Compressive Sensing

Publicações Matemáticas

Compressive Sensing

Adriana Schulz
IMPA

Eduardo A. B. da Silva
UFRJ

Luiz Velho
IMPA

impa

27º Colóquio Brasileiro de Matemática

Impresso no Brasil / Printed in Brazil

Capa: Noni Geiger / Sérgio R. Vaz

27º Colóquio Brasileiro de Matemática

- A Mathematical Introduction to Population Dynamics - Howard Weiss
- Algebraic Stacks and Moduli of Vector Bundles - Frank Neumann
- An Invitation to Web Geometry - Jorge Vitório Pereira e Luc Pirio
- Bolhas Especulativas em Equilíbrio Geral - Rodrigo Novinski e Mário Rui Páscoa
- C*-algebras and Dynamical Systems - Jean Renault
- **Compressive Sensing - Adriana Schulz, Eduardo A. B. da Silva e Luiz Velho**
- Differential Equations of Classical Geometry, a Qualitative Theory - Ronaldo Garcia e Jorge Sotomayor
- Dynamics of Partial Actions - Alexander Arbieto e Carlos Morales
- Introduction to Evolution Equations in Geometry - Bianca Santoro
- Introduction to Intersection Theory - Jean-Paul Brasselet
- Introdução à Análise Harmônica e Aplicações - Adán J. Corcho Fernandez e Marcos Petrúcio de A. Cavalcante
- Introdução aos Métodos de Decomposição de Domínio - Juan Galvis
- Problema de Cauchy para Operadores Diferenciais Parciais - Marcelo Rempel Ebert e José Ruidival dos Santos Filho
- Simulação de Fluidos sem Malha: Uma Introdução ao Método SPH - Afonso Paiva, Fabiano Petronetto, Geovan Tavares e Thomas Lewiner
- Teoria Ergódica para Autômatos Celulares Algébricos - Marcelo Sobottka
- Uma Iniciação aos Sistemas Dinâmicos Estocásticos - Paulo Ruffino
- Uma Introdução à Geometria de Contato e Aplicações à Dinâmica Hamiltoniana - Umberto L. Hryniewicz e Pedro A. S. Salomão
- Viscosity Solutions of Hamilton-Jacobi Equations - Diogo Gomes

ISBN: 978-85-244-0294-4

Distribuição: IMPA
Estrada Dona Castorina, 110
22460-320 Rio de Janeiro, RJ
E-mail: ddic@impa.br
http://www.impa.br

Preface

Compressive sensing (CS) is a novel idea that rethinks data acquisition. The theory was so revolutionary when it was created in 2004 that an early paper outlining it was initially rejected on the basis that its claims appeared impossible to be substantiated.

The impact of compressive sensing goes far beyond the research labs and enters a more organic social level. This new area was able to establish a true synergy between many disciplines of science, technology and engineering. Usually such groups are far apart due to the cultural differences of their respective fields. Now, thanks to compressive sensing, it is frequent to see pure mathematicians, applied mathematicians, computer scientists, and hardware engineers coming together to share ideas about the theory and its applications [1].

We were motivated to study CS, not only because it is a novel idea that has had a great impact in the academic community, but also because it is a very rich theory that covers interesting mathematical tools as well as notions of acquisition, compression, dimensional reduction and optimization. Furthermore, the applications of compressive sensing are very much related to audio-visual media, vision and graphics, our main research focus.

This work started as the final project of one of the authors, Adriana Schulz, at the Department of Electronics and Computer Engineering of POLI/UFRJ under the co-supervision of the two other authors, Eduardo da Silva and Luiz Velho. Motivated by the extreme success of this study, the authors continued pursuing further research in the area. Finally, this book was written as the course notes for an intermediate level tutorial for the 27th Brazilian Mathematics Colloquium.

The intention of this book is to develop a presentation of the fundamental aspects involved in CS which may be used as a bibliographic guide for those who are initiating on this field. We were careful to elaborate examples of applications in different acquisition scenarios, which allowed us to answer a few interesting questions and evaluate the performance of the technique. We also illustrate applications in image processing, graphics and vision. Some of these applications are related to emerging subareas, known as image-based modeling and rendering, which combine methods from all the aforementioned fields.

Contents

List of Figures

Chapter 1

Introduction

Acquisition and reconstruction are essential in every signal processing system and sampling theorems are responsible for the bridge between continuous and discrete domains. The most important theorem that sets a limit to the sampling rate guaranteeing signal recovery is the Shannon-Nyquist theorem for band-limited signals.

We know, however, that natural and manmade signals tend to be compressible, i.e., if point sampled many of the acquired coefficients will be redundant. Hence, a lot of effort has been made in order to rewrite the sampled data reducing the number of bits required to represent it. These schemes perform what is referred to as compression.

The sample-then-compress framework is very efficient and is used in many applications with a good performance. However, the fact that we are able to compress the acquired data, suggests that Nyquist was a pessimist, who considered the worst case scenario in which all that is known is that the signals are band-limited. But what if, instead of considering the Nyquist rate, we would try to recover the data by sensing at the information rate?

This is what compressive sensing is about. It comes out as a new paradigm for data acquisition that rises against the common knowledge of the filed. In truth, it gives stable and robust algorithms that allows sensing at rates much smaller then the Nyquist limit and recovering the signals with little corruption.

The basic idea is that compressibility translates in the existence

of a representation in which the signal is sparse (most coefficients are zero). Therefore, while taking only a small number of samples would make the recovery problem ill-posed (an infinite number of solutions would be available), the compressibility property allows us to search in all possible solutions the one that makes the recovered signal sparse.

Of course, there is a twist in the word "sample". We cannot point sample the signal and hope to reconstruct it with a very small number of measurements because, once it is sparse, most of our acquired data will be zero. Instead, we measure the signal by calculating its inner product against different test functions.

Compressive sensing is intriguing not only because it proves that it is possible to reconstruct a signal with a very small number of measurements but also because it is *nonadaptive*. By this we mean that the algorithm is completely blind, not needing to guess characteristics of the original object (apart from sparsity). Moreover, the solution is obtained by means of a linear program that solves a convex optimization problem.

1.1 Organization

In Chapter 2, we consider the classic methods for image compression which apply the sample-then-compress framework. We study schemes that make use of transforms (as the DCT and Wavelets) in order to exploit signal redundancy and map the data coefficients that are less correlated and, therefore, sparse.

Growing in abstraction levels, this compression paradigm is related in Chapter 3 to signal representation and reconstruction models. The latter are then studied with emphasis in approximation theory.

With the former analysis, the stage is set for the investigation of compressive sensing. Nevertheless, before we examine the fundamental theorems, some effort is made in Chapter 4 to intuitively justify the combination of sensing and compression in a single procedure.

Based on the definition of the reconstruction algorithm, we must establish the characteristics that, when imposed to the acquisition model, guarantee good performances. Hence, in Chapter 5, a few parameters are defined and several theorems that evaluate CS in dif-

ferent contexts are exposed.

In Chapter 6, we verify the CS theory by means of examples. We consider applications for image compression in scenarios where the signal is either sparse or only approximately sparse, as well as when measurements are corrupted by Gaussian and quantization noise.

In Chapter 7, we discuss new applications in Computer Graphics, Vision and related fields.

Chapter 2

Image Compression

During the last decades we have been experiencing a multimedia revolution that has enabled the access to large amounts of data even in adverse situations. A key ingredient that has made these technologies possible is the ability to express information in a compact form.

Data compression, therefore, aims at reducing the number of bits required to represent a signal by exploiting structures in the data (such as sparsity and redundancy) and characteristics of the users (such as the limited perceptual abilities of human beings).

To evaluate compression efficiency, it can be taken into account properties of the algorithm (complexity, velocity, memory consumption), the amount of compression, and how closely the reconstruction resembles the original signal.

In this work, we will focus on the *rate-distortion* criteria, that evaluates the trade-offs between the average number of bits used to represent each signal sample value and a quantification of the difference between the original signal and its reconstruction after compression.

Figure 2.1 illustrates a *rate-distortion* function $R(D)$ that specifies the lowest rate at which the output of a source can be encoded while keeping the distortion less than or equal to D. This function is very useful because it defines a bound and therefore a way to determine optimality given a particular source. It will not always be possible to design optimal compression schemes and thus the goal of many

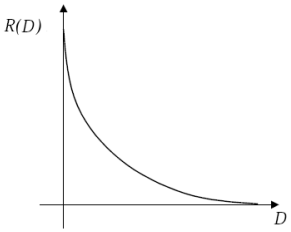


Figure 2.1: The *rate-distortion* function.

researchers in this area is to improve performance by approaching the $R(D)$ curve.

In this chapter, we will overview the basic elements of compression techniques and some popular standards for image compression.

2.1 Transform Coding

Most signals observed in nature are, in some way, compressible. This is not surprising if we consider that redundancy plays an important role in facilitating human perception. For example, it is easier and more pleasant to read a text with repetitions, listen to songs that do not have many abrupt variations, and watch videos with trivial differences between frames. The same thing occurs with images, where adjacent pixels tent do be very similar. In Figure 2.2, one can compare a redundant image (left) with a non-redundant one (right).

The existence of redundancy indicates that storing an image as a matrix in which each coefficient is the intensity of the correspondent pixel is inefficient because many pixel values will be equivalent.

The solution is to find a sparse representation, i.e., a representation in which the information is concentrated in only a few significant coefficients, the rest being zero valued. If this is accomplished, the number of coefficients that needs to be stored (or transmitted) will be largely reduced.

Transform coding [3] is the name given to data compression techniques that change the signal representations to minimize redundancy. Figure 2.3 introduces the three basic operations of transform

(a) Image *lena*. (b) White Gaussian noise.

Figure 2.2: In the image *lena*, pixels that are not in the boundary region are very similar to adjacent ones. The white noise, however, is not compressible. (Extracted from [2].)

coding.

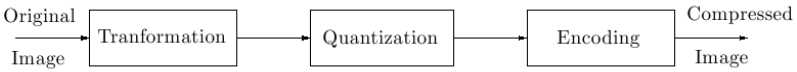


Figure 2.3: Transform coding operations.

The *transformation* of the image into a set of less redundant coefficients is the first step of the compression procedure. Simultaneously, it minimizes the correlation among coefficients and maximizes the energy concentration. Nevertheless, obtaining a matrix with many zeros is not enough to reduce the number of bits required for signal reconstruction.

It is interesting to point out that pixel values usually range between 0 and 255, i.e, each pixel is represented by 8 bits. After applying a transformation, however, the coefficients can assume arbitrary floating-point values. Moreover, transformations often generate many very small coefficients instead of just zero-valued ones.

Both of these problems are solved during the *quantization* step, which aims at representing a large range of values by a relatively small set of symbols. Though this strongly reduces the rate, it often

leads to information loss.

The last step aims at mapping the symbols in the smallest stream of bits possible. This procedure, called *encoding*, takes into account the statistical characteristics of the symbols and the positions of the significant (non-zero) coefficients in the matrix.

A simple illustration of a coding scheme that uses a transformation operation is the *Differential Pulse Coded Modulation* (DPCM) [2]. The fact that, in most natural images, adjacent pixels tend to have similar values indicates that a reasonable compression scheme would involve transmitting the difference between adjacent pixel instead of the original values.

This is the procedure of the DPCM, which uses as an estimate the value of the adjacent right pixel and transmits only the difference between the two. The advantage is that the values will now concentrate around zero and therefore more efficient quantization and coding schemes can be employed.

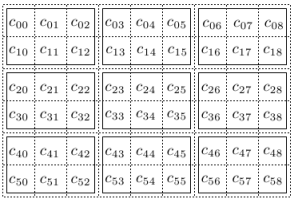
Notice that, without quantization and coding, this procedure, instead of reducing the output bit stream, enlarges it, because the pixel values which before transformation were between $\{0, 255\}$, range between $\{-255, 255\}$ after it.

In the following sections, we will study in more detail and will exemplify these three basic operations.

2.2 Transformation

From what was just mentioned, we conclude that the goal of the transformation step is to exploit information redundancy so as to adapt the signal in order to facilitate efficient quantization and encoding.

