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

April 2020

Acknowledgments

Abstract

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This study explores the use of compressive sensing (CS)—an emergent sampling theorem that allows reconstruction of signals from much fewer samples than required by the Nyquist-Shannon sampling theorem (NST)—as a viable method for compression, encryption, and/or enhancement. In this framework, the computational burden of encoding/decoding a signal is shifted from the sampling device to the device performing reconstruction, decompression, or other modes of post-processing. As such, there exist many ways to reconstruct a signal from compressive measurements.

CS has found its applications in simple audio signals containing stable frequencies, such as pure tones [1, 2], and dynamic frequencies, such as speech [3–5], images [6–8], and grayscale videos [9, 10]. The formulation of a sensing matrix in CS requires a basis conforming to some uniform uncertainty principle, and most common starting points would be partial discrete cosine transforms (DCT) or partial discrete wavelet transforms (DWT). Recent studies, however, have shown that learned bases perform much better on more complex signals [11–13], i.e., those that would be typically encountered in real life situations. The learning algorithms associated with the construction of these bases range from classical iterative methods, which have long been used in optimization problems, to the more contemporary machine learning methods.

1.1 Related literature

In 2004, Candès, Romberg, Tao [14], and Donoho [15] asked the question,

> With the recent breakthroughs in lossy compression technologies, we now know that most of the data we acquire can be thrown away with minimal perceptual loss. Why bother to acquire all the data when we can just directly measure the part that will not be thrown away?

This was eventually answered in many different ways by the same people, ultimately birthing the field which we now know as compressive sensing. The methods in CS apply concepts from time-frequency uncertainty principles [16] and sparse representations, which were studied rigorously by Donoho and Elad [17]. CS can be viewed as a strategic undersampling method: the signal is sampled at random locations in the real domain, and the ratio of the indices where it is sampled to the size of the signal can be associated with some quasi-frequency which may be lower than the Nyquist rate.

Linh-Trung et al. [18] demonstrated the use of deterministic chaos filters to acquire samples instead of random distributions. Sampling using a Gaussian-Logistic map was applied to acoustic signals in [1]. Normally, a deterministic chaotic function will need one or more initialization values as a "seed", and the sequence of numbers produced by different combinations of initial values rapidly diverge from each other. This phenomenon led to investigating the use of compressive sensing as an encryption algorithm. Simultaneous compression and encryption was achieved by [6], and it was found that the initial values were sensitive to perturbations on the order of 10^{-15}. Their image compression-encryption model via compressive sensing was shown to have a key space on the order of 10^{34}, making it extremely resistant to brute force and other types of attacks. This was extended in [7] to utilize a higher-dimensional variant of the Lorenz attractor, subsequently expanding the key space to the order of 10^{83}. In the methods above, sampling was performed in the signal domain (i.e., temporal domain for audio, spatial domain for images), and the reconstruction was performed in the frequency domain with a DCT or similar basis. [2] proposed a method to perform both sampling and reconstruction in the time domain using differential evolution.

Audio signals, compared to images, are much more densely packed with information. Whereas images are not naturally bandlimited and rather, are dependent on the spatial resolution and bit depth of the imaging device, audio size scales proportionally with time and takes on a wider range of values. The accepted frequency range of human hearing is from 20 Hz to 20 kHz, so by the NST, a sampling frequency of at least 40 kHz is needed to ensure that an audio sample is recorded correctly. Any meaningful audio recording, especially those containing speech, will certainly have a duration of a few seconds up to a few hours, so one cannot straightforwardly apply methodologies used for images or recordings with relatively static frequencies, as the first challenge this would pose for electronic systems is insufficient memory to process the entire signal all at once. Low [3, 4] circumvented this problem by transforming the signal to the modulation domain, i.e., the signal's spectrogram, essentially raising a one-dimensional signal to N-dimensions, where the value of N is dependent on the desired spectrogram resolution, number of subbands, and percent overlap between adjacent subbands. In such signals, recordings with an observed noise floor could be easily be denoised, which is an inherent property of CS [19].

