Implementation of Compressed Sampling in Voice Signal and Image

A thesis paper submitted to the Department of Electrical and Electronic Engineering, Khulna University of Engineering and Technology, Khulna, Bangladesh, in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering.

Shantanu Sen Gupta

Roll: 1203016

Department of Electrical and Electronic Engineering
Khulna University of Engineering and Technology, Khulna- 9203, Bangladesh.

May, 2017

Implementation of Compressed Sampling in Voice Signal and Image

Author

Shantanu Sen Gupta
Roll No. 1203016

Supervisor

Dr. Md. Mahbub Hasan
Associate Professor
Department of Electrical and Electronic Engineering,
Khulna University of Engineering & Technology,
Khulna-9203.

Department of Electrical and Electronic Engineering
Khulna University of Engineering & Technology
Khulna-9203

May 2017

DECLARATION OF AUTHORSHIP

This is to certify that the thesis work entitled “Implementation of Compressed Sampling in Voice Signal and Image” has been carried out by Shantanu Sen Gupta, Roll no. 1203016 in the Department of Electrical and Electronic Engineering, Khulna University of Engineering & Technology, Khulna-9203, Bangladesh. The above thesis work or any part of this work has not been previously presented to another examination board and has not been published anywhere.

Signature of supervisor   Signature of candidate

ACKNOWLEDGEMENT

First and foremost, I would like to express my gratitude to Almighty God, the most merciful for his endless blessings to accomplish this work successfully.

It has been an honor and a pleasure to have Dr. Md. Mahbub Hasan, Associate Professor, Department of Electrical and Electronic Engineering, Khulna University of Engineering and Technology (KUET), as supervisor. In addition to his huge knowledge and experience, I could also enjoy his support and patience during the hardest moments of the research work and writing of the thesis. His broad technical skills and positive way of criticizing have been very instructive and will remain with me as a model for the future. In the end, I feel proud to have had a very positive and instructive cooperation not only with the professor figure but also with the person, Dr. Md. Mahbub Hasan.

The major acknowledgements are due to my family who gave their best for bringing me up with the lessons that are not included in any type of book. Their endless encouragement and support remain in my heart. For them, my deep love and affection forever.

Finally, I am also grateful to all of my teachers and friends for their inspiration and unconditional support for all the time.

Author

ABSTRACT

Bandwidth is a major issue in today's era. In conventional signal processing, Nyquist-Shannon sampling theorem requires more bandwidth. This thesis work presents a new approach of signal acquisition and reconstruction named 'Compressed Sampling'. This work mainly focuses on signal reconstruction. Voice and image signals are used to reconstruct. Two reconstruction algorithms namely- l_1 and OMP (Orthogonal Matching Pursuit) are exploited here. The whole experiment is conducted using MATLAB. Experimental results are presented to validate the capability of compressed sampling to reconstruct the signals.

TABLE OF CONTENT

Declaration of Authorship i

Acknowledgement ii

Abstract iii

List of contents iv

List of figures vii

List of abbreviation x

List of table xi

CHAPTER 1 INTRODUCTION

1.1 Background 1

1.2 Objectives 3

1.3 Methodology of the Thesis Work 3

1.4 Thesis Organization 4

CHAPTER 2 BACKGROUND STUDY OF SIGNAL ACQUISITION AND RECONSTRUCTION

2.1 Signal 6

2.2 Signal Acquisition 6

2.3 Nyquist-Shannon Sampling Theorem 7

2.4 Quantization 11

2.5 Encoding 13

2.6 Decoding and Reconstruction 13

CHAPTER 3 COMPRESSED SAMPLING AND ITS APPLICATIONS

3.1 Compressive Sensing 15

3.2 Sparsity 17

3.3 Incoherence 20

3.4 Design Sensing Matrix 21

3.4.1 Random Sensing Matrices 22

3.4.2 Deterministic Sensing Matrices 22

3.5 Applications of compressive sensing 23

3.5.1 Single-Pixel Camera 23

3.5.2 Magnetic Resonance Imaging 25

CHAPTER 4 RECONSTRUCTION ALGORITHMS

4.1 Types 26

4.2 Optimization Methods 26

4.3 Greedy Methods 27

4.4 Thresholding-Based Methods 28

CHAPTER 5 PROPOSED METHODOLOGY FOR VOICE SIGNAL AND IMAGE RECONSTRUCTION

5.1 Introduction 31

5.2 Programming Approach 32

CHAPTER 6 RESULT AND DISCUSSION

6.1 Introduction 34

6.2 Voice Signal Reconstruction 34

6.3 Image Reconstruction 42

6.4 Analysis of l_1 and OMP Reconstruction Algorithm 48

CHAPTER 7 RESULT AND DISCUSSION

7.1 Conclusion 50

7.2 Scopes 50

REFERENCES 52

LIST OF FIGURE

Figure 1.1	Uniformly sampled data at Nyquist rate (2x Fourier bandwidth)	1
Figure 1.2	Compress data (signal-dependent, nonlinear)	2
Figure 1.3	Flow Chart of Proposed Work	4
Figure 2.1	Digital data acquisition system block diagram	7
Figure 2.2	Signal reconstruction using different sampling rate	10
	(a) A sine wave	
	(b) Sampling at 1 time per cycle	
	(c) Sampling at 1.5 times per cycle	
	(d) Sampling at 2 times per cycle	
	(e) Sampling at many times per cycle	
Figure 2.3	Quantization process	11
Figure 2.4	Different bit resolution with different level of quantization compared to analog	12
	(a) 2-bit resolution with four levels of quantization	
	(b) 3-bit resolution with four levels of quantization	
Figure 2.5	Signal acquisition and reconstruction exploiting Nyquist-Shannon theorem	14
Figure 3.1	Block Diagram for Compressive Sensing	16
Figure 3.2	Antonella, Niels, and Paulina.	17
Figure 3.3	Schematic representation of a single-pixel camera	24
Figure 5.1	Basic flow chart of proposed work	31
Figure 5.2	Flow chart of voice signal and image reconstruction	32
Figure 6.1	Original and reconstructed signals for ‘barking of a dog’	36
	(a) Original signal	
	(b) Reconstructed signal using l_1 minimization	

	(c) Reconstructed signal using OMP	
Figure 6.2	Original and reconstructed signals for ‘Voice signal of Dipan’	37
	(a) Original signal	
	(b) Reconstructed signal using l_1 minimization	
	(c) Reconstructed signal using OMP	
Figure 6.3	Original and reconstructed signals for ‘Voice signal of Hasan’	39
	(a) Original signal	
	(b) Reconstructed signal using l_1 minimization	
	(c) Reconstructed signal using OMP	
Figure 6.4	Original and reconstructed signals for ‘Voice signal of Nayan’	40
	(a) Original signal	
	(b) Reconstructed signal using l_1 minimization	
	(c) Reconstructed signal using OMP	
Figure 6.5	Original and reconstructed signals for ‘Voice signal of Shohag’	42
	(a) Original signal	
	(b) Reconstructed signal using l_1 minimization	
	(c) Reconstructed signal using OMP	
Figure 6.6	Original and reconstructed images for ‘Einstein’	43
	(a) Original image	
	(b) Reconstructed image using l_1 minimization	
	(c) Reconstructed image using OMP	
Figure 6.7	Original and reconstructed images for ‘MAC’	44

(a) Original image

(b) Reconstructed image using l_1 minimization

(c) Reconstructed image using OMP

Figure 6.8 Original and reconstructed images for ‘Nature’ 45

(a) Original image

(b) Reconstructed image using l_1 minimization

(c) Reconstructed image using OMP

Figure 6.9 Original and reconstructed images for ‘Rabindranath’ 46

(a) Original image

(b) Reconstructed image using l_1 minimization

(c) Reconstructed image using OMP

Figure 6.10 Original and reconstructed images for ‘Shakespeare’ 47

(a) Original image

(b) Reconstructed image using l_1 minimization

(c) Reconstructed image using OMP

LIST OF SYMBOLS & ABBREVIATIONS

CS	-	Compressive Sensing
IEEE	-	Institution of Electrical Engineering
MATLAB	-	Matrix Laboratory
OMP	-	Orthogonal Matching Pursuit
CoSaMP	-	Compressive Sampling Matching Pursuit
IHT	-	Iterative Hard Thresholding
HTP	-	Hard Thresholding Pursuit

LIST OF TABLE

Table 6.1 Comparison between l1 minimization and OMP algorithm for voice signal 48

Table 6.2 Comparison between l1 minimization and OMP algorithm for images 49

CHAPTER 1

INTRODUCTION

1.1 Background

Success of digital data acquisition is placing increasing pressure on signal/image processing hardware and software to support. Because of higher resolution (ADCs, Imaging System, Microarrays), increasing number of sensors (image data bases, camera arrays, distributed wireless sensor networks) and increasing numbers of modalities (acoustic, RF, visual), deluge of data had been occurred. This phenomenon has risen several questions such as how to acquire, store, fuse and efficiently process this huge amount of data? [1] The foundation of digital data acquisition is Nyquist-Shannon sampling theorem which states that a signal can be perfectly reconstructed when it is sampled at the rate of two times of the signal's maximum frequency. This figure illustrates that the more sample we take, the more accurate will be the reconstructed signal. The more the sampling rate, the more data have to be transmitted. If these huge amounts of data have to be transmitted along a channel then a lot of bandwidth is required. Figure 1.1 shows the typical diagram of long established data acquisition paradigm.

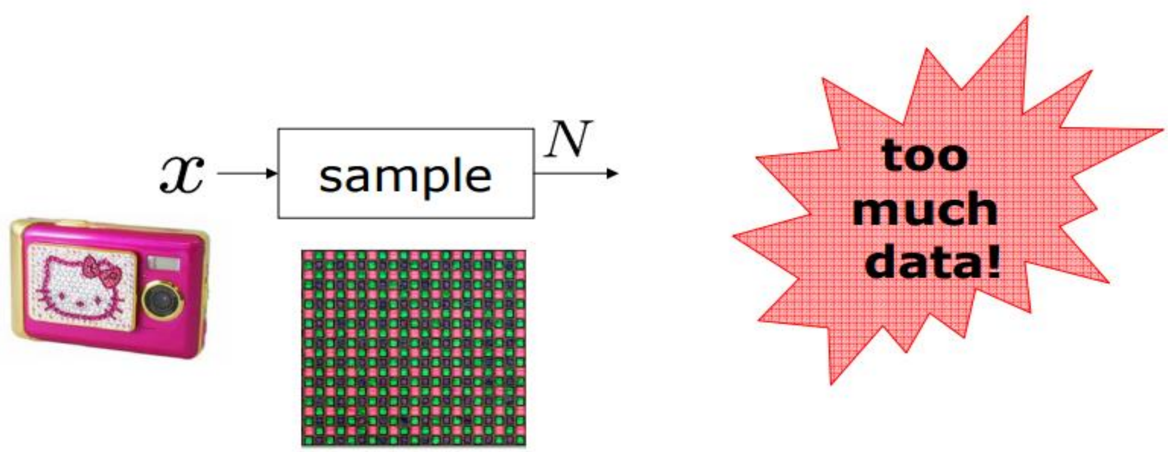


Figure 1.1: Uniformly sampled data at Nyquist rate (2x Fourier bandwidth)

So, to overcome this difficulty compressive sensing had been introduced. Compressed sensing (also known as compressive sensing, compressive sampling, or sparse sampling) is a signal processing technique for efficiently acquiring and reconstructing a signal, by finding solutions to underdetermined linear systems. This is based on the principle that, through optimization, the sparsity of a signal can be exploited to recover it from far fewer samples than required by the Shannon-Nyquist sampling theorem. There are two conditions under which recovery is possible. The first one is sparsity which requires the signal to be sparse in some domain. The second one is incoherence which is applied through the isometric property which is sufficient for sparse signals. The main idea is that with prior knowledge about constraints on the signal's frequencies, fewer samples are needed to reconstruct the signal. Basic compressive sensing paradigm is as follows in Figure 1.2

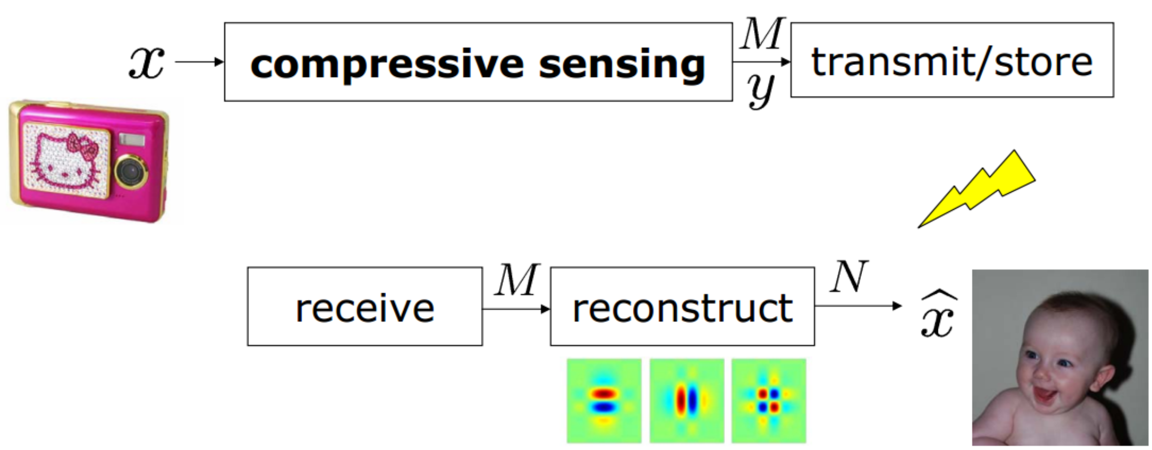


Figure 1.2: Compress data (signal-dependent, nonlinear)

1.2 Objectives

- To acquire knowledge about signal acquisition and reconstruction technique.
- To acquire knowledge about Nyquist-Shannon sampling theorem and its disadvantages.
- To introduce with Compressive Sensing and its different reconstruction algorithms.
- To use compressive sensing in voice signal and image reconstruction.

1.3 Methodology of the Thesis Work

In order to meet the objectives, the following process as shown in Figure 1.3 must be followed. For the first part of the thesis work, the understanding about signal acquisition and reconstruction is compulsory. Different ideas and topics related to the signal acquisition and reconstruction were identified. Then, after the method of the thesis work was determined, the preliminary result should be done.

For the second part of the thesis, a deep study should be done. Every single term related to compressive sensing should be mastered. Then, the simulation was implemented. The result obtained from the simulation was compared for the original signal and image. Lastly, the analysis and then the report writing should be done when the simulation was successful.