These are usually linear transforms that are applied to a sequence of inputs. In images, we have to partition the array of pixels into blocks of size N which will then be mapped to a transform sequence, as shown in Figure 2.4. The size of N is dictated by practical considerations. While large blocks will allow a greater number of zero coefficients, transform complexity grows more than linearly with N and statistical characteristics change abruptly (images are not stationary signals but we can assume stationary in a block if N is small).



$$v_0 = (c_{00}, c_{01}, c_{02}, c_{10}, c_{11}, c_{12})$$
$$v_1 = (c_{03}, c_{04}, c_{05}, c_{13}, c_{14}, c_{15})$$
$$v_2 = (c_{06}, c_{07}, c_{08}, c_{16}, c_{17}, c_{18})$$
$$v_3 = (c_{30}, c_{31}, c_{32}, c_{40}, c_{41}, c_{42})$$
$$v_4 = (c_{33}, c_{34}, c_{35}, c_{43}, c_{44}, c_{45})$$
$$v_5 = (c_{36}, c_{37}, c_{38}, c_{46}, c_{47}, c_{48})$$
$$v_6 = (c_{50}, c_{51}, c_{52}, c_{60}, c_{61}, c_{62})$$
$$v_7 = (c_{53}, c_{54}, c_{55}, c_{63}, c_{64}, c_{65})$$
$$v_8 = (c_{56}, c_{57}, c_{58}, c_{66}, c_{67}, c_{68})$$

Figure 2.4: Partition of an image array into blocks of size $N = 6$ and the sequence of correspondent vectors.

Let us now analyze three very common transforms and their applications in image compression.

2.2.1 Karhunen-Loève Transform (KLT)

KLT [4] is referred by many authors as PCA (Principal Components Analysis). In general, if we partition an image into blocks of size N and then represent each block as a vector in $\mathbb{R}^N$, the correlation between the coordinates will be very large, as shown in Figure 2.5.

The idea of KLT is to rotate the axes in order to minimize the correlation, which can be interpreted as redundancy between coefficients, and consequently increase energy concentration.

The basis vectors of the KLT are given by the orthonormalized eigenvectors of its autocorrelation matrix. This indicates a drawback to this technique: it is functionally dependent on the input data.

2.2.2 Discrete Cosine Transform (DCT)

The DCT [4] is very similar to the Fourier transform in the sense that it provides a spectral analysis of the signal. It has, however, a few properties, that make it interesting for compression applications.

The cosine transform is very closely related to the KLT of a first-order stationary Markov sequence when the correlation parameter is close to 1 and therefore, provides excellent energy compaction for highly correlated data.

Moreover, it is a real transform that can be implemented by a fast algorithm and is data independent.

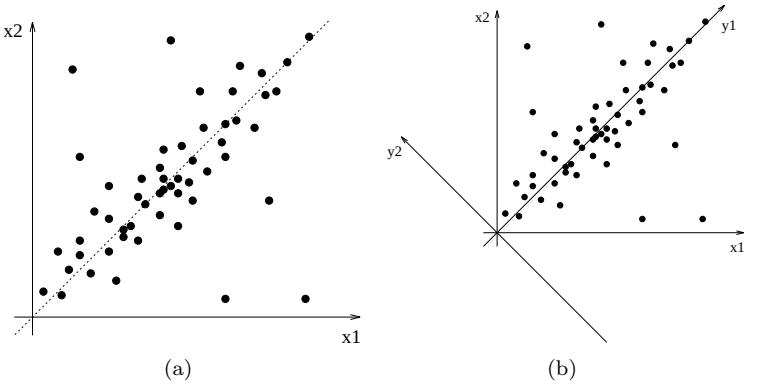


Figure 2.5: Each image block is represented in (a) as a vector in $\mathbb{R}^2$, and the on the KLT transform shown in (b) each vector $[a\ b]^T = ax_1 + bx_2$ will be represented by $[c\ d]^T = cy_1 + dy_2$. (Extracted from [2].)

We represent an image in the DCT domain by a matrix where each coefficient is given by

$$X_{k_1,k_2} = \alpha_1(k_1)\alpha_2(k_2) \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1,n_2} \cos\left[\beta_1 \cdot k_1\right] \cos\left[\beta_2 \cdot k_2\right]$$

where x_{n_1,n_2} is the value of the pixel at (n_1, n_2),

$$\beta_i = \frac{\pi}{N_i}\left(n_i + \frac{1}{2}\right)$$

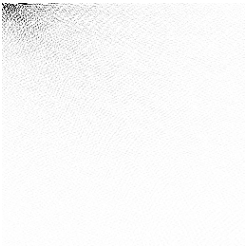
and

$$\begin{cases} \alpha_i(k) = \sqrt{\frac{1}{N_i}}, & \text{if } k = 0 \\ \alpha_i(k) = \sqrt{\frac{2}{N_i}}, & \text{if } k \neq 0 \end{cases}$$

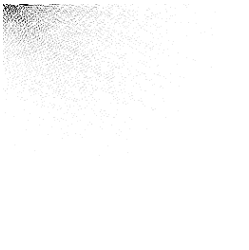
Notice that the first coefficient corresponds to the average signal level (DC value) of the signal and greater frequencies are associated with higher coefficients.

Figure 2.6 illustrates the transformation applied to the image *lena*. To simplify the example, block partitioning was not used. A better

(a) Original image.



(b) DCT transform of (a).



(c) Most significant coefficients.

(d) Image reconstructed form (c).

Figure 2.6: Example of image reconstructed with 1 out of 10 coefficients: we set to zero the smallest values of the DCT transform and reconstruct the image by applying an inverse DCT. We observe that, since many DCT coefficients are close to zero, the distortion is rather small.

result would have been achieved if we had applied the DCT individually to $N \times N$ blocks.

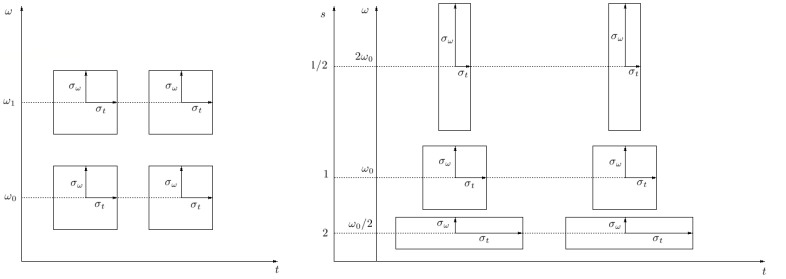
2.2.3 Discrete Wavelet Transform (DWT)

While the time domain describes the way a signal varies in time and its Fourier transform sheds light to the frequencies distribution, the

Wavelet transform can be interpreted as a way to extract information from a signal concerning both time and frequency. A first approach to achieve simultaneously both features is to apply the Fourier transform to windows of the original signal $x(t)$. This is known as the *Short Term Fourier Transform* (STFT) [5], and can be defined as

$$X_F(\omega, t) = \int_{\infty}^{-\infty} x(\tau) g(\tau - t) e^{-j\omega\tau} d\tau \tag{2.1}$$

where $g(t)$ is a window function centered in zero, with variance in time[1] σ_t^2, and variance in frequency σ_ω^2.



(a) STFT (b) Wavelet Transform

Figure 2.7: Time × frequency plane for the STFT and Wavelet transform. (Extracted from [6].)

Notice from Figure 2.7(a) and Equation 2.1 that the information in (ω_0, t_0) mostly depends on the values of signal $x(t)$ in the intervals $[\omega_0 - \sigma_\omega, \omega_0 + \sigma_\omega]$ and $[t_0 - \sigma_t, t_0 + \sigma_t]$. The smaller σ_t^2 the better a feature can be localized in the time domain, while the smaller the σ_ω^2 the better the frequency resolution of the STFT. However, the *uncertainty principle* states that we cannot find a window function $g(t)$ that allows for both σ_t^2 and σ_ω^2 to be arbitrarily small, i.e., it is

[1] We calculate variance as follows

$$\sigma_t^2 = \frac{\int_{-\infty}^{\infty} t^2 g(t) dt}{\int_{-\infty}^{\infty} g(t) dt}$$

impossible to obtain precise localization in both domains simultaneously.

Therefore, a fixed window function implies a predetermined resolution in which information is obtained. Images, however, as well as most natural signals, combine features of different detail levels. Therefore, a major drawback in the STFT is that the size of the window function is invariant.

The Wavelet transform tries to solve this problem by introducing the concept of scale. A scale is closely related to the width of the window and represents a measure of the amount of detail in the signal. The Wavelet transform of a signal $x(t)$ is the decomposition of $x(t)$ on the basis composed by translated and scaled version of a mother function $\Phi(t)$. The mother function scaled by s and translated by t is described as follows:

$$\Phi_{s,t}(\tau) = \frac{1}{\sqrt{s}}\Phi(\frac{\tau - t}{s})$$

where $\frac{1}{\sqrt{s}}$ is a normalization factor.

The function $\Phi_{s,t}(\tau)$ dilates and contracts with s, varying inversely to its Fourier transform, as shown in Figure 2.8. Therefore, the interval of the signal $x(t)$ that contributes to its Wavelet transform at (s,t) varies as shown in Figure 2.7(b).

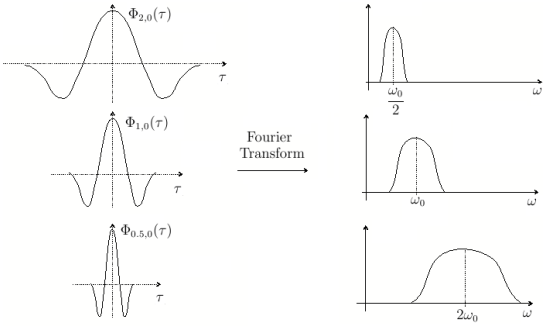


Figure 2.8: Scaled wavelet functions and their Fourier transforms. (Extracted from [6].)

The values of the transformed coefficients for a given scale inform

how much there is of the signal at a given resolution level. In small scales, refinement signal details are explored, while in large ones, coarse details are analyzed.

The redundancy generated by mapping a one dimensional signal in a two dimensional function indicates that recovery will still be possible after discretization is done. A common partition of the time $\times$ frequency grid is shown in Figure 2.9 and is known as a dyadic lattice:

$$(s,t) \in \{(2^m, n2^m t_0), n, m \in \mathbb{Z}\}$$

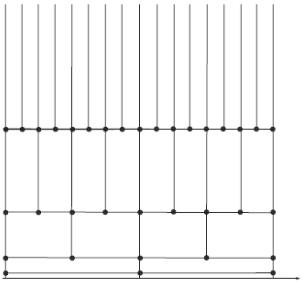


Figure 2.9: The discrete grid of the DWT. (Extracted from [5].)

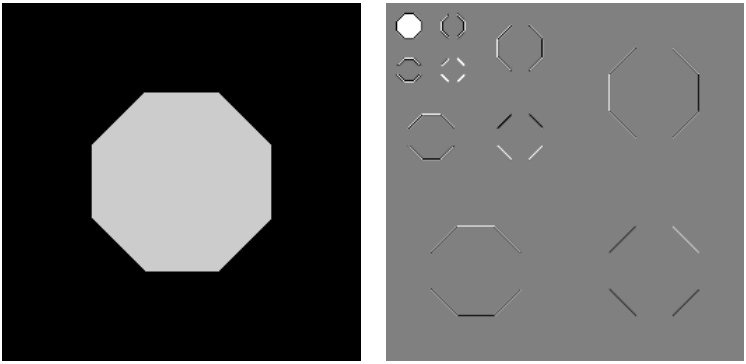
In terms of signal processing, a Wavelet transform is equivalent to filtering a signal in different subbands, each representing the signal information in a different resolution. This conclusion can be drawn from Figure 2.8, where the scaled Wavelet function is represented in the frequency domain by band-pass filters.

A common way to generate this subband decomposition is by dividing a signal into low and high-pass bands and then filtering again the low-pass channel in low and high-pass channels. The process of dividing the resulting low-pass channel is repeated until a predetermined number of stages is reached.

At each step, the low-pass filtering corresponds to a smoothing of the signal and the removal of details, whereas the high-pass corresponds to the differences between the scales.

In images, the DWT is applied both to rows and columns, as shown in Figure 2.10. In this Figure we notice that most of the

coefficients are close to zero and that the horizontal, vertical and diagonal bands are closely related. These features, allied to the ability of dividing the information in detail levels, make the DWT interesting for compression applications.