Due to the large size of video information as a consequence of its high dimensionality, it is possible, but impractical, to apply image CS techniques on an entire frame-by-frame basis. Correlations between adjacent frames are utilized instead, and can be obtained using dictionary learning [11] or principal components analysis [9]. For the same reason, the application of CS to grayscale videos presuppose the use of machine learning methods. Iliadis [20, 21] came up with two different deep neural network architectures whose inputs and outputs are patches derived from grayscale videos. This idea was utilized in [22] who modified the architecture into a residual network containing several convolutional layers. The original design was targeted towards image reconstruction, but could easily be extended to videos.

In the same vein, neural network methods could also be used in CS of speech. Advances in natural language processing were primarily made using recurrent neural networks (RNN). In [23], a speech signal was first modeled by their proposed RNN architecture based on a noise-constrained least squares estimate, and final recovery is done via Kalman filtering. A new simple recurrent unit (SRU) network was created in [24] which maps the relation between noisy and clean speech recordings for speech enhancement.

1.2 Novelty

This study aims to provide a generalization for applying CS techniques to signals of arbitrary dimensions. Previous studies worked exclusively with either audio or image sequences as the target for CS, and due to the computational demands, the focus of most of the research in the field has been to optimize the computational complexity for real-time applications, and improve signal reconstruction quality. In the establishment of CS methods, two different general frameworks to compressively sample signals arise, namely, one-dimensional CS (1DCS) and two-dimensional CS (2DCS). It is shown that an N-dimensional signal can be decomposed into factors of one-dimensional and two-dimensional signals, and can be processed using methods appropriate for each type of signal. Furthermore, it is shown that N is bound not only by the type of signals being worked with, but also the computational power of the decoding/decompressing device. In particular, large values of N are useful in encryption, where a signal is first raised to a high dimension in a certain basis, the sensing matrix is derived from another high-dimensional basis, and the result is cast back to either one or two dimensions to yield the encrypted message.

1.3 Thesis overview

The next chapter establishes the relevant mathematical concepts and notation to be used throughout this study, algorithms used in signal reconstruction, and appropriate metrics per type of signal. Chapters 3–5 respectively focus on two-dimensional CS, one-dimensional CS, and N-dimensional CS. The reason behind the ordering of Chapters 3 & 4 will become apparent as the usage of spectrograms are introduced. Each of these chapters are self-contained methodologies, results, and discussions to emphasize that the methods can work independently of each other, save for the generalization to N-dimensions. The study is concluded and recommendations for future studies are presented in Chapter 6.

Chapter 2

Preliminaries

The trend of both curiosity and profit-driven human development has caused a surge in the amount of openly accessible raw data. More often than not, the data is generated much faster than it can be processed into something interpretable or useful. In the endeavor of keeping up with the inflow of information, there are two major factors that significantly hinder our progress. First, Moore's law implicitly sets a physical limit to the number of transistors that can be placed on a chip, consequently limiting how powerful and how fast electronic systems can become (barring a paradigm shift in the fundamental design of semiconductors). The second is the Nyquist-Shannon sampling theorem (NST), which limits the range of frequencies a recording device can successfully capture. This states that given that you know a signal's highest frequency component f_B, sampling it at a rate f_S that is at least twice this frequency is sufficient to capture all of the pertinent information regarding that signal: that is $f_S \geq 2f_B$, where f_B is known as the Nyquist rate [25], and also as the signal bandwidth. 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 least 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.