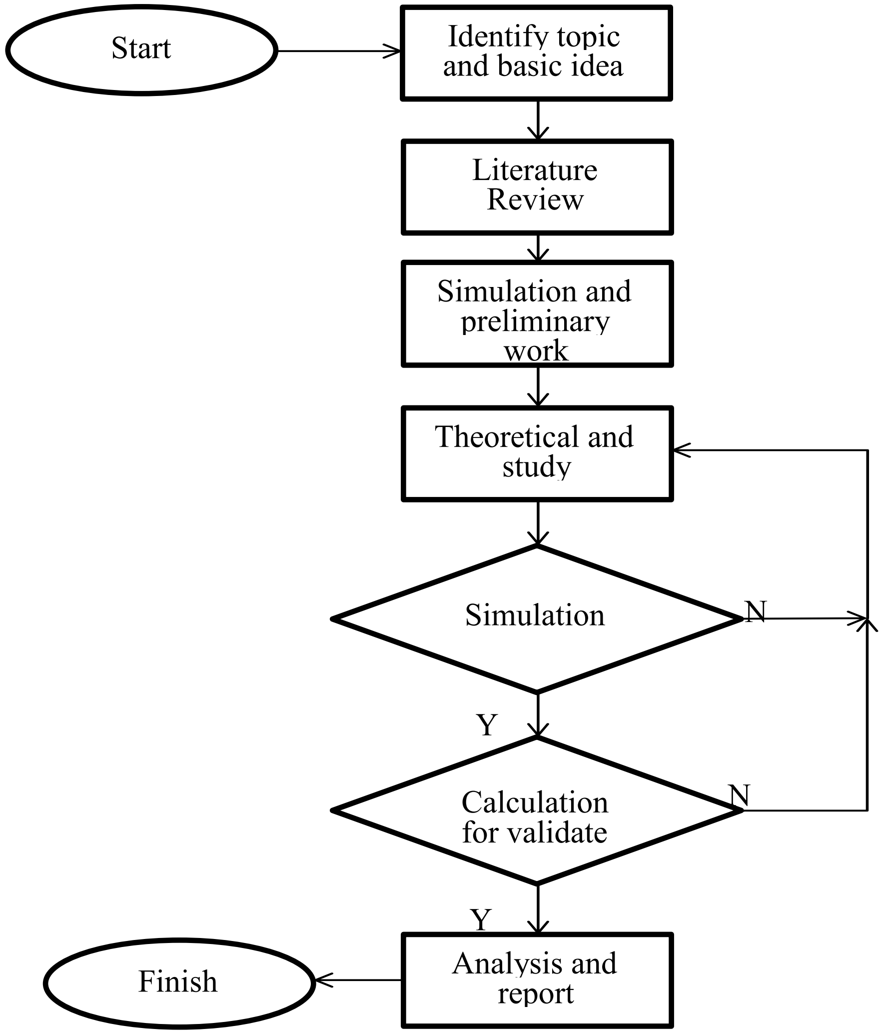


Figure 1.3: Flow Chart of Proposed Work

1.4 Thesis Organization

In **Chapter 1,** the introduction and thesis objectives are described. It also includes the literature review of the effect of Nyquist-Shannon sampling theorem on the data transmission system. In **Chapter 2,** conventional way of signal acquisition and reconstruction is discussed. **Chapter3** contains a detailed discussion on compressive sensing and its different dependencies, mathematical background, applications and past, present and future of this technology. **Chapter 4** consists of different types of reconstruction algorithms associated with compressive sensing. **Chapter 5** explains the methodology of the experiments. Detailed block diagram of each experiment is provided with the pseudo code of reconstruction algorithm. Results and wave shapes found from experiments by using MATLAB and relevant discussion are presented in **Chapter 6.** The thesis summary and the scope for future work are described in **Chapter 7**.

CHAPTER 2

BACKGROUND STUDY OF SIGNAL ACQUISITION AND RECONSTRUCTION

2.1 Signal

Signal is a detectable physical quantity or impulse (as a voltage, current, or magnetic field strength) by which messages or information can be transmitted. It is a function representing some variable that contains some information about the behavior of a natural or artificial system. Signals are one part of the whole. Signals are meaningless without systems to interpret them, and systems are useless without signals to process. [2]

2.2 Signal Acquisition

Signal acquisition or Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. Data acquisition systems, abbreviated by the acronyms DAS or DAQ, typically convert analog waveforms into digital values for processing. The components of data acquisition systems include:

- Sensors, to convert physical parameters to electrical signals.
- Signal conditioning circuitry, to convert sensor signals into a form that can be converted to digital values.
- Analog-to-digital converters, to convert conditioned sensor signals to digital values.

In conventional way, the sampling rate must fulfill the Nyquist-Shannon sampling rate. Figure 2.1 shows the digital data acquisition system block diagram.

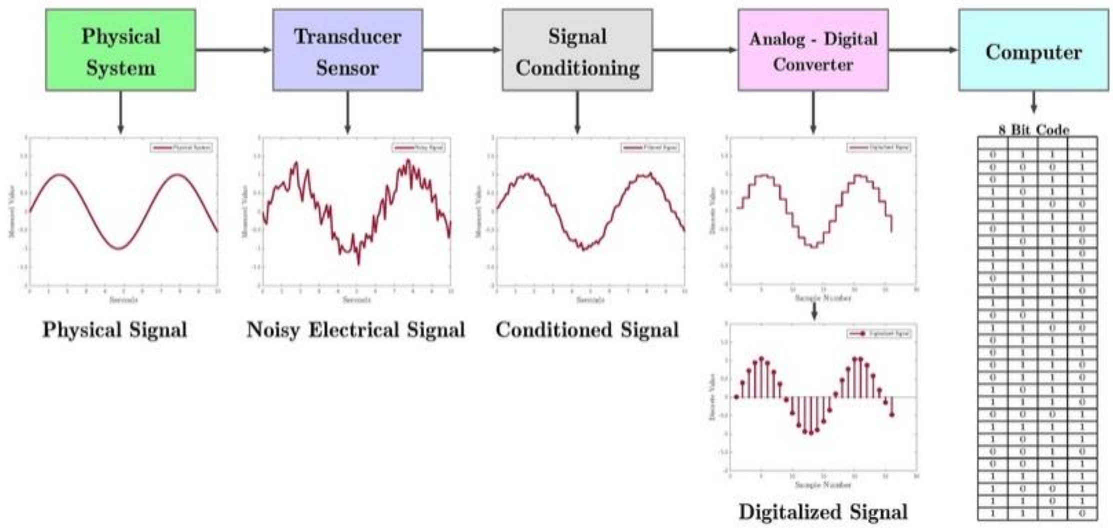


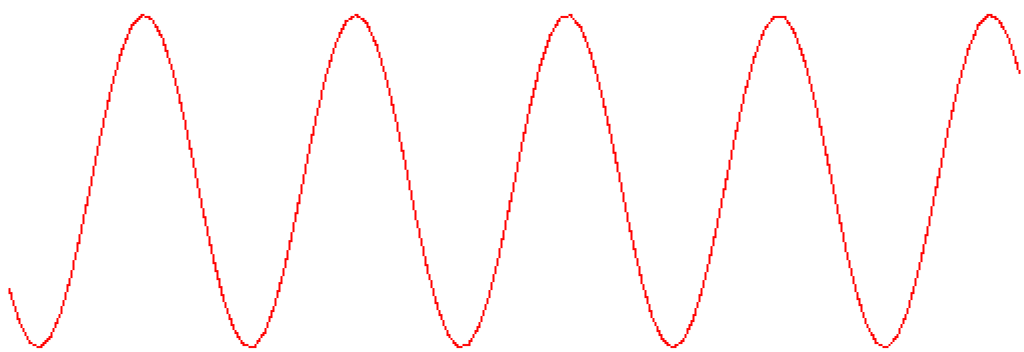
Figure 2.1: Digital data acquisition system block diagram

2.3 Nyquist-Shannon Sampling Theorem

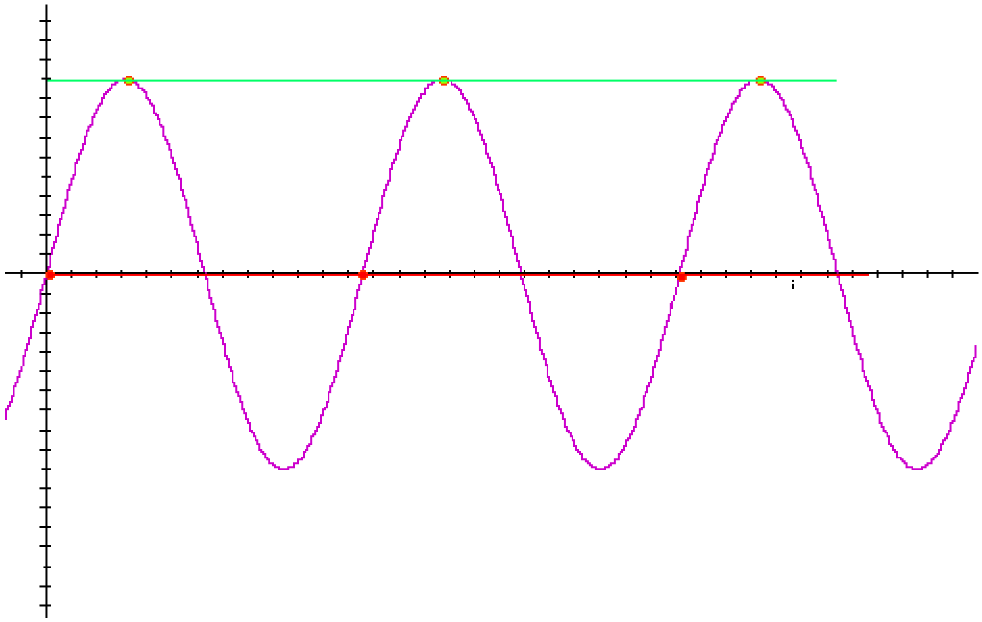
In the field of digital signal processing, the sampling theorem is a fundamental bridge between continuous-time signals (often called "analog signals") and discrete-time signals (often called "digital signals"). It establishes a sufficient condition for a sample rate that permits a discrete sequence of samples to capture all the information from a continuous-time signal of finite bandwidth.

Sampling is a process of converting a signal (for example, a function of continuous time and/or space) into a numeric sequence (a function of discrete time and/or space).

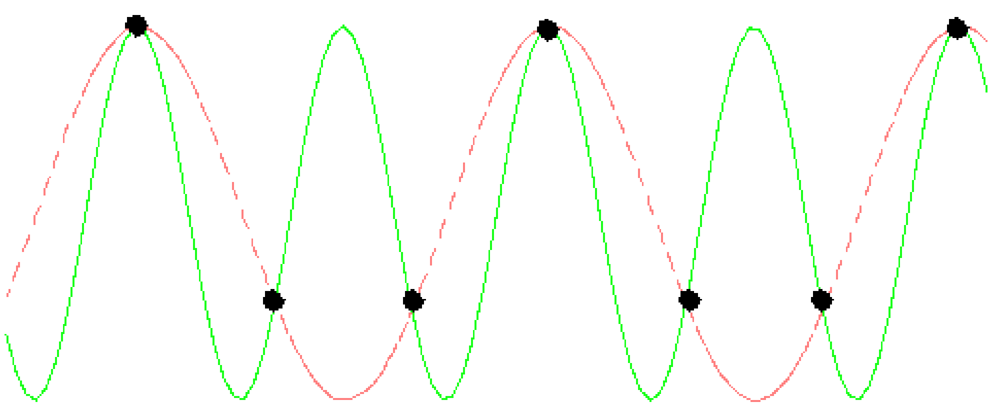
If a function x(t) contains no frequencies higher than B hertz, it is completely determined by giving its ordinates at a series of points spaced 1/(2B) seconds apart. A sufficient sample-rate is therefore 2B samples/second, or anything larger. Equivalently, for a given sample rate fs, perfect reconstruction is guaranteed possible for a bandlimit $B < f_s/2$. When the bandlimit is too high (or there is no bandlimit), the reconstruction exhibits imperfections known as aliasing. Modern statements of the theorem are sometimes careful to explicitly state that x(t) must contain no sinusoidal component at exactly frequency B, or that B must be strictly less than ½ the sample rate. The two thresholds, 2B and $f_s/2$ are respectively called the Nyquist rate and Nyquist frequency. And respectively, they are attributes of x(t) and of the sampling equipment. The condition described by these inequalities is called the Nyquist criterion, or sometimes the Raabe condition. The theorem is also applicable to functions of other domains, such as space, in the case of a digitized image. The only change, in the case of other domains, is the units of measure applied to t, fs, and B. The symbol T = 1/fs is customarily used to represent the interval between samples and is called the sample period or sampling interval. And the samples of function x(t) are commonly denoted by x[n] = x(nT) (alternatively "x_n" in older signal processing literature), for all integer values of n. Figure 2.2 shows a simple sine wave and it's reconstruction using different amount of sample. [3]



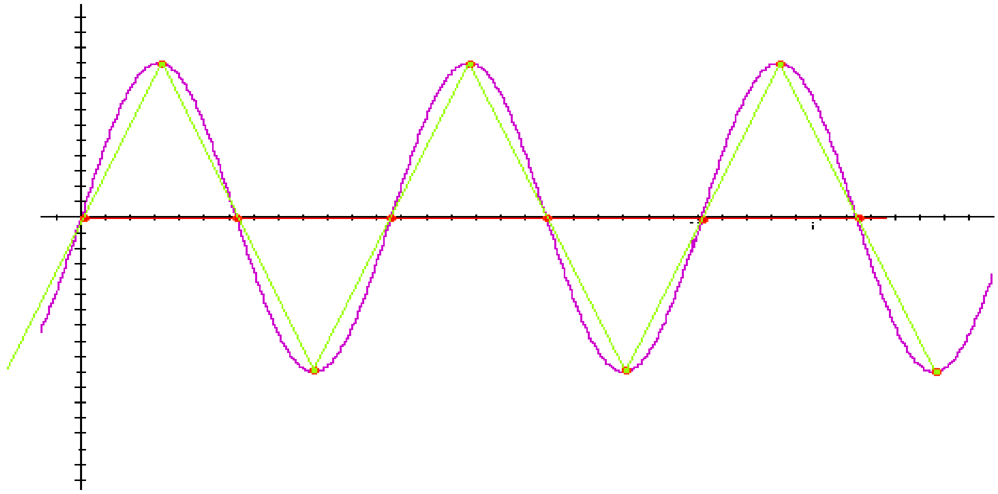
(a) A sine wave



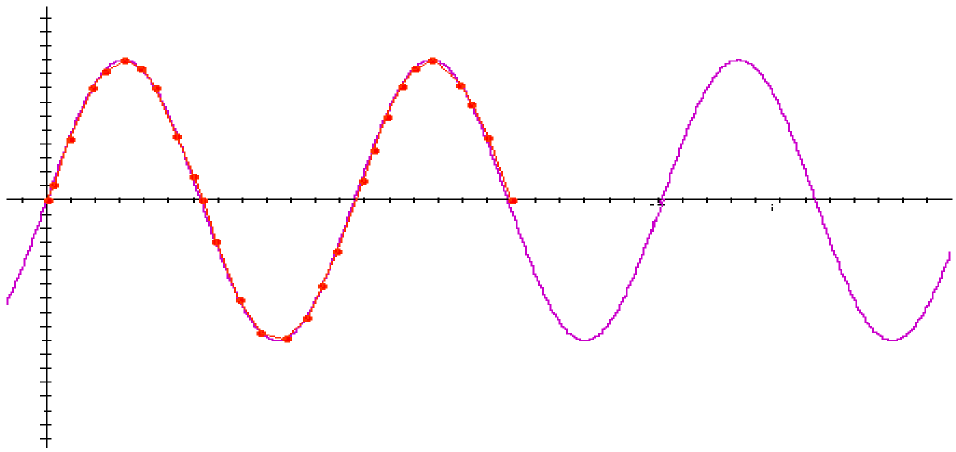
(b) Sampling at 1 time per cycle



(c) Sampling at 1.5 times per cycle



(d) Sampling at 2 times per cycle



(e) Sampling at many times per cycle

Figure 2.2: Signal reconstruction using different sampling rate

2.4 Quantization

Quantization, in mathematics and digital signal processing, is the process of mapping a large set of input values to a (countable) smaller set. Rounding and truncation are typical examples of quantization processes. Quantization is involved to some degree in nearly all digital signal processing, as the process of representing a signal in digital form ordinarily involves rounding. Quantization also forms the core of essentially all lossy compression algorithms. The difference between an input value and its quantized value (such as round-off error) is referred to as quantization error. A device or algorithmic function that performs quantization is called a quantizer. An analog-to-digital converter is an example of a quantizer. Figure 2.3 shows a original signal, its quantized signal form and quantization noise signal.