(a) Original image (b) Wavelet Transform

Figure 2.10: Example of 2D Wavelet transform of three stages. In (b) the coefficients are represented on a grayscale, white corresponding to positive values, back to negative and gray to zero values. (Extracted from [7].)

2.3 Quantization

Quantization [3] consists in representing a source output using one of a finite (and usually small) number of codewords. Since the number of codewords and the characteristics of the quantizer are closely related to the level of compression and the loss in fidelity, it is essential to bear in mind a rate-distortion criteria during this procedure.

Here we present two kinds of quantizers that differ in terms of the set of inputs and outputs, that can be either scalars or vectors.

2.3.1 Scalar Quantization

Scalar quantization consists in dividing a scalar input range into intervals and assigning for each one a codeword and an output value.

Figure 2.11 is an example of a linear quantizer, where all intervals have the same size, called *quantization step*.

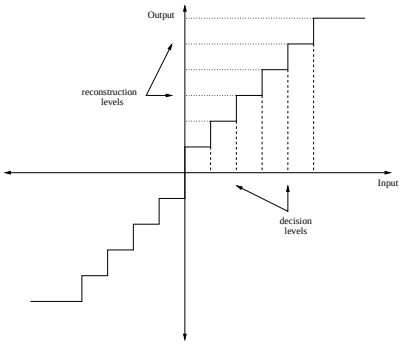


Figure 2.11: Linear quantizer input-output map.

In many applications it is not efficient to establish constant distances between decision and reconstruction levels. If this does not happen the quantization is called non-linear. In most image compression standards, however, the latter is not used because entropy coding combined with linear quantization provides a very similar performance and is less complex to implement.

2.3.2 Vector Quantization

From what has been studied up until now and from basic results in information theory, it is clear that encoding a sequence of outputs instead of individual samples separately is more efficient according to a rate-distortion criteria.

In this case, instead of quantizing each image pixel, we divide images into blocks of size N and represent each one as a vector in $\mathbb{R}^N$. The output of the quantizer is a finite set of vectors called *codebook* and each block of the source output is associated to the closest vector in the codebook, usually by applying the Euclidean

norm.

The process of finding the optimal codebook of size k for a given source set of vectors $\mathcal{S}$ involves choosing the k vectors of the codebook, and the k *quantization cells* - each quantization cell corresponds to the subset of $\mathcal{S}$ that is associated to the k^{th} code-vector. This procedure is not analytical because it involves two related considerations:

- Given the quantization cells, the best codebook is constructed by extracting the centers of each cell.
- Given the codebook, the best quantization cells are found by assigning each element in $\mathcal{S}$ to its closest vector in the codebook.

Hence, there are many algorithms for finding the best codebook given certain input data. Here we will describe one of the simplest, yet very popular, referred to as LBG:

1. Initialize the codebook by selecting k vectors at random.
2. Specify the quantization cells, i.e., assign to each source output the closest vector in the codebook.
3. Reset the codebook by selecting the centers of each quantization cell.
4. Return to step 2 unless a finalization condition is reached.

2.4 Encoding

We refer to coding [8] as the process of assigning binary representations to the output of a source, here referred to as alphabet. For example, the ASCII code uses 8 bits and each of the 2^8 possible combinations is associated to one of 256 letters or punctuation marks. This is a so called *fixed-length code* because all symbols are represented by the same number of bits.

To minimize the average number of bits per symbol, we should use fewer bits to represent symbols that occur more often. This is done in the Morse code, as illustrated in Figure 2.12. Note that the smallest codeword is associated to the letter E, which is the most used in the English language.

A .−	J .−−−	S ...	2 ..−−−
B −...	K −.−	T −	3 ...−−
C −.−.	L .−..	U ..−	4 ....−
D −..	M −−	V ...−	5 .....
E .	N −.	W .−−	6 −....
F ..−.	O −−−	X −..−	7 −−...
G −−.	P .−−.	Y −.−−	8 −−−..
H	Q −−.−	Z −−..	9 −−−−.
I ..	R .−.	1 .−−−−	0 −−−−−

Figure 2.12: Morse code.

We measure efficiency in terms of rate minimization by comparing the average symbol length with the alphabet's entropy, which is a measurement of the average information per source symbol.

Let $\mathcal{S} = \{s_1, \dots s_K\}$ be a given alphabet where each symbol has the probability of occurrence $p_k = P(\mathcal{S} = s_k)$. The entropy is given by:

$$H(\mathcal{S}) = \sum_{k=1}^{K} p_k \cdot \log\left(\frac{1}{p_k}\right) \tag{2.2}$$

and the average code length by:

$$\bar{L} = \sum_{k=1}^{K} p_k \cdot l_k$$

where l_k is the size of the codeword associated to the symbol s_k.

In this case, coding efficiency is measured by:

$$\eta = \frac{H(\mathcal{S})}{\bar{L}}$$

The Shannon Theorem guarantees $\bar{L} \geq H(\mathcal{S})$ and therefore the optimal code occurs when $\eta = 1$.

Along with minimizing rate, efficient codes must be uniquely decodable, i.e., there must be no ambiguity between codewords. It is also desirable that the decoding be instantaneous, which means that the decoder knows the moment a code is complete without having to wait until the beginning of the next codeword.

Now we will outline two coding procedures that are often employed in image compression standards.

2.4.1 Huffman Code

David Huffman developed an instantaneous code where the average symbol length is very close to the entropy. It is based on two observations:

- Symbols with greater probability of occurrence should have smaller codewords.
- The two symbols that occur least frequently should have the same length.

We will demonstrate this coding procedure by an example. Let $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$ be an alphabet where the probability of occurrence of each symbol is respectively $\{0.5, 0.25, 0.125, 0.125\}$.

The symbols are arranged in order of decreasing probability and the last two symbols are combined iteratively until only one symbol is left. Figure 2.13 illustrates this procedure and the decision tree generated by the coding strategy.

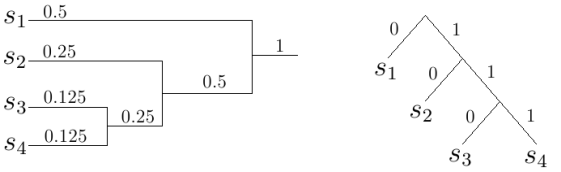


Figure 2.13: Huffman code.

Table 2.1: Associated codewords generated by the Huffman coding.

Symbol	Codeword
s_1	0
s_2	10
s_2	110
s_2	111

Table 2.1 displays the codewords associated to each symbol. Notice that in this case, since the distribution of probabilities is dyadic,

the code is optimal, i.e., $\eta = 1$.

2.4.2 Arithmetic Code

Though very successful in many circumstances, the Huffman code becomes inefficient when a single symbol has a very large probability of occurrence. This is often the case in small alphabets, where the obligation of using an integer number of bits to represent each symbol, limits the reduction of the average code length.

In this case, a better performance would be achieved by blocking groups of symbols together and generating codes capable of characterizing entire sequences of symbols by a unique identifier. This is the proposition of the arithmetic code, which maps each sequence into the unit interval $[0, 1)$. We will illustrate the encoding procedure with an example.

Let $\mathcal{S} = \{s_1, s_2, s_3\}$ be a given alphabet where each symbol has the probability of occurrence $p_1 = 0.5$, $p_2 = 0.2$, $p_3 = 0.3$. The first step consists in dividing the unit interval into regions that are associated with each symbol. The size of each region is, of course, directly related to the symbol probability, since larger regions will require a smaller number of decimal figures to be represented.

If the first symbol to be encoded is s_1, then the code will be a number in $[0, 0.5)$ and this interval will be divided according to the alphabet's probability distribution. This process is repeated iteratively as shown in Figure 2.14, which considers the sequence (s_1, s_3, s_2), and the transmitted code is a number between 0.425 and 0.455, for example the mean 0.44. The decoder procedure is also done iteratively dividing the interval and finding the associated symbols.

There are, however, two problems associated with arithmetic coding:

- There is no information provided as to when the decoding should stop.
- The binary representation of a real value with infinite precision can be infinitely long.

The first problem can be solved either by informing the decoder the size of the sequence or by associating a region of the unit interval

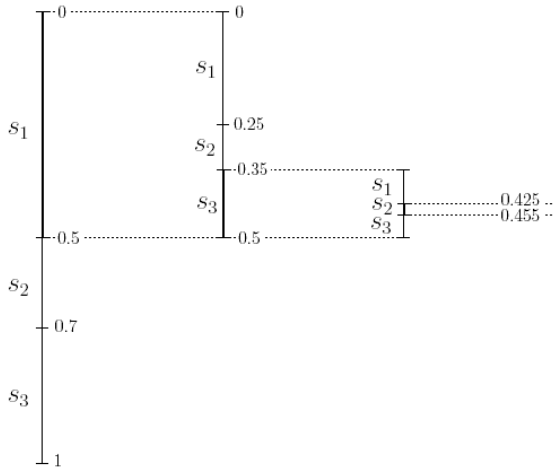


Figure 2.14: Example of arithmetic encoding.

with an *end-of-transmission* symbol. Figure 2.15 illustrates the EOT symbol, that brings the decoding procedure to a stop as soon as it is detected.

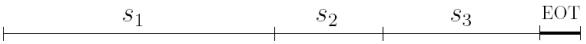


Figure 2.15: The *end-of-transmission* symbol.

There are several approaches to solve the second problem. The simplest one would be to encode each decimal symbol at a time, i.e., when we reach an interval small enough to make the n^{th} digit stop varying, it is transmitted.

2.5 Standards

In this section we will illustrate image compression by describing two very important standards: JPEG and JPEG2000.

2.5.1 JPEG

The JPEG [3] standard uses a very popular compression technique that involves DCT transform, followed by scalar quantization and Huffman coding.

The procedure starts by dividing the image into blocks of size 8×8 which are transformed by a forward DCT. This transformation isolates, in general, the important image components in the upper left portion of the matrix.

The calculated coefficients are quantized by uniform scalar quantization, where the step size varies increasingly as we move from DC coefficients to higher-order coefficients. The variation of the step size is related to the perception of the human visual system to errors in different spatial frequencies. Since the human eye is less sensitive to higher spatial frequencies, we can accept greater quantization errors for the coefficients that represent them. The following matrix shows the weight of each quantization step, i.e., the quantization step of the coefficient c_{ij} is $q_{\text{global}}Q_{ij}$, where q_{global} is a parameter associated with the compression rate.

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

The DC values are encoded separately from the AC ones because they vary little between adjacent blocks and, thus, it is interesting to encode the difference between neighbors. Therefore the DC values, i.e., the fist coefficient of each transformed block, are coded using DPCM followed by a Huffman entropy encoder.

To understand the coding of the AC coefficients it is important to analyze some properties of the matrix that stores the quantized coefficients of a typical DCT-transformed image block:

$$C = \begin{bmatrix} 42 & 26 & 10 & 0 & 0 & 0 & 0 & 0 \\ -3 & -2 & 0 & 2 & -1 & 0 & 0 & 0 \\ -21 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Notice that, not only is the matrix sparse, but also most of the nonzero coefficients are located on its upper-left corner. These characteristics suggest a scanning in a diagonal zigzag pattern, as shown in Figure 2.16.

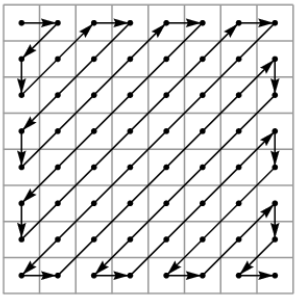


Figure 2.16: The zigzag scanning pattern.

The JPEG standard uses *run-length* encoding; i.e., each nonzero value that is scanned in the above fashion is stored as a sequence of pairs (run, length); the first indicating the number of preceding zeros and the second the values of the component. These pairs are then encoded using a Huffman code.

A drawback of dividing the image into blocks is that coding artifacts may be generated at block edges. This effect, called *blockiness*, is illustrated in Figure 2.17 .

Figure 2.17: Example of the blocking effect generated by a JPEG compression with very high rate.

2.5.2 JPEG2000

JPEG2000 [9] gains up to about 20% compression performance for medium compression rates in comparison to the first JPEG standard, but has, however, notably higher computational and memory demands. It involves a Wavelet transform followed by scallar quantization and arithmetic coding.

