

COMPRESSIVE SENSING: APPLICATIONS FROM 1-D
TO N-D

Kenneth V. Domingo

kdomingo@nip.upd.edu.ph

AN UNDERGRADUATE THESIS SUBMITTED TO
NATIONAL INSTITUTE OF PHYSICS
COLLEGE OF SCIENCE
UNIVERSITY OF THE PHILIPPINES
DILIMAN, QUEZON CITY

In Partial Fulfillment of the Requirements

for the Degree of

BACHELOR OF SCIENCE IN APPLIED PHYSICS

JUNE 2020

Acknowledgments

Abstract

Contents

1	Introduction	3
1.1	Related literature	4
1.2	Novelty	6
1.3	Thesis overview	8
2	Preliminaries	9
2.1	Sampling paradigms	10
2.2	Sparsity	12
2.3	Incoherence	13
2.4	Reconstruction strategies	14
3	Random sampling-based compressive sensing	16
3.1	Test case: Sinusoid	16
3.2	Effect of random distribution on reconstruction error	19
3.2.1	Uniform	19
3.2.2	Gaussian	20
3.2.3	Poisson	20
3.2.4	Triangular	20

3.2.5	Results & discussion	21
4	Image compressive sensing	23
4.1	Test case: Sinusoidal pattern	24
4.2	Image with multiple sinusoids	26
4.2.1	Pre-processing	26
4.2.2	Processing	26
4.2.3	Reconstruction evaluation	26
4.3	Simultaneous compression & encryption	29
4.3.1	Methodology	30
4.3.2	Results & discussion	31
4.3.3	Correlation analysis	31
4.3.4	Key sensitivity analysis	32
5	Audio compressive sensing	37
5.1	Test case: Sinusoid redux	37
5.2	Comparison of algorithms	38
5.3	Speech	39
5.3.1	Sparse transformation	39
5.3.2	Pre-processing	40
5.3.3	Processing	41
5.3.4	Reconstruction evaluation	41
5.3.5	Error space analysis	42
6	Conclusions	46
A	Codes and Implementations	51

List of Figures

2.1	Original 512×512 , 8-bit image (left), and a random subset (for better visibility) of its D8 DWT coefficients (middle). Most of the signal energy is concentrated in just a few terms. By discarding all but the 25,000 highest coefficients and performing the inverse transform, the resulting image (right) is perceptually no different from the original.	13
3.1	Original test signal (blue) and random measurements (orange).	17
3.2	Original signal (leftmost), LASSO reconstruction (middle), and CVXPY reconstruction (rightmost). The top row shows the time domain representation, while the bottom row shows the frequency domain representation.	19
3.3	Probability densities of the different random distributions used in this section, corresponding to the signal indices.	21
3.4	Evaluated MSE for each random distribution as a function of compression ratio, average over 10 iterations.	22

4.1	Test 64×64 pixel 2D sinusoid patterns corresponding to vertical sinusoids, horizontal sinusoids, diagonal sinusoids, and egg tray pattern. All frequency components are 4 Hz.	25
4.2	<i>Relativity</i> by M.C. Escher, a complex image consisting of various sinusoidal patterns.	28
4.3	Reconstructed <i>Relativity</i> from 50% of samples from each patch. . . .	29
4.4	Extracted and reconstructed patches from <i>Relativity</i> using 40% of samples.	34
4.5	Simultaneous compression and encryption achieved with compressive sensing: original image (left), encrypted image (middle), and decrypted/reconstructed image (right).	35
4.6	Test image Lena (first) with the encrypted representation (second), the decryption result when the correct keys are used but x_{01} is perturbed by a value of 10^{-15} (third), and the decryption result when the correct keys are used but x_{02} is perturbed by a value of 10^{-15}	35
4.7	MSE curves resulting from evaluation of reconstruction error for tiny perturbations in the initial values Δx_{01} and Δx_{02}	35
4.8	SSIM curves resulting from evaluation of reconstruction error for tiny perturbations in the initial values Δx_{01} and Δx_{02}	36
5.1	330 Hz guitar signal representation in the time domain (left column) and frequency domain (right column).	43
5.2	Comparison of the performance of LASSO, OMP, and SL0.	44

5.3 Test speech signal in the time domain (top row) and modulation domain (bottom row).	44
5.4 PESQ and SNR _{seg} error space maps as a function of compression ratio and number of subbands.	45

List of Tables

4.1 Correlation coefficients of test Lena image.	32
--	----

List of changes

Deleted: The trend of both curiosity and profit-driven [...]	3
Replaced: ,	3
Deleted: —as a viable method for compression, encryp [...]	3
Deleted: Recent studies, however, have shown that [...]	4
Added: both asked and answered the questions that [...]	4
Deleted: CS can be viewed as a strategic undersampl [...]	4
Added: Linh-Trung et al.	5
Deleted: Sampling using a Gaussian-Logistic map was [...]	5
Replaced: initial value parameters	5
Deleted: of numbers	5
Added: In this study, a logistic map was used to [...]	5
Added: including those in this study,	5
Deleted: Audio signals, compared to images, have hi [...]	5
Replaced: greater than	5
Replaced: completely	5
Replaced: significant duration	6
Deleted: or recordings with relatively static frequencies	6
Deleted: i.e., the signal's spectrogram,	6
Deleted: , and percent overlap between adjacent subbands	6
Added: This method was adopted in this study, and [...]	6
Added: , for applications such as compression, encr [...]	6
Replaced: Contemporary	6
Replaced: signals	6
Deleted: as the target for CS	6

Replaced: and	6
Deleted: of most of the research in the field has been to	6
Added: on constructing effective sensing matrices,	6
Replaced: ing	6
Deleted: signal	6
Deleted: to compressively sample signals	6
Deleted: , namely, one-dimensional CS (1DCS) and [...]	7
Added: for image and audio signals	7
Deleted: It is shown that an N -dimensional signal [...]	7
Added: Furthermore, current research tend to evalu [...]	7
Added: Finally, this study is conducted in order to [...]	7
Added: Chapter 3 establishes basic workflows and [...]	8
Replaced: 4 & 5	8
Replaced: image-based CS and audio-based CS	8

Chapter 1

Introduction

1 The trend of both curiosity and profit-driven human development has caused a
2 surge in the amount of openly accessible raw data. More often than not, the data
3 is generated much faster than it can be processed into something interpretable
4 or useful. In the endeavor of keeping up with the inflow of information, there
5 are two major factors that significantly hinder our progress. First, Moore's law
6 implicitly sets a physical limit to the number of transistors that can be placed on
7 a chip, consequently limiting how powerful and how fast electronic systems can
8 become (barring a paradigm shift in the fundamental design of semiconductors).
9 The second is the Nyquist-Shannon sampling theorem (NST), which limits the
10 range of frequencies a recording device can successfully capture.

11 This study explores the use of compressive sensing (CS), — an emergent
12 sampling theorem that allows reconstruction of signals from much fewer samples
13 than required by the Nyquist-Shannon sampling theorem (NST) —as a viable
14 method for compression, encryption, and/or enhancement. In this framework, the
15 computational burden of encoding/decoding a signal is shifted from the sampling

¹⁶ device to the device performing reconstruction, decompression, or other modes
¹⁷ of post-processing. As such, there exist many ways to reconstruct a signal from
¹⁸ compressive measurements.

¹⁹ CS has found its applications in simple audio signals containing stable
²⁰ frequencies (such as pure tones [1, 2]) and dynamic frequencies (such as speech
²¹ [3–5]), images [6–8], and grayscale videos [9, 10]. The formulation of a sensing
²² matrix in CS requires a basis conforming to some uniform uncertainty principle,
²³ and most common starting points would be partial discrete cosine transform (DCT)
²⁴ matrices or partial discrete wavelet transform (DWT) matrices. ~~Recent studies,~~
²⁵ ~~however, have shown that learned bases perform much better on more complex~~
²⁶ ~~signals, i.e., those that would be typically encountered in real life situations. The~~
²⁷ ~~learning algorithms associated with the construction of these bases range from~~
²⁸ ~~classical iterative methods, which have long been used in optimization problems,~~
²⁹ ~~to the more contemporary machine learning methods.~~

³⁰ 1.1 Related literature

³¹ In 2004, Candès, Romberg, Tao [11], and Donoho [12] ~~both asked and answered~~
³² ~~the questions that birthed~~ the field which we now know as compressive sensing.
³³ The methods in CS apply concepts from time-frequency uncertainty principles [13]
³⁴ and sparse representations, which were studied rigorously by Donoho and Elad [14].
³⁵ ~~CS can be viewed as a strategic undersampling method: the signal is sampled at~~
³⁶ ~~random points in the real domain, and the ratio of the indices where it is sampled~~
³⁷ ~~to the size of the signal can be associated with some quasi-frequency which may~~
³⁸ ~~be lower than the Nyquist rate.~~

39 Linh-Trung et al. [15] demonstrated the use of deterministic chaos
40 filters to acquire samples instead of random distributions. Sampling using
41 a Gaussian Logistic map was applied to acoustic signals in [1]. Normally,
42 a deterministic chaotic function needs one or more initial value parameters
43 initialization values as a “seed”, and the sequence of numbers produced by different
44 combinations of initial values rapidly diverge from each other. This phenomenon
45 led to investigation of the use of CS as an encryption algorithm. Simultaneous
46 compression and encryption was achieved by [7], and it was found that the produced
47 sequences were sensitive to initial value perturbations on the order of 10^{-15} . Their
48 image compression-encryption model via CS was shown to have a key space on
49 the order of 10^{83} , making it extremely resistant to brute force and other types of
50 attacks. In this study, a logistic map was used to encode and construct the sensing
51 matrix. The encryption system exhibited similar key sensitivity and robustness
52 characteristics mentioned in the former. In the methods above, including those in
53 this study, sampling was performed in the signal domain (i.e., temporal domain
54 for audio, spatial domain for images), and the reconstruction was performed in the
55 frequency domain.

56 Audio signals, compared to images, have higher information density.
57 Whereas images are not naturally bandlimited and rather, are dependent on
58 the spatial resolution and bit depth of the imaging device, audio size scales
59 proportionally with time and takes on a wider range of values. The accepted
60 frequency range of human hearing is from 20 Hz to 20 kHz, so by the NST, a
61 sampling frequency greater than at least 40 kHz is needed to ensure that an audio
62 sample is recorded completely correctly. Any meaningful audio recording, especially
63 those containing speech, will certainly have a significant duration of a few

64 ~~seconds up to a few hours~~, so one cannot straightforwardly apply methodologies
65 used for images ~~or recordings with relatively static frequencies~~. The first challenge
66 this would pose for electronic systems is insufficient memory to process the entire
67 signal all at once. Low circumvented this problem [3, 4] by transforming the
68 signal to the modulation domain, ~~i.e., the signal's spectrogram~~, essentially raising
69 a one-dimensional signal to N -dimensions, where N is dependent on the desired
70 spectrogram resolution and number of subbands, ~~and percent overlap between~~
71 ~~adjacent subbands~~. This method was adopted in this study, and additionally, each
72 subband was multiplied with a window function to suppress potential boundary
73 artifacts when reconstructing. In earlier experiments, reconstruction would often
74 completely fail when windowing was not used. In the cases, however, that were
75 successful, the reconstruction exhibited severe boundary artifacts in the form of
76 distortion, aliasing, or noise.

77 1.2 Novelty

78 This study aims to provide a generalization for applying CS techniques to signals
79 of arbitrary dimensions, ~~for applications such as compression, encryption, and~~
80 ~~enhancement. Contemporary Previous~~ CS research work exclusively on either audio
81 or image ~~signals sequences as the target for CS~~, and—due to the computational
82 demands—and the focus ~~of most of the research in the field has been to~~ on
83 ~~constructing effective sensing matrices, optimizing~~ the computational complexity
84 for real-time applications, and improving ~~signal~~ reconstruction quality. In the
85 establishment of CS methods, two different general frameworks ~~to compressively~~
86 ~~sample signals~~ arise, namely, one-dimensional CS (1DCS) and two-dimensional

87 CS (2DCS) for image and audio signals. It is shown that an N -dimensional
88 signal can be decomposed into factors of one-dimensional and two-dimensional
89 signals, and can be processed using methods appropriate for each type of signal.
90 Furthermore, it is shown that N is bound not only by the type of signals being
91 worked with, but also the computational power of the decoding/decompressing
92 device. In particular, large values of N are useful in encryption, where a signal is
93 first raised to a high dimension in a certain basis, the sensing matrix is derived
94 from another high-dimensional basis, and the result is cast back to either one or
95 two dimensions to yield the encrypted message.