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

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{2.1}$$

or in the literature of signal processing [26], by correlating them with a waveform basis

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

In conventional sampling, $\mathbf{a}_k$ are Dirac basis vectors which turn $\mathbf{y}$ into a vector containing samples of $\mathbf{x}$ in the temporal or spatial domain; if $\mathbf{a}_k$ are Fourier basis vectors (i.e., sinusoids), then $\mathbf{y}$ is a vector of Fourier coefficients. If the signal has been sampled sufficiently in the sense that the number of measurements m is equal to the dimension n of the signal, then $\mathbf{A}$ is a square matrix, and the original signal $\mathbf{x}$ can be reconstructed from the information vector $\mathbf{y}$ by inversion of (2.1). However, the process of recovering $\mathbf{x} \in \mathbb{R}^n$ from $\mathbf{y} \in \mathbb{R}^m$ becomes ill-posed when we consider the undersampled case ($m \ll n$), as the sensing matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$—whose row vectors are denoted as $\mathbf{a}_m$—causes the system to become underdetermined: there exist infinitely many candidate solutions $\hat{\mathbf{x}}$ which satisfy (2.1). To add to this, we also consider the possibility that the measurements are not perfect, and are contaminated with noise. How then do we recover a signal from measurements which are incomplete and most likely inaccurate? The answer lies in enforcing constraints based on models of natural signals, as well as constraints based on optimization techniques.

2.1 Sparsity

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

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

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

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

and its corresponding inverse is

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

where the cosine term corresponds to $\boldsymbol{\psi}(k)$ in (2.3). We can express (2.4) more conveniently as $\mathbf{f} = \mathbf{\Psi}\mathbf{x}$, where $\mathbf{\Psi} \in \mathbb{R}^{n\times n}$ is the sparsifying matrix. Figure 2.1 shows this sparsifying process in action: given a test image, taking its Daubechies 8 discrete wavelet transform (D8 DWT) and zooming into a random subset shows that most of the signal energy is concentrated in just a few of the coefficients. All the other coefficients, when compared to the k highest coefficients, are practically zero; such a signal is referred to as k-sparse. The compressed image resulting from discarding all but the 25,000 highest coefficients and performing the 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 [27].

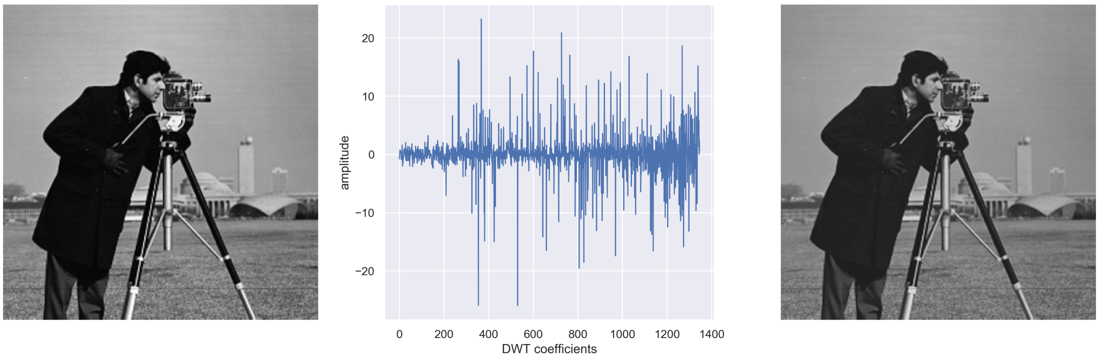

Figure 2.1: Original 512 × 512, 8-bit image (left), and a random subset (for better visibility) of its D8 DWT coefficients (middle). Most of the signal energy is concentrated in just a few terms. By discarding all but the 25,000 highest coefficients and performing the inverse transform, the resulting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

2.2 Incoherence

Suppose we have two matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ which are involved in the sensing of a signal. As before, $\mathbf{\Psi}$ is the sparsifying matrix which converts the signal into a sparse representation, and $\mathbf{\Phi}$ is the actual sensing matrix. The coherence between these two bases is expressed as

$$\mu(\mathbf{\Phi}, \mathbf{\Psi}) = \sqrt{n} \max_{0 \le i,j < n} |\langle \boldsymbol{\varphi}_i | \boldsymbol{\psi}_j \rangle| \tag{2.6}$$