The Wavelet transform is applied to the tilled image, where the size of the tile can vary widely, being possible to consider the whole image as one single tile. This is important because small tiles can generate blocking effects, as in the JPEG standard.

The Wavelet coefficients are quantized by a uniform scalar quantizer with step size varying between subbands considering the human visual sensibility to different scaled informations. Each bit plane[2] of the quantized coefficients is then encoded using a process called *Embedded Block Coding with Optimal Truncation* (EBCOT).

As studied in section 2.2.3 Wavelet transform divide the image

[2]A bit plane of a digital discrete signal is a set of bits having the same position in the respective binary numbers. For example, for 8-bit data representation there are 8 bitplanes: the first one contains the set of the most significant bits and the 8^{th} contains the least significant bits.

into subbands that represent approximation scales. Notice, however, that some Wavelet coefficients in different subbands represent the same spacial location in the image. In Figure 2.10(b), it is noteworthy that the vertical subbands approximate scaled versions of each other, the same being true for horizontal and diagonal bands. This means that there exists a relation between the Wavelets coefficients illustrated in Figure 2.18.

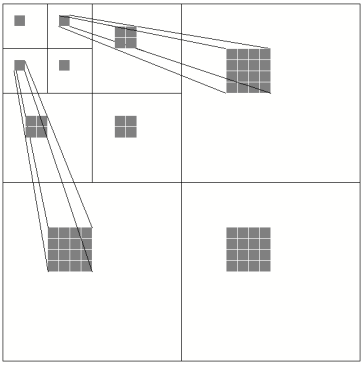


Figure 2.18: Related Wavelet coefficients.

Many algorithms, as the EZW and the SPHT codes, exploit the similarity among bands of the same orientation in order to reduce the size of the encoded image. JPEG2000 coding, however, does not exploit inter-subband redundancies. Instead, the EBCOT algorithm partitions each subband into small rectangular blocks called *code-blocks* and encodes each one independently.

Though there is an efficiency loss for not exploiting the correlation between subbands, this is compensated for because this method produces bit streams that are SNR and resolution scalable. For each codeblock a separate highly scalable bit stream is generated and may be independently truncated to any of a collection of different lengths.

The bits generated by the EBCOT algorithm are then encoded using an arithmetic code.

2.6 Classification of Compression Techniques

Many authors distinguish compression techniques as lossless or lossy, the former referring to invertible representations and the latter to representations in which some of the information is lost. Since quantization involves distortion effects, it is clear that we have focused our study in lossy compression schemes. In terms of the rate-distortion criteria, lossless compression would occur when the function $R(D)$ crosses the y-axis, i.e., when the distortion is zero.

For images we are usually interested in lossy techniques because they allow lower rates and the human visual system is not sensitive to small distortions. An exception to this rule would be when dealing with medical images, where the slightest error can result in a wrong diagnosis.

Another form of classification is linear and non-linear compression. To illustrate the difference between the two we will discuss the JPEG standard for image compression.

As shown in Section 2.5.1, the DCT transform results in a sparse matrix where the significant coefficients are concentrated in the upper-left corner and an encoding procedure called *run-length* coding makes use of these properties in order to reduce the size of the output stream of bits. Another approach would be to consider that all components in the lower-right corner are small, and so store only N values that belong to the region of the matrix that is usually significant, as shown in Figure 2.19.

This would not be as efficient as the run-length coding because some high-frequency information might be lost and zero-valued coefficients would be unnecessarily stored. However this approach is interesting because the compression technique does not depend on the image, i.e., we do not need to know *a priori* where the significant coefficients are before we begin encoding. This is what is referred to in literature as linear compression. In other words, if A and B are images and $\hat{A}$ and $\hat{B}$ are their compressed forms, the compression of $A + B$ will result in $\hat{A} + \hat{B}$. In non-linear compression, however, the location of the significant coefficients must be known before the reconstruction can be accomplished and, therefore, the linearity does

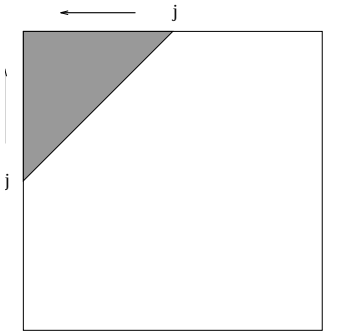
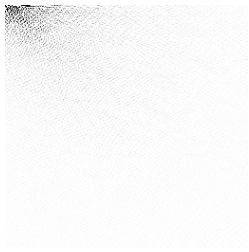


Figure 2.19: Example of region of matrix that would be selected as being significant in a linear compression scheme. (Extracted from [2].)

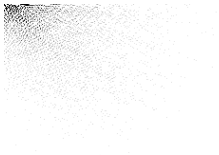
not hold.

In Figure 2.20 we compare the reconstruction of image *lena* with 1 out of 10 coefficients using non-linear and linear DCT compression and are able o conclude that the latter scheme is much less efficient. In 2.20(c) we set to zero the smallest values of the DCT transform and in 2.20(e) we set to zero the DCT coefficients that are not on the upper-left corner of the transformed matrix. Images 2.20(d) and 2.20(f) are reconstructed by applying an inverse DCT to 2.20(c) and 2.20(e), respectively.

(a) Original image.



(b) DCT transform of (a).



(c) Most significant coefficients.

(d) Reconstruction from (c).

(e) Coefficients on upper-left corner.

(f) Reconstruction form (e).

Figure 2.20: Example of image reconstructed with 1 out of 10 coefficients

Chapter 3

Signal Representations

Representation is a key aspect in signal processing. It refers to describing a signal completely and unambiguously as a sequence of enumerable coefficients. The importance of this procedure can be associated with the continuous nature of existing signals, which has to be overcome before digital processing.

Discretization, however, is not the only benefit we are searching for. Good signal representations can enable a series of procedures as analysis, noise filtering and compression. The idea behind this is that depending on how we describe a signal some of its aspects can be highlighted, i.e., we can distribute the information of interest between specific components and therefore ease access to them [10].

In this chapter we will overview different ways of representing signals and analyze their basic characteristics and how signals can be reconstructed from them.

3.1 Parallel to Image Compression

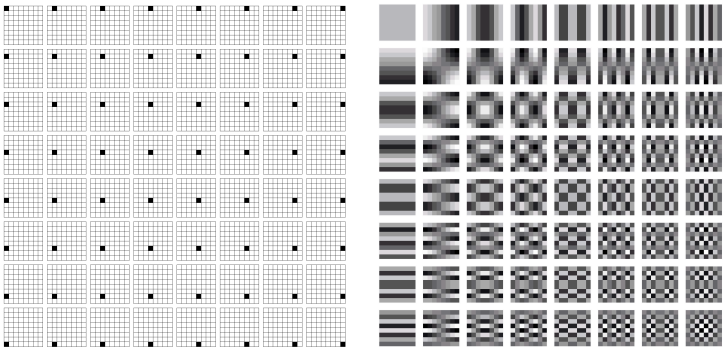
In the former chapter, we discussed transform coding as a method for compressing images by representing the same information in a smaller

number of coefficients. It is interesting to point out, however, that when we exploit redundancy to map the image data to less correlated coefficients, we are actually choosing a new way to represent the signal.

We can interpret an $n \times n$ image block as a vector in $\mathbb{R}^N$, where $N = n^2$. In the bit-map representation, each of the N canonical basis vectors would corespond to the information of a single pixel.

Since each orthonormal basis is a rotation of each other, the DCT transform is, therefore, no more than the rotation of this basis. Notice that the DCT expands the original image in sequence of cosines, i.e., the transformation is actually the projection in a new orthonormal basis.

The bit-map (canonical) basis is equivalent to *Dirac* functions in a two dimensional space, as shown in Figure 3.1(a), while the DCT basis is illustraded in Figure 3.1(b).



(a) 8×8 bit map basis. (b) 8×8 DCT basis.

Figure 3.1: Waveforms that compose the bit map and DCT bases.

Notice, however, that the DCT preserves many properties such as invertibility and orthogonality, which cannot be guaranteed for arbitrary representations. In the next section, we will, therefore, define such representations in a more abstract and generalized manner.

3.2 Signal Decompositions

We define a signal representation [5] by a function $R : \mathbf{H} \rightarrow \mathcal{S}$ that maps a Hilbert space[1] $\mathbf{H}$ into a space of sequences. For a given signal, $x \in \mathbf{H}$, its representation $R(x)$ is a sequence:

$$R(x) = (s_1, s_2, s_3...) \in \mathcal{S}$$

where s_n is a pair (α_n, g_{γ_n}), the first representing a coefficient and the second a waveform.

Associated with R is a set of functions $\mathcal{D} = (g_\lambda)_{\lambda \in \Gamma}$ called *dictionary*. Notice that the dictionary may be uncountable, however, the $(g_{\gamma_n})_{n \in \mathbb{Z}}$ used in the representation of a particular signal X consists of a countable subset.

In some cases, the function R is invertible and the signal x will be perfectly reconstructed from its representation $R(x)$. We then say that the representation is *exact* and the original signal is reconstructed by the linear combination

$$x = \sum_{n \in \mathbb{Z}} \alpha_n g_{\gamma_n}$$

Nevertheless, when the representation is not exact, we make use of techniques to approximate the reconstruction of x.

The dimension N of the signal space $\mathbf{H}$ is associated with the number of elements of the dictionary that are needed to span the space. A good representation scheme requires the use of a *complete* dictionary, i.e., any function in $\mathbf{H}$ can be expanded by a combination of the waveforms $(g_\lambda)_{\lambda \in \Gamma}$. It is noteworthy, however, that the size of the dictionary may be larger than N. In this case, we say that the dictionary is redundant because there is more than one way to represent the same signal. It is important to point out that, is some cases, we deal with infinite dimensions.

[1] A Hilbert space is an inner product space which, as a metric space, is complete, i.e., an abstract vector space in which distances and angles can be measured and which is complete, meaning that if a sequence of vectors approaches a limit, then that limit is guaranteed to be in the space as well.

The key point in signal decompositions is thus to obtain the sequence of dictionary waveforms $(g_{\lambda_n})_{n\in\mathbb{Z}}$ and their corresponding coefficients $(\alpha_n)_{n\in\mathbb{Z}}$. There are many methods that do so, exploiting signal properties, as mentioned earlier. We will now distinguish between two representation models: basis and frames.

3.2.1 Basis

A basis [11] is a set of linearly independent elements $(\phi_\lambda)_{\lambda\in\Gamma}$ that span the Hilbert space **H**. By linear independence we mean that no function can be expressed as a linear combination of the others - this implies that the set is minimal.

Orthogonal Basis

We define an orthonormal basis as a collection of functions $\{\phi_\lambda; \lambda \in \Gamma\}$ that are complete in the sense that they span **H** and satisfy:

$$\int_{-\infty}^{\infty} \phi_i(t)\bar{\phi}_j(t)dt = \delta(i-j), \quad \forall i,j \in \Gamma$$

where $\bar{\phi} = \mathrm{Re}\{\phi\} - j\mathrm{Im}\{\phi\}$ is de complex conjugate.

In this case, the representation is exact and the reconstruction is given by

$$x = \sum_{\lambda\in\Gamma} \langle x, \phi_\lambda\rangle \phi_\lambda$$

where the inner product $\langle x, \phi_\lambda\rangle = \int_{-\infty}^{\infty} x(t)\bar{\phi}_\lambda(t)dt$ is interpreted as the projection of the signal of interest in the base function ϕ_λ.

3.2.2 Frames

Frames [11] are a generalization of the concept of basis in a linear space. While a set of vectors forms a basis in $\mathbb{R}^M$ if they span $\mathbb{R}^M$ and are linearly independent, a set of $N \geq M$ vectors form a frame if they span $\mathbb{R}^M$.

More formally, a frame is a family of vectors $(\phi_\lambda)_{\lambda\in\Gamma}$ that characterizes any signal x in a Hilbert space $\mathbf{H}$ from its inner product $\{\langle x,\phi_\lambda\rangle\}_{\lambda\in\Gamma}$, where the index set Γ might be finite or infinite.