96 Furthermore, current research tend to evaluate signal reconstruction quality
97 using statistical metrics, such as mean-squared error (MSE) and its variants.
98 Arguably, the final interpreter of all signals are humans, and it is important to be
99 able to tell how well any compressive/reconstructive algorithm performs just by
100 looking at the metrics without directly observing the signal contents. In light of
101 this, the study also aims to evaluate the reconstruction quality of CS algorithms
102 using perceptually accurate metrics. This class of objective metrics are usually
103 built upon now-obsolete subjective scoring systems, and allows human observers
104 to make an informed estimate of the signal quality without directly accessing the
105 signal itself.

106 Finally, this study is conducted in order to lay out a unified, standardized
107 workflow for similar applications of CS on signals with arbitrary content. This
108 includes signals containing a combination of audio and images, such as color videos
109 and hyperspectral images.

¹¹⁰ **1.3 Thesis overview**

¹¹¹ The next chapter establishes the relevant mathematical concepts and notation
¹¹² to be used throughout this study, algorithms used in signal reconstruction, and
¹¹³ appropriate metrics per type of signal. Chapter 3 establishes basic workflows
¹¹⁴ and studies the effect of random sampling on CS reconstruction. Chapters 4 & 5
¹¹⁵ 3–5 respectively focus on image-based CS and audio-based CS~~two-dimensional~~
¹¹⁶ CS, one-dimensional CS, and N-dimensional CS. Each of these chapters are
¹¹⁷ self-contained methodologies, results, and discussions to emphasize that the
¹¹⁸ methods can work independently of each other. Conclusions of the study and
¹¹⁹ recommendations for future studies are presented in Chapter 6.

₁₂₀ **Chapter 2**

₁₂₁ **Preliminaries**

₁₂₂ The trend of both curiosity and profit-driven human development has caused a
₁₂₃ surge in the amount of openly accessible raw data. More often than not, the data
₁₂₄ is generated much faster than it can be processed into something interpretable
₁₂₅ or useful. In the endeavor of keeping up with the inflow of information, there
₁₂₆ are two major factors that significantly hinder our progress. First, Moore's law
₁₂₇ implicitly sets a physical limit to the number of transistors that can be placed on
₁₂₈ a chip, consequently limiting how powerful and how fast electronic systems can
₁₂₉ become (barring a paradigm shift in the fundamental design of semiconductors).
₁₃₀ The second is the Nyquist-Shannon sampling theorem (NST), which limits the
₁₃₁ range of frequencies a recording device can successfully capture. This states that
₁₃₂ given that you know a signal's highest frequency component f_B , sampling it at a
₁₃₃ rate f_S that is greater than twice this frequency is sufficient to capture all of the
₁₃₄ pertinent information regarding that signal: that is $f_S > 2f_B$, where f_B is known
₁₃₅ as the Nyquist frequency or the bandwidth; twice this value is the Nyquist rate
₁₃₆ [16]. For signals that are not naturally bandlimited, such as images, the ability

reproduce a signal is dependent on the device's resolution and still follows the same principle: there should be at more than twice the number of pixels codimensional with the image's highest spatial frequency. For practical day-to-day use, the NST will suffice. However, issues arise when bandwidth and storage are at a premium. Typically, after sensing a signal, not all of the raw data is stored. Rather, this data is converted to a compressed format by systematically discarding values such that the loss of information is virtually imperceptible. Thus, the process of acquiring massive amounts of data followed by compression is extremely wasteful. CS aims to directly acquire the parts of the signal that would otherwise survive this compression stage in the classical sampling scheme.

2.1 Sampling paradigms

Consider a signal $\mathbf{x} \in \mathbb{R}^n$; this notation indicates that \mathbf{x} is a vector of cardinality n , containing elements over the field of real numbers (\mathbf{x} can also easily be a complex vector, but for the purposes of this chapter, it is sufficient to emphasize that we are working with real-valued signals). The process of acquisition or sensing this signal can be modeled as a linear system, where the physical signal properties we wish to capture are transformed into digital values by applying a linear transformation

$$\mathbf{y} = \mathbf{Ax} \tag{2.1}$$

or in the literature of signal processing [17], by correlating them with a waveform basis

$$y_k = \langle \mathbf{x} | \mathbf{a}_k \rangle, \quad k \in \mathbb{N} \leq n \tag{2.2}$$

156 In conventional sampling, \mathbf{a}_k are Dirac basis vectors which turn \mathbf{y} into a
157 vector containing samples of \mathbf{x} in the temporal or spatial domain; if \mathbf{a}_k are Fourier
158 basis vectors (i.e., sinusoids), then \mathbf{y} is a vector of Fourier coefficients. If the signal
159 has been sampled sufficiently in the sense that the number of measurements m is
160 equal to the dimension n of the signal, then \mathbf{A} is a square matrix, and the original
161 signal \mathbf{x} can be reconstructed from the information vector \mathbf{y} by inversion of (2.1).
162 However, the process of recovering $\mathbf{x} \in \mathbb{R}^n$ from $\mathbf{y} \in \mathbb{R}^m$ becomes ill-posed when we
163 consider the undersampled case ($m \ll n$), as the sensing matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ —whose
164 row vectors are denoted as \mathbf{a}_m —causes the system to become underdetermined:
165 there exist infinitely many candidate solutions $\hat{\mathbf{x}}$ which satisfy (2.1). To add to
166 this, we also consider the possibility that the measurements are not perfect, and
167 are contaminated with noise. How then do we recover a signal from measurements
168 which are incomplete and most likely inaccurate? The answer lies in enforcing
169 constraints based on models of natural signals, as well as constraints based on
170 optimization techniques. CS can be viewed as a strategic undersampling method:
171 the signal is sampled at random points, and the ratio of the number of indices where
172 it is sampled to the size of the signal can be associated with some quasi-frequency
173 which may be lower than the Nyquist rate.

¹⁷⁴ **2.2 Sparsity**

¹⁷⁵ Most natural signals, especially those with some underlying periodicity, can be
¹⁷⁶ represented sparsely when expressed in the appropriate basis. This process of
¹⁷⁷ “sparsifying” can be expressed as

$$f = \langle \mathbf{x} | \psi(k) \rangle \quad (2.3)$$

¹⁷⁸ Similar to (2.2), this involves correlating the signal with the appropriate
¹⁷⁹ basis function to yield a representation in the sparse domain. Image information,
¹⁸⁰ for example, are commonly expressed in the DCT domain by

$$f_k = \sum_{n=0}^{N-1} x_n \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right], \quad 0 \leq k < N \quad (2.4)$$

¹⁸¹ and its corresponding inverse is

$$x_k = \frac{1}{2} f_0 + \sum_{n=1}^{N-1} f_n \cos \left[\frac{\pi}{N} \left(k + \frac{1}{2} \right) n \right], \quad 0 \leq k < N \quad (2.5)$$

¹⁸² where the cosine term corresponds to $\psi(k)$ in (2.3). We can express (2.4) more
¹⁸³ conveniently as $\mathbf{f} = \Psi \mathbf{x}$, where $\Psi \in \mathbb{R}^{n \times n}$ is the sparsifying matrix. Figure 2.1
¹⁸⁴ shows this sparsifying process in action: given a test image, taking its Daubechies-8
¹⁸⁵ discrete wavelet transform (D8 DWT) and zooming into a random subset shows
¹⁸⁶ that most of the signal energy is concentrated in just a few of the coefficients.
¹⁸⁷ All the other coefficients, when compared to the k highest coefficients, are
¹⁸⁸ practically zero; such a signal is referred to as k -sparse. The compressed image
¹⁸⁹ resulting from discarding all but the 25,000 highest coefficients and performing the
¹⁹⁰ inverse transform shows that any difference from the original image is virtually

imperceptible. A similar concept is used in JPEG compression, wherein an image is divided into 8×8 blocks, and in each block, a certain number of DCT coefficients are discarded depending on the desired quality factor Q [18].

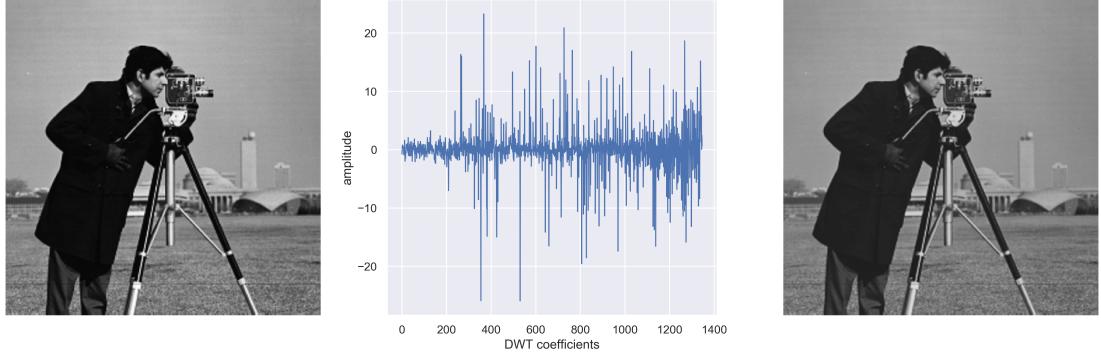


Figure 2.1: Original 512×512 , 8-bit image (left), and a random subset (for better visibility) of its D8 DWT coefficients (middle). Most of the signal energy is concentrated in just a few terms. By discarding all but the 25,000 highest coefficients and performing the inverse transform, the resulting image (right) is perceptually no different from the original.

2.3 Incoherence

Suppose we have two matrices Φ and Ψ which are involved in the sensing of a signal. As before, Ψ is the sparsifying matrix which converts the signal into a sparse representation, and Φ is the actual sensing matrix. The coherence between these two bases is expressed as

$$\mu(\Phi, \Psi) = \sqrt{n} \max_{0 \leq i, j < n} |\langle \varphi_i | \psi_j \rangle| \quad (2.6)$$

In other words, the coherence is the measure of the largest correlation between the column vectors of Φ and Ψ . In compressive sensing, we are interested in low-coherence basis pairs (i.e., basis pairs for which $\mu \rightarrow 1$). For example, in

202 the classical sampling scheme, Φ is the Dirac basis $\varphi_k(t) = \delta(t - k)$, and Ψ is the
203 DCT basis (2.4). This basis pair in particular is also called a time-frequency pair
204 and achieves maximum incoherence ($\mu = 1$) regardless of the number of dimensions
205 [13]. Additionally, any orthonormal basis Φ containing independent identically
206 distributed (i.i.d.) entries are also largely incoherent with a fixed basis Ψ [17].
207 The consequence of this is that CS performs most efficiently when sensing with
208 incoherent and random systems.

209 **2.4 Reconstruction strategies**

210 Another measure of the sparsity of a signal is its ℓ_0 norm, denoted $\|\mathbf{x}\|_0$, which
211 simply counts the number of non-zero coefficients of \mathbf{x} . As such, the goal of the
212 reconstruction stage in CS is to find the sparsest representation of the vector \mathbf{x} in
213 terms of the sensing matrix Φ by solving the combinatorial optimization problem

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{y} = \Phi \mathbf{x} \quad (2.7)$$

214 which, as the name implies, requires one to enumerate all possible k -element
215 combinations of the columns of Φ , and determining the smallest combination
216 which approximates the signal the closest. However, this process quickly becomes
217 intractable even for a modestly-sized signal. This requirement is therefore relaxed
218 by instead minimizing the ℓ_1 norm

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{y} = \Phi \mathbf{x} \quad (2.8)$$

219 where the ℓ_1 norm is defined as

$$\|\mathbf{x}\|_1 = \sum_{i=0}^{N-1} |x_i| \quad (2.9)$$

220 and is commonly called the taxicab or Manhattan distance. Most signals
221 encountered in practical situations, however, are not sparse but rather,
222 approximately sparse. As mentioned earlier, any signal measurement will inevitably
223 include some form of noise. Though ℓ_1 minimization can definitely still be used
224 (by casting it as a convex problem, as in the case of [19, 20]), other algorithms opt
225 for an ℓ_1 -regularized least squares approach as in the case of LASSO [21], whose
226 objective is

$$\min_{\mathbf{x}} \frac{1}{2m} \|\mathbf{y} - \Phi \mathbf{x}\|_2^2 + \alpha \|\mathbf{x}\|_1 \quad (2.10)$$

227 where $0 \leq \alpha \leq 1$ is the ℓ_1 regularization parameter. Greedy algorithms are also a
228 popular approach in this problem, the most common being the sparsity-constrained
229 orthogonal matching pursuit (OMP) [22], which has the objective

$$\min_{\mathbf{x}} \|\mathbf{y} - \Phi \mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq k \quad (2.11)$$

230 This method enforces the constraint that the reconstructed signal should be, at
231 most, k -sparse in the selected coding dictionary Φ . There exist a plethora of
232 algorithms dedicated to the decoding phase of CS. The ones mentioned above are
233 primarily used in this study.