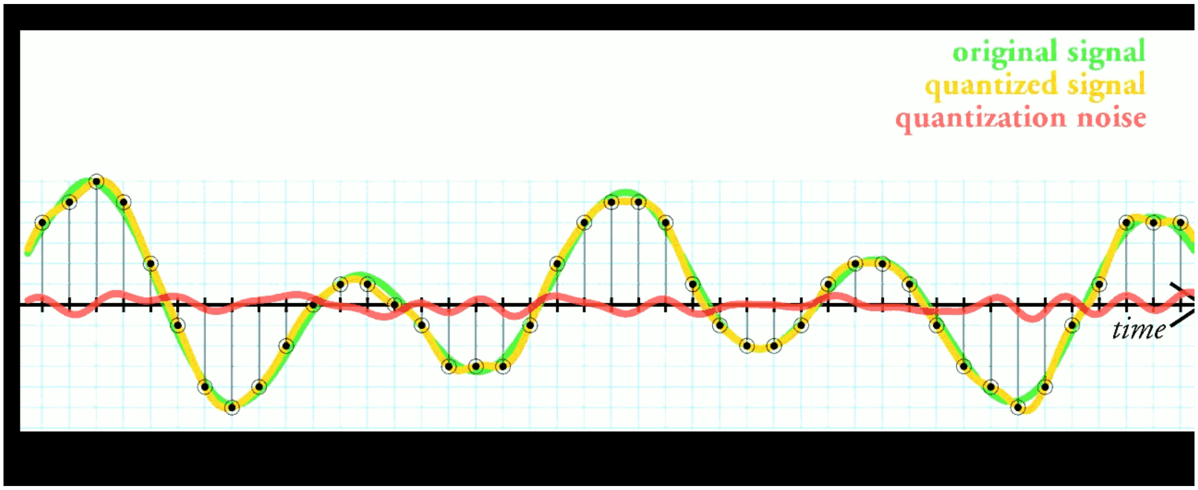
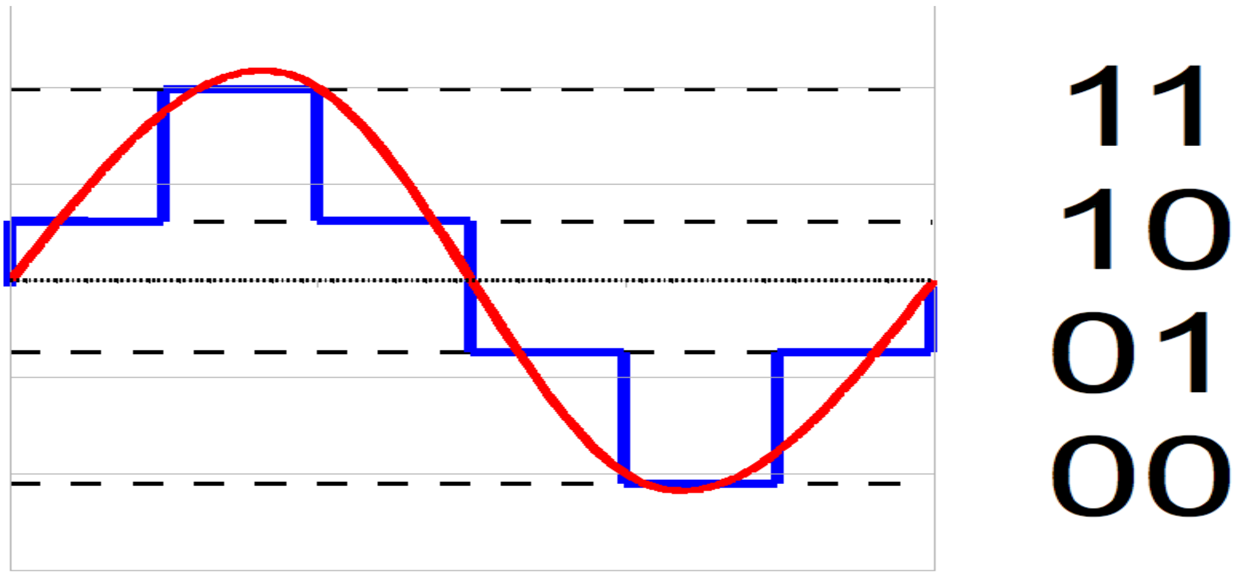
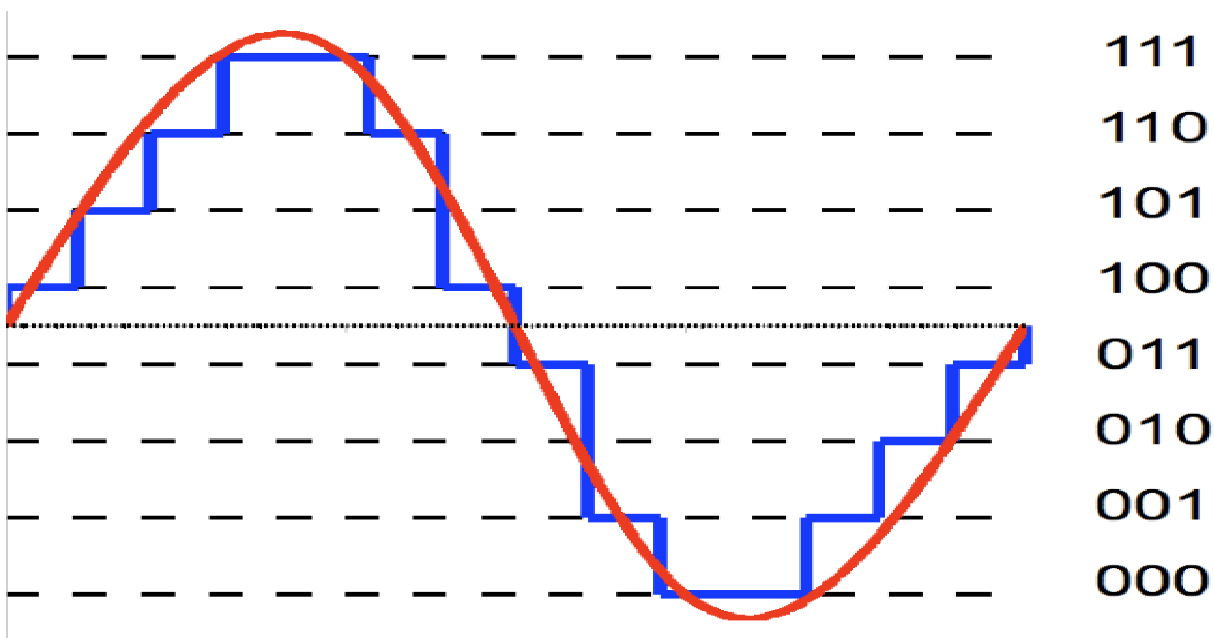


Figure 2.3: Quantization process (This example shows the original analog signal (green), the quantized signal (black dots), the signal reconstructed from the quantized signal (yellow) and the difference between the original signal and the reconstructed signal (red).)

Figure 2.4 shows quantization signal for different resolution.



(a) 2-bit resolution with four levels of quantization



(b) 3-bit resolution with four levels of quantization

Figure 2.4: Different bit resolution with different level of quantization compared to analog

2.5 Encoding

The quantized sample values need to be encoded as bit patterns. Some systems assign positive and negative values to samples, some just shift the curve to the positive part and assign only positive values. In other words, some systems us an unsigned integer to represent a sample, while others use signed integers to do so. However, the signed integers don't have to be in two's complement, they can be sign and magnitude values. The leftmost bit is used to represent the sign (0 for positive values and 1 for negative values), and the rest of the bits are used to represent the absolute values.

The system needs to decide how many bits should be allocated for each sample. Although in the past only eight bits were assigned to sound samples, today 16, 24 or even 32 bits per sample is normal. The number of bits per sample is sometimes referred to as the bit depth.

Today the dominant standard for storing audio is MP3 (short for MPEG Layer3). This standard is a modification of the MPEG (Motion Picture Experts Group) compression method used for video. It uses 44,100 samples per second and 16 bits per sample. This result is a signal with a bit rate 705,600 bits per second, which is compressed using a compression method that discards information that cannot be detected by the human ear. This is called lossy compression, as opposed to loss-less compression.

Several standards for image encoding are in use. JPEG (Joint Photographic Experts Group) uses the True-color scheme, but compresses the image to reduce the number of bits. GIF (Graphic Interchange Format), on the other hand, uses the indexed color scheme. [4]

2.6 Decoding and Reconstruction

Reconstruction is the process of creating an analog voltage (or current) from samples. A digital-to-analog converter takes a series of binary numbers and recreates the voltage (or current) levels that corresponds to that binary number. Then this signal is filtered by a lowpass filter. This process is analogous to interpolating between points on a graph, but it can be shown that under certain conditions the original analog signal can be reconstructed exactly from its samples. Unfortunately, the conditions for exact reconstruction cannot be achieved in practice, and so in practice the reconstruction is an approximation to the original analog signal. [5]

So, the full block diagram of signal acquisition and reconstruction can be shown as Figure 2.5

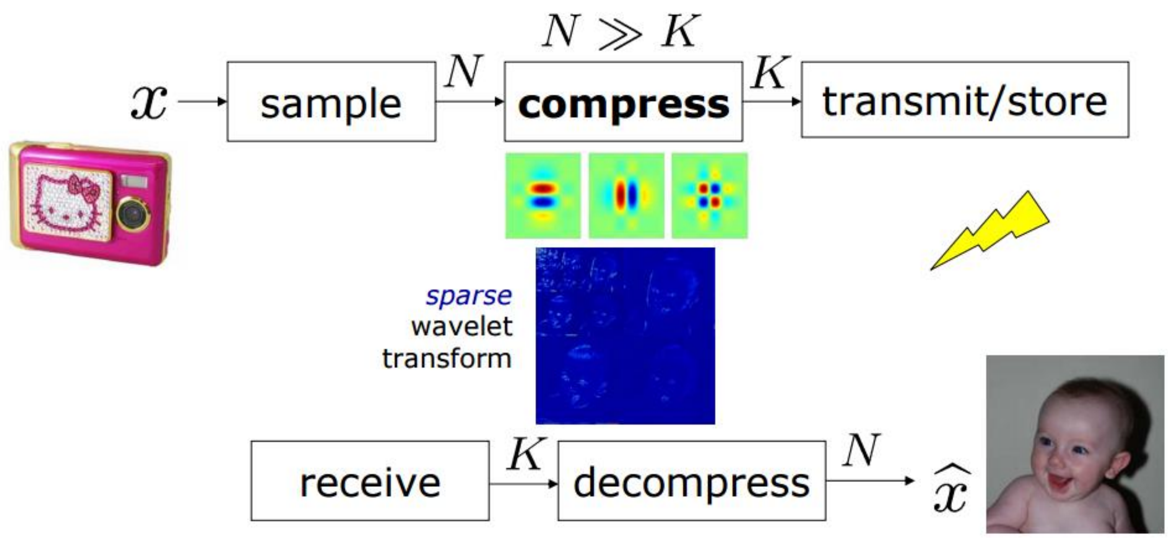


Figure 2.5: Signal acquisition and reconstruction exploiting Nyquist-Shannon theorem

CHAPTER 3

COMPRESSED SAMPLING AND ITS APPLICATIONS

3.1 Compressive Sensing

While the Nyquist-Shannon sampling theorem states that a certain minimum number of samples is required in order to perfectly capture an arbitrary band-limited signal; but when the signal is sparse in a known basis we can vastly reduce the number of measurements that need to be stored. This is the fundamental idea behind CS: rather than first sampling at a high rate and then compressing the sampled data, we would like to find ways to directly sense the data in a compressed form, at a lower sampling rate. CS differs from classical sampling in three important respects. First, sampling theory typically considers infinite length, continuous-time signals. In contrast, CS is a mathematical theory focused on measuring finite-dimensional vectors in R^N. Second, rather than sampling the signal at specific points in time, CS systems typically acquire measurements in the form of inner products between the signal and more general test functions. Thirdly, the two frameworks differ in the manner in which they deal with signal recovery. In the Nyquist-Shannon framework, signal recovery is achieved through sinc interpolation - a linear process that requires little computation and has a simple interpretation. In CS, however, signal recovery from the compressive measurements is typically achieved using highly nonlinear methods.

The block diagram for signal processing using compressive sensing is shown in figure 3.1. The scene under observation is capture using some sensing matrix, which maps the signal from N-dimensional space to M-dimensions, where M<<N. Thus, it captures the signal in a compressed form, rather than sampling at Nyquist rate and then compressing. Finally, the M dimensional data needs to be reconstructed back to the N dimensional space using efficient reconstruction algorithms. [6]

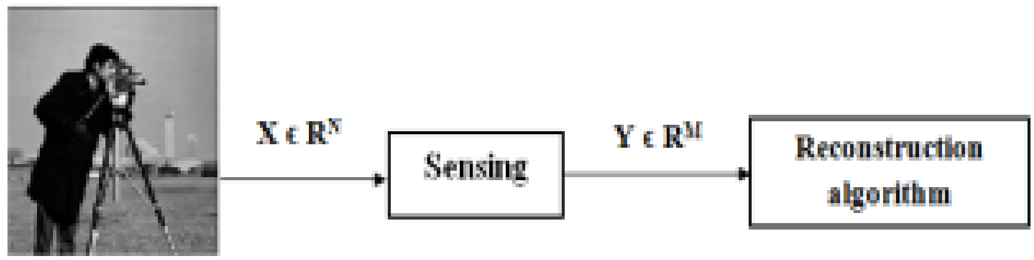


Figure 3.1: Block Diagram for Compressive Sensing

In mathematical terms, the observed data $y \epsilon C^m$ is connected to the signal $x \epsilon C^N$ of interest via

$$Ax = y.$$

The matrix $A \epsilon C^{MxN}$ models the linear measurement (information) process. Then one tries to recover the vector $x \epsilon C^N$by solving the above linear system. Traditional wisdom suggests that the number m of measurements, i.e., the amount of measured data, must be at least as large as the signal length N (the number of components of x). This principle is the basis for most devices used in current technology, such as analog-to-digital conversion, medical imaging, radar, and mobile communication. Indeed, if m < N, then classical linear algebra indicates that the linear system is underdetermined and that there are infinitely many solutions (provided, of course, that there exists at least one). In other words, without additional information, it is impossible to recover x from y in the case m < N. This fact also relates to the Shannon sampling theorem, which states that the sampling rate of a continuous-time signal must be twice its highest frequency in order to ensure reconstruction. Thus, it came as a surprise that under certain assumptions it is actually possible to reconstruct signals when the number m of available measurements is smaller than the signal length N. Even more surprisingly, efficient algorithms do exist for the reconstruction. The underlying assumption which makes all this possible is sparsity. The research area associated to this phenomenon has become known as compressive sensing, compressed sensing, compressive sampling, or sparse recovery. [7] Cs relies on two principles: *sparsity*, which pertains to the signals of interest, and *incoherence*, which pertains to the sensing modality. [8]

3.2 Sparsity

A signal is called sparse if most of its components are zero. As empirically observed, many real-world signals are compressible in the sense that they are well approximated by sparse signals—often after an appropriate change of basis. This explains why compression techniques such as JPEG, MPEG, or MP3 work so well in practice. For instance, JPEG relies on the sparsity of images in the discrete cosine basis or wavelet basis and achieves compression by only storing the largest discrete cosine or wavelet coefficients. The other coefficients are simply set to zero. Refer to Fig. 3.2 for an illustration of the fact that natural images are sparse in the wavelet domain. [7]

Figure 3.2: Antonella, Niels, and Paulina. Top: Original Image. Bottom: Reconstruction using 1% of the largest absolute wavelet coefficients, i.e., 99 % of the coefficients are set to zero [7]

Let us consider again the acquisition of a signal and the resulting measured data. With the additional knowledge that the signal is sparse or compressible, the traditional approach of taking at least as many measurements as the signal length seems to waste resources: At first, substantial efforts are devoted to measuring all entries of the signal and then most coefficients are discarded in the compressed version. Instead, one would want to acquire the compressed version of a signal “directly” via significantly fewer measured data than the signal length—exploiting the sparsity or compressibility of the signal. In other words, we would like to compressively sense a compressible signal! This constitutes the basic goal of compressive sensing.