In other words, the coherence is the measure of the largest correlation between the column vectors of $\mathbf{\Phi}$ and $\mathbf{\Psi}$. In compressive sensing, we are interested in low-coherence basis pairs (i.e., basis pairs for which $\mu \to 1$). For example, in the classical sampling scheme, $\mathbf{\Phi}$ is the Dirac basis $\boldsymbol{\varphi}_k(t) = \delta(t-k)$, and $\mathbf{\Psi}$ is the DCT basis (2.4). This basis pair in particular is also called a time-frequency pair and achieves maximum incoherence ($\mu = 1$) regardless of the number of dimensions [16]. Additionally, any orthonormal basis $\mathbf{\Phi}$ containing independent identically distributed (i.i.d.) entries are also largely incoherent with a fixed basis $\mathbf{\Psi}$ [26]. The consequence of this is that CS performs most efficiently when sensing with incoherent and random systems.

2.3 Reconstruction strategies

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

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

which, as the name implies, requires one to enumerate all possible k-element combinations of the columns of $\mathbf{\Phi}$, and determining the smallest combination which approximates the signal the closest. However, this process quickly becomes intractable even for a modestly-sized signal. This requirement is therefore relaxed by instead minimizing the ℓ_1 norm

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

where the ℓ_1 norm is defined as

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

and is commonly called the taxicab or Manhattan distance. Most signals encountered in practical situations, however, are not sparse but rather, approximately sparse. As mentioned earlier, any signal measurement will inevitably include some form of noise. Though ℓ_1 minimization can definitely still be used (by casting it as a convex problem, as in the case of [28, 29]), other algorithms opt for an ℓ_1-regularized least squares approach as in the case of LASSO [30], whose objective is

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

where $0 \leq \alpha \leq 1$ is the ℓ_1 regularization parameter. Greedy algorithms are also a popular approach in this problem, the most common being the sparsity-constrained orthogonal matching pursuit (OMP) [31], which has the objective

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

This method enforces the constraint that the reconstructed signal should be, at most, k-sparse in the selected coding dictionary $\mathbf{\Phi}$. There exist a plethora of algorithms dedicated to the decoding phase of CS. The ones mentioned above are primarily used in this study.

Chapter 3

Two-dimensional compressive sensing (2DCS)

3.1 Methodology

The compressive sensing (CS) of two-dimensional, spatial signals is the more intuitive of its applications as it is easier to visualize. As we will see later on, the CS of image signals can be simplified either by flattening it to one dimension or by patches. The general workflow that arises from image CS is as follows:

1. Define the compression ratio m/n, where n is the signal size, and m is the desired size of the compressed signal.

2. Draw m random indices from the signal without replacement and store this as a vector ξ.

3. Extract the row vectors of the desired $n \times n$ sparsifying basis $\mathbf{\Psi}$ corresponding to the indices in ξ, and stack these to form the sensing matrix $\mathbf{\Phi}$.

4. With the desired reconstruction algorithm, perform the minimization (2.8) to obtain the reconstructed signal $\hat{\mathbf{x}}$.

In the case of high-definition images, it is usually more practical and yields better results if the image is first divided into smaller, manageable patches and apply the above steps per patch, then stitch the patches back together at the end to form the final reconstructed image.

3.2 Test case: Sinusoidal pattern

Image signals are commonly expressed sparsely as a linear superposition of a finite number of 2D sinusoidal patterns. In Fig. 3.1, 64×64 pixel sinusoidal patterns were generated, corresponding to $\sin(x)$ (sine wave traveling horizontally), $\sin(y)$ (sine wave traveling vertically), $\sin(x + y)$ (sine wave traveling diagonally), and $\sin(x) + \sin(y)$ (egg tray pattern). In each case, all frequency components are 4 Hz. Figure 3.2 visualizes the compressed image when a random sample of 5% is taken from the signal. The actual compressed signal that is seen by the reconstruction algorithm is a one-dimensional vector containing the sampling points. For reconstruction, the algorithm used is CVXPY, which casts the minimization problem 2.8 as a convex optimization problem, and directly minimizes the ℓ_1 norm [28, 29]. From the 5% of samples taken from the original image, the reconstructed signals are shown in Fig. 3.3, each of which has a mean-squared error (MSE) of no more than 0.06. Exact reconstruction is attained for the pure horizontal and pure vertical sine waves, as well as the egg tray pattern. Additional artifacts are present in the reconstructed diagonal sine wave, particularly at the boundaries.