²³⁴ Chapter 3

²³⁵ Random sampling-based ²³⁶ compressive sensing

²³⁷ In this chapter, I lay out the groundwork for performing basic compressive sensing
²³⁸ techniques which will be repeatedly used and built upon in the following chapters.
²³⁹ I also investigate various random properties that take place in the construction
²⁴⁰ of sensing matrices and their potential effect on the reconstruction quality. In
²⁴¹ order to quantify these properties, I focus primarily on one-dimensional sinusoids,
²⁴² better visualized as audio. In particular, these are signals containing a few known
²⁴³ frequency components that do not vary appreciably, if at all, through time.

²⁴⁴ 3.1 Test case: Sinusoid

²⁴⁵ For the signals of interest, I use the Fourier domain as the sparse representation. I
²⁴⁶ synthesized a C₅ piano note (523 Hz)—corresponding to a Nyquist rate of 1046
²⁴⁷ Hz—using Guitar Pro, with the standard sampling rate of 44.1 kHz and a duration
²⁴⁸ of 1 second. Due to the number of samples, I only worked with the first 1/8th

249 second, corresponding to 5512 samples. This will be the original signal; let's call
 250 this signal \mathbf{x} . I then compressively sampled this portion by taking 300 uniformly
 251 distributed random measurements, equivalent to a 5% compression ratio; this will
 252 be our compressed vector \mathbf{y} . Figure 3.1 visualizes how these measurements are
 253 distributed in time.

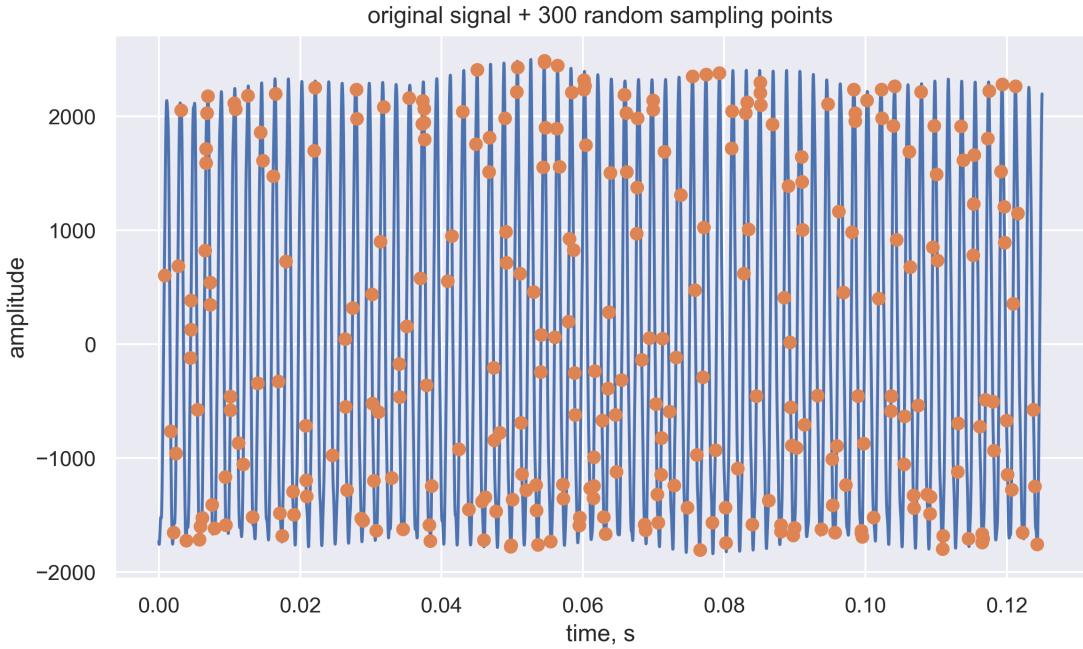


Figure 3.1: Original test signal (blue) and random measurements (orange).

254 In this model, sampling points consist of a discrete set of indices containing
 255 the signal amplitude corresponding to an instantaneous point in time. The random
 256 measurements actually point to these indices, which subset the chosen sparsifying
 257 basis: in this case, the DCT domain, for simplicity. The same random indices are
 258 used to select the rows of the DCT matrix of shape $n \times n$, where n is the signal
 259 dimension. I then stack these rows to form the sensing matrix \mathbf{A} ; essentially a
 260 partial DCT matrix of shape $m \times n$, where m is the number of random measurements.

261 Essentially, I am simulating a sensing device that not only purposely undersamples,
262 but also samples at random, intermittent points in time.

263 In Chapter 2, I discussed an overview of popular algorithms used in CS.
264 For the purposes of comparison in this chapter, I will be focusing on a gradient-
265 based method (LASSO), and a convex optimization-based method (CVXPY). The
266 optimization objectives for these two become

$$\text{LASSO} : \min_{\hat{\mathbf{x}}} \frac{1}{2m} \|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}\|_2^2 + \alpha \|\hat{\mathbf{x}}\|_1 \quad (3.1)$$

$$\text{CVXPY} : \min_{\hat{\mathbf{x}}} \|\hat{\mathbf{x}}\|_1 \quad \text{subject to} \quad \mathbf{A}\hat{\mathbf{x}} = \mathbf{y} \quad (3.2)$$

267 where $\hat{\mathbf{x}}$ is the candidate solution, and the optimum value of α was automatically
268 determined via 10-fold cross validation. A detailed implementation is shown
269 in Appendix A. Figure 3.2 shows a comparison of the original signal with the
270 reconstructions from the two algorithms in the time and frequency domains. In the
271 time domain, both algorithms appear to have been able to successfully reconstruct
272 the signal, though the CVXPY recovery shows many artifacts. The LASSO
273 algorithm yields a mean-squared error (MSE) of 0.002, while CVXPY yields an
274 MSE of 0.074. In the frequency domain, however, many frequencies are erroneously
275 being recovered by the LASSO method, and more so with the CVXPY method.
276 Because LASSO's α is a hyperparameter, it will need additional tuning to yield a
277 more optimal value.

3.2. EFFECT OF RANDOM DISTRIBUTION ON RECONSTRUCTION ERROR

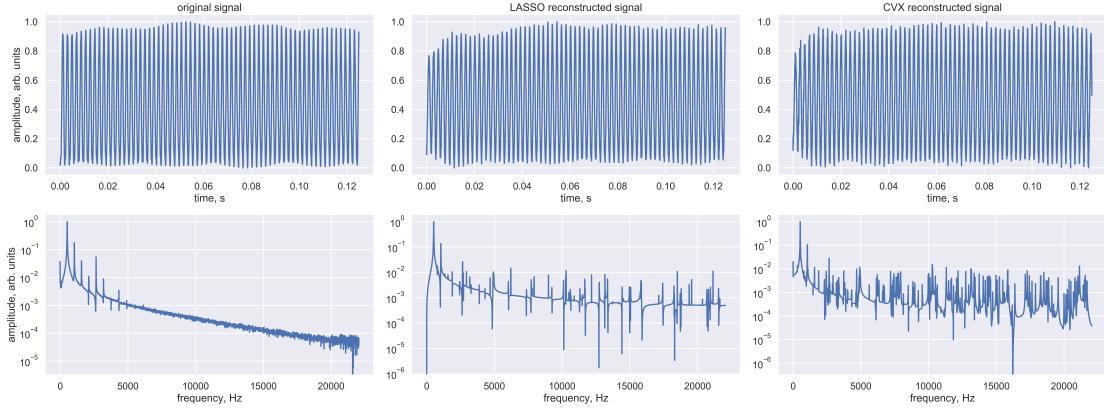


Figure 3.2: Original signal (leftmost), LASSO reconstructed signal (middle), and CVXPY reconstructed signal (rightmost). The top row shows the time domain representation, while the bottom row shows the frequency domain representation.

278 3.2 Effect of random distribution on reconstruction error

279 So far, I have worked solely with uniformly-distributed random sampling. Here,
 280 I investigate and compare the quality of reconstruction (in MSE) in terms of
 281 the random distribution. I will be working with two common distributions: the
 282 Gaussian and Poisson distributions, as well as the triangular distribution, which is
 283 commonly used in audio and image dithering. For each distribution, I generate
 284 i.i.d. random variables and use these to compressively sample the signal. I then
 285 evaluate the reconstruction MSE and take the average over 10 iterations to obtain
 286 error bars.

287 3.2.1 Uniform

288 The uniform distribution is given by

$$U(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b, \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

²⁸⁹ where a and b are the lower and upper bounds, respectively.

²⁹⁰ 3.2.2 Gaussian

²⁹¹ The Gaussian/normal distribution can be generated by

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \quad (3.4)$$

²⁹² where the parameters $\mu : \mu \in \mathbb{R}$ & $\sigma^2 : \sigma > 0$ are the distribution's mean and
²⁹³ variance, respectively.

²⁹⁴ 3.2.3 Poisson

²⁹⁵ The Poisson distribution can be generated by

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x!} \quad (3.5)$$

²⁹⁶ where the parameter $\lambda : \lambda > 0$ is the distribution's mean and variance.

²⁹⁷ 3.2.4 Triangular

²⁹⁸ The triangular distribution is generated by

$$T(x) = \begin{cases} 0 & x < a, \\ \frac{2(x-a)}{(b-a)(c-a)} & a \leq x < c, \\ \frac{2}{b-a} & x = c, \\ \frac{2(b-x)}{(b-a)(b-c)} & c < x \leq b, \\ 0 & x > b \end{cases} \quad (3.6)$$

299 where $a : a \in (-\infty, +\infty)$ is the lower bound, $b : b > a$ is the upper bound, and
300 $c : a \leq c \leq b$ is the mode.

301 **3.2.5 Results & discussion**

302 Due to the computational requirements, I will only work with the first $1/32$ seconds
303 of the signal, corresponding to 1378 samples. Figure 3.3 shows the probability
304 density for each distribution.

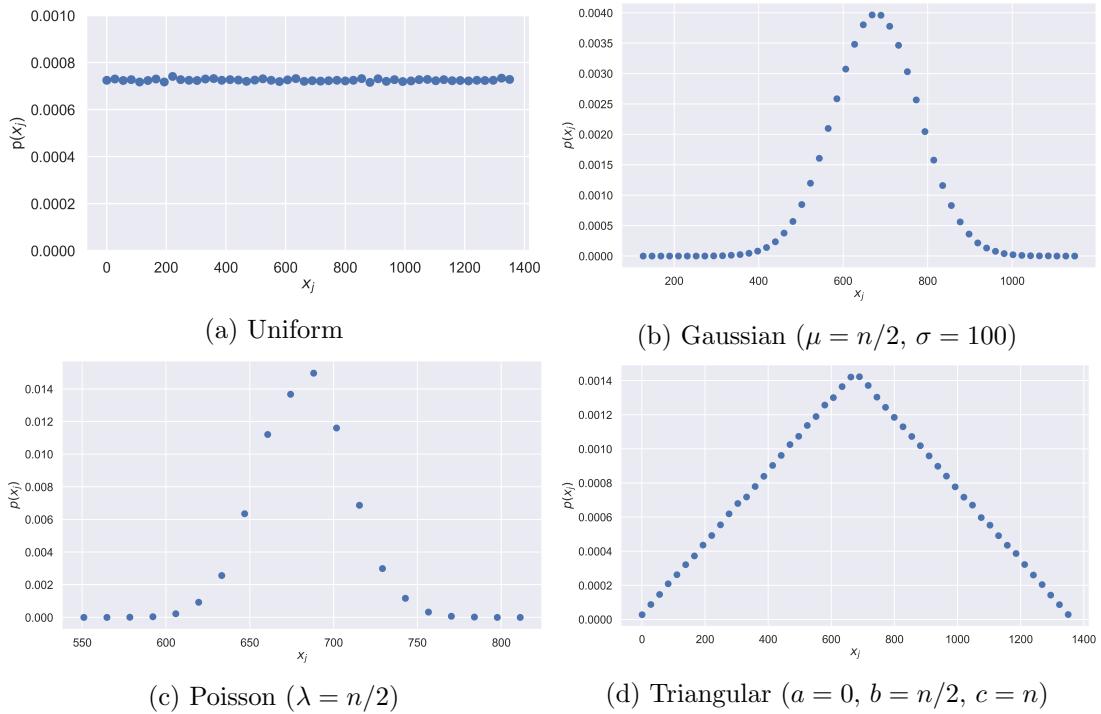


Figure 3.3: Probability densities of the different random distributions used in this section, corresponding to the signal indices.