We emphasize that the main difficulty here lies in the locations of the nonzero entries of the vector **X** not being known beforehand. If they were, one would simply reduce the matrix **A** to the columns indexed by this location set. The resulting system of linear equations then becomes overdetermined and one can solve for the nonzero entries of the signal. Not knowing the nonzero locations of the vector to be reconstructed introduces some nonlinearity since s-sparse vectors (those having at most s nonzero coefficients) form a nonlinear set. Indeed, adding two s-sparse vectors gives a 2s-sparse vector in general. Thus, any successful reconstruction method will necessarily be nonlinear.

Intuitively, the complexity or “intrinsic” information content of a compressible signal is much smaller than its signal length (otherwise compression would not be possible). So one may argue that the required amount of data (number of measurements) should be proportional to this intrinsic information content rather than the signal length. Nevertheless, it is not immediately clear how to achieve the reconstruction in this scenario.

Looking closer at the standard compressive sensing problem consisting in the reconstruction of a sparse vector $x \epsilon C^N$ from undetermined measurements **y=Ax** $\epsilon\ C^m, m < N$, one essentially identifies two questions:

- How should one design the linear measurement process? In other words, what matrices $A \epsilon\ C^{mxN}$ are suitable.
- How can we reconstruct x from y=Ax? In other words, what are efficient reconstruction algorithms?

These two questions are not entirely independent, as the reconstruction algorithm needs to take **A** into account, but we will see that one can often separate the analysis of the matrix **A** from the analysis of the algorithm.

Let us notice that the first question is by far not trivial. In fact, compressive sensing is not fitted for arbitrary matrices $A \epsilon C^{mxN}$. For instance, if A is made of rows of the identity matrix, then y = Ax simply picks some entries of x, and hence, it contains mostly zero entries. In particular, no information is obtained about the nonzero entries of x not caught in y, and the reconstruction appears impossible for such a matrix A. Therefore, compressive sensing is not only concerned with the recovery algorithm—the first question on the design of the measurement matrix is equally important and delicate. We also emphasize that the matrix A should ideally be designed for all signals x simultaneously, with a measurement process which is non-adaptive in the sense that the type of measurements for the datum y_j (i.e. j th row of A) does not depend on the previously observed data $y_1, \dots \dots . y_{j-1}$ As it turns out, adaptive measurements do not provide better theoretical performance in general. [7]

Mathematically speaking, we have a vector $f \epsilon R^N$ (such as the n-pixel) which we expand in an orthonormal basis (such as a wavelet basis) ψ = [$\Psi_1, \Psi_{2\dots\dots}$] as follows:

$$f(t) = \sum_{i=1}^{n} x_i \Psi_i(t) \quad \dots\dots\dots\dots\dots\dots\dots\dots.. (1)$$

where x is the coefficient sequence of f, $x_i = \langle f, \Psi_i \rangle$. It will be convenient to express f as Ψx (where Ψ is the nxn matrix). The implication of sparsity is now clear: when a signal has a sparse expansion, one can discard the small coefficients without much perceptual loss. Formally, consider $f_s(t)$ obtained by keeping only the terms corresponding to the S largest values of (x_i) in the expansion (1). By definition, $f_s := \Psi x_s$ where here and below, x_s is the vector of coefficients (x_i) with all but the largest S set to zero. This vector is sparse in a strict sense since all but a few of its entries are zero; we will call S-sparse such objects with at most S nonzero entries.

3.3 Incoherence

Incoherence extends the duality between time and frequency and expresses the idea that objects having a sparse representation in Ψ must be spread out in the domain in which they are acquired, just as a Dirac or a spike in the time domain is spread out in the frequency domain. Put differently, incoherence says that unlike the signal of interest, the sampling/sensing waveforms have an extremely dense representation in Ψ.

Suppose we are given a pair (ϕ, Ψ) of orthobases of R^N The first basis Φ is used for sensing the object f and the second is used to represent f. The restriction to pairs of orthobases is not essential and will merely simplify our treatment.

The coherence between the sensing basis ϕ and the representation basis Ψ is:

$$\mu(\Phi, \Psi) = \sqrt{n} \,.\, \max_{1 \le k, j \le n} \left| \langle \varphi_k \,,\ \Psi_j \rangle \right| \ldots\ldots\ldots\ldots\ldots (2)$$

In plain English, the coherence measures the largest correlation between any two elements of $\phi \; and \; \psi$. If $\phi \; and \; \psi$ contain correlated elements, the coherence is large. Otherwise, it is small. As for how large and how small, it follows from linear algebra that $\mu(\Phi, \Psi) \in [1, \sqrt{n}]$.

Compressive sampling is mainly concerned with low coherence pairs, and we now give examples of such pairs. In our first example, Φ is the canonical or spike basis $\varphi_k(t) = \delta(1 - k)$ and ψ is the Fourier basis, $\psi_j(t) = n^{-1/2} e^{i2\pi jt/n}$. Since, ϕ is the sensing matrix, this corresponds to the classical sampling scheme in time or space. The time-frequency pair obeys $\mu(\phi, \psi) = 1$ and, therefore, we have maximal incoherence. [8]

3.4 Design Sensing Matrix

The sensing mechanisms collect information about a signal x(t) by linear functional recordings,

$$y_k = \langle x, a_k \rangle \; k = 1,2,3 \ldots, M \quad \ldots\ldots\ldots (3)$$

That is, we simply correlate the object we wish to acquire with the waveforms $a_k(t)$. This is a standard setup. If the sensing waveforms are Dirac delta functions (spikes), for example, then y is a vector of sampled values of x in the time or space domain. If the sensing waveforms are sinusoids, then y is a vector of Fourier coefficients; this is the sensing modality used in magnetic resonance imaging (MRI).

There are two main theoretical questions in CS. First, how should we design the sensing matrix **A** to ensure that it preserves the information in the signal x? Second, how can we recover the original signal x from measurements y? In the case where our data is sparse or compressible, we will see that we can design matrices A with M<< N that ensure that we will be able to recover the original signal accurately and efficiently using a variety of practical algorithms. To recover a unique k -sparse vector (a vector with allmost k < N nonzero entries) x; restrictions are imposed on **A** like satisfying the Null Space Property (NUS), the Restricted Isometry Property (RIP) and or some desired Coherence. [6] So, design process has 2 major steps:

- Selecting the number of measurements ‘M’ to be taken
- Selecting the type of matrix that satisfies RIP and other necessary conditions

Sensing or measurement matrices are mainly two types:

- ➢ Random Matrix
- ➢ Deterministic Matrix

3.4.1 Random Sensing Matrices

Three types of random matrices are mainly used as sensing matrix:

1. Gaussian Independent Identically Distribution (i.i.d.) matrix
2. Uniform random ortho projections matrix
3. Bernoulli matrix

These matrices are used in medical image processing, geophysical data analysis, communications and other various signal processing problems. Although these matrices also show very high probability in reconstruction, they also have many drawbacks such as excessive complexity in reconstruction, significant space requirement for storage, and no efficient algorithm to verify whether a sensing matrix satisfies RIP property with small RIC value. Hence, exploiting specific structures of deterministic sensing matrices is required to solve these problems of the random sensing matrices.

3.4.2 Deterministic Sensing Matrices

Recently, several deterministic sensing matrices have been proposed. We can classify them into two categories. First are those matrices which are based on coherence. Second are those matrices which are based on RIP or some weaker RIPs. Some proposed deterministic matrices are:

- **Chirp Sensing Matrices**

  For this type, we can get 2×4 deterministic sensing matrix A as

$$A_{\text{Chirp}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{i\pi} & e^{i\pi} & e^{i\pi + i2\pi} \end{bmatrix}$$

- **Second order reed-muller sensing matrices**

  For this type, we can get 4 × 8 deterministic sensing matrix A as

$$A_{RM} = \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

- **Binary BCH matrices**

  For this type, we can get 4 × 8 deterministic sensing matrix A as

$$A_{bin} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

The advantages of these matrices, in addition to their deterministic constructions, are the simplicity in sampling and recovery process as well as small storage requirement. The only disadvantage is that a priori information on location of nonzero components should be known.

3.5 Applications of compressive sensing

There are several fields of interest where compressive sensing technology can be used. Here only two fields will be briefly discussed and others will be named only.

3.5.1 Single-Pixel Camera

Compressive sensing techniques are implemented in a device called the single-pixel camera. The idea is to correlate in hardware a real-world image with independent realizations of Bernoulli random vectors and to measure these correlations (inner products) on a single pixel. It suffices to measure only a small number of such random inner products in order to reconstruct images via sparse recovery methods.

For the purpose of this exposition, images are represented via gray values of pixels collected in the vector $z \epsilon R^N$, where $N = N_1N_2$ and N_1, N_2 the width and height of the image in pixels. Images are not usually sparse in the canonical (pixel) basis, but they are often sparse after a suitable transformation, for instance, a wavelet transform or discrete cosine transform. This means that one can write $z = Wx,$ where $x \epsilon R^N$ is a sparse or compressible vector and $W \epsilon R^{N\times N}$ is a unitary matrix representing the transform.

The crucial ingredient of the single-pixel camera is a microarray consisting of a large number of small mirrors that can be turned on or off individually. The light from the image is reflected on this microarray and a lens combines all the reflected beams in one sensor, the single pixel of the camera, shown in figure 3.3.

Although the single-pixel camera is more a proof of concept than a new trend in camera design, it is quite conceivable that similar devices will be used for different imaging tasks. In particular, for certain wavelengths outside the visible spectrum, it is impossible or at least very expensive to build chips with millions of sensor pixels on an area of only several square millimeters. In such a context, the potential of a technology based on compressive sensing is expected to really pay off. [7]

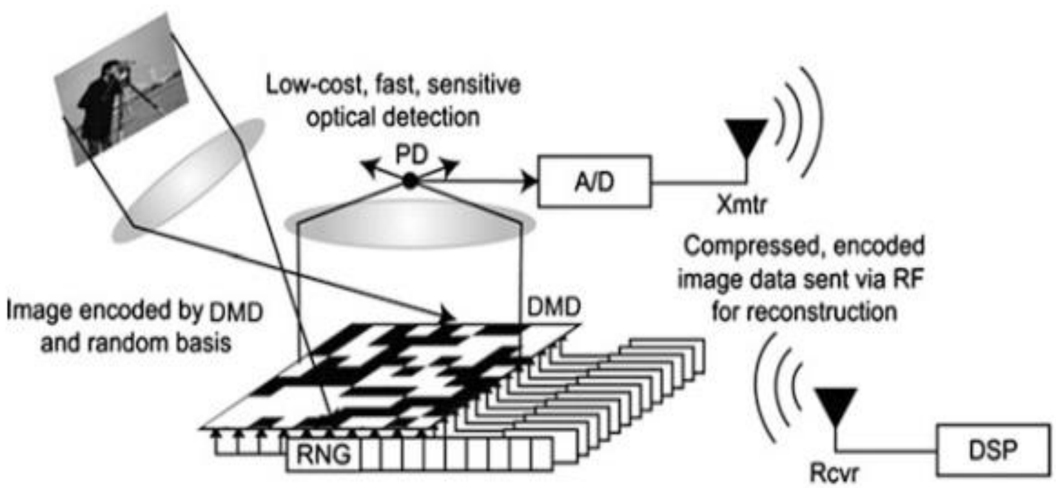


Figure 3.3: Schematic representation of a single-pixel camera (Image courtesy of Rice University) [7]

3.5.2 Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) is a common technology in medical imaging used for various tasks such as brain imaging, angiography (examination of blood vessels), and dynamic heart imaging. In traditional approaches (essentially based on the Shannon sampling theorem), the measurement time to produce high-resolution images can be excessive (several minutes or hours depending on the task) in clinical situations. For instance, heart patients cannot be expected to hold their breath for too long a time, and children are too impatient to sit still for more than about two minutes. In such situations, the use of compressive sensing to achieve high resolution images based on few samples appears promising.

MRI relies on the interaction of a strong magnetic field with the hydrogen nuclei (protons) contained in the body's water molecules. A static magnetic field polarizes the spin of the protons resulting in a magnetic moment. Applying an additional radio frequency excitation field produces a precessing magnetization transverse to the static field. The precession frequency depends linearly on the strength of the magnetic field. The generated electromagnetic field can be detected by sensors. Imposing further magnetic fields with a spatially dependent strength, the precession frequency depends on the spatial position as well. Exploiting the fact that the transverse magnetization depends on the physical properties of the tissue (for instance, proton density) allows one to reconstruct an image of the body from the measured signal. [7]

CHAPTER 4

RECONSTRUCTION ALGORITHMS

4.1 Types

The algorithms are divided into three categories:

- Optimization Methods
- Greedy Methods
- Thresholding-based Methods

4.2 Optimization Methods

An optimization problem is a problem of the type

$$\text{minimize } F_0(x) \qquad \text{subject to } F_i(x) \leq b_i, i \in [n]$$

Where the function $F_0 : R^N \rightarrow R$ is called an objective function and the functions $F_1, F_2, \ldots F_n : R^N \rightarrow R$ are called constraint functions. This general framework also encompasses equality constraints of the type $G_i(x) = c_i$ and $- G_i(x) = - c_i$. If $F_1, F_2, \ldots. F_n$ are all convex functions, then the problem is called a convex optimization problem. .If $F_1, F_2, \ldots. F_n$ are all linear functions, then the problem is called a linear program. Our sparse recovery problem is in fact an optimization problem, since it translates into

$$\text{minimize } \|z\|_0 \qquad \text{subject to } Az = y$$

This is a nonconvex problem and it is NP-hard in general. However, keeping in mind that $\|z\|_q^q$ approaches $\|z\|_0$ as q>0

$$\text{minimize } \|z\|_q \qquad \text{subject to } Az = y$$

For q>1, even 1-sparse vectors are not solutions of (P_q). For 0<q<1, (P_q) is again a nonconvex problem, which is also NP-hard in general. But for the critical value q = 1, it becomes the following convex problem:

$$\text{minimize } \|z\|_1 \qquad \text{subject to } Az = y$$

This principle is usually called l_1-minimization or basis pursuit. There are several specific algorithms to solve the optimization problem.