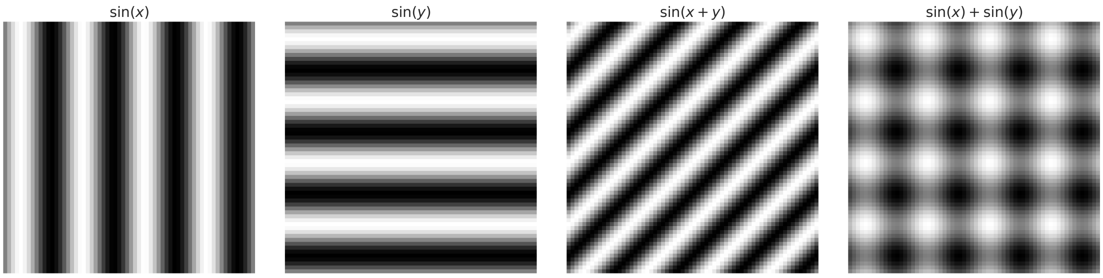

Figure 3.1: Test 2D sinusoid patterns.

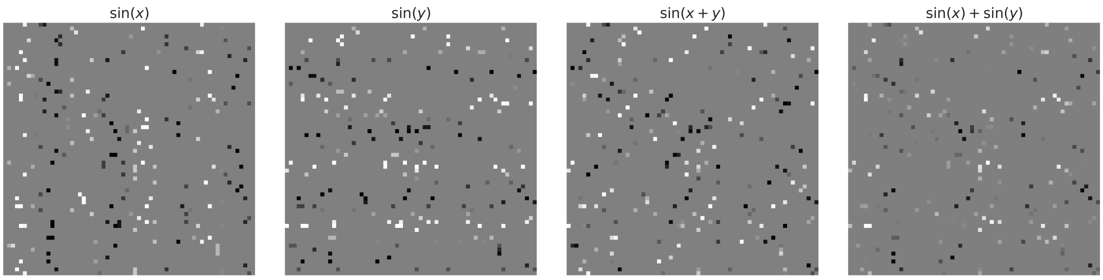

Figure 3.2: Visualization of compressed sinusoid patterns.

3.3 Image with multiple sinusoids

For this section, the image used is M.C. Escher's *Relativity*, an example of a more complex image consisting of various sinusoidal patterns. The original image has dimensions of 1600×981 pixels, for a total of 1,569,600 pixels. This would require the construction of a $1{,}569{,}600 \times 1{,}569{,}600$ sparsifying matrix, or $\approx 2 \times 10^{12}$ entries. Assuming that the matrix would be stored as 32-bit floating point numbers, this

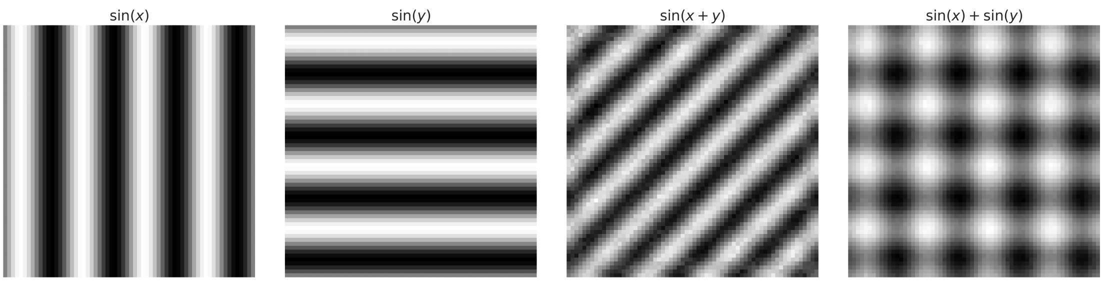

Figure 3.3: Reconstructed sinusoid patterns from 5% of samples from original image.