Frame Theory, developed by Duffin and Schaeffer, sets a condition for the frame to define a complete and stable signal representation:

Definition 1. *The sequence $(\phi_\lambda)_{\lambda\in\Gamma}$ is a frame of $\mathbf{H}$ if there exist two constants $A>0$ and $B>0$ such that for any $x\in\mathbf{H}$*

$$A\|x\|^2 \le \sum_{\lambda\in\Gamma} |\langle x,\phi_\lambda\rangle|^2 \le B\|x\|^2$$

When $A=B$ the frame is said to be tight.

It is noteworthy that a frame representation may be redundant, and, considering $\|\phi_\lambda\|=1, \forall\lambda\in\Gamma$, this redundancy can be measured by the frame bounds A and B. The following example will be used to illustrate frame redundancy:

Example 1. *Let (e_1,e_2) be an orthonormal basis of a two-dimensional plane $\mathbf{H}$. The three vectors:*

$$\phi_1 = e_1 \ , \ \phi_2 = -\frac{e_1}{2} + \frac{\sqrt{3}}{2}e_2 \ , \ \phi_3 = -\frac{e_1}{2} - \frac{\sqrt{3}}{2}e_2$$

have equal angles of $\frac{2\pi}{3}$ between any two vectors. For any $x\in\mathbf{H}$

$$\begin{aligned}
&\sum_{n\in\Gamma} |\langle x,\phi_n\rangle|^2 \\
&= |\langle x,e_1\rangle|^2 + |-\frac{1}{2}\langle x,e_1\rangle + \frac{\sqrt{3}}{2}\langle x,e_2\rangle|^2 + |-\frac{1}{2}\langle x,e_1\rangle - \frac{\sqrt{3}}{2}\langle x,e_2\rangle|^2 \\
&= \frac{3}{2}|\langle x,e_1\rangle + \langle x,e_2\rangle|^2 \\
&= \frac{3}{2}\|x\|^2
\end{aligned}$$

These three vectors thus define a tight frame with $A=B=\frac{3}{2}$. The frame bound $\frac{3}{2}$ gives the redundancy ratio, i.e., three vectors in a two-dimensional space.

3.3 Uniform Point Sampling

In this section we will introduce the simplest method for representing a function and analyze some of its characteristics.

Point sampling discretizes a signal $x(t)$ by taking a partition $t_1 < t_2 < \cdots < t_N$ of the domain interval I. The subsequent representation is given by the vector:

$$x_n = (x(t_1), x(t_2), \ldots, x(t_N)) \in \mathbb{R}^N$$

This way, the space of real functions defined on the interval I is represented by the Euclidean space $\mathbb{R}^N$. Point sampling is called uniform if $t_n = nt_s,\ \forall n$.

What remains to be investigated is if uniform point sampling is an exact representation and how can the original function x be recovered from x_n.

The Shannon theorem guarantees that a band-limited signal can be perfectly reconstructed if the sampling rate is $1/(2\omega_0)$ seconds, where ω_0 is the highest frequency in the original signal. We will not demonstrate this theorem here, but we will try to convince the reader with the following observations. Additional material regarding this theorem can be found in [12].

It is intuitive that sampling a signal in the time domain is equivalent to multiplying it by a *Dirac* comb. The Fourier transform of a *Dirac* comb is also a *Dirac* comb and therefore, in the frequency domain, the band-limited spectrum of the signal is being convolved by a *Dirac* comb, see Figure 3.2.

By observing these pictures it is easy to see that if the sampling rate ω_s is greater than $2\omega_0$, then the signal in the frequency domain can be recovered by an ideal low pass filter, as shown in Figure 3.3.

Since the Fourier transform of the Gate function is a *sinc* function, the reconstruction of the signal in the time domain is no more than an interpolation of the sampled vector by *sinc* functions.

On the other hand, if this limit of $2\omega_0$, called the *Nyquist rate*, is not respected, then repeated spectrums will overlap and it will be impossible to recover the signal by a low pass filtering. This phenomenon is called *aliasing*, and is illustrated in Figure 3.4.

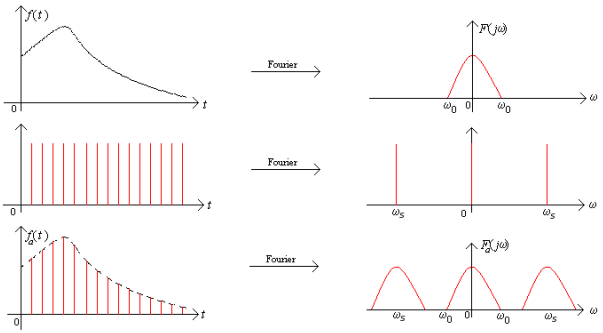


Figure 3.2: Sampling in time and the consequences in the frequency domain.

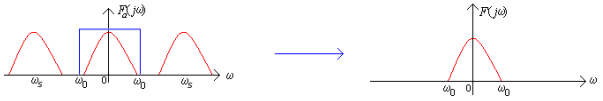


Figure 3.3: Extracting the repeated spectrums.

Notice that point sampling involves representing a signal as sequence of values

$$R(x) = (\alpha_n)_{n\in\mathbb{Z}}$$

where α_n is the projection of the signal on a delayed *Dirac*

$$\alpha_n = \langle x, \delta(t - nt_s)\rangle = \int_{-\infty}^{\infty} x\delta(t - nt_s) = x(nt_s).$$

This representation is an invertible function, once the original signal can be reconstructed by an interpolation of *sinc* functions. The exact reconstruction is then given by

$$x = \sum_{n\in\mathbb{Z}} \alpha_n h(t - nt_s)$$

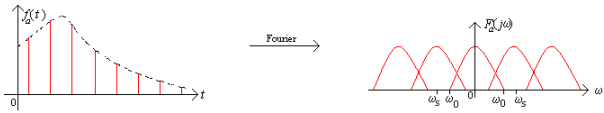


Figure 3.4: Undersampling in time and the consequences in the frequency domain.

where $h = \text{sinc}\left(\frac{t}{t_s}\right)$ is a scaled *sinc* function.

This is a very interesting example, because the projection waveforms used for representation are different from the reconstruction waveforms (dictionary).

3.3.1 Oversampling

If the sampling rate ω_s is greater than $2\omega_0$, we observe information redundancy, i.e., the number of samples is larger than it has to be to enable reconstruction of the signals. This can be usefull for many applications because it minimizes noise errors and allows the use of less complex anti-aliasing filters.

In this case, however, the scaled *sinc* functions that can be used to reconstruct this signal are not necessarily orthogonal. Note that

$$\langle h(t), h(t - nt_s)\rangle = \langle H(j\omega), H(j\omega)e^{-j\omega t_s}\rangle$$

where $H(j\omega)$ is a Gate function of badwidth $2/t_0$, $t_0 = 1/\omega_0$, and $t_s = 1/\omega_s$. Therefore, if $t_s = t_0/2$, then $\langle H(j\omega), H(j\omega)e^{-j\omega t_s}\rangle = 0$ and the basis is orthogonal.

However, when we oversample, this does not occur. Actually, the set $(h(t - nt_s))_{n\in\mathbb{Z}}$ becomes complete and redundant. In terms of what has been just described, this set is a frame.

3.3.2 Undersampling

In many applications, however, the signal of interest is not band limited or it is necessary to sample in a rate smaller than the Nyquist

limit. In this case, uniform sampling will undoubtedly produce aliasing.

In signal processing this problem is usually solved by applying an anti-aliasing filter. Since the periodic spectrum will overlap, to minimize the distortion effect, frequencies higher than ω_s are eliminated before sampling starts. This is accomplished by a low-pass filter known as the *anti-aliasing filter*. Figure 3.5 illustrates this procedure.

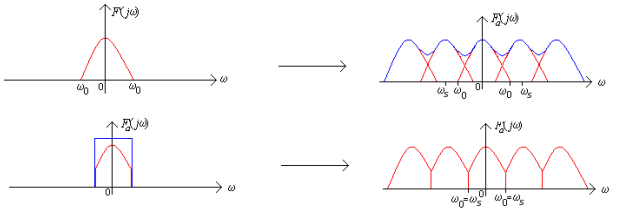


Figure 3.5: Anti-aliasing filter.

Let us now analyze this problem using the concepts of representation and reconstruction. There are two problems with undersampling. The first is that high frequency information is lost and the second is that the low frequencies are distorted due to spectrum superpositions. Since the first problem cannot be solved using uniform sampling at such a low rate, we will focus on avoiding the second.

The idea is to smoothen the signal before sampling, i.e., to extract high frequencies by applying a low-pass filter. Filtering the high frequency information and then projecting the result signal on a delayed *Dirac* function is equivalt to projecting the original signal on a small pulse waveform $v(t)$, as shown in Figure 3.6.

It is interesting to point out that this kind of sampling is actually common and easier to implement than the *Dirac* comb. A camera, for instance, when acquiring an image, sets for each pixel an average of the surrounding values. This is not only a good procedure because it minimizes distortion effects, but also because it is easier to implement on hardware. Point sampling in a camera doesn't gather much light, and therefore the signal to noise ratio will be inadequate. Moreover,

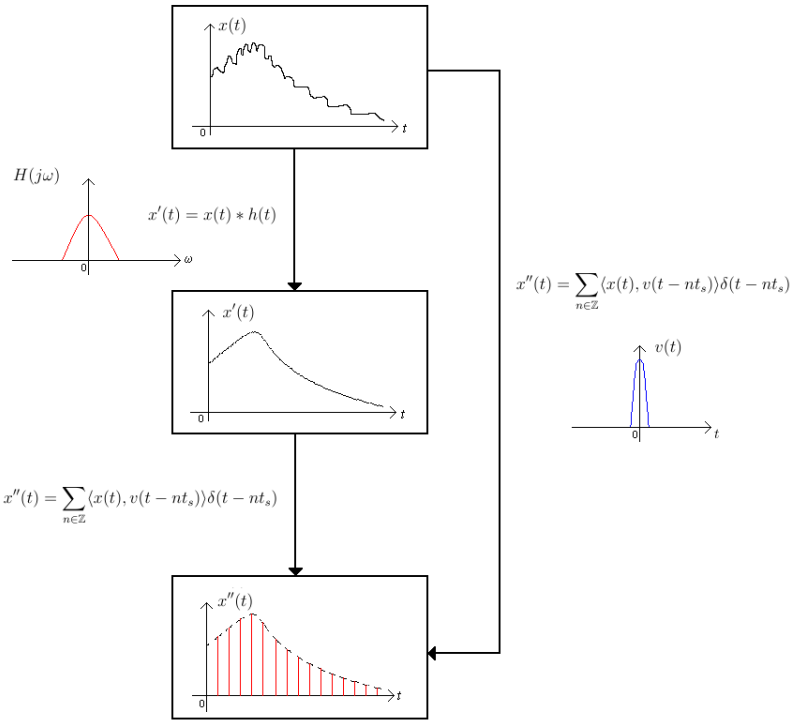


Figure 3.6: Undersampling.

sampling by *Diracs* would require a very precise sensing mechanism, and usually electron beams have Gaussian intensity functions.

Consider that $v(t)$ is a scaled *sinc* function. In this case, we are projecting the signal on a basis of delayed *sincs* $(v_n)_{n\in\mathbb{Z}}$, where

$$v_n(t) = \text{sinc}\left(\frac{t - nt_s}{t_s}\right)$$

This is, in fact, an orthogonal basis and, therefore, we can reconstruct the signal by

$$\hat{x} = \sum_{n\in\mathbb{Z}} \langle x, v_n\rangle v_n$$

If t_s is such that the Nyquist limit is respected, then reconstruction is exact ($\hat{x} = x$); however, if t_s is large, then we are taking a signal of a Hilbert space and projecting it in the subspace spanned by $(e_n)_{n\in\mathbb{Z}}$. Notice that this projection is taking a vector from a subspace of higher dimension and projecting it in a subspace of lower dimension and, therefore, this is a form of compression.

3.4 Approximation Theory

Being able to represent signals using different bases is usefull in signal processing because it allows to approximate certain types of signals using just a few vectors.

In this section we will exploit in a more formal way what was just illustrated by the undersampling problem.