<u>Basis Pursuit</u>

Input: measurement matrix A, measurement vector y.

Instruction:

$$x^{\#} = argmin \, \|z\|_1 \qquad \text{subject to } Az = y$$

Output: the vector $x^{\#}$

<u>Quadratically Constrained Basis Pursuit</u>

Input: measurement matrix A, measurement vector y, noise level n

Instruction:

$$x^{\#} = argmin \, \|z\|_1 \qquad \text{subject to } \|Az-y\|_2 \leq n$$

Output: the vector $x^{\#}$

4.3 Greedy Methods

In this section, we introduce two iterative greedy algorithms commonly used in compressive sensing. The first algorithm, called Orthogonal Matching Pursuit (OMP), adds one index to a target support S^n at each iteration and updates a target vector x^n as the vector supported on the target support S^n that best fits the measurements. The algorithm is formally described as follows:

<u>Orthogonal Matching Pursuit (OMP)</u>

Input: measurement matrix A, measurement vector y.

Initialization: $S^0 = \phi$, $x^0 = 0$

Iteration: repeat until a stopping criterion is met at $n = \bar{n}$:

$$S^{n+1} = S^n \cup \{j_{n+1}\}, \quad j_{n+1} := \underset{j \in [N]}{argmax} \{|(A * (y - Ax^n))_j|\}, \qquad (\text{OMP})_1$$

$$x^{n+1} = \underset{z \in C^N}{argmin} \{||y - Az||_2, supp(z) \subset S^{n+1}\} \quad (OMP)_2$$

Output: the $\bar{n}$ sparse vector $x^{\#} = x^{\bar{n}}$

A weakness of the orthogonal matching pursuit algorithm is that, once an incorrect index has been selected in a target support S^n, it remains an all the subsequent target supports S^n for $n^{/} \geq n$. Hence, if an incorrect index has been selected, s iterations of the orthogonal matching pursuit are not enough to recover a vector with sparsity s. A possible solution is to increase the number of iterations. The following algorithm, called Compressive Sampling Matching Pursuit Algorithm, proposes another strategy when an estimation of the sparsity s is available.

Compressive Sampling Matching Pursuit (CoSaMP)

Input: measurement matrix A, measurement vector y, sparsity level s

Initialization: s-sparse vector x^0, typically $x^0 = 0$

Iteration: repeat until a stopping criterion is met at $n = \bar{n}$:

$$U^{n+1} = supp(x^n) \cup L_2(A * (y - Ax^n)) \quad (CoSaMP_1)$$

$$u^{n+1} = \underset{z \in C^N}{argmin} \{||y - Az||_2, supp(z) \subset U^{n+1}\} \quad (CoSaMP_2)$$

$$x^{n+1} = H_s(u^{n+1}) \quad (CoSaMP_3)$$

Output: the $\bar{n}$ sparse vector $x^{\#} = x^{\bar{n}}$

4.4 Thresholding-Based Methods

In this section, we describe further algorithms involving the hard thresholding operator H_s. The intuition for these algorithms, which justifies categorizing them in a different family, relies on the approximate inversion of the action on sparse vectors of the measurement matrix A by the action of its adjoint A^*. Thus, the basic thresholding algorithm consists in determining the support of the s-sparse vector $x \in C^N$ to be recovered from the measurement vector $y = Ax \in C^m$ as the indices of s largest absolute entries of

A*y and then in finding the vector with this support that best fits the measurement. Formally, the algorithm reads as follows.

Basic Thresholding

Input: measurement matrix A, measurement vector y, sparsity level s

Instruction:

$$S^{\#} = Ls(A^*y)$$

$$x^{\#} = \underset{z \in C^N}{argmin} \{||y - Az||_2, supp(z) \subset S^{\#}\}$$

Output: the s-sparse vector $x^{\#}$

The more elaborate iterative hard thresholding algorithm is an iterative algorithm to solve the rectangular system Az = y, knowing that the solution is s-sparse. We shall solve the square system A*Az = A*y instead, which can be interpreted as the fixed-point equation z = (Id – A*A)z + A*y. Classical iterative methods suggest the fixed-point iteration x^{n+1} = (Id – A*A)x^n + A*y. Since Since we target s-sparse vectors, we only keep the s largest absolute entries of (Id – A*A)x^n + A*y = x^n + A * (y-Ax^n) at each iteration. The resulting algorithm reads as follows.

Iterative Hard Thresholding (IHT)

Input: measurement matrix A, measurement vector y, sparsity level s

Initialization: Initialization: s-sparse vector x^0, typically $x^0 = 0$

Iteration: repeat until a stopping criterion is met at $n = \bar{n}$:

$$x^{n+1} = H_s(x^n + A * (y-Ax^n))$$

Output: the s-sparse vector $x^{\#} = x^{\bar{n}}$

The iterative hard thresholding algorithm does not require the computation of any orthogonal projection. If we are willing to pay the price of the orthogonal projections, like in the greedy methods, it makes sense to look at the vector with the same support as x^{n+1} that best fits the measurements. This leads to the hard thresholding pursuit algorithm defined below.

Hard Thresholding Pursuit (HTP)

Input: measurement matrix A, measurement vector y, sparsity level s

Initialization: Initialization: s-sparse vector x0, typically x0 = 0

Iteration: repeat until a stopping criterion is met at n = $\bar{n}$:

$S^{n+1} = L_s(x^n + A * (y - Ax^n))$

$x^{n+1} = \underset{z \in C^N}{argmin} \{||y - Az||_2, supp(z) \subset S^{n+1}\}$

Output: the s-sparse vector $x^{\#} = x^{\bar{n}}$

CHAPTER 5

PROPOSED METHODOLOGY FOR VOICE SIGNAL AND IMAGE RECONSTRUCTION

5.1 Introduction

Based on the theory about compressive sensing and reconstruction algorithm in the chapter 3 and chapter 4, two reconstruction algorithms have been developed by MATLAB programming. These algorithms can reconstruct the voice signals and images. The basic block diagram of proposed work is given below:

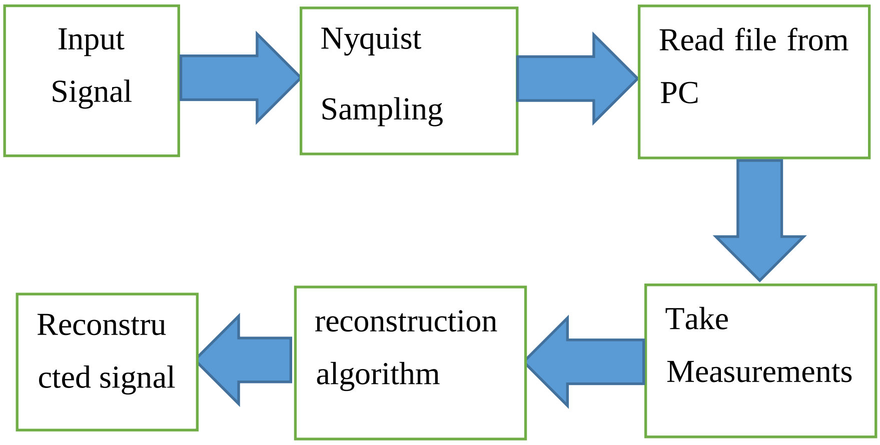


Figure 5.1: Basic flow chart of proposed work

5.2 Programming Approach

Based on the scope of thesis work, the concept of compressive sensing and its reconstruction has been tested on different voice signals and images. The Block diagram of voice signal and image reconstruction is shown below:

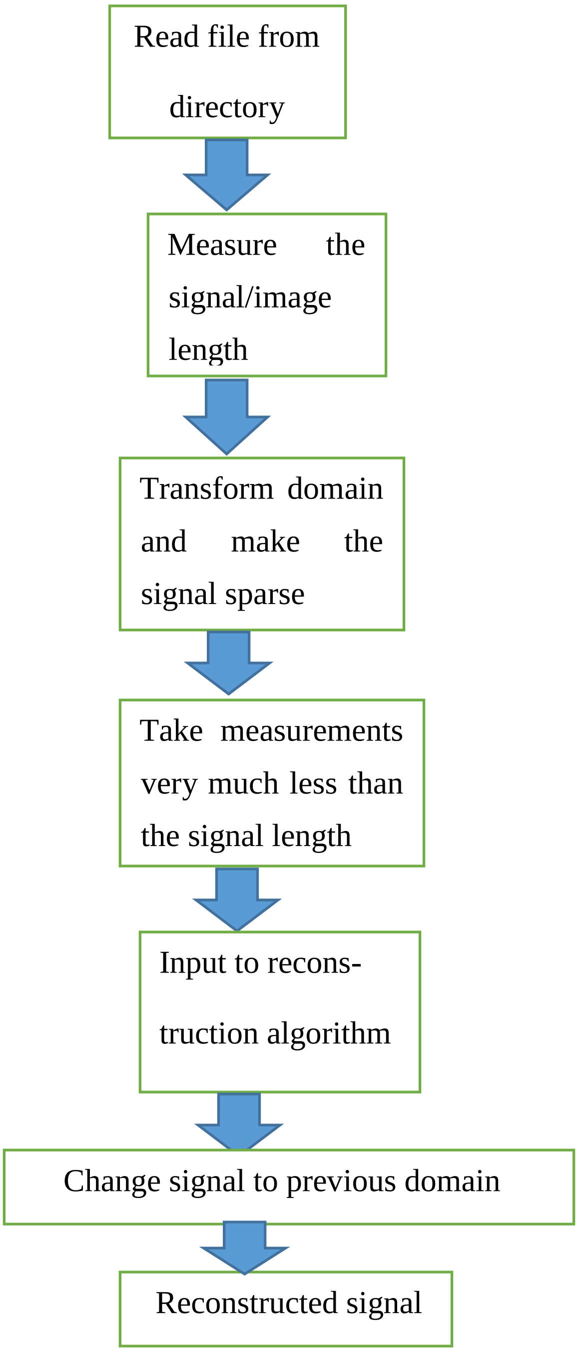


Figure 5.2: Flow chart of voice signal and image reconstruction

In the Figure 5.2, the method shown in the block diagram has been exploited for five different voice signals and five different images. Every time the signal / image read from the file which was samples in Nyquist rate. Then the representation domain has been changed (in this experiment time domain to frequency domain) to make it sparse. For voice signal, FFT (Fast Fourier Transform) has been used to change domain and DCT (Discrete Cosine Transform) has been used to change domain of image. Then 'm' measurements has been taken where m = 3*log(k); k= number of non-zero entries in sparse signal. Then two reconstruction algorithm- l_1 and Orthogonal Matching Pursuit (OMP) have been implemented to reconstruct the original signal. The reconstructed signal is in frequency domain. To change it to time domain we have exploited IFFT (Inverse Fast Fourier Transform) and IDCT (Inverse Discrete Cosine Transform) for voice signal and image respectively.

CHAPTER 6

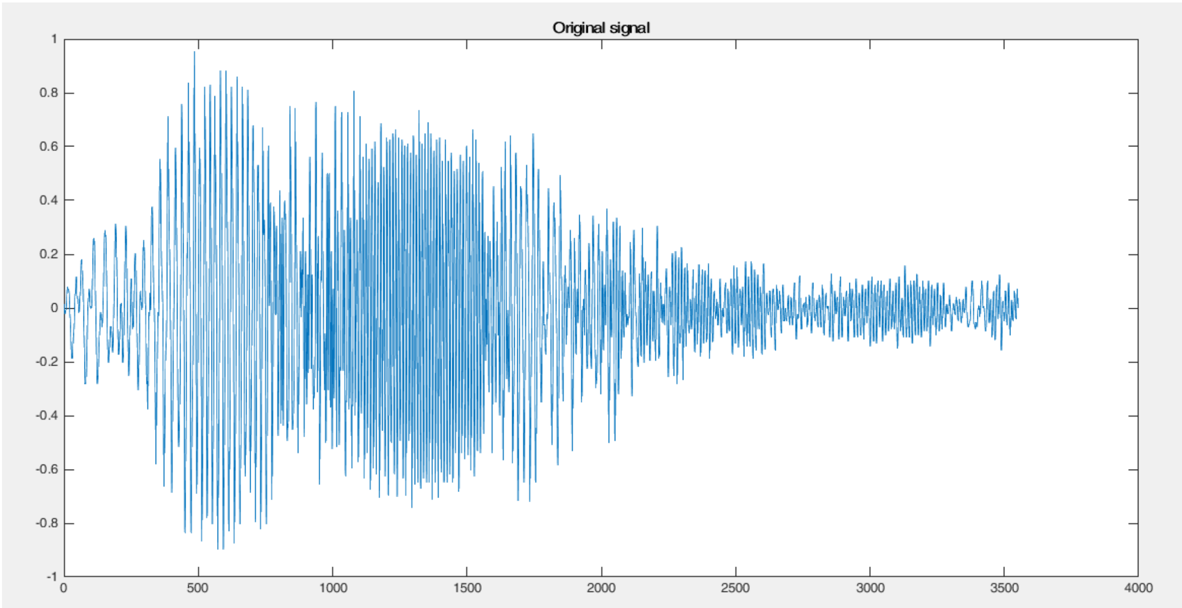
RESULT AND DISCUSSION

6.1 Introduction

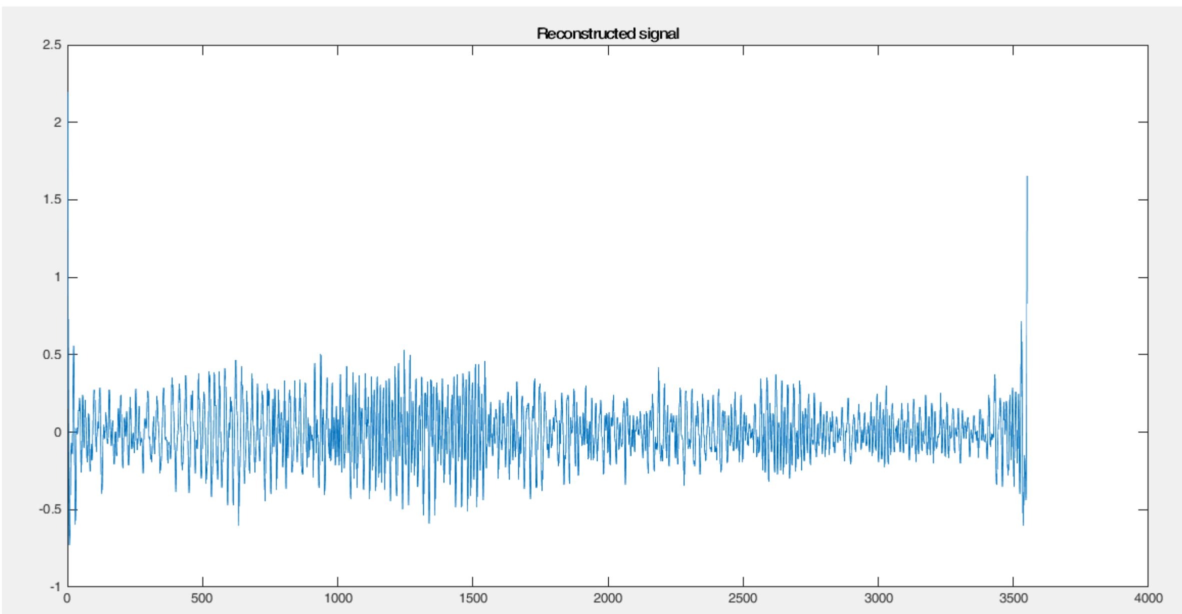
In this chapter, the results of different reconstruction algorithm associated with compressive sensing exploited in signal reconstruction will be displayed. The result consists of the voice signal reconstruction using l_1 minimization and Orthogonal Matching Pursuit (OMP) algorithm. We have used the same algorithms for image reconstruction.

