomic decomposition by basis pursuit,” *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[26] Tony Cai, Lie Wang, and Xu Guangwu, “Stable recovery of sparse signals and an oracle inequality,” *IEEE Transactions on Information Theory*, vol. 7, no. 56, pp. 3516–3522, 2010.

[27] Christoph Studer, Patrick Kuppinger, Graeme Pope, and Helmut Bolcskei, “Recovery of sparsely corrupted signals,” *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3115–3130, 2012.

[28] Christoph Studer and Richard G Baraniuk, “Stable restoration and separation of approximately sparse signals,” *Applied and Computational Harmonic Analysis*, vol. 37, no. 1, pp. 12–35, 2014.

[29] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, MIT press, 2009.