305 Figure 3.4 shows the MSE evaluated for each random distribution as a
306 function of the fraction of total samples, more aptly referred to as the compression
307 ratio. From this, we can observe that the uniform and triangular distributions give
308 the lowest reconstruction error, but the latter has a more consistent performance

3.2. EFFECT OF RANDOM DISTRIBUTION ON RECONSTRUCTION ERROR

309 across a wide range of compression ratios. They are followed, in order, by the
310 Poisson and Gaussian distributions. One reason for the former's performance is
311 that they are able to completely span the signal with appreciable probability near
312 the bounds, while the latter's probability near the bounds are quickly approaching
313 zero.

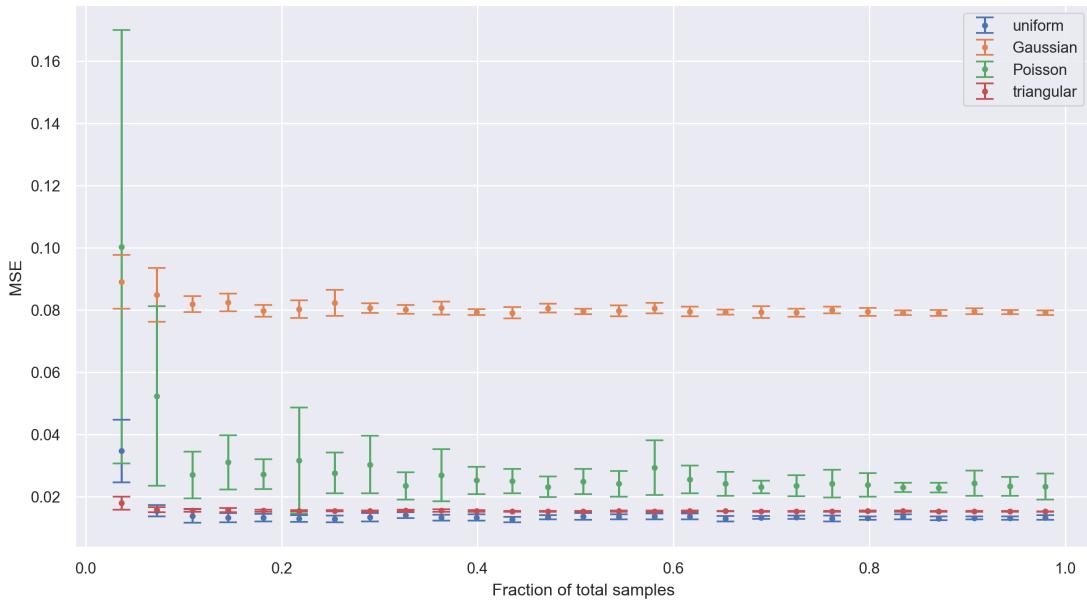


Figure 3.4: Evaluated MSE for each random distribution as a function of compression ratio, average over 10 iterations.

314 In line with these findings, I will be using uniformly-distributed random
315 variables throughout this study unless otherwise stated. In a later chapters, I will
316 be exploring more on recovering exact frequencies and their harmonics beyond the
317 Nyquist rate in real signals, as well as signals with multiple frequencies.

318 **Chapter 4**

319 **Image compressive sensing**

320 One of the more intuitive applications of CS lies in spatial signals as it is easier to
321 visualize. In this scheme, the process can be simplified either by flattening it to
322 one dimension and processing it in its entirety, or maintaining its dimensionality
323 and processing it by patches. The general workflow that arises from image CS is
324 as follows:

- 325 1. Define the compression ratio m/n , where n is the signal size, and m is the
326 desired size of the compressed signal.
- 327 2. Draw m random indices from the signal without replacement and store this
328 as a sample sequence ξ .
- 329 3. Extract the row vectors of the desired $n \times n$ sparsifying basis Ψ indexed by
330 ξ , and stack these to form the sensing matrix Φ (i.e., $\Phi = \Psi_\xi$)
- 331 4. With the desired reconstruction algorithm, perform the optimization (2.8) to
332 obtain the reconstructed signal \hat{x} .

333 In the case of high-definition images (whose shortest side is at least 720
334 pixels), it is usually more practical and yields better results if the image is processed
335 in patches.

336 4.1 Test case: Sinusoidal pattern

337 As mentioned in Chapter 2, the most commonly used sparse representation domain
338 for images is the Fourier domain, referred to in some fields as k -space. In this space,
339 signals are represented as a linear superposition of a finite number of sinusoidal
340 patterns. In Fig. 4.1a, 64×64 pixel sinusoidal patterns are generated, corresponding
341 to sine waves traveling horizontally, vertically, and diagonally, as well as an egg
342 tray pattern. In each case, all frequency components are 4 Hz. Figure 4.1b
343 visualizes the compressed image when a random sample of 5% is taken from the
344 signal. The actual compressed signal that is seen by the reconstruction algorithm
345 is a one-dimensional sequence containing only the information from the points
346 being sampled. Orthogonal matching pursuit (OMP) was used for reconstruction,
347 which is a greedy algorithm that finds the combination of basis vectors which best
348 represents the signal (similar to matching pursuit), but in addition, the residual at
349 each iteration is recomputed using an orthogonal projection on the set of previously
350 selected basis vectors [23]. Its objective function is

$$\arg \min_{\mathbf{x}} \|\mathbf{y} - \Phi \mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq \gamma \quad (4.1)$$

351 where γ is a hyperparameter which controls the maximum allowable number of
352 non-zero coefficients. The **Scikit-learn** implementation sets this value to 10%
353 of the number of samples by default [21]. Evaluation of the mean-squared error

354 (MSE) for the pure horizontal and pure vertical sine waves, as well as the egg tray
 355 pattern yields a value that is practically negligible ($\approx 10^{-31}$); the reconstruction is
 356 exact. On the other hand, the reconstructed diagonal sine wave yields an MSE of
 357 10^{-3} —still quite small, but mild distortion can be observed at the image boundaries.
 358 This is due to the fact that the information at hand is finite, and so is the window
 359 size which, in this case, is the same size as the signal itself.

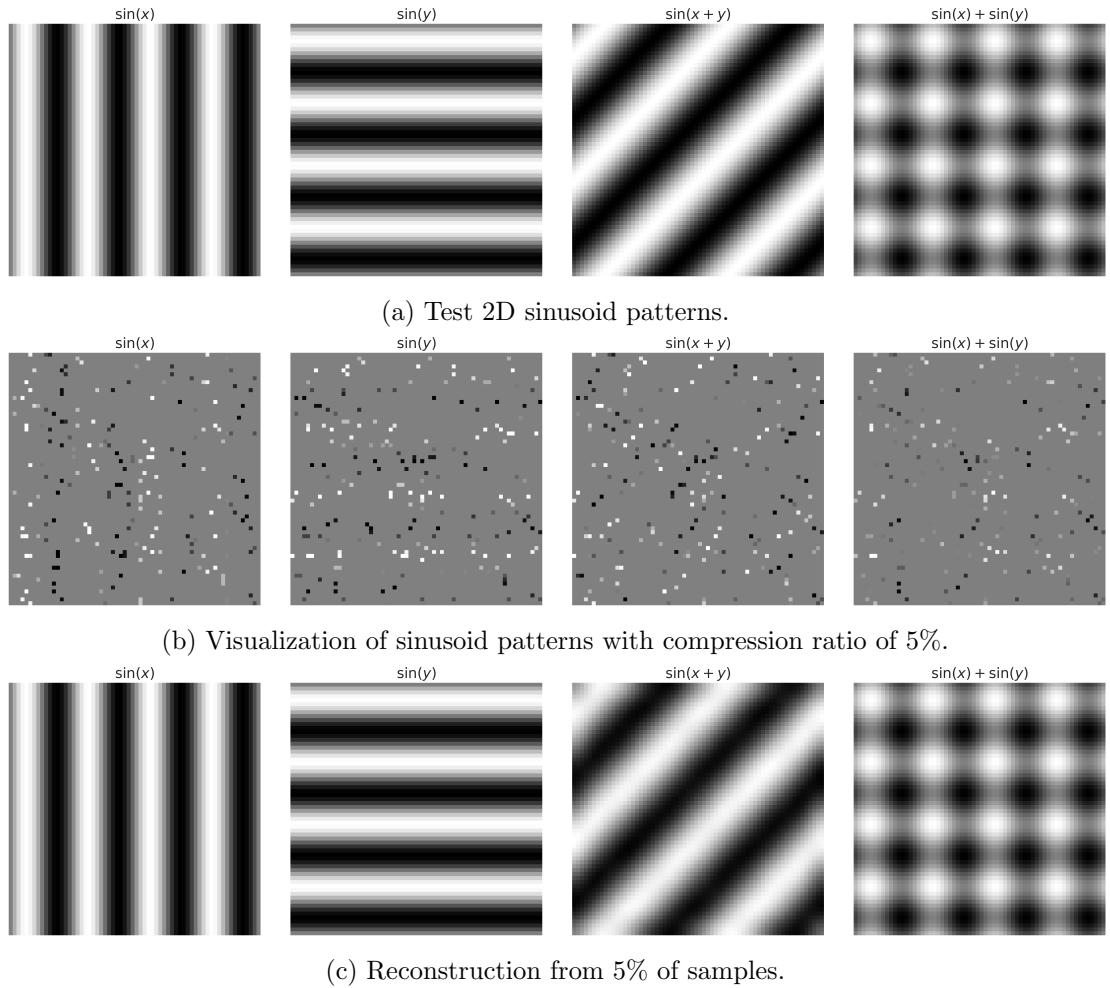


Figure 4.1: Test 64×64 pixel 2D sinusoid patterns corresponding to vertical sinusoids, horizontal sinusoids, diagonal sinusoids, and egg tray pattern. All frequency components are 4 Hz.

360 **4.2 Image with multiple sinusoids**

361 **4.2.1 Pre-processing**

362 For this section, the image used is M.C. Escher's *Relativity*, an example of a
363 more complex image but consisting of dominant sinusoidal patterns that are made
364 apparent when you zoom in. The original image has dimensions of 1600×981
365 pixels, for a total of 1,569,600 pixels. Following the procedure with the previous
366 section, this would require the construction of a $1,569,600 \times 1,569,600$ sparsifying
367 matrix containing $\approx 2 \times 10^{12}$ entries. Assuming that the matrix would be stored as
368 32-bit floating point numbers, this process alone would take up ≈ 8 GB of memory,
369 and it would be highly impractical to process similarly-sized images as a whole.
370 The workaround is to split it into smaller, manageable patches.

371 **4.2.2 Processing**

372 For this image in particular, it was first resized to 1600×976 pixels so that
373 it could be equally divided into a grid of 16×16 , each with a dimension of
374 100×61 pixels. After compressively sampling each patch at 40% compression ratio,
375 reconstruction was performed using the Embedded Conic Solver (ECOS) of the
376 Convex Optimization Python library (CVXPY), which recasts (2.8) as a convex
377 problem and directly minimizes the ℓ_1 norm [19, 20, 24] and thus, is significantly
378 slower compared to OMP.

379 **4.2.3 Reconstruction evaluation**

380 To quantify the reconstruction quality, the Structural Similarity Index (SSIM) [25]
381 was used. This is a perception-based model that takes into account perceptual

382 factors such as luminance, contrast, and structure. SSIM is calculated on windows
383 in the image, and is defined as

$$\text{SSIM}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{(2\mu_{\mathbf{x}}\mu_{\hat{\mathbf{x}}} + c_1)(2\sigma_{\mathbf{x}\hat{\mathbf{x}}} + c_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\hat{\mathbf{x}}}^2 + c_1)(\sigma_{\mathbf{x}}^2 + \sigma_{\hat{\mathbf{x}}}^2 + c_2)} \quad (4.2)$$

384 where $\mathbf{x}, \hat{\mathbf{x}}$ are the original and reconstructed signals, respectively, μ are the image
385 means, σ are the image standard deviations, and c are constants to stabilize division
386 with a small denominator. SSIM values of 0.8 and above are considered acceptable.