3.4.1 Approximation on a Linear Basis

Given a signal x and an orthogonal basis $\mathcal{B} = (\phi_\lambda)_{\lambda\in\Gamma}$, an approximation projects x over M basis vectors

$$x_M = \sum_{n\in I_M} \langle x, \phi_n\rangle \phi_n \tag{3.1}$$

The choice of the M vectors can be done *a priori* or *a posteriori* (depending on the signal x). In the first case, the approximation is called linear and, in the second, non-linear.

Though linear approximations are simpler to implement, the distortion generated will highly depend on the original signal, whereas in the non-liner case we can adapt the projection vector to minimize the approximation error.

In this context, we can discuss DCT linear and non-linear compression studied in Section 2.6. The DCT involves projecting the signal into a basis that makes it sparse and the run-length coding involves choosing from this new basis the most significant vectors. In this non-linear procedure, we need to save each coefficient value and its 'position', which refers to the vectors of this new basis that are most important to represent the signal. In linear compression,

the significant vectors are known *a priori*, and we only need to store the coordinate values, which are the projections of the signal on each base vector.

3.4.2 Approximation on Overcomplete Dictionaries

Linear expansion in a single basis is not always efficient because the information will be diluted across the whole basis. In overcomplete dictionaries [13], however, we are able to express the same signal using a smaller number of coefficients. Mallat illustrated this idea [14] by comparing signal representations to language vocabularies. While a small vocabulary may be sufficient to express any idea, it will sometimes require the use of full sentences to replace unavailable words otherwise available in large dictionaries.

Therefore, a good compression scheme involves finding the best representation of an image using a redundant dictionary. It is noteworthy that a trade-off considering the dictionary's size must be analyzed because, while a big dictionary guarantees a small number of values necessary to represent a given signal, it also demands a large number of bits to specify each vector.

Due to redundancy there are, however, innumerable ways to represent the same signal. The intention of most of the developed techniques is to find a representation which concentrates the energy in a small number of coefficients.

What we are looking for is a sparse representation, i.e., a representation with a larger number of zero coefficients. We can reduce this problem to the one of finding, for a given N-dimensional signal x, a P-sized dictionary $\mathcal{D} = \{g_1, g_2, \ldots, g_P\}$, and a value M, $M < N < P$, the representation

$$x_M = \sum_{m=0}^{M-1} \alpha_{p_m} g_{p_m} \tag{3.2}$$

that minimizes $\|x - x_M\|$.

This problem, however, is combinatorial and NP-hard. Thus, a series of pursuit methods were developed to reduce computational complexity by searching efficient but non-optimal approximations. To

illustrate how the latter perform, we will overview two very popular algorithms.

Basis Pursuits

Basis pursuits [15] consists in solving the following convex optimization problem with inequality constraints

$$\min \|\boldsymbol{\alpha}\|_1, \text{ subject to } \sum_{p=0}^{P-1} \alpha_p g_p = x$$

where $\boldsymbol{\alpha}$ is a vector of dimension P containing the α_p coefficients.

This is more a principle than an algorithm, and there are many computational solutions to this problem, the most popular ones using linear programming.

The idea behind this technique is that the l_1-norm enhances sparsity, as will be discussed in Chapter 4.

Therefore a good approximation strategy results from extracting the M largest coefficients of the optimal P-sized $\boldsymbol{\alpha}$ vector.

Matching Pursuits

Matching pursuit [14] is a greedy algorithm that decomposes a signal into a linear expansion of waveforms that are selected from a redundant dictionary.

At each step, the dictionary element that best matches the signal structure is chosen and the projection of the signal on it is stored. This process is repeated M times using the residual which results from the subtraction.

The advantage of this technique is that it is less computationaly expensive than Basis Pursuits and very powerful in terms of performance. It also shares many interesting properties such as energy conservation and invertibility when $M = P$. However, since it maximizes the projection at each step without considering the overall signal structure, it is suboptimal.

Chapter 4

Compressive Sensing: An Overview

Up until now we have been following the *sample-then-compress* framework, i.e., for a given image, we find a sparse representation and then encode the significant coefficients. The shortcomings of this approach are that before a compressing scheme can be applied, the encoder must:

- store a large number of samples;
- compute all the transform coefficients; and
- find the locations of the large coefficients.

This is what usually happens in popular image acquisition instruments. Common digital cameras sample using a large number of mega-pixels, but store the images in a compressed form, for example, the JPEG standard. This indicates that we only need a small percentage of the measured coefficients to reconstruct the signal and, therefore, efficiency is lost.

This suggests that a smarter and cheaper method could be used to improve performance. In this context, compressive sensing appears. It involves sampling the original signal in a rate smaller than the Nyquist limit and reconstructing it by means of an optimization procedure.

In this chapter we will study the main concepts of this novel idea and how it first came to existence. We will leave the formalization of the theory involved for the next chapter.

4.1 Essential Aspects

What we want is to build an acquisition scheme that captures the image already in a compressed form. Consider the DCT based compression scheme. If we knew *a priori* which were the most significant DCT coefficients (consider, for instance, a linear compression scheme), we could then simply measure their values without the need of exploiting each pixel information.

Note that the word *sample* here has a new meaning. It refers no longer to point samples, but rather to more general linear measurements of the signal. Each measurement y_m in the acquisition system is an inner product of the signal x against a different test function ϕ_m (for example, a row of the DCT transform matrix)

$$y_1 = \langle x, \phi_1 \rangle \,, \quad y_2 = \langle x, \phi_2 \rangle \,, \quad \dots \quad , \quad y_M = \langle x, \phi_M \rangle$$

where M is the number of measurements.

However, as we have seen in the previous chapters, linear approximations usually have performances that are far from optimal, illustrating that this *a priori* knowledge is hard to obtain. Accordingly, though it is true that x is sparse in some domain, we can not know exactly which are the significant coefficients. Moreover, it is desirable to obtain a *nonadaptive* solution to the problem, so as to be able to use the same mechanism to capture information from any signal.

4.1.1 The Algebraic Problem

Let s be the signal represented in a sparse domain, i.e,

$$s = \Psi x$$

where x is the original signal and Ψ is a transformation that makes s sparse, for example, the DCT.

To take a small number of measurements is to multiply x by a fat[1] matrix Φ_Ω as shown in Figure 4.1, where each row is a measurement function ϕ_m.

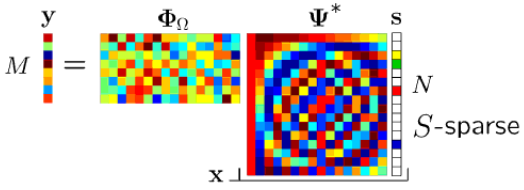


Figure 4.1: The acquisition matrix. (Extracted from [16].)

$$\begin{aligned} y &= \Phi_\Omega x \\ x &= \Psi^* s \iff s = \Psi x \\ y &= \Theta_\Omega s, \text{ where } \Theta_\Omega = \Phi_\Omega \cdot \Psi^* \end{aligned}$$

The reconstruction problem involves finding x so that $y = \Phi_\Omega x$, or, analogously, s so that $y = \Theta_\Omega s$. This problem, however, is ill posed because there is an infinite number of possible solutions. All the same, not all solutions satisfy the sparsity property of s and, therefore, a simple choice would consist of searching among all possible solutions the one that makes s the sparsest.

[1] We use the term *fat* to refer to a matrix where the number of rows exceeds the number of columns.

4.1.2 Sparsity and the l_1 Norm

Sparsity can be described by the l_0 norm

$$\|\alpha\|_{l_0} = \sharp\{i : \alpha(i) \neq 0\}$$

where $\sharp$ denotes the number of elements in the set.

Hence, the solution we want is

$$\min_x \|\Psi x\|_{l_0} \quad \text{subject to} \quad \Phi_\Omega x = y$$

Or, alternatively

$$\min_s \|s\|_{l_0} \quad \text{subject to} \quad \Theta_\Omega s = y$$

Yet, this problem is combinatorial and NP-hard; however it has been observed that sparse signals have small l_1 norms relative to their energy. We will motivate the relation between the l_0 and the l_1 norm by the 2-dimensional example in Figure 4.2.

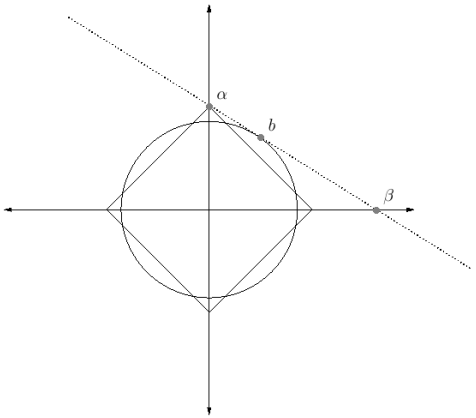


Figure 4.2: Sparsity and the l_1 norm.

Suppose we wish to find the signal s that has minimum l_0 norm, given that s respects a linear equation that constrains its position in $\mathbb{R}^2$ to the dotted line. Note that if we minimize the l_2 norm the

optimal solution will be given by $s = b$, which is not sparse and far from the l_0 solutions α and β. However, the l_1 minimization would result in $s = \alpha$, which is the exact solution we wanted.

The l_1 norm is convex, which makes optimization problem computationally tractable. Hence, all the following analyses and results will be given considering l_1 minimization.

4.1.3 The Recovery Algorithm

We can now understand the idea of compressive sensing in terms of its recovery algorithm. This theory involves undersampling a signal and then recovering it by the convex optimization problem

$$\min_s \|s\|_{l_1} \quad \text{subject to} \quad \Theta_\Omega s = y$$

Though we have understood why this is a good procedure, we still have to analyze its efficiency. How can we know for sure that the sparsest solution is the one that reconstructs the original signal s? What do we need to assume about the sensing matrix and the number of samples? What kind of results can we guarantee?

A series of theorems and definitions have been proposed to formalize this idea and to specify sufficient conditions that guarantee good results. These will be studied with some care in the following chapter. We will, nevertheless, take some time to introduce the first theorem proposed in this field. Though it is much weaker than the ones that will be considered in the future, it sheds light to many interesting ideas, as well as how the researchers first came up with CS.

4.2 The Fourier Sampling Theorem

4.2.1 The Magnetic Resonance Imaging Problem

The classical tomography problem consists in reconstructing a 2D image x from samples of its Fourier transform $\hat{x}(\omega)$ on the star shaped domain Ω illustrated by Figure 4.3.

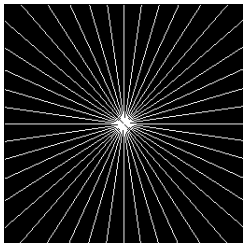
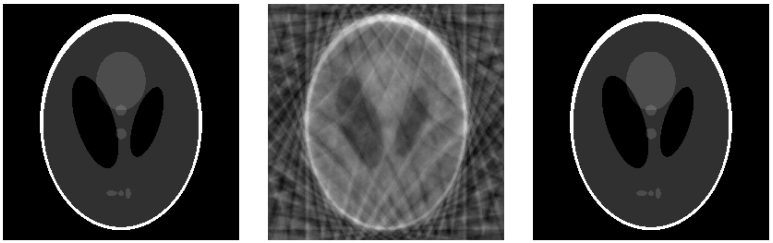


Figure 4.3: Sampling domain Ω in the frequency plane. (Extracted from [17].)

The most common algorithm, called *filtered backprojection*, assumes the non-sampled Fourier coefficients to be zero, in this way reconstructing the image with minimal energy. An image reconstructed by this procedure is shown in Figure 4.4 and illustrates how this mechanism has a bad performance.



(a) Original image. (b) Reconstruction with filtered backprojection. (c) Reconstruction using convex optimization.

Figure 4.4: First CS experiment applied to the Logan-Shepp phantom test image. (Extracted from [17].)

The solution proposed by [17] involves guessing the missing Fourier coefficients by means of a convex optimization based on the total-variation norm [2]

[2]The total-variation (TV) norm can be interpreted as the l_1-norm of the (ap-

$$\min_y \|y\|_{TV} \quad \text{subject to} \quad \hat{y}(\omega) = \hat{x}(\omega), \forall \omega \in \Omega$$

This was implemented with some numerical constants and resulted in the *exact* reconstruction of the original image. This surprising result led the researches to formalize a new sampling theorem.