6.2 Voice Signal Reconstruction

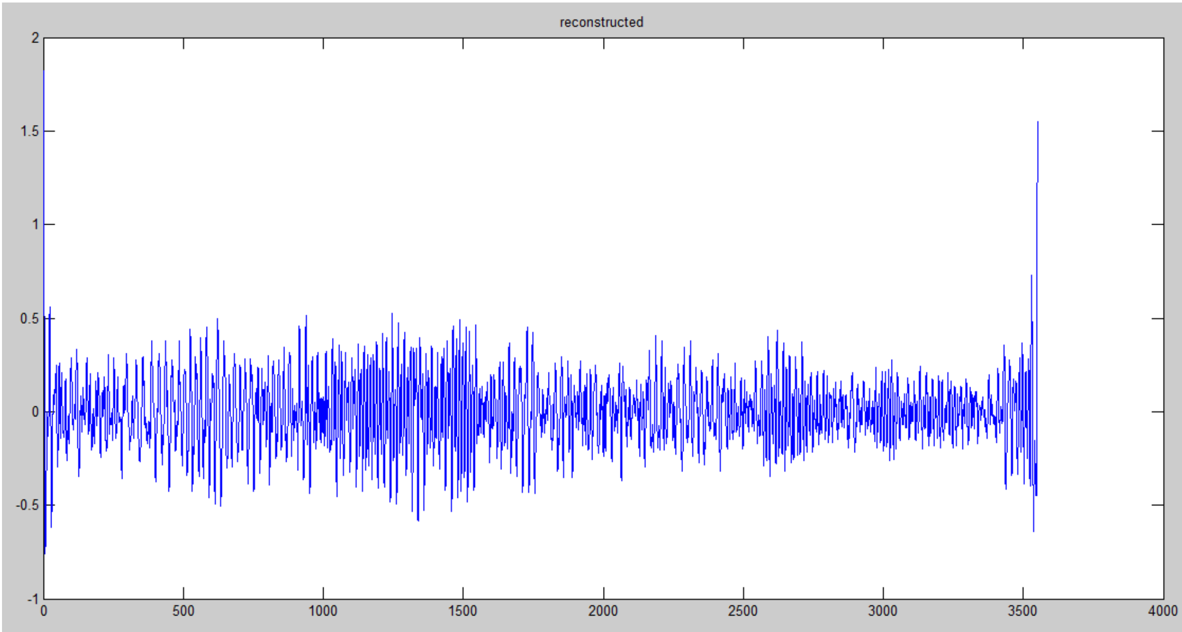
We have firstly read the voice signal file from computer. The original signal is shown in figure 5.1 ~ 5.5 (a). Then we have transformed it to the frequency domain using fast fourier transform (FFT). Then measurements have been taken from this sparse signal. After we used l_1 minimization algorithm to reconstruct the original signal shown in figure 5.1 ~ 5.5 (b). In another process, we have used OMP algorithm to reconstruct the original signal shown in figure 5.1 ~ 5.5 (c). Through figure 5.1 ~ 5.5 five different voice signals will be shown for original signal, reconstruction using l_1 minimization and reconstruction using OMP algorithm.



(a) Original signal

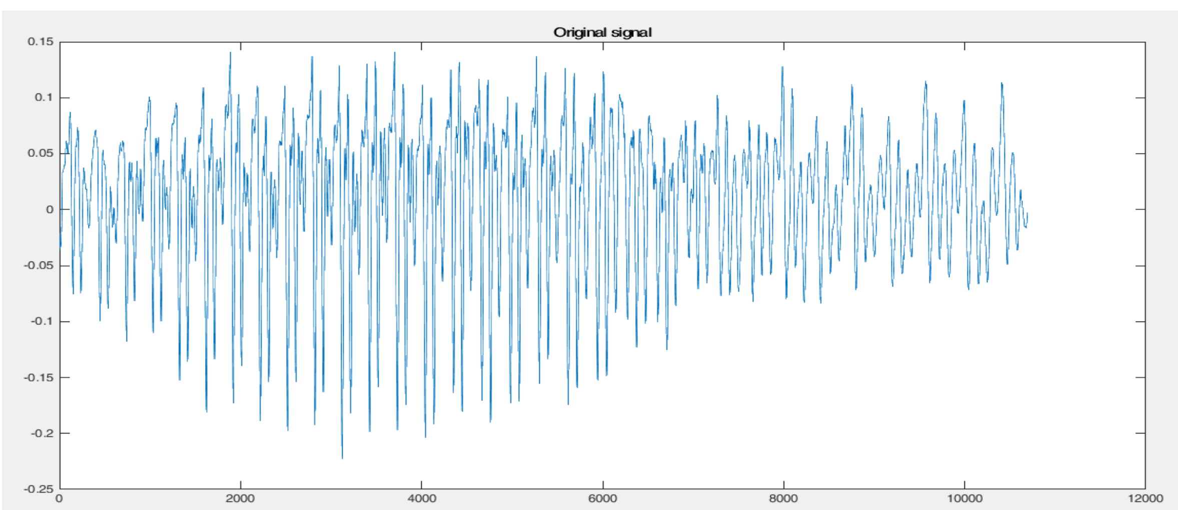


(b) Reconstructed signal using l_1 minimization

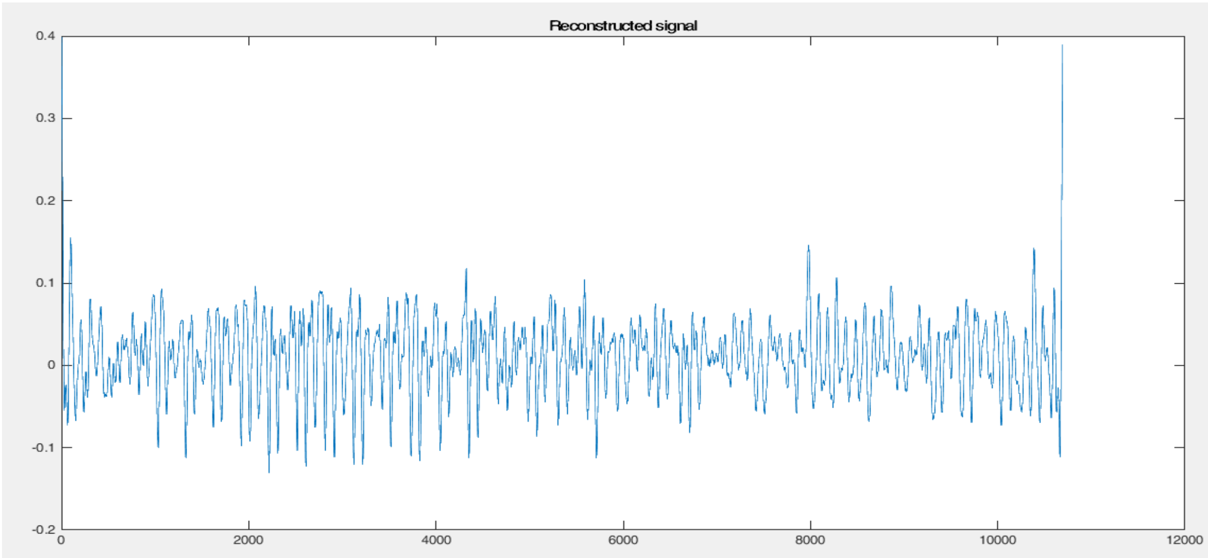


(c) Reconstructed signal using OMP

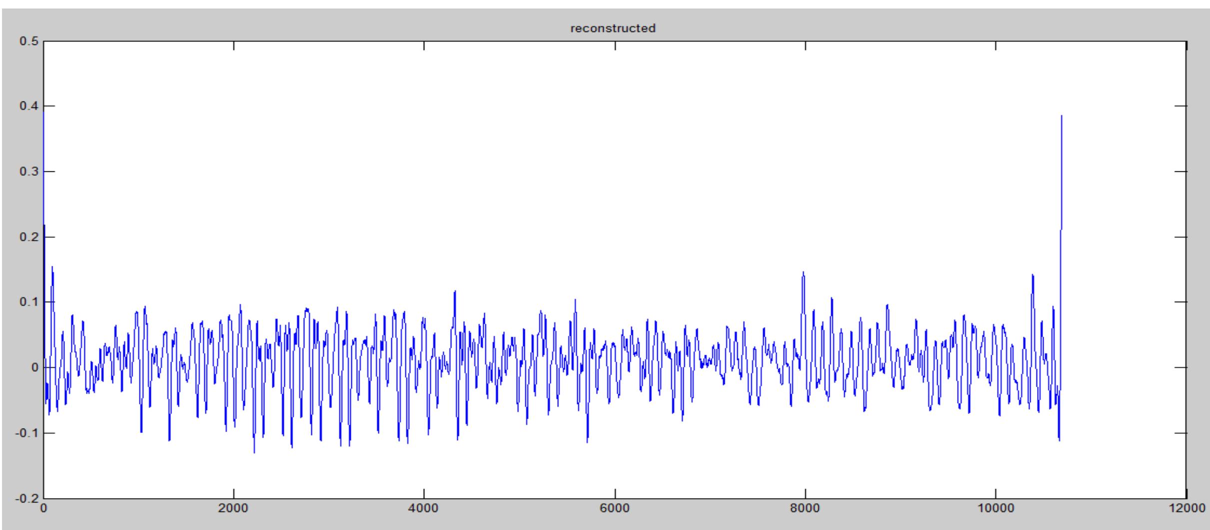
Figure 6.1: Original and reconstructed signals for ‘barking of a dog’



(a) Original signal

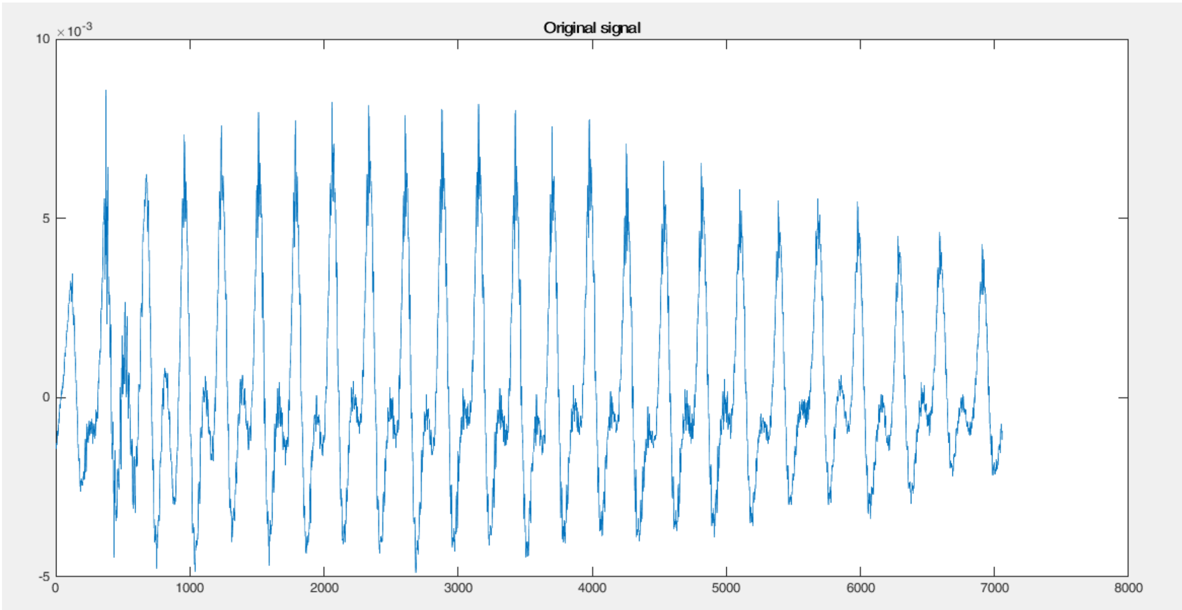


(b) Reconstructed signal using l_1 minimization

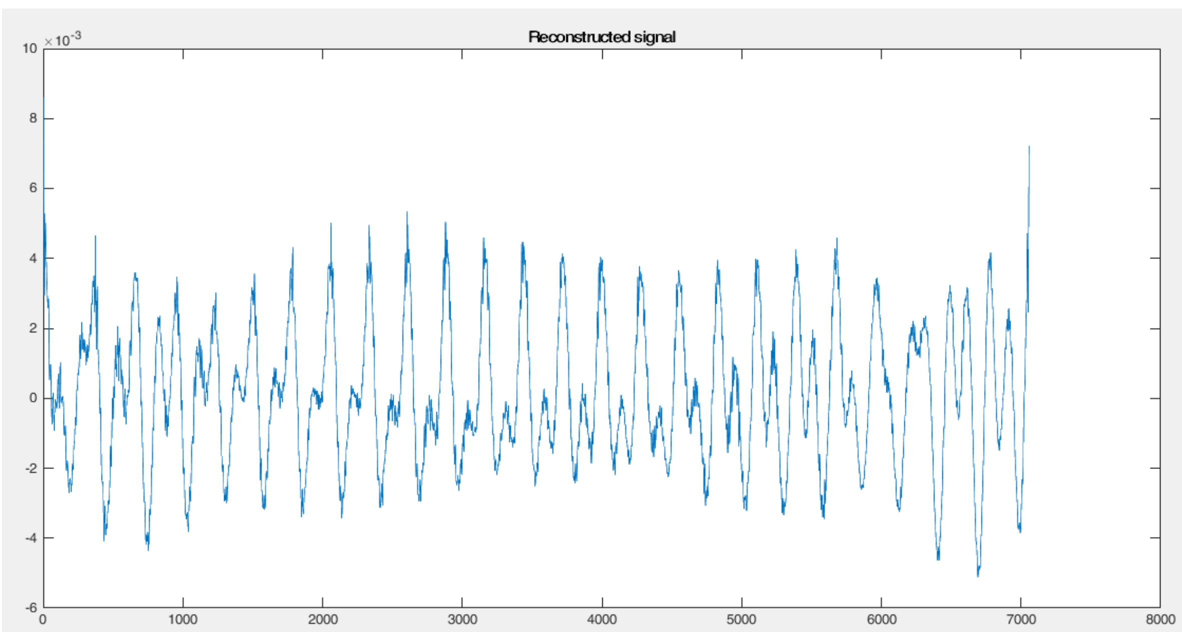


(c) Reconstructed signal using OMP

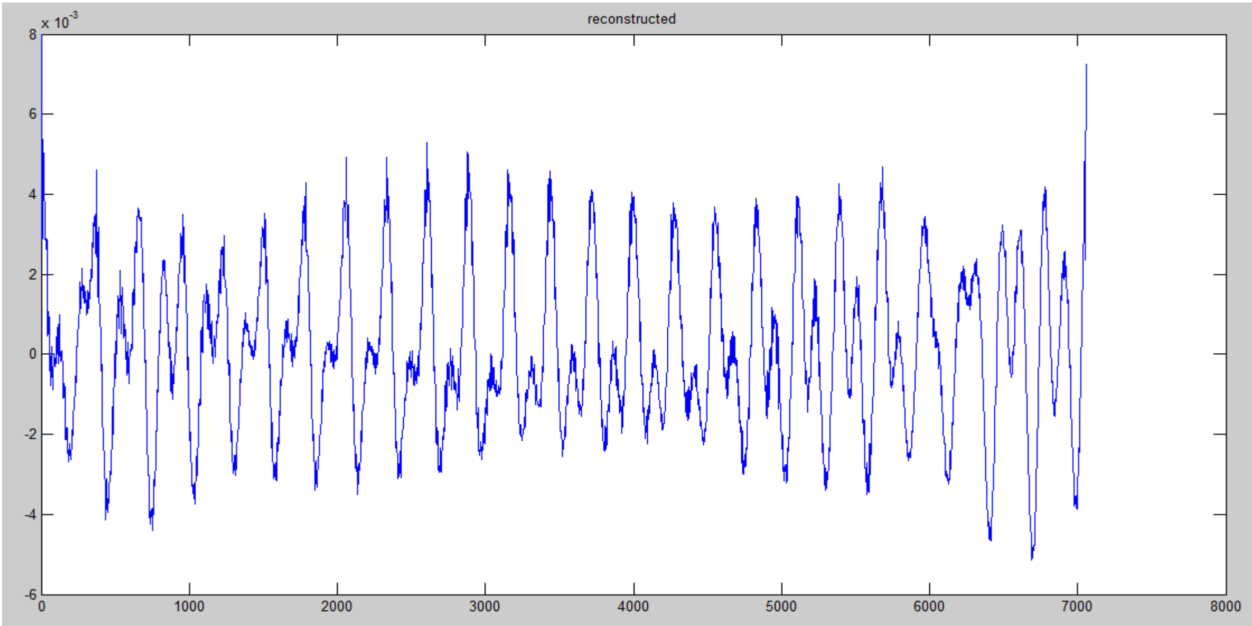
Figure 6.2: Original and reconstructed signals for 'Voice signal of Dipan'