387 After stitching all patches at the end, the reconstructed image is shown
388 in Fig. 4.3. Evaluation of SSIM yields a value of 0.88, way above the acceptable
389 threshold. Selected patches with the aforementioned dominant patterns are
390 shown with their reconstructed counterparts in Fig. 4.4, corresponding to patches
391 dominated by horizontal sinusoids, vertical sinusoids, diagonal sinusoids, multiple
392 sinusoids, and patches with no dominant pattern.

393 We can observe that at this compression rate, the patches with a single
394 apparent sinusoidal pattern (Figs. 4.4a-4.4c) are successfully recovered, with some
395 noise present especially for the patch with a dominant diagonal pattern (similar
396 to the previous section). The patch with multiple sinusoid patterns (Fig. 4.4d),
397 although still recognizable, is laden with a lot of noise. Lastly, the patch with no
398 apparent pattern (Fig. 4.4e) is barely recognizable, except for the portions where a
399 dominant sinusoidal pattern is partially present in the frame.

400 From this, the following information can be gleaned. First, reconstruction
401 performs better on smaller patches, and when the patch in question contains as
402 few frequency components as possible (such is the case with the patches with only
403 one dominant pattern). Second, the patch with no dominant pattern—upon closer

404 visual inspection—can be classified as being successfully recovered; however, the
405 reconstruction noise is almost at the same level as the signal itself, which makes
406 them indistinguishable. This can be attributed to the fact that the patches with no
407 apparent dominant pattern are actually composed of a superposition of sinusoids
408 residing primarily in the high-frequency region of k -space. Since the sampling
409 points are uniformly distributed throughout the spatial domain, so are they in
410 the frequency domain. Thus, the information in the high-frequency region is not
411 sufficiently captured, and a higher compression ratio is required to be able to better
412 recover these high-frequency regions. Another solution would be, as mentioned
413 earlier, to make the patches smaller so that lesser frequencies are captured in one
414 patch.

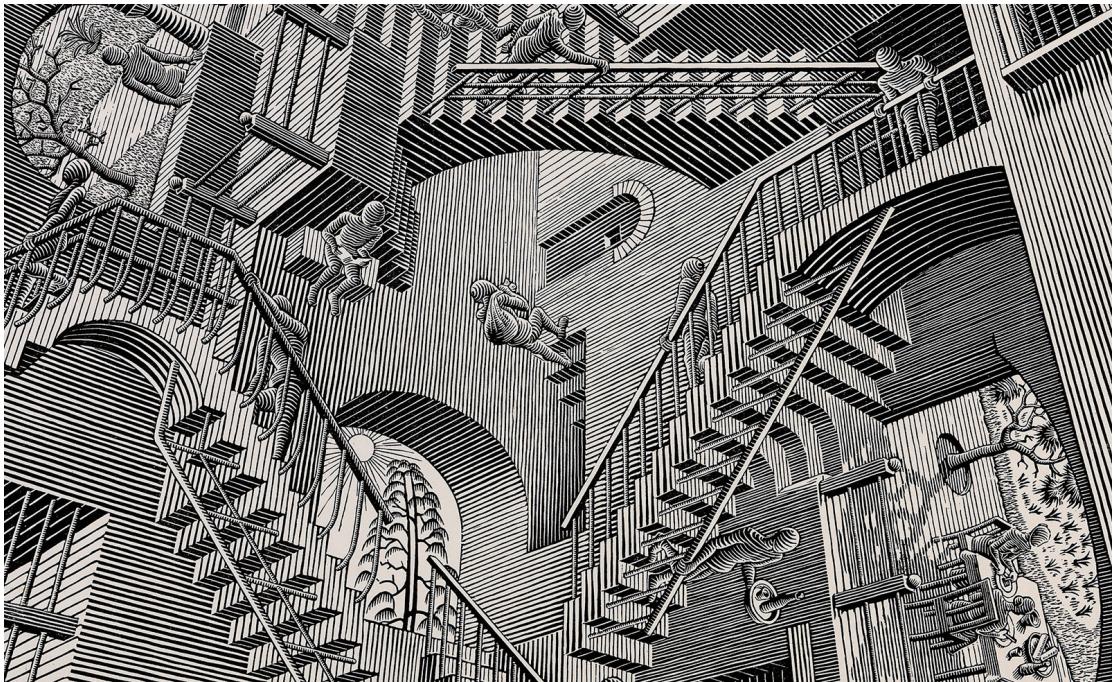


Figure 4.2: *Relativity* by M.C. Escher, a complex image consisting of various sinusoidal patterns.

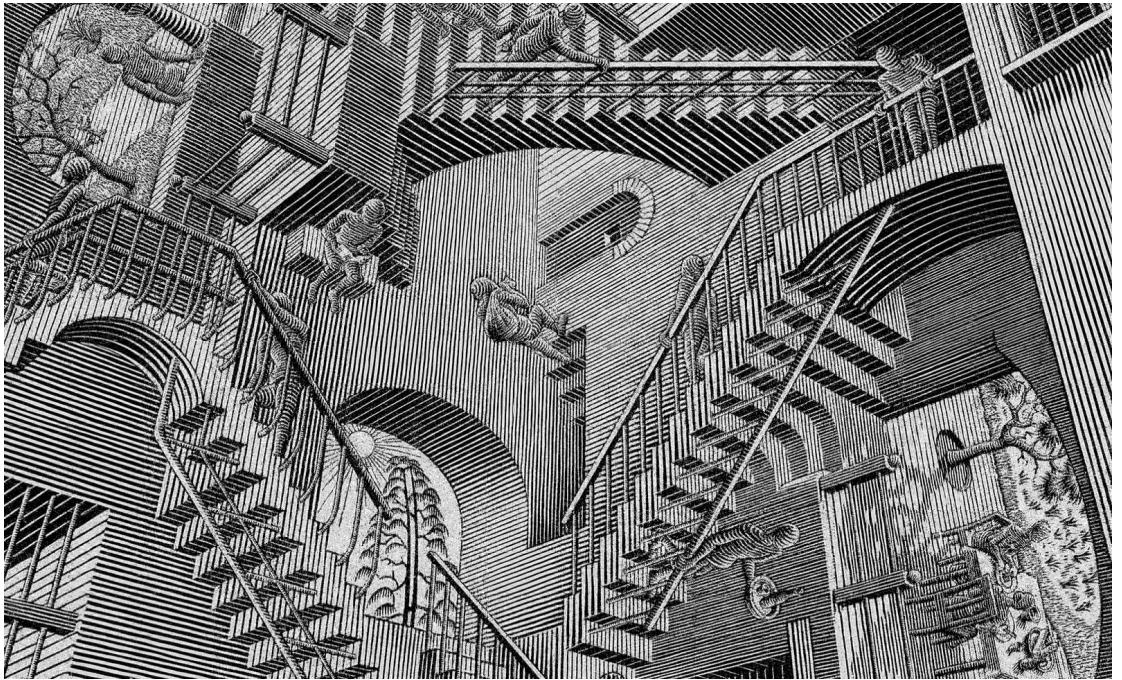


Figure 4.3: Reconstructed *Relativity* from 50% of samples from each patch.

415 4.3 Simultaneous compression & encryption

416 Because of the way compressively sampled images are coded with the sensing
417 matrix, the use of CS as an encryption algorithm arises naturally. Consider the
418 logistic map

$$x_{n+1} = rx_n(1 - x_n) \quad (4.3)$$

419 which is often used as an archetypal example of deterministic chaotic behavior for
420 values of $r \in [3.57, 4)$. In this regime, the sequences produced by varying the initial
421 parameter x_0 rapidly diverge from each other. Thus, this can become an encryption
422 system by treating the parameters r and x_0 as an encryption key set, and (4.3) as
423 the hash function. One key set is then needed for each signal dimension. Thus,

424 two key sets are needed for grayscale image applications (six, in the case of an
425 RGB image and a unique key set for each color channel).

426 **4.3.1 Methodology**

427 The construction of the sensing matrix Φ differs from the general workflow, and is
428 as follows:

- 429 1. From (4.3), generate a sequence of length $2m$ with the initial key pair r_1 and
430 x_{01} . Discard the first n elements to avoid the transient response and store
431 the latter n elements as a sequence s .
- 432 2. Explicitly generate the index sequence of s and store it as sequence
433 $p = [0, 1, \dots, m - 1]$.
- 434 3. Sort p according to ascending values of s .
- 435 4. Generate the first sensing matrix Φ_1 by extracting and stacking rows of a
436 Hadamard matrix H of order N indexed by the first m elements of p , i.e.,

$$\Phi_1 = \begin{bmatrix} H_{p_1} & H_{p_2} & \cdots & H_{p_m} \end{bmatrix}^\top \quad (4.4)$$

437 where H_{p_i} denotes the p_i th row vector of H .

- 438 5. Any other sensing matrices can be constructed in the same way.

439 The usage of a Hadamard matrix implies that the image must first be
440 reshaped to have dimensions that are integer multiples of 4. The original image
441 is first reshaped to 256×256 pixels, and is sparsified by transforming it to the
442 DCT domain. The desired compressed dimension is set to $m = 192$, corresponding

⁴⁴³ to a compression ratio of 75%, and the keys are set to values of $r_1 = r_2 = 3.99$,
⁴⁴⁴ $x_{01} = 0.11$, and $x_{02} = 0.24$.

⁴⁴⁵ 4.3.2 Results & discussion

⁴⁴⁶ Figure 4.5 shows the application of this to the Lena test image. Visual inspection
⁴⁴⁷ of the encrypted representation shows horizontal and vertical bands distributed
⁴⁴⁸ throughout the representation space, and is indicative that a simple inverse Fourier
⁴⁴⁹ transform will not recover any meaningful information. Assuming that the receiver
⁴⁵⁰ knows the encryption scheme, recovery of the original message is successful if the
⁴⁵¹ same keys r_1, r_2, x_{01}, x_{02} as the encryption stage are used, which will allow the
⁴⁵² receiver to construct the exact same sensing matrices Φ_1, Φ_2 and perform the
⁴⁵³ inverse operation on the encrypted message. In the decrypted image, encryption
⁴⁵⁴ artifacts can be observed, as indicated by some visible banding, but is nonetheless
⁴⁵⁵ recognizable; evaluation of the MSE and SSIM yields values of 0.02 and 0.82,
⁴⁵⁶ respectively.

⁴⁵⁷ 4.3.3 Correlation analysis

⁴⁵⁸ Correlation is a statistical measure of obtaining useful information from a given
⁴⁵⁹ signal by analyzing adjacent pixels. The correlation coefficients can be obtained by

$$C = \frac{\sum_{i=1}^N (x_i - \langle \mathbf{x} \rangle)(y_i - \langle \mathbf{y} \rangle)}{\sqrt{\sum_{i=1}^N (x_i - \langle \mathbf{x} \rangle)^2 \sum_{i=1}^N (y_i - \langle \mathbf{y} \rangle)^2}} \quad (4.5)$$

⁴⁶⁰ where \mathbf{x} and \mathbf{y} are the original and encrypted signals, respectively. For the purposes
⁴⁶¹ of encryption, a correlation coefficient closer to 0 is better. Table 4.1 shows the
⁴⁶² computed correlations for the original and encrypted Lena images in the horizontal,

Table 4.1: Correlation coefficients of test Lena image.