4.2.2 New Sampling Theorem

Theorem 1 (Fourier Sampling Theorem [17])**.** *Assume that* $x \in \mathbb{R}^N$ *is S-sparse and that we are given* M *Fourier coefficients with frequencies selected uniformly at random*[3]*. Suppose that the number of measurements*[4] *obeys*

$$M \geq C \cdot S \cdot \log N$$

where C *is a relatively small constant. Then minimizing*

$$\min_s \|s\|_{l_1} \quad \textit{subject to} \quad \Theta_\Omega s = y$$

reconstructs x *exactly with overwhelming probability.*

This theorem differs from usual constraint specifications because it involves probabilistic results. The reason for this rather unorthodox approach is that we cannot obtain powerful results if we consider all measurable sets of size M, as there are some special sparse signals that vanish nearly everywhere in the Fourier domain.

To illustrate this, consider the discrete *Dirac* comb in $\mathbb{R}^N$, where N is a perfect square and the signal spikes are equally spaced by $\sqrt{N}$, as shown in Figure 4.5.

---

propriately discretized) gradient.

[3]In this case, we denote by Φ the $N \times N$ Fourier transform matrix and by Φ_Ω the fat matrix created by extracting N rows of Φ.

[4]It is common in literature to denote the set that supports the signal by T and the sampling set by Ω. Therefore, $S = |T|$ and $M = |\Omega|$.

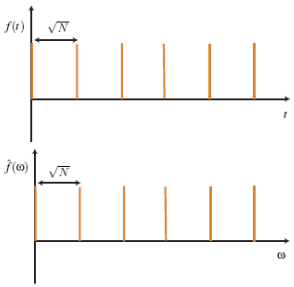


Figure 4.5: Comb filter. (Extracted from [18].)

Let Ω be the set of all frequencies but the multiples of $\sqrt{N}$. Then the observed signal in the Fourier domain is equal to zero and the reconstruction is identically zero. Note that the problem here does not really have anything to do with l_1 minimization once the signal cannot be reconstructed from its Fourier samples using any possible method.

Another interesting point to analyze is whether it would be possible to recover an arbitrary signal from less than $CS \log N$ samples using another algorithm. To motivate that this solution is tight we will use the same example of the *Dirac* comb. If x is as shown in Figure 4.5, to be able to recover it from $\hat{x}$, the observation set Ω must contain at least on spike. Supposing that

$$|T| < |\Omega| < \frac{N}{2} \iff \sqrt{N} < M < \frac{N}{2}$$

and choosing Ω uniformly at random, the probability that no spike is chosen is given by [17]

$$P = \frac{\binom{N-\sqrt{N}}{M}}{\binom{N}{M}} \geq \left(1 - \frac{2M}{N}\right)^{\sqrt{N}}$$

Therefore, for the probability of unsuccessful recovery to be smaller that $N^{-\delta}$, it must be true that

$$\sqrt{N} \cdot \log\left(1 - \frac{2M}{N}\right) \leq -\delta \log N$$

Since $M < \frac{N}{2}$, $\log\left(1 - \frac{2M}{N}\right) \approx -\frac{2M}{N}$ and we obtain the solution

$$M \geq Const \cdot \delta \cdot \sqrt{N} \cdot \log N$$

Hence, we conclude that the above theorem identifies a fundamental limit, and thus no recovery can be successfully achieved with significantly fewer observations.

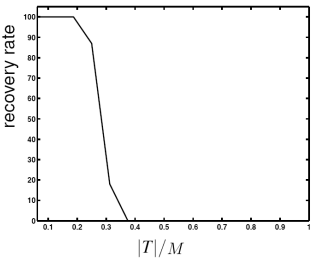


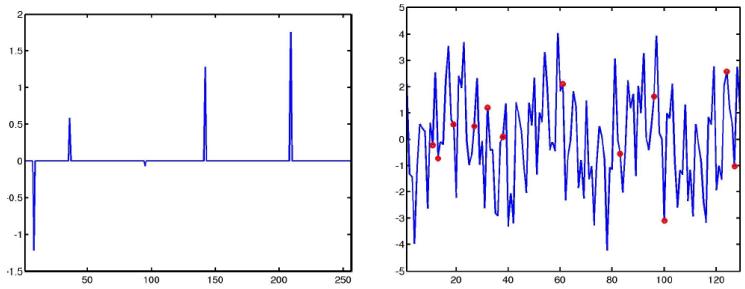
Figure 4.6: Numerical example. (Extracted from [18].)

A final illustration is given in Figure 4.6, which shows how the recovery rate decreases when the number of samples decreases in relation to the set that supports the signal. To build this graph signals of size $N = 1024$ were used and $|T|$ spikes were randomly placed.

4.2.3 Relashionship with Nyquist Sampling Theorem

Consider the signal in Figure 4.7(a). To follow the Nyquist sampling scheme, we would have to consider the size of the signal band in the frequency domain and sample it at twice that rate. In CS theory, on the other hand, we don't have to consider the signal band at all. All that is relevant is the number of nonzero coefficients which, multiplied by a log factor, gives us the sensing rate.

When sampling in the Fourier domain, the measurements are as shown by the dots of Figure 4.7(b), and reconstruction involves an



(a) Sparse signal in the time domain. (b) Fourier transform of (a) and CS measurements (dots).

Figure 4.7: CS intrepolation ploblem. (Extracted from [18].)

interpolation procedure that returns the curve. Notice, however, that this problem cannot be solved by a simple interpolation formula, as is done in the Nyquist sampling theorem with the *sinc* function. Instead, we reach the interpolated result by means of a convex optimization procedure that minimizes the l_1 norm of the sparse signal.

This problem was solved by [18] and the recovery is exact.

4.3 Uncertainty Principles

Though CS may seem like a great breakthrough, the basic principles around it have been known for quite some time. In fact, we can consider this novel idea as an extension of the theory about uncertainty principles.

We have already mentioned in our study of the Wavelet transform in Section 2.2.3 that a function and its Fourier transform cannot both be highly concentrated. We can extend this uncertainty principle to functions x that are not concentrated in an *interval*. Instead, if x is practically zero outside a measurable set T and its Fourier transform $\hat{x}$ is practically zero outside a measurable set Ω, then [19]

$$|T| \cdot |\Omega| \geq 1 - \delta$$

where δ is an oscillation parameter related to the *practically zero* definition.

In the discrete case, if $x \in \mathbb{R}^N$ has N_t nonzero components and $\hat{x}$ is not zero at N_ω, the uncertainty principle states that

$$N_t \cdot N_\omega \geq N$$

where the lower bound $N_t N_\omega = N$ is reached in the case where x is a *Dirac* comb. Note that this happens in the example shown in Figure 4.5, where $N_t = \sqrt{N}$ and $N_\omega = \sqrt{N}$.

In most common studies, uncertainty principles are used to prove that certain things are impossible, for example, obtaing good resolutions simultaneously in the time and frequency domains. However, in this approach, we make use of this theorem to allow recovery of signals despite amounts of missing information.

Donoho and Stark showed in [19] that it is possible to recover a bandlimited signal when sampled with missing elements. Consider that the signal x, where $\hat{x} \in \Omega$, is observed in the time domain but a subset T^c of the information is lost. Then the observed signal $r(t)$ is such that

$$r(t) = \begin{cases} x(t) + n(t), & \text{if } t \in T \\ 0, & \text{if } t \in T^c \end{cases}$$

where $n(t)$ is a noise signal.

It can be demonstrated that x can be recovered from r, provided that $|T^c||\Omega| < 1$.

Intuitively, consider the signal h, $\hat{h} \in \Omega$, completely concentrated on T^c. The problem of reconstructing x from r derives from the fact that x and $x + h$ cannot be distinguished and therefore the reconstruction error can be arbitrary large. However, such function h cannot exist because if it did the uncertainty principle would require $|T^c||\Omega| \geq 1$. Hence, a stable reconstruction to the above problem can be achieved.

4.4 Extensions

The practical relevance of Theorem 1 has two limitations. The first one is that it restricts the sampling domain to Fourier and we are not always at liberty to choose the types of measurements we use to acquire a signal. The second is that completely unstructured measurement systems are computationally hard.

In view of these shortcomings, a significant amount of effort has been given to make CS theory useful for practical applications. Not only have researches expanded this result, but they also described conditions that guarantee good performances in adverse situations.

Chapter 5

Compressive Sensing: Theoretical Aspects

In the previous chapter we introduced a sampling theory that embbeds compression. We will now provide some key mathematical insights underlying this new approach.

Two different results will be discussed:

- Basic CS - theory that stipulates constraints for the exact recovery of sparse signals.

- Robust CS - expansion of the former results to allow CS to be used in applications where the signal is not exactly sparse or the measurements are corrupted by noise.

This chapter also includes some important considerations for the design of efficient sensing matrices.

5.1 Basic CS

Basic CS deals with analyzing the constraints that guarantee perfect reconstruction by means of an l_1 optimization, considering that there exists a domain in which the signal x is S-sparse and that the acquired measurements are not corrupted by noise.

The first concept that needs to be extended from the discussed *Fourier Sampling Theorem* is that the domain where x is sparse and the domain where the samples are taken may vary in different applications, not necessarily being time and frequency. Therefore, it is of utmost importance to develop a way of determining if a sampling domain is efficient, given that the signal is sparse after it is multiplied by Ψ, where Ψ is, for example, a wavelet transform. [1]

[1] Notation review:

We use x to refer to an input signal and s to denote its S-sparse representation. T is the set that supports s and is of size $|T| = S$ and Ω is the random measurement subset of size $|\Omega| = M$.

We denote by Φ the matrix that spans $\mathbb{R}^N$, where each row is a measurement function ϕ_m to be applied to the signal x. Therefore, the sensing problem is

$$y = \Phi_\Omega x$$

where Φ_Ω is a fat matrix created by randomly selecting M rows of Φ. Since x is sparse in the Ψ domain, the sparse representation of x is given by

$$s = \Psi x$$

And therefore, since Ψ is unitary (orthonormal transform),

$$y = \Phi_\Omega \Psi^* s$$
$$\Rightarrow y = \Theta_\Omega s, \text{ where } \Theta_\Omega = \Phi_\Omega \Psi^*$$

We also denote $\Theta = \Phi\Psi^*$ and $\Theta_{\Omega T}$ is the submatrix created by extracting the columns of Θ_Ω corresponding to the indexes of T. Note that Θ is $N \times N$, Θ_Ω is $M \times N$, and $\Theta_{\Omega T}$ is $M \times S$.

5.1.1 Incoherence

Coherence [20] is a measurement of the correlation between the sensing waveforms ϕ_k and the waveforms where the signal is supposed to be sparse ψ_k. Assuming both have unit l_2 norm, the definition is as follows.

Definition 2 (Coherence between Ψ and Φ [21])**.**

$$\mu(\Phi, \Psi) = \sqrt{N} \max_{i,j} |\langle \phi_i, \psi_j \rangle| \quad , \quad \|\phi_i\|_{l_2} \quad \|\psi_i\|_{l_2} = 1$$

Note that $\mu(\Phi, \Psi)$ measures the minimum angle between the sensing waveforms and the sparsity waveforms. Therefore, if we look at the waveforms as vectors in R^N, then high incoherencies mean that these vectors are far apart, i.e., nearly orthogonal.

From linear algebra we get

$$1 \leq \mu(\Phi, \Psi) \leq \sqrt{N}$$

Demostration: The upper bound comes from the Cauchy-Schwarz inequality

$$|\langle \phi_i, \psi_j \rangle|^2 \leq \|\phi_i\|^2 \cdot \|\phi_j\|^2 \;\Rightarrow\; \mu(\Phi, \Psi) \leq \sqrt{N}$$

and the lower bound can be derived if we consider that Ψ is an orthogonal basis

$$\sum_j |\langle \phi_i, \psi_j \rangle|^2 = 1 \;\Rightarrow\; \max_j |\langle \phi_i, \psi_j \rangle| \geq \frac{1}{\sqrt{N}} \;\Rightarrow\; \mu(\Phi, \Psi) \geq 1$$

□

Therefore, the time and the frequency domains are maximally incoherent, since the Fourier basis $\psi_k(t) = \frac{1}{\sqrt{N}} e^{\frac{2\pi j k}{N}}$ and the canonical basis $\phi_k(t) = \delta(t - k)$ yield $\mu = 1$. This is very good because better results are achieved when coherence is small, i.e., when both domains are poorly correlated.