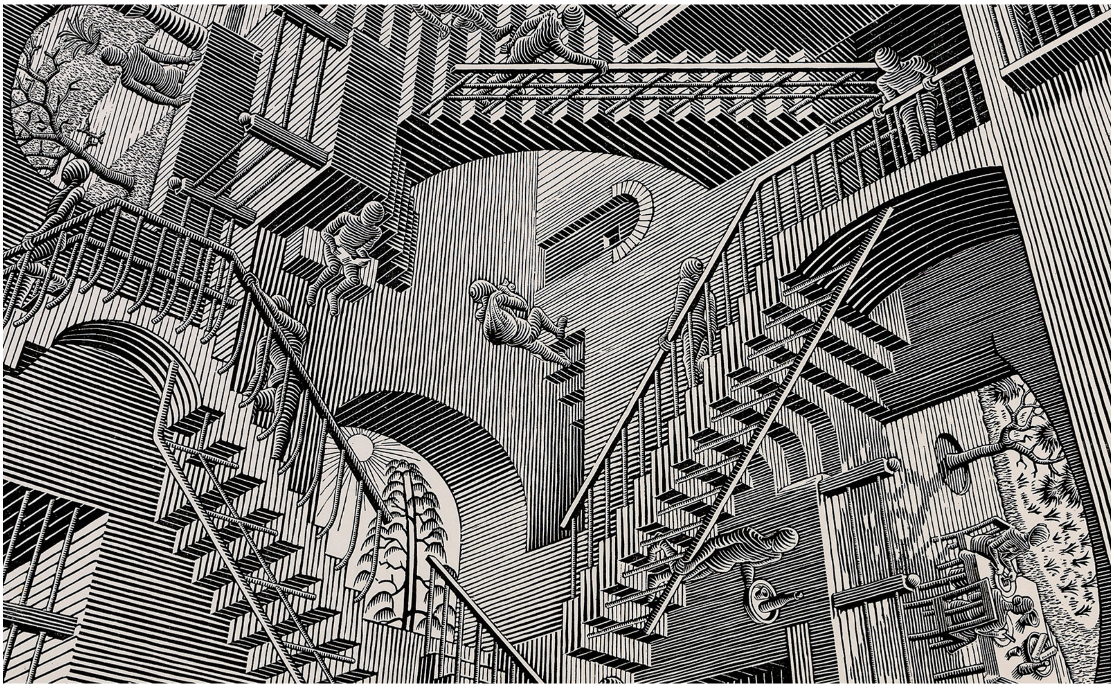

Figure 3.4: *Relativity* by M.C. Escher, a complex image consisting of various sinusoidal patterns.

process alone would take up ≈ 8 GB of RAM, and it would be highly impractical to process similar images as a whole. A workaround is to split it into smaller, manageable patches. For this image in particular, it was first resized to 1600×976 pixels so that it could be equally divided into a grid of 16×16, each with a dimension of 100×61 pixels. After compressively sensing each patch and stitching all patches at the end, the reconstructed image is shown in Fig. **??**. Selected patches are shown side-by-side with their reconstructed counterparts in Fig. **??**, corresponding to patches dominated by horizontal sinusoids, vertical sinusoids, diagonal sinusoids, and patches with no dominant pattern.