	horizontal	vertical	diagonal
original	0.94	0.97	0.91
encrypted	0.42	0.02	-0.05

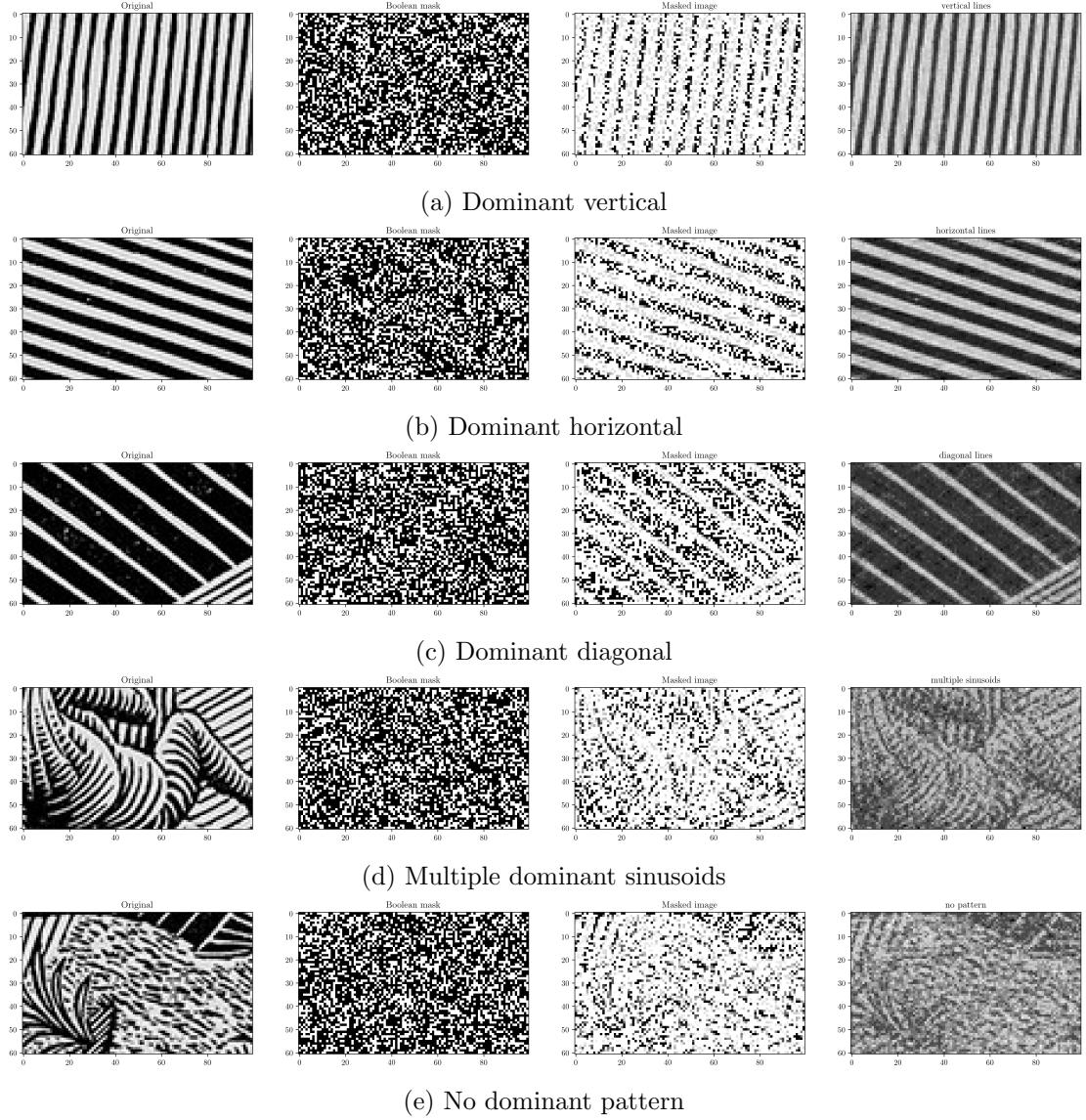
463 vertical, and diagonal directions. The correlation of the encrypted signal is much
 464 lower than that of the original, which means that a potential attacker cannot gain
 465 any useful information by employing statistical analyses.

466 4.3.4 Key sensitivity analysis

467 With the knowledge that the hash function (4.3) is chaotic, the sensitivity of the
 468 encryption system to the keys can be tested by perturbing the initial keys with
 469 small values. Figure 4.6 shows the decryption results when all the correct keys are
 470 used, except for x_{01} , which is perturbed by a tiny value $\approx 10^{-15}$ (third image), and
 471 similarly when the correct x_{01} is used but x_{02} is perturbed by the same amount
 472 (last image).

473 Additionally, Fig. 4.7 shows the MSE curves evaluated by varying values of
 474 the perturbation Δx_{01} and Δx_{02} . Perturbations are generated as 100 equally-spaced
 475 values in the range $\pm 10^{-14}$, plus zero. This specific value was chosen because
 476 the resolution of a double precision floating point number in Python on a 64-bit
 477 CPU architecture is 10^{-16} . The MSE generally oscillates at some high value, and
 478 exhibits a sharp dip when the MSE is evaluated for the correct keys ($\Delta x_0 = 0$).
 479 The same is done for Fig. 4.8, which shows the SSIM curves. Similarly, values
 480 oscillate at a low value, corresponding to recovered signals with no meaningful

₄₈₁ data. Maximum SSIM of 0.83 is achieved only for the correct keys. This shows
₄₈₂ that brute force attacks are intractable against this kind of encryption system.


 Figure 4.4: Extracted and reconstructed patches from *Relativity* using 40% of samples.

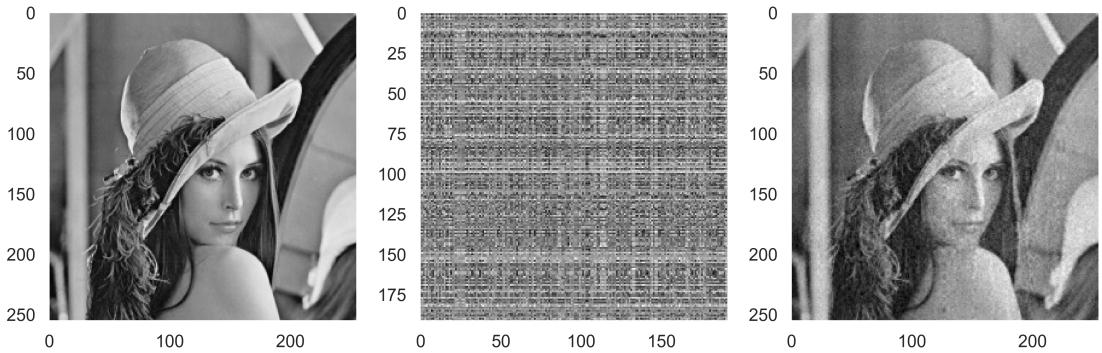


Figure 4.5: Simultaneous compression and encryption achieved with compressive sensing: original image (left), encrypted image (middle), and decrypted/reconstructed image (right).

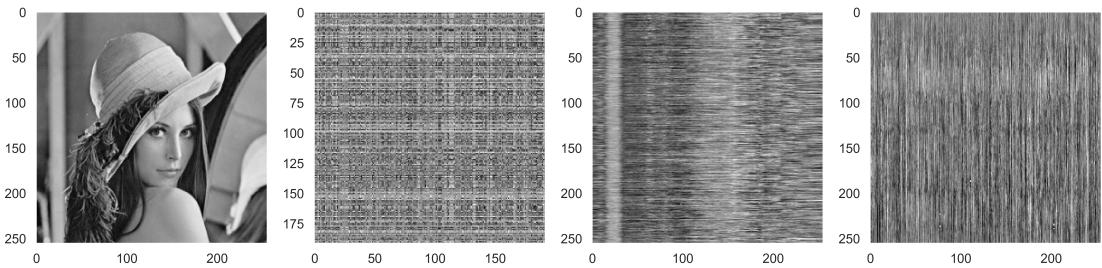


Figure 4.6: Test image Lena (first) with the encrypted representation (second), the decryption result when the correct keys are used but x_{01} is perturbed by a value of 10^{-15} (third), and the decryption result when the correct keys are used but x_{02} is perturbed by a value of 10^{-15} .

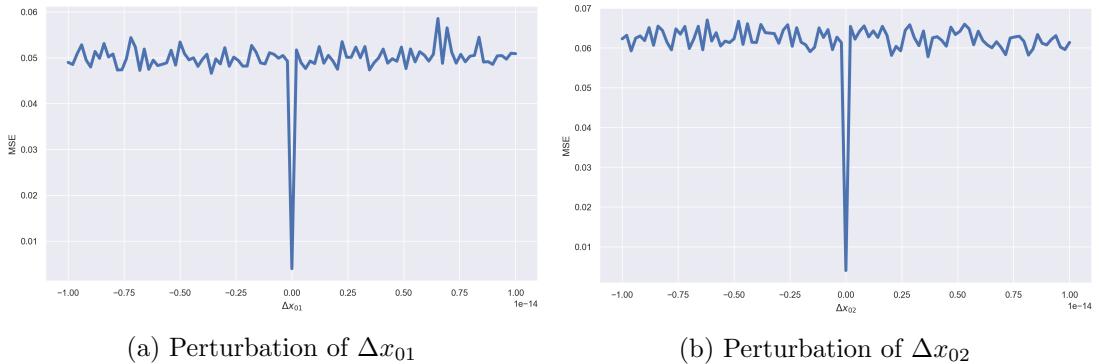


Figure 4.7: MSE curves resulting from evaluation of reconstruction error for tiny perturbations in the initial values Δx_{01} and Δx_{02} .

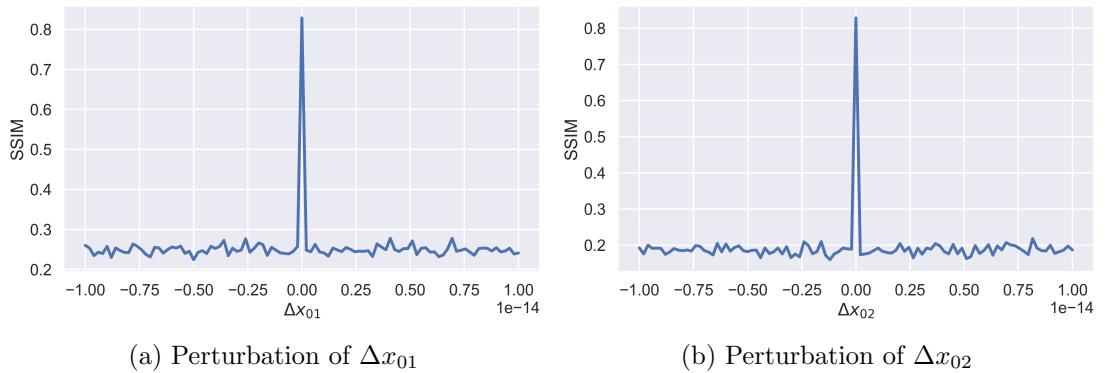


Figure 4.8: SSIM curves resulting from evaluation of reconstruction error for tiny perturbations in the initial values Δx_{01} and Δx_{02} .

483 **Chapter 5**

484 **Audio compressive sensing**

485 In this chapter, I apply CS to audio signals. These type of signals act as the
486 bridge to N -dimensional CS as they are one-dimensional when represented in the
487 time domain, but are projected to higher dimensions when represented in another
488 domain, such as the spectrogram/modulation domain. Unlike images, audio signals
489 are a tad harder to compressively sample. Due to their relatively higher information
490 density, the effects of undersampling are easily observed.

491 **5.1 Test case: Sinusoid redux**

492 In this test case, I recorded a guitar playing a single E₄ (330 Hz) note at the
493 standard 44.1 kHz sampling rate for 4 seconds. Since the Nyquist rate of the
494 actual signal is 660 Hz, the recording can be downsampled to a practical 8 kHz
495 for processing. The signal waveform and frequency content is shown in Fig. 5.1a.
496 The base frequency is dominant in the frequency spectrum, and several harmonics
497 can be observed. The goal here is to be able to recover the harmonics that have a
498 frequency higher than the compressive sampling rate.

499 The compressed signal is shown in Fig. 5.1b, which was compressively
 500 sampled with a quasi-frequency of 1000 Hz (1000 i.i.d. random samples per second),
 501 corresponding to a 12.5% compression ratio. The waveform envelope still resembles
 502 that of the original, but due to the random nature of sampling, the periodicity is
 503 not preserved, and is reflected in the seemingly random frequency content.

504 Following a similar process shown in Chapter 3, I chose DCT to be the
 505 sparse representation domain, and LASSO as the optimization algorithm. The
 506 reconstructed signal is shown in Fig. 5.1c. For this case, I am concerned only with
 507 the frequency components that are recovered, and not so much with the magnitude.
 508 Thus, the reconstruction quality can be quantified using the cosine similarity

$$\text{similarity} = \cos \theta = \frac{\mathbf{x} \cdot \hat{\mathbf{x}}}{\|\mathbf{x}\|_2 \|\hat{\mathbf{x}}\|_2} \quad (5.1)$$

509 which allows us to compare two signals' frequency content directly in the time
 510 domain. A cosine similarity value of 0.8 and above indicates acceptable quality; a
 511 value of 1.0 indicates perfect reconstruction.