(a) Original signal

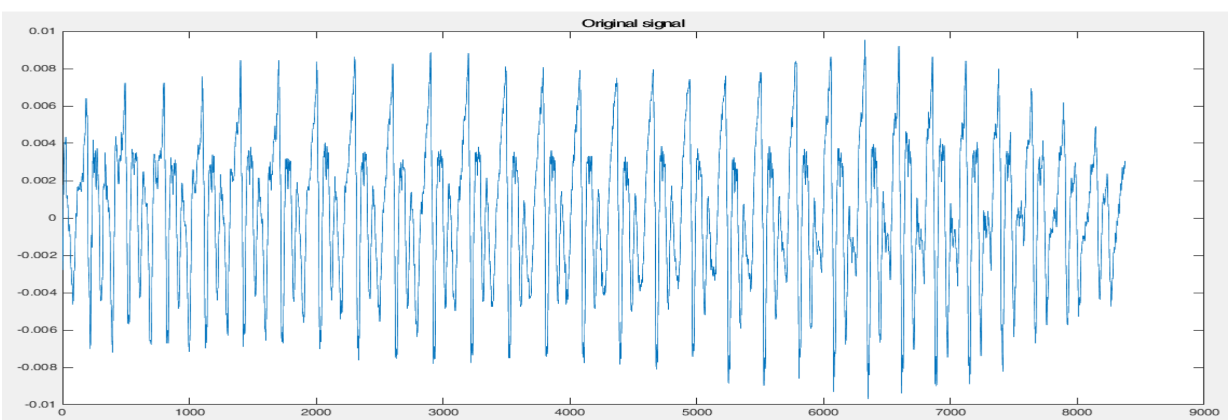


(b) Reconstructed signal using l_1 minimization

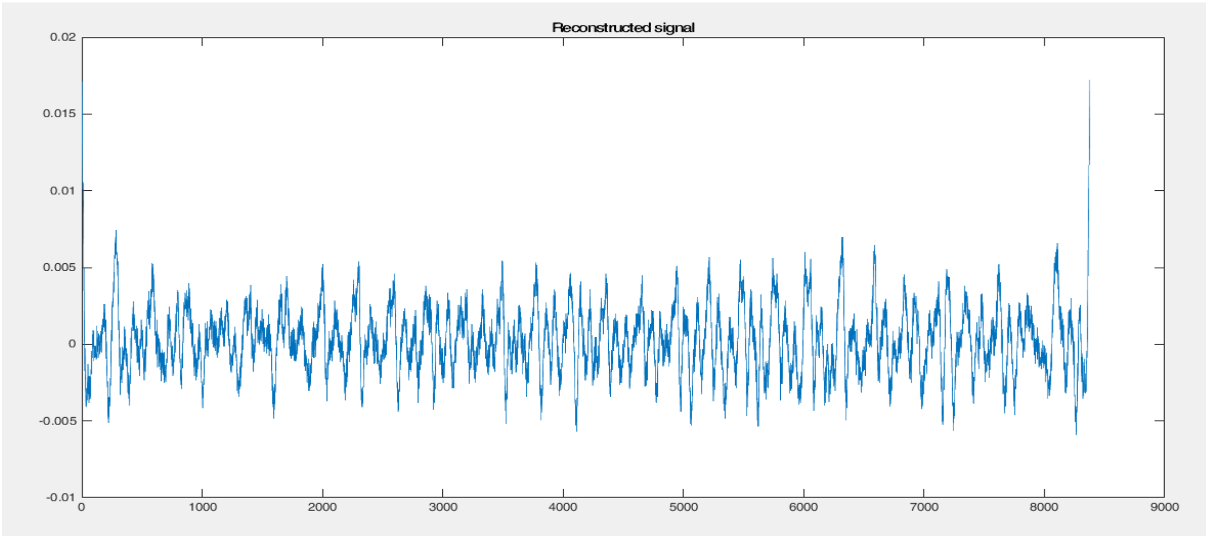


(c) Reconstructed signal using OMP

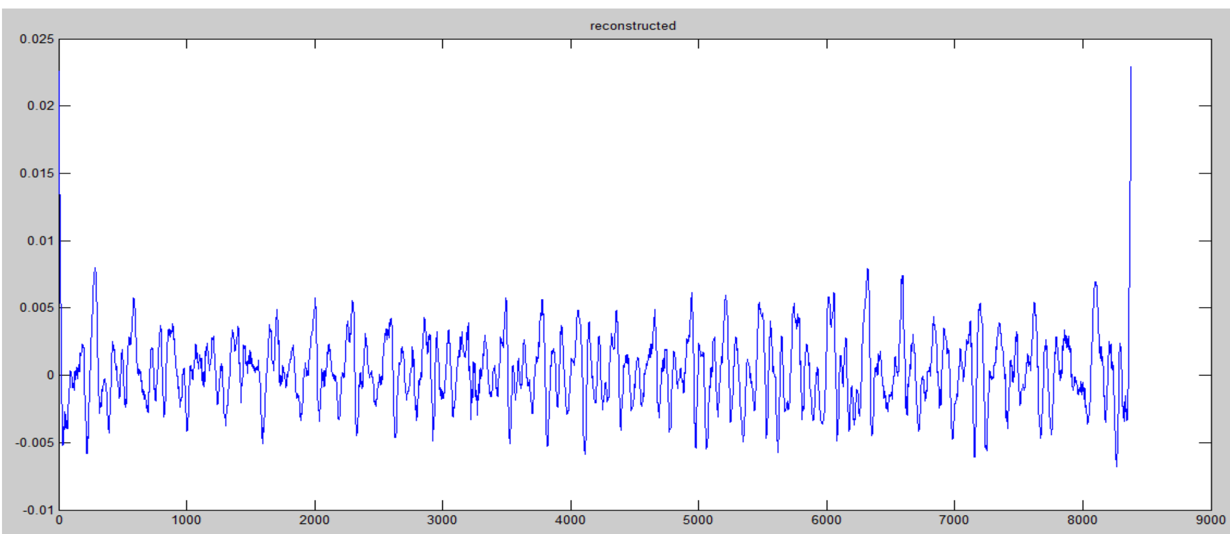
Figure 6.3: Original and reconstructed signals for ‘Voice signal of Hasan’. (a) (b) and (c)



(a) Original signal

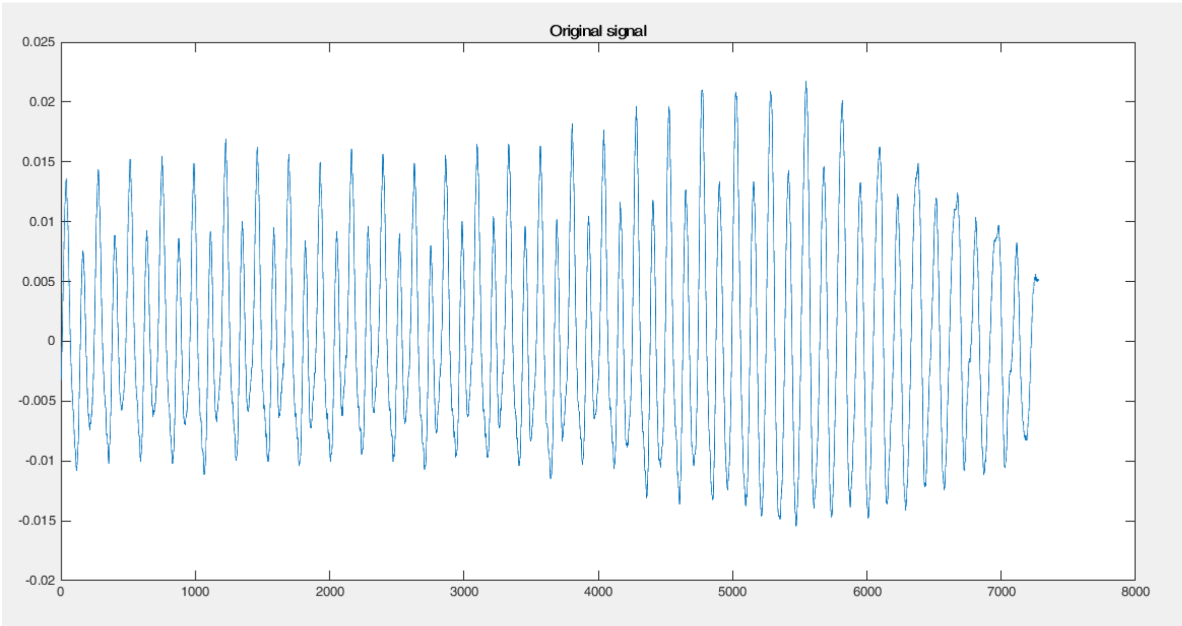


(b) Reconstructed signal using l_1 minimization

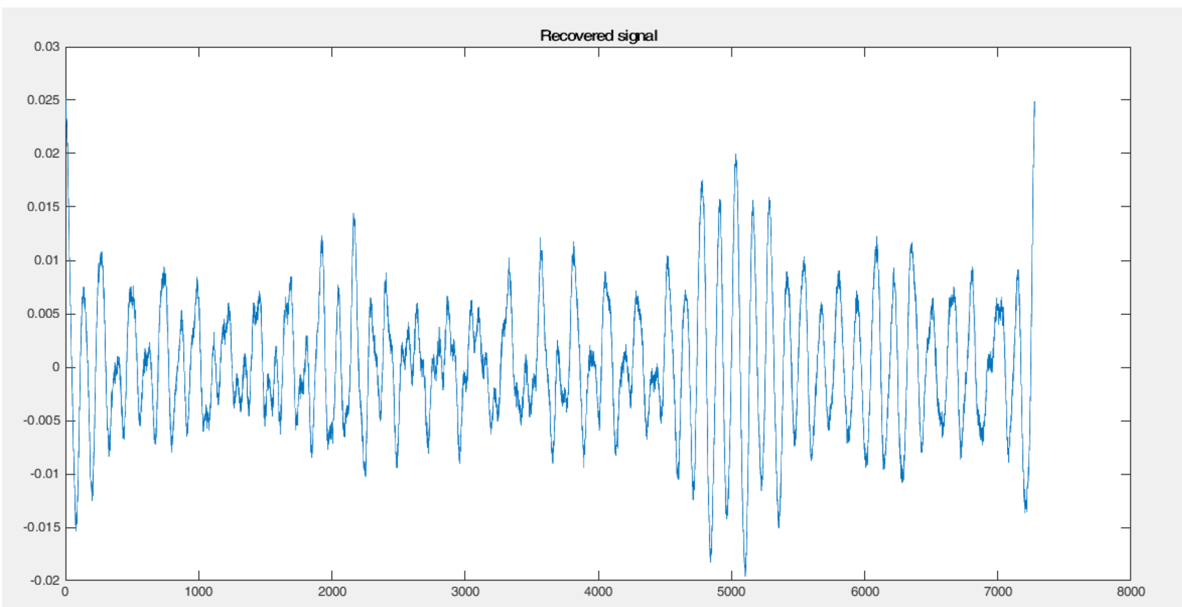


(c) Reconstructed signal using OMP

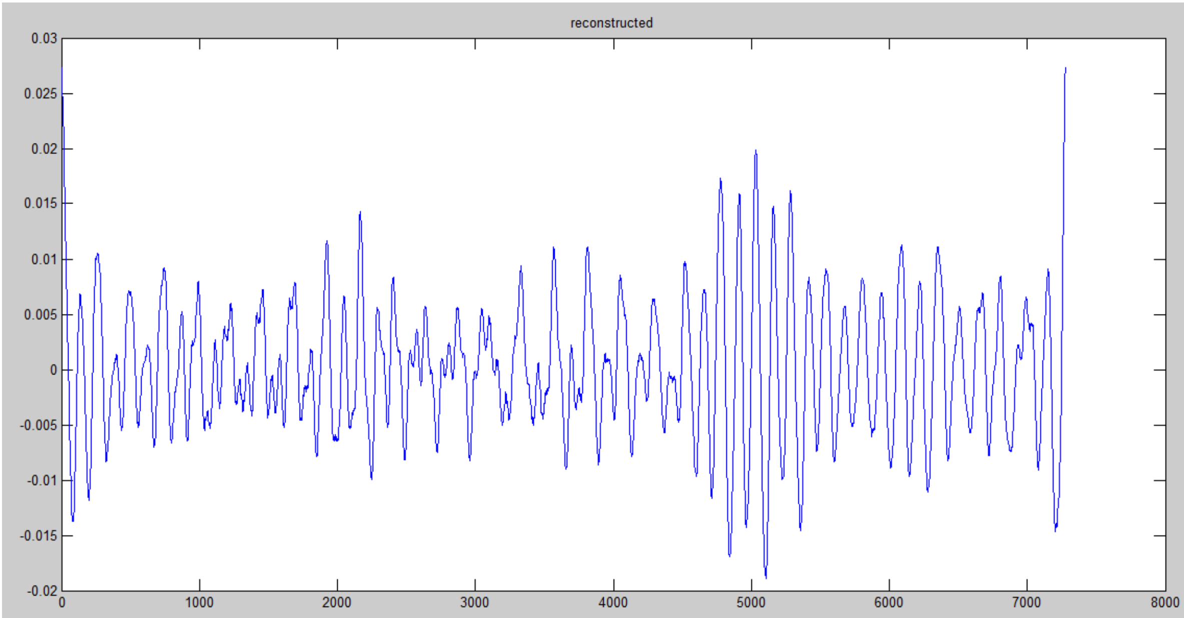
Figure 6.4: Original and reconstructed signals for ‘Voice signal of Nayan’. (a) (b) and (c)