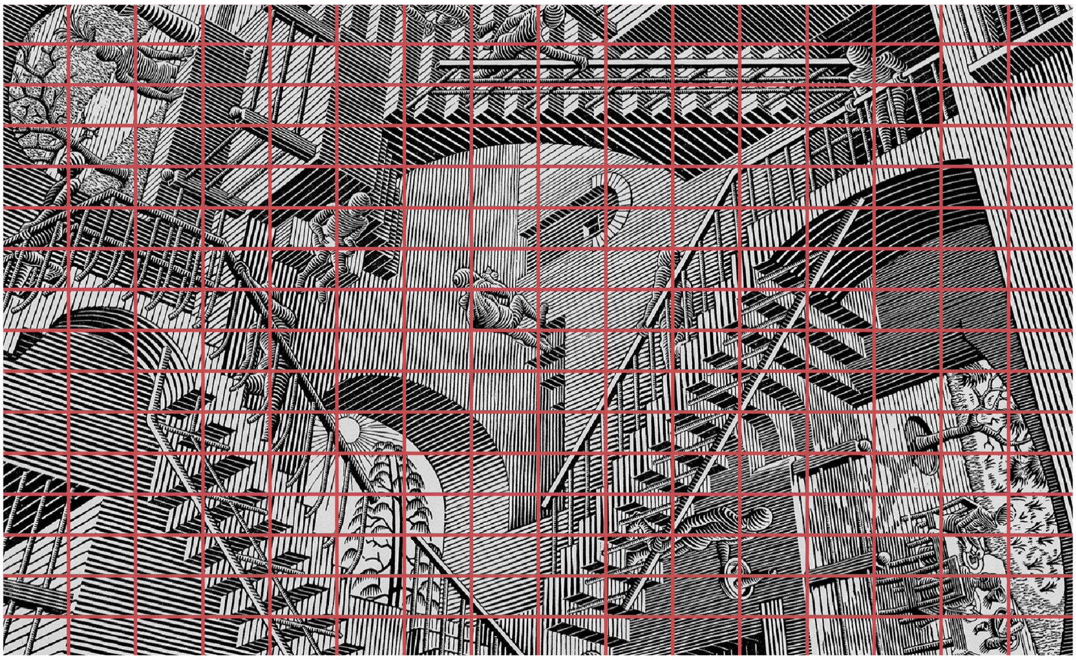

Figure 3.5: Test image divided into a grid of 16×16.

3.4 Simultaneous compression & encryption

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

$$x_{n+1} = rx_n(1 - x_n) \tag{3.1}$$

which is often used as an archetypal example of deterministic chaotic behavior when $r \in [3.57, 4]$. Within these values, the sequences produced by varying the initial value x_0 rapidly diverge from each other. Thus, this can become an encryption system by treating the parameters r and x_0 as the encryption keys, and (3.1) as the hash function.

Chapter 4

One-dimensional compressive sensing (1DCS)

Chapter 5

Generalization to N-dimensions

Chapter 6

Conclusions

Bibliography

[1] Magnel Rose Mathew and B. Premanand. Sub-Nyquist Sampling of Acoustic Signals Based on Chaotic Compressed Sensing. *Procedia Technol.*, 24:941–948, 2016.

[2] Imrich Andráš, Pavol Dolinský, Linus Michaeli, and Ján Šaliga. A time domain reconstruction method of randomly sampled frequency sparse signal. *Meas. J. Int. Meas. Confed.*, 127(December 2016):68–77, 2018.

[3] Siow Yong Low, Duc Son Pham, and Svetha Venkatesh. Compressive speech enhancement. *Speech Commun.*, 55(6):757–768, 2013.

[4] Siow Yong Low. Compressive speech enhancement in the modulation domain. *Speech Commun.*, 102(August):87–99, 2018.

[5] Vinayak Abrol, Pulkit Sharma, and Anil Kumar Sao. Voiced/nonvoiced detection in compressively sensed speech signals. *Speech Commun.*, 72:194–207, 2015.

[6] Yan Mo, Aidi Zhang, Fen Zheng, and Nanrun Zhou. An image compression-encryption algorithm based on 2-D compressive sensing. *J. Comput. Inf. Syst.*, 9(24):10057–10064, 2013.

[7] Nanrun Zhou, Shumin Pan, Shan Cheng, and Zhihong Zhou. Image compression-encryption scheme based on hyper-chaotic system and 2D compressive sensing. *Opt. Laser Technol.*, 82:121–133, 2016.