512 5.2 Comparison of algorithms

513 Following the same procedure as the previous section, my aim now is to compare
 514 the performance of three different reconstruction algorithms in terms of average
 515 runtime and reconstruction quality. The algorithms used are LASSO and OMP,
 516 which were described in Chapter 2. Additionally, the Smoothed L₀ Norm (SL0)
 517 [26] is used, which approximates the ℓ_0 norm using a Gaussian of the form

$$\lim_{\sigma \rightarrow 0} x \exp \left(-\frac{x^2}{2\sigma^2} \right) \quad (5.2)$$

518 While all algorithms have polynomial time complexity [27–29], OMP shows
519 the worst scaling with respect to time; LASSO and SL0 show similar performance
520 over time (Fig. 5.2a). In terms of reconstruction quality (cosine similarity), LASSO
521 is able to breach the 0.8 threshold at 30% compression ratio, while SL0 achieves
522 this at 50% compression. On the other hand, OMP shows a nonlinear trend with a
523 large error, which is indicative of unstable performance for low compression ratios
524 (Fig. 5.2b).

525 **5.3 Speech**

526 In order to show its practical merits, we will inevitably have to deal with increasingly
527 large and complex signals. Audio recordings containing speech will encompass
528 a wide range of frequencies, so such signals can only be downsampled so much
529 before essential information is lost to aliasing. Unlike images, large audio signals
530 cannot simply be chopped into smaller, manageable pieces. The effects of aliasing
531 are amplified due to the high information density, and CS' violation of periodic
532 constraints introduce artifacts in the vicinity of where the signal was sliced. This
533 is the motivation for transforming the signal first into the modulation domain
534 (spectrogram).

535 **5.3.1 Sparse transformation**

536 In obtaining the spectrogram representation, first define a short length sampling
537 window, typically only a few milliseconds in duration, as well as the overlap between
538 adjacent frames. The latter is crucial in suppressing boundary artifacts as it ensures
539 that some information from the current measurement is carried over to the next

540 measurement. The signal is then divided into frames by sliding this window across
 541 the entire signal. Each frame is multiplied with a window function; in this case, I
 542 used the Hann window, defined as

$$w[n] = \sin^2\left(\frac{\pi n}{N}\right) \quad (5.3)$$

543 where $N + 1$ is the length of the window, and $n : 0 \leq n \leq N$ is the frame index.
 544 Finally, each frame undergoes a Fourier transformation. The entire process is also
 545 called a short-time Fourier transform, and is summarized as

$$X(\omega, p) = \sum_{p=0}^{P-1} x[p]w[p - kR]e^{-i\omega p} \quad (5.4)$$

546 where $x[p]$ is the p th signal frame, $w[p - kR]$ is the window function with hop size
 547 R and time index k , and ω is the angular frequency.

548 5.3.2 Pre-processing

549 Test signals were obtained from the TIMIT Acoustic-Phonetic Continuous Speech
 550 Corpus [30], which contains speech recordings in **WAV** format. The recordings are
 551 of English speakers grouped by region, sex, and unique spoken sentence. All files
 552 have a sampling rate of 16 kHz and are, on average, 3 seconds long. I chose a test
 553 signal at random, specifically the **DR8/MJLN0/SA1.wav** file. This indicates that the
 554 speaker was from dialect region 8 (nomadic), was male with speaker code **JLN0**,
 555 and spoke unique sentence **SA1**, which reads

556 She had your dark suit in greasy wash water all year.

557 Before proceeding, I downsampled the file to 8 kHz. The representation of
 558 the signal in the time and modulation domains are shown in Fig. 5.3a.

559 **5.3.3 Processing**

560 I compressively sampled the signal with a compression ratio of 40%, using
 561 1024 frames and 75% frame overlap. Following the results from Sec. 5.2, I
 562 used the LASSO algorithm for reconstruction, once again obtaining the optimal
 563 regularization parameter α by 5-fold cross validation.

564 **5.3.4 Reconstruction evaluation**

565 The reconstruction quality was quantified using the International Telecommu-
 566 nication Standardization Sector (ITU-T) recommendation P.862 [31], otherwise
 567 known as the Perceptual Evaluation of Speech Quality (PESQ). This metric is a
 568 full-reference, perceptually intuitive scoring system which models the now-obsolete
 569 mean opinion scores (MOS). This algorithm performs a series of standardized
 570 tests modeled after qualitative metrics, analyzes and compares the original and
 571 reconstructed signals, and returns a value from 1.0 (bad) to 5.0 (perfect). Because
 572 real reconstructed signals are rarely exactly the same as the original, the PESQ
 573 values are usually thresholded up to 4.5 (excellent). PESQ values of 3.0 and above
 574 indicate acceptable quality.

575 For a more quantitative test, I also used the average segmental signal-to-
 576 noise ratio (SNR_{seg}) [32], defined as

$$\text{SNR}_{\text{seg}} = \frac{10}{B} \sum_{b=0}^{B-1} \log_{10} \frac{\sum_{i=Nb}^{Nb+N-1} x_i^2}{\sum_{i=Nb}^{Nb+N-1} (x_i - \hat{x}_i)^2} \quad (5.5)$$

577 where N is the frame length, B is the number of frames, x_i are the original signal
578 samples, and \hat{x}_i are the reconstructed signal samples.

579 Figure 5.3b shows the reconstructed signal. Qualitative comparison in the
580 time domain shows that the original and reconstructed waveforms are structurally
581 similar. In the modulation domain, the dynamic range of the latter seems to have
582 diminished, but the dominant frequencies can still be observed. Evaluation of the
583 PESQ and SNR_{seg} yields values of 2.50 and 0.07, respectively. At face value, I can
584 immediately tell from the PESQ that the reconstructed signal quality is slightly
585 below average; listening to the reconstructed recording reveals a noticeable level of
586 noise in the background. However, the same distinction cannot be made for the
587 SNR_{seg} since its bounds are not well-defined.

588 **5.3.5 Error space analysis**

589 Using the same test signal, I generated the error space maps by compressively
590 sampling the signal and evaluating the metrics for varying compression ratios \in
591 $[0.1, 0.9]$ in increments of 0.1, and varying number of frames $\in \{128, 256, 512, 1024\}$,
592 while keeping the frame overlap constant at 75%. Figure 5.4 shows the PESQ
593 and SNR_{seg} maps. The former exhibits a sensitivity to the compression ratio, and
594 achieves the acceptable threshold of 3.0 at around 60% compression. The latter
595 shows sensitivity towards the number of frames (as it is an *average* metric) with
596 some additional degradation below 40% compression ratio. It achieves a maximum
597 value of 0.08 at around 1024 frames.

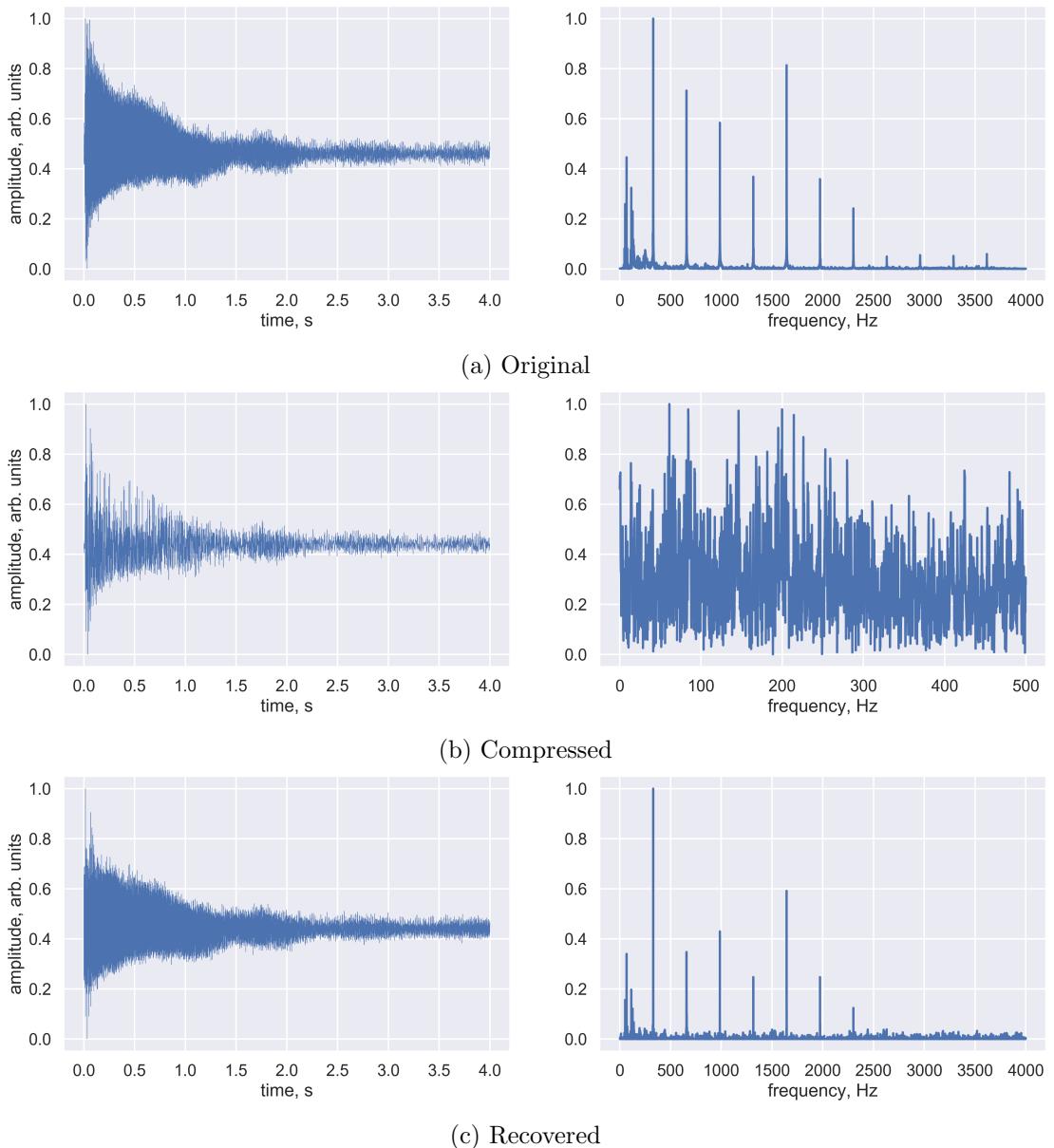


Figure 5.1: 330 Hz guitar signal representation in the time domain (left column) and frequency domain (right column).

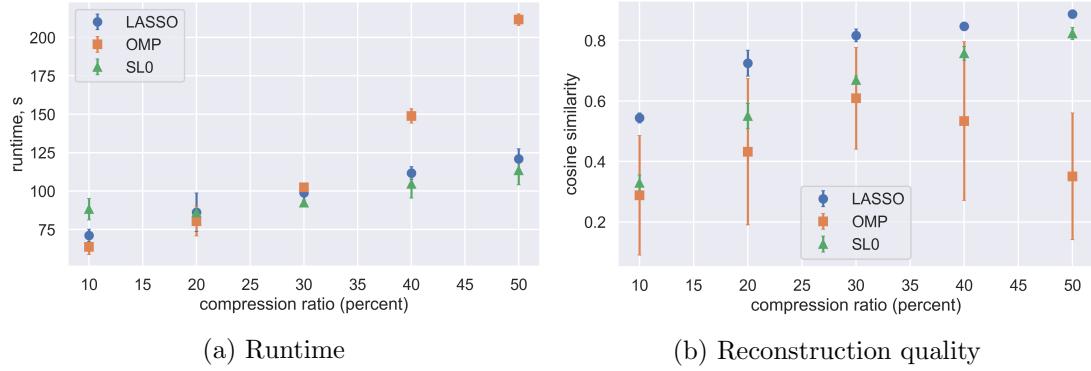


Figure 5.2: Comparison of the performance of LASSO, OMP, and SL0.