(a) Original signal



(b) Reconstructed signal using l_1 minimization

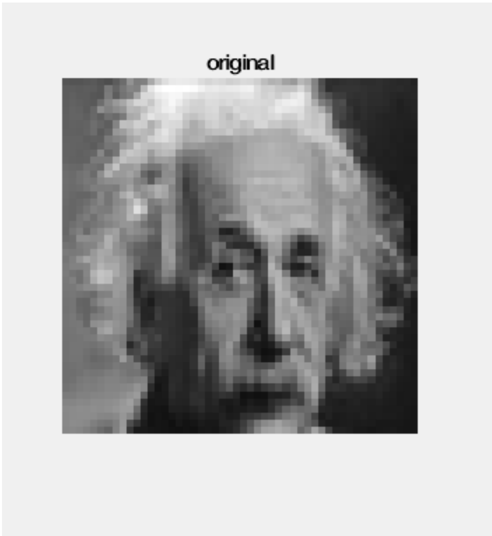


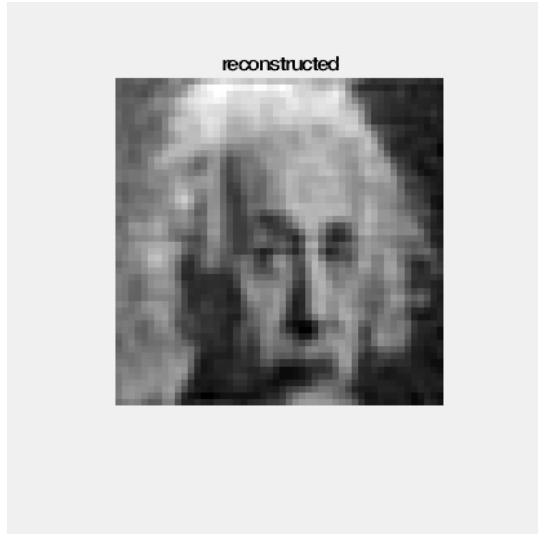
(c) Reconstructed signal using OMP

Figure 6.5: Original and reconstructed signals for 'Voice signal of Shohag'

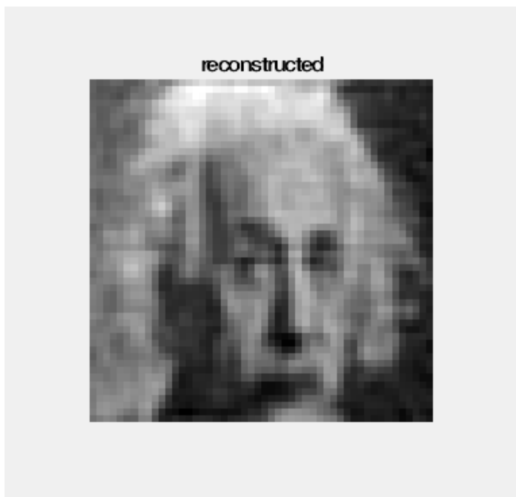
6.3 Image Reconstruction

We have firstly read the Image file from computer. The original image (gray scale) is shown in figure 5.6 ~ 5.10 (a). Then we have transformed it to the frequency domain it using Discrete Cosine Transform (DCT) after converting the image into gray scale. Then measurements have been taken from this sparse signal. After we used l_1 minimization algorithm to reconstruct the original image shown in figure 5.6 ~ 5.10 (b). In another process, we have used OMP algorithm to reconstruct the original image shown in figure 5.6 ~ 5.10 (c). Through figure 5.6 ~ 5.10 five different images will be shown for original image, reconstruction using l_1 minimization and reconstruction using OMP algorithm.



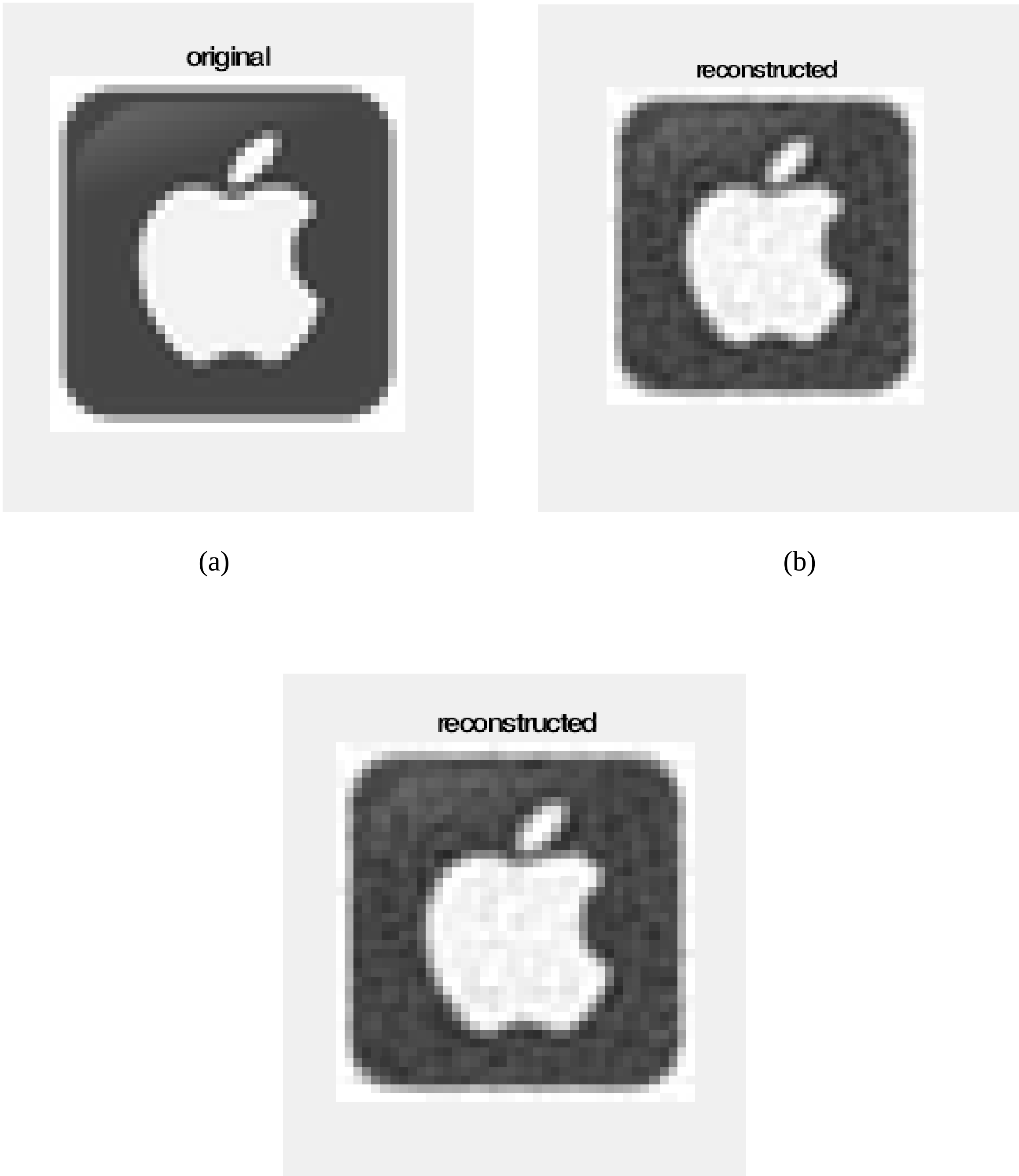


(a) (b)



(c)

Figure 6.6: Original and reconstructed images for ‘Einstein’. (a) Original Image (b) Reconstructed image using l_1 minimization and (c) Reconstructed image using OMP



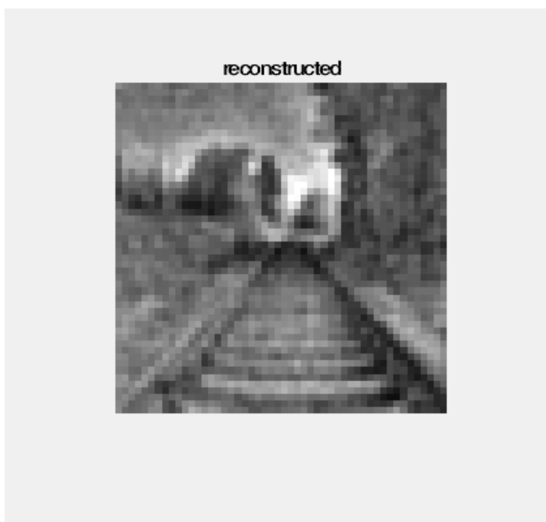
(a) (b)

(c)

Figure 6.7: Original and reconstructed images for ‘MAC’. (a) Original image (b) Reconstructed image using l_1 minimization and (c) Reconstructed image using OMP

(a)

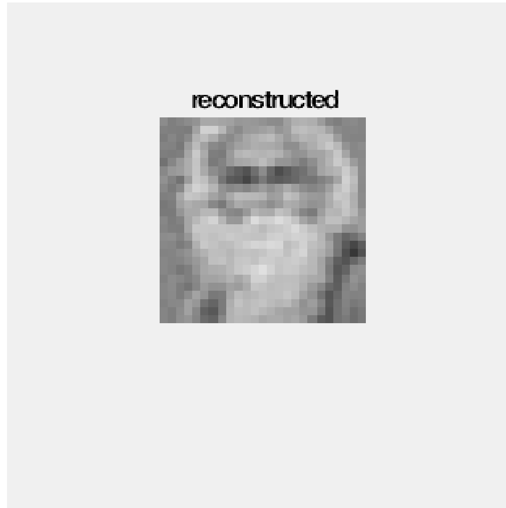
(b)



(c)

Figure 6.8: Original and reconstructed images for ‘Nature’. (a) Original image (b) Reconstructed image using l_1 minimization and (c) Reconstructed image using OMP

(a)

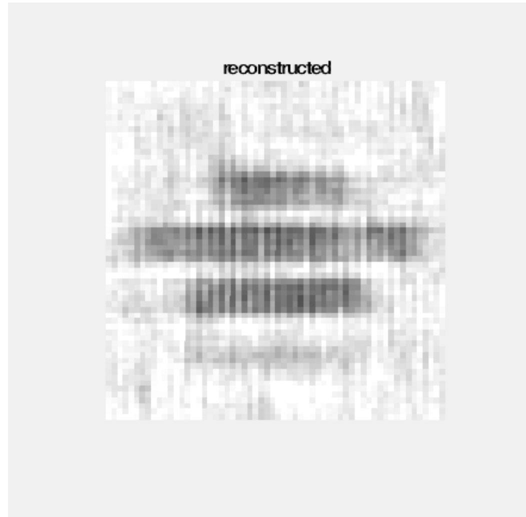


(b)

(c)

Figure 6.9: Original and reconstructed images for ‘Rabindranath’. (a) Original image (b) Reconstructed image using l_1 minimization and (c) Reconstructed image using OMP

(a)



(b)

(c)

Figure 6.10: Original and reconstructed images for ‘Shakespeare’. (a) Original image (b) Reconstructed image using l_1 minimization and (c) Reconstructed image using OMP

6.4 Analysis of l_1 and OMP Reconstruction Algorithm

Table 6.1 shows the comparison between two different reconstruction algorithms in different aspects which have been exploited here for voice signals.

Table 6.1: Comparison between l_1 minimization and OMP algorithm for voice signal

	l_1 minimization algorithm				Orthogonal Matching Pursuit (OMP) algorithm			
	MSE	T	N	m	MSE	T	N	m
Figure 6.1	0.0425	115.605918	3551	1276	0.0444	46.69	3551	1276
Figure 6.2	0.0020	22765.500648	10694	3319	0.0020	1225.602547	10694	3319
Figure 6.3	0.0000031869	2441.327106	7056	2377	0.000003189	171.607675	7056	2377
Figure 6.4	0.0000078336	1662.706321	8379	1572	0.0000072773	70.373476	8379	1572
Figure 6.5	0.000034839	531.469725	7276	1175	0.000034362	32.309083	7276	1175

Table 6.2 shows the comparison between two different reconstruction algorithms in different aspects which have been exploited here for images.

Table 6.2: Comparison between l_1 minimization and OMP algorithm for images

	l_1 minimization algorithm				Orthogonal Matching Pursuit (OMP) algorithm			
	MSE	T	N	m	MSE	T	N	m
Figure 6.6	0.0015	9.513646	2500	1110	0.0015	2.856373	2500	1110
Figure 6.7	0.0016	4.350230	1600	1062	0.0016	2.12701	1600	1062
Figure 6.8	0.0017	16.453836	2500	1653	0.0017	7.875102	2500	1653
Figure 6.9	0.00087943	0.592825	625	450	0.0008794	0.284703	625	450
Figure6.10	0.0163	12.913639	3600	900	0.0248	8.906033	3600	900

CHAPTER 7

CONCLUSION AND SCOPES

7.1. Conclusion

Based on this thesis, several methods for reconstructing the voice signal and image have been investigated. The methods are l_1 and Orthogonal Matching Pursuit (OMP) algorithm. In order to investigate whether these algorithms are able to reconstruct the signal we perform coding in MATLAB.

Based on the programming that had been done, it can be proved that both l_1 and OMP algorithm can reconstruct the voice signal and image almost perfectly. These two algorithms' performance is different in reconstruction time and mean square error (MSE). Large size image and voice signals could not be used because of the error 'Max array size-- Out of memory'.

7.2 Scopes

For future work and research, there are several recommendations that can be carried out by using this project as a platform. The recommendations are as follows:

1) Programing of MRI signal reconstruction can be done using MATLAB. MRI signal acquisition can be accelerated using compressive sensing.
2) A common approach to make the recovery problem amenable to compressive sensing framework, is to discretize the continuous domain. This will result in what is often called the gridding error or basis mismatch. Developing rigorous compressive sensing

framework for signals which are sparse in a continuous domain is also a great challenge.

3) Design of sensing matrix is an open research area.
4) The concept of compressive sensing has inspired the development of new data acquisition hardware. By now we have seen compressive sensing “in action” in a variety of applications, such as MRI, astronomy, and analog-to-digital conversion. The construction of compressive sensing based hardware is still a great challenge.

REFERENCES

[1] Richard Baraniuk, Justin Romberg, Michael Wakin, “Tutorial on Compressive Sensing”.

[2] Wikibooks, “Signals and Systems/Definition of Signals and Systems”, Avaiable: https://en.wikibooks.org/wiki/Signals_and_Systems/Definition_of_Signals_and_Systems

[3] “Nyquist's Sampling Theorem”, Available: https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node149.html

[4] Behrouz A. Forouzan, Firouz Mosharraf, “Foundations of Computer Science”.

[5] Wikibooks, “Digital Signal Processing/Sampling and Reconstruction” , Available: https://en.wikibooks.org/wiki/Digital_Signal_Processing/Sampling_and_Reconstruction

[6] Usham Dias, Milind Rane, S. R. Bandewar, “Survey of Compressive Sensing,”, International Journal of Scientific & Engineering Research, Volume 3, Issue 2, February 2012.

[7] S. Foucart, H. Rauhut, “A Mathematical Introduction to Compressive Sensing,” Applied and Numerical Harmonic Analysis, DOI 10.1007/978-0-8176-4948-7.

[8] Emmanuel J. Candès, Michael B. Wakin, “An Introduction to Compressive Sampling,” IEEE Signal Processing Magazine, Volume 25, Issue 2, pp 21-30, March 2008.

[9] Thu L. N. Nguyen, Yoan Shin, “Deterministic Sensing Matrices in Compressive Sensing: A Survey,” The Scientific World Journal, Volume 2013 (2013), Article ID 192795, November 5, 2013.