[8] Roland A. Romero, Giovanni A. Tapang, and Caesar A. Saloma. Compressive sensing on the Fourier domain as a method for increasing image signal-to-noise ratio. In *Proceedings of the Samahang Pisika ng Pilipinas Physics Conference*, volume 34, pages SPP–2016–PA–14, University of the Philippines Visayas, Iloilo City, Philippines, 2016.

[9] Sheng Liu, Mingming Gu, Qingchun Zhang, and Bing Li. Principal component analysis algorithm in video compressed sensing. *Optik (Stuttg).*, 125(3):1149–1153, 2014.

[10] Jian Chen, Kai Xiong Su, Wei Xing Wang, and Cheng Dong Lan. Residual distributed compressive video sensing based on double side information. *Zidonghua Xuebao/Acta Autom. Sin.*, 40(10):2316–2323, 2014.

[11] Haixiao Liu, Bin Song, Hao Qin, and Zhiliang Qiu. Dictionary learning based reconstruction for distributed compressed video sensing. *J. Vis. Commun. Image Represent.*, 24(8):1232–1242, 2013.

[12] Pulkit Sharma, Vinayak Abrol, Nivedita, and Anil Kumar Sao. Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation. *Comput. Speech Lang.*, 52:191–208, 2018.

[13] Nasser Eslahi, Ali Aghagolzadeh, and Seyed Mehdi Hosseini Andargoli. Image/video compressive sensing recovery using joint adaptive sparsity measure. *Neurocomputing*, 200:88–109, 2016.

[14] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Commun. Pure Appl. Math.*, 59(8):1207–1223, 2006.

[15] David L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

[16] D.L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Trans. Inf. Theory*, 47(7):2845–2862, aug 2001.

[17] David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via L1 minimization. *Proc. Natl. Acad. Sci. U. S. A.*, 100(5):2197–2202, 2003.

[18] Nguyen Linh-Trung, Dinh Van Phong, Zahir M. Hussain, Huu Tue Huynh, Victoria L. Morgan, and John C. Gore. Compressed sensing using chaos filters. *Proc. 2008 Australas. Telecommun. Networks Appl. Conf. ATNAC 2008*, pages 219–223, 2008.

[19] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, 2007.

[20] Michael Iliadis, Leonidas Spinoulas, and Aggelos K. Katsaggelos. Deep fully-connected networks for video compressive sensing. *Digit. Signal Process. A Rev. J.*, 72:9–18, 2018.

[21] Michael Iliadis, Leonidas Spinoulas, and Aggelos K. Katsaggelos. DeepBinaryMask: Learning a binary mask for video compressive sensing. *Digit. Signal Process. A Rev. J.*, 96:102591, 2020.

[22] Hantao Yao, Feng Dai, Shiliang Zhang, Yongdong Zhang, Qi Tian, and Changsheng Xu. DR2-Net: Deep Residual Reconstruction Network for image compressive sensing. *Neurocomputing*, 359:483–493, 2019.

[23] Youshen Xia and Jun Wang. Low-dimensional recurrent neural network-based Kalman filter for speech enhancement. *Neural Networks*, 67:131–139, 2015.

[24] Xingyue Cui, Zhe Chen, and Fuliang Yin. Speech enhancement based on simple recurrent unit network. *Appl. Acoust.*, 157:107019, 2020.

[25] Claude E Shannon. Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers*, 37(1):10–21, 1 1949.

[26] Emmanuel J. Candes and Michael B. Wakin. An introduction to compressive sampling: A sensing/sampling paradigm that goes against the common knowledge in data acquisition. *IEEE Signal Process. Mag.*, 25(2):21–30, 2008.

[27] CCITT Study Group VIII and Joint Photographics Expert Group. *T.81 – Digital compression and coding of continuous-tone still images – Requirements and guidelines*, 1982.

[28] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[29] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos,

D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[31] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. *CS Tech.*, pages 1–15, 2008.