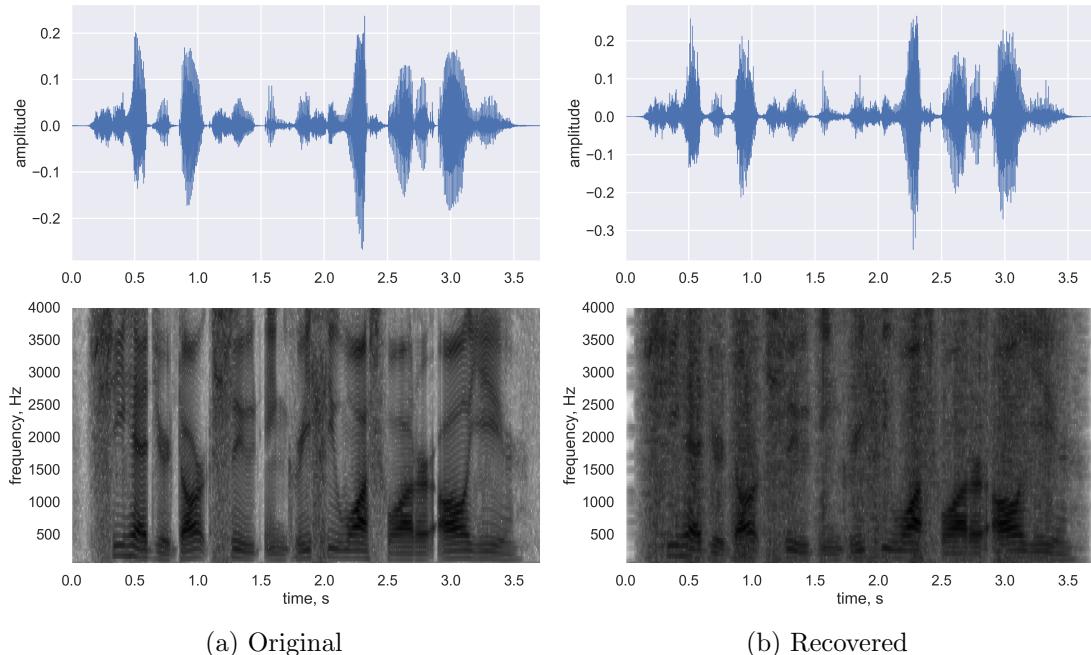


Figure 5.3: Test speech signal in the time domain (top row) and modulation domain (bottom row).

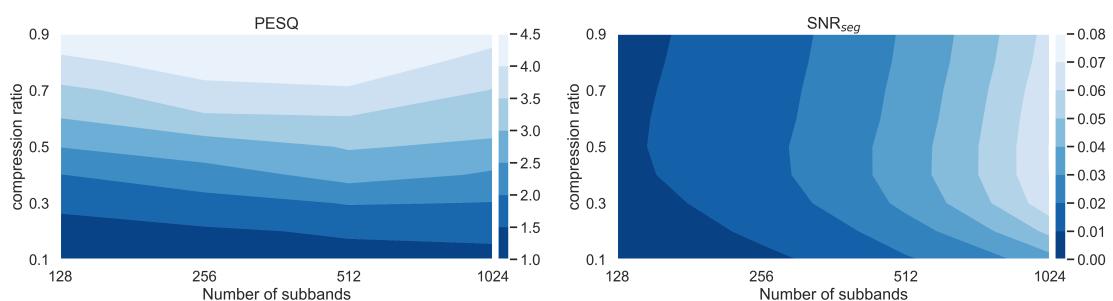


Figure 5.4: PESQ and SNR_{seg} error space maps as a function of compression ratio and number of subbands.

⁵⁹⁸ Chapter 6

⁵⁹⁹ Conclusions

600 **Bibliography**

- 601 [1] M. R. Mathew and B. Premanand, Sub-Nyquist Sampling of Acoustic Signals
602 Based on Chaotic Compressed Sensing, *Procedia Technol.* **24**, 941 (2016),
603 ISSN 22120173.
- 604 [2] I. Andráš, P. Dolinský, L. Michaeli, and J. Šaliga, A time domain reconstruction
605 method of randomly sampled frequency sparse signal, *Meas. J. Int. Meas.*
606 *Confed.* **127**, 68 (2018), ISSN 02632241.
- 607 [3] S. Y. Low, D. S. Pham, and S. Venkatesh, Compressive speech enhancement,
608 *Speech Commun.* **55**, 757 (2013), ISSN 01676393.
- 609 [4] S. Y. Low, Compressive speech enhancement in the modulation domain, *Speech*
610 *Commun.* **102**, 87 (2018), ISSN 01676393.
- 611 [5] V. Abrol, P. Sharma, and A. K. Sao, Voiced/nonvoiced detection in
612 compressively sensed speech signals, *Speech Commun.* **72**, 194 (2015), ISSN
613 01676393.
- 614 [6] Y. Mo, A. Zhang, F. Zheng, and N. Zhou, An image compression-encryption
615 algorithm based on 2-D compressive sensing, *J. Comput. Inf. Syst.* **9**, 10057
616 (2013), ISSN 15539105.

- 617 [7] N. Zhou, S. Pan, S. Cheng, and Z. Zhou, Image compression-encryption
618 scheme based on hyper-chaotic system and 2D compressive sensing, *Opt. Laser*
619 *Technol.* **82**, 121 (2016), ISSN 00303992.
- 620 [8] R. A. Romero, G. A. Tapang, and C. A. Saloma, in *Proceedings of*
621 *the Samahang Pisika ng Pilipinas Physics Conference* (University of the
622 Philippines Visayas, Iloilo City, Philippines, 2016), vol. 34, SPP–2016–PA–14.
- 623 [9] S. Liu, M. Gu, Q. Zhang, and B. Li, Principal component analysis algorithm
624 in video compressed sensing, *Optik (Stuttg.)*. **125**, 1149 (2014), ISSN 00304026.
- 625 [10] J. Chen, K. X. Su, W. X. Wang, and C. D. Lan, Residual distributed
626 compressive video sensing based on double side information, *Zidonghua*
627 *Xuebao/Acta Autom. Sin.* **40**, 2316 (2014), ISSN 02544156.
- 628 [11] E. J. Candès, J. K. Romberg, and T. Tao, Stable signal recovery from
629 incomplete and inaccurate measurements, *Commun. Pure Appl. Math.* **59**,
630 1207 (2006), ISSN 00103640, arXiv:0503066v2.
- 631 [12] D. Donoho, M. Elad, and V. Temlyakov, Stable recovery of sparse overcomplete
632 representations in the presence of noise, *IEEE Trans. Inf. Theory* **52**, 6 (2006),
633 ISSN 0018-9448.
- 634 [13] D. Donoho and X. Huo, Uncertainty principles and ideal atomic decomposition,
635 *IEEE Trans. Inf. Theory* **47**, 2845 (2001), ISSN 00189448.
- 636 [14] D. L. Donoho and M. Elad, Optimally sparse representation in general
637 (nonorthogonal) dictionaries via L1 minimization, *Proc. Natl. Acad. Sci. U. S.*
638 *A.* **100**, 2197 (2003), ISSN 00278424.

- 639 [15] N. Linh-Trung, D. Van Phong, Z. M. Hussain, H. T. Huynh, V. L. Morgan,
640 and J. C. Gore, Compressed sensing using chaos filters, *Proc. 2008 Australas.*
641 *Telecommun. Networks Appl. Conf. ATNAC 2008* 219–223 (2008).
- 642 [16] C. E. Shannon, Communication in the presence of noise, *Proceedings of the*
643 *Institute of Radio Engineers* **37**, 10 (1949).
- 644 [17] E. J. Candes and M. B. Wakin, An introduction to compressive sampling: A
645 sensing/sampling paradigm that goes against the common knowledge in data
646 acquisition, *IEEE Signal Process. Mag.* **25**, 21 (2008), ISSN 10535888.
- 647 [18] CCITT Study Group VIII and Joint Photographic Experts Group, *T.81 –*
648 *Digital compression and coding of continuous-tone still images – Requirements*
649 *and guidelines* (1982).
- 650 [19] S. Diamond and S. Boyd, CVXPY: A Python-embedded modeling language
651 for convex optimization, *Journal of Machine Learning Research* **17**, 1 (2016).
- 652 [20] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, A rewriting system for
653 convex optimization problems, *Journal of Control and Decision* **5**, 42 (2018).
- 654 [21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel,
655 M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn:
656 Machine learning in Python, *Journal of Machine Learning Research* **12**, 2825
657 (2011).
- 658 [22] R. Rubinstein, M. Zibulevsky, and M. Elad, Efficient implementation of the
659 K-SVD algorithm using batch orthogonal matching pursuit, *CS Tech.* 1–15
660 (2008).
- 661 [23] S. G. Mallat and Z. Zhang, Matching Pursuits With Time-Frequency
662 Dictionaries (1993).

- 663 [24] A. Domahidi, E. Chu, and S. Boyd, in *European Control Conference (ECC)*
664 (2013), 3071–3076.
- 665 [25] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, Image Quality Assessment:
666 From Error Visibility to Structural Similarity, *IEEE Trans. Image Process.*
667 **13**, 600 (2004), ISSN 1057-7149.
- 668 [26] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, A fast approach for
669 overcomplete sparse decomposition based on smoothed L0 norm, *IEEE Trans.*
670 *Signal Process.* **57**, 289 (2009), ISSN 1053587X, arXiv:arXiv:0809.2508v2.
- 671 [27] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, Least angle regression,
672 *The Annals of Statistics* **32**, 407 (????).
- 673 [28] B. L. Sturm and M. G. Christensen, in *20th European Signal Processing*
674 *Conference (EUSIPCO 2012)* (2012), 220–224.
- 675 [29] J. Xiang, H. Yue, X. Yin, and L. Wang, A new smoothed L0 regularization
676 approach for sparse signal recovery, *Mathematical Problems in Engineering*
677 (2019).
- 678 [30] J. S. Garafolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L.
679 Dahlgren, and V. Zue, *TIMIT Acoustic-Phonetic Continuous Speech Corpus*
680 *LDC93S1* (Linguistic Data Consortium, 1993).
- 681 [31] Telecommunication Standardization Sector of ITU, Perceptual evaluation of
682 speech quality (PESQ): An objective method for end-to-end speech quality
683 assessment of narrow-band telephone networks and speech codecs, *ITU-T*
684 *Recommendation P.862 (02/01)* (2001).
- 685 [32] P. C. Loizou, *Speech Enhancement: Theory and Practice* (CRC Press, 2013),
686 2nd ed.

687 Appendix A

688 Codes and Implementations

Listing A.1: Code for compressive sensing of 1D test sinusoids.

```
689 import numpy as np
690 import numpy.random as rand
691 import scipy.fftpack as fft
692
693 # load recording
694 signal = np.loadtxt("piano.txt").astype("float32")
695
696 # define parameters
697 samprate = 44.1e3
698 duration = 1/8
699 N = int(duration*samprate)
700 M = 300
701 t = np.linspace(0, duration, N)
702
703 # extract short portion of recording
704 sig_start = 40000
705 x = signal[sig_start:sig_start + N]
706
707 # simulate compressive measurements
708 yi = rand.randint(0, N, M)
709 yi = np.sort(yi)
710 y = x[yi]
711
712 # L1 optimization using CVX ECOS
713 import cvxpy as cvx
714 xhat_cvx = cvx.Variable(N)
715 objective = cvx.Minimize(cvx.Norm(xhat_cvx, 1))
```

```

2817 constraints = [A*xhat_cvx == y]
2918 prob = cvx.Problem(objective, constraints)
3019 result = prob.solve(verbose=True, solver="ECOS")
3120 x_cvx = np.array(xhat_cvx.value)
3221 x_cvx = np.squeeze(x_cvx)
3322 x_cvx = fft.dct(x_cvx, norm="ortho", axis=0)
3423
3524 # L1-regularized L2 optimization using LASSO
3625 from sklearn.linear_model import Lasso, LassoCV
3726 lasso = LassoCV(cv=10, random_state=0, verbose=True, n_jobs=-1)
3827 lasso.fit(A, y)
3928 x_lasso = fft.idct(lasso.coef_)

```

Listing A.2: Code for generating different random distributions.

```

730 from sklearn.metrics import mean_squared_error
1  # define parameters
2  samprate = 44.1e3
3  duration = 1/32
4  N = int(duration*samprate)
5  M = np.arange(50, N+1, 50)
6  t = np.linspace(0, duration, N)
7
8  # define normalization function
9  def normalize(x):
10     x = x.astype(float)
11     x /= x.max()
12     return x
13
14  # evaluate errors
15  y_uniform_errs = []
16  for m in M:
17      trial_err = []
18      for i in range(10):
19          yi = rand.uniform(0, N, m)
20          yi = np.sort(yi)
21          y_uniform = signal[yi]
22          d = np.identity(N)
23          d = fft.dct(d)
24          A = d[yi]
25          lasso = Lasso(alpha=0.1)
26          lasso.fit(A, y_uniform)
27          xhat_uniform = fft.idct(lasso.coef_)
28          mse = mean_squared_error(normalize(signal), normalize(xhat_uniform))
29          trial_err.append(mse)
30
31  y_uniform_errs.append(trial_err)

```