We can perceive this observation if we notice that sampling in the sparse domain directly returns many zero-valued coefficients. The

advantage of incoherence is that if we measure a series of random combinations of the entries, we learn something new about the sparse vector with every measurement.

We can also define incoherence based on the matrix Θ.

Definition 3 (Mutual Coherence [22])**.**

$$\mu(\Theta) = \sqrt{N} \max_{i,j} |\Theta_{i,j}|$$

Notice that this is equivalent to Definition 2

$$\Theta = \begin{bmatrix} \phi_1^T \\ \vdots \\ \phi_N^T \end{bmatrix} \begin{bmatrix} \psi_1^* & \dots & \psi_N^* \end{bmatrix} = \begin{bmatrix} \phi_1^T\psi_1^* & \dots & \phi_1^T\psi_N^* \\ \vdots & \ddots & \vdots \\ \phi_N^T\psi_1^* & \dots & \phi_N^T\psi_N^* \end{bmatrix}$$

And, since each row (or column) of Θ has necessarily an unitary l_2-norm [2], μ will take a value between 1 and $\sqrt{N}$.

In terms of the matrix Θ, μ can be interpreted as a rough measure of how concentrated the rows of Θ are. From the above comment we notice that if there is a coincident vector ϕ_i and ψ_j, the i^{th} row of Θ will be maximally concentrated, i.e., $\Theta_{i,j} = 1$ and $\Theta_{i,k} = 0, \forall k \neq i$. On the other hand, the best recovery possibility occurs if ϕ_i is spread out in the Ψ domain, i.e., when the row is diluted: $\Theta_{i,k} = \frac{1}{\sqrt{N}}, \forall k$.

5.1.2 Result Theorem

Theorem 2 ([20])**.** *Let Θ be an $N \times N$ orthogonal matrix and $\mu(\Theta)$ be as defined previously. Fix a subset T of the signal domain. Choose a subset Ω of the measurement domain of size M, and a sign sequence z on T uniformly at random. Suppose that*

$$M \geq C_0 \cdot |T| \cdot \mu^2(\Theta) \cdot \log(N)$$

[2] The rows have unitary l_2-norm if we consider Ψ orthonormal and the columns have unitary l_2-norm if we consider Φ orthonormal.

for some fixed numerical constant C_0. Then for every signal s supported on T with signs matching z, the recovery from $y = \Theta_\Omega s$ by solving

$$\hat{s} = \min_{s^*} \|s^*\|_{l_1} \quad \textit{subject to} \quad \Theta_\Omega s^* = y$$

Is exact ($\hat{s} = s$) with overwhelming probability.

Theorem 2 extends the previous *Fourier Sampling Theorem* with the exception that the latter holds for each sign sequence. The need to randomize the signs comes from an artifact that was used to demonstrate the thesis. It is highly probable that it still holds without this constraint, however researchers have not been able to prove this up until now [18].

We will not demonstrate this theorem here, but we will give two examples that serve as insights to its tightness.

To show that this is a fundamental limit, consider Ψ the time and Φ the frequency domain. Then, $\mu = 1$ and the above theorem results in the *Fourier Sampling Theorem*, which we have proven to be tight.

On the other hand, consider that Φ and Ψ are the same, i.e., $\mu^2(\Phi, \Psi) = N$ and we want to recover a signal that is 1-sparse. The theorem says that we actually need to measure every coefficient to guarantee recovery. This is intuitive because since each measurement informs only one of the ψ_k coefficients, unless we measure the nonzero coefficient, the information will vanish. Therefore, to reconstruct x with probability greater than $1-\delta$, we need to see all ϕ_k components.

The latter result is maintained without the need to assume $\Phi = \Psi$, as long as we consider both orthogonal. In fact, if there exists two coefficients i and j, such that $|\langle \phi_i, \psi_j \rangle| = 1$, then $\mu(\Phi, \Psi) = \sqrt{N}$ and the number of measurements needed to recover a 1-sparse signal x is N. To see this result intuitively, note that $\theta_{i,j} = 1$, $\theta_{i,k} = 0, \forall k \neq j$

and $\theta_{k,j} = 0, \forall k \neq j$. Therefore, $y = \Theta s$ can be rewritten as:

$$y = \begin{bmatrix} * & \dots & * & 0 & * & \dots & * \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ * & \dots & * & 0 & * & \dots & * \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ * & \dots & * & 0 & * & \dots & * \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ * & \dots & * & 0 & * & \dots & * \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ * \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Notice that unless ϕ_j is chosen, i.e., unless $j \in \Omega$ we will not obtain any information because $\Theta_\Omega s = 0$. Therefore, to guarantee recovery we must sample with the hole matrix $\Theta_\Omega = \Theta$.

5.2 Restricted Isometries

In this section, we will define strict conditions that when imposed to the matrix Θ guarantee that CS is efficient.

5.2.1 An Uncertainty Principle

Below is an intermediate result that follows directly from incoherence.

Theorem 3 ([18]). *Let Θ, T, and Ω be as in Theorem 2. Suppose that the number of measurements M obeys*

$$M \geq \cdot|T| \cdot \mu^2(\Theta) \cdot max\left(C_1 log|T|, C_2 \log\left(3/\delta\right)\right),$$

for some positive constants C_1, C_2. Then

$$P\left(\left\| \frac{N}{M} \Theta^*_{\Omega T} \Theta_{\Omega T} - I \right\| \geq 1/2\right) \leq \delta$$

The above equation means that all the eigenvalues of $\frac{N}{M}\Theta^*_{\Omega T}\Theta_{\Omega T}$ are between $\frac{1}{2}$ and $\frac{3}{2}$. To see that this is an uncertainty principle, let $s \in \mathbb{R}^N$ be a sequence supported on T, and suppose that

$\|\frac{N}{M}\Theta^*_{\Omega T}\Theta_{\Omega T} - I\| \leq 1/2$ (which is very likely the case). It follows that

$$\frac{1}{2}\cdot\frac{M}{N}\cdot\|s\|^2_{l_2} \quad\leq\quad \|\Theta_\Omega s\|^2_{l_2} \quad\leq\quad \frac{3}{2}\cdot\frac{M}{N}\cdot\|s\|^2_{l_2}$$

This asserts that the portion of the energy of s that will be concentrated on the set Ω is essentially proportional to M. Notice that $\|s\|^2_{l_2} = \|\Theta s\|^2_{l_2}$ and, therefore, we can rewrite the equation as

$$\frac{1}{2}\cdot\frac{M}{N}\cdot\|\bar{s}\|^2_{l_2} \quad\leq\quad \|\bar{s}_\Omega\|^2_{l_2} \quad\leq\quad \frac{3}{2}\cdot\frac{M}{N}\cdot\|\bar{s}\|^2_{l_2}$$

where $\bar{s} = \Theta s$ and $\bar{s}_\Omega$ is $\bar{s}$ restricted to set Ω, $\bar{s}_\Omega = \Theta_\Omega s$.

Hence, the relation says that the energy of the signal restricted of the set Ω is much smaller than the energy of the signal. This is an uncertainty relation because it means that if a signal is S-sparse (if the signal is concentrated on T), then it cannot be concentrated on the set Ω. If fact, this relation is quantized because there is a fixed value M/N to which the concentration in each domain is proportional.

Though usually uncertainty principles are considered bad, this one actually makes recovery possible. We can only take less measurements because the energy is diluted in the Φ domain and, thus, by taking random measurements, we are able to obtain a considerate amount of information about the signal.

5.2.2 The Restricted Isometry Property

Based on the intermediate result presented in Section 5.2.1, Candès and Tao defined in [22] the restricted isometry property. A refined approach appears in [23].

Definition 4 (Restricted Isometry Constant [23])**.** *For each integer $S = 1, 2, \ldots, N$ we define the S-restricted isometry constant δ_S of a matrix Θ_Ω as the smallest number such that*

$$(1-\delta_S)\|s\|^2_{l_2} \leq \|\Theta_{\Omega T}s\|^2_{l_2} \leq (1+\delta_S)\|s\|^2_{l_2}$$

for all S-sparse vectors.

The restricted isometry is a property of the measurement matrix Θ_Ω that refers to the existence and boundary of δ_S. The RIP establishes a condition which, if obeyed by Θ_Ω, guarantees recovery of sparse vectors. Notice that the constant δ_S is intrinsic to the structure of Θ_Ω and, therefore, by setting constraints to its size, we can quantify the efficiency of the sensing matrix.

The reason we call this RIP is straightforward: the energy of the signal restricted to the set Ω is proportional to the size of Ω. Nevertheless, some authors describe this as an Uniform Uncertainty principle (UUP). The relation to the uncertainty principles has already been established in Section 5.2.1 and involves guaranteing that the signal cannot be concentrated simultaneously on both sets. This condition, however, is stronger than Theorem 3 because it is valid for every set T (every S-sparse vector). Hence, it is called *uniform*.

We will now try to illustrate what this property means in terms of linear algebra. By undersampling we get an ill posed problem and, from the infinite number of solutions, we are going to choose the one that makes s the sparsest. However, how can we know for sure that this solution is unique? How can we force that there will be no other solution that is as sparse as s or sparser? As mentioned earlier, we can only guarantee this if we have incoherent measurements, i.e., if the sensing matrix has some properties.

First of all, note that if Θ_Ω has linear dependent columns, two different sparse vectors can result in the same measurement.

Demostration:

$$\Theta_\Omega \cdot c = \sum_{j=1}^{N} c_j \cdot v_j, \quad \text{where } v_j \text{ is a column of } \Theta_\Omega$$

Let $c \neq 0$ be a vector such that $\sum_{j=1}^{N} c_j \cdot v_j = 0$ (this is always possible because the columns are l.d.). Then, if we partition the set of indexes $I = \{1, 2, \ldots, N\}$ into two disjoint sets $I_1 \cup I_2 = I$, it results that

$$\Theta_\Omega \cdot c = \sum_{j \in I_1} c_j \cdot v_j = \sum_{j \in I_2} -c_j \cdot v_j$$

And we measure the vectors a and b defined as follows

$$a = \begin{cases} a_j = c_j, \text{ if } j \in I_1 \\ a_j = 0, \text{ if } j \in I_2 \end{cases} \qquad b = \begin{cases} b_j = -c_j, \text{ if } j \in I_2 \\ b_j = 0, \text{ if } j \in I_1 \end{cases}$$

by Θ_Ω, we obtain the same result $y = \Theta_\Omega a = \Theta_\Omega b$. □

Hence, we conclude that the existence of linear dependent columns lead to equivalent measurements for two different input signals and, therefore, recovery can only be guaranteed if the columns are linear independent. However, we cannot impose linear independence because the matrix is fat, i.e., the number of columns is larger than the number of rows. Here again sparsity comes to the rescue. All we need is that the columns of Θ_Ω behave like an l.i. system for sparse linear combinations involving no more than S vectors. That is exactly what the RIP gives us, it says that for every T of size no bigger than S, $\Theta_{\Omega T}$ is approximately orthogonal.

It can be easily shown that, if $\delta_{2S} < 1$ for $S \geq 1$, for any T such that $|T| \leq S$, there is a unique s with $\|s\|_{l_0} \leq S$ and obeying $y = \Theta_\Omega s$.

Demostration: Suppose for contradiction that there are two S-sparse signals s_1 and s_2 such that $\Theta_\Omega s_1 = \Theta_\Omega s_2 = y$. Then, let h be such that $h = s_1 - s_2$. It is clear that h is $2S$-sparse and that

$$\Theta_\Omega h = \Theta_\Omega(s_1 - s_2) = \Theta_\Omega s_1 - \Theta_\Omega s_2 = 0.$$

The RIP states that

$$(1 - \delta_{2S})\|h\|^2 \leq \|\Theta_{\Omega T} h\|^2 = 0$$

Since $\delta_{2S} < 1$, $(1-\delta_{2S}) > 0$ and, therefore we must have $\|h\|^2 = 0$ contradicting the hypothesis that s_1 and s_2 were distinct. □

We should point out that these results are general in the sense that they are not considering that the recovery algorithm is based on the l_1 norm.

5.2.3 Result for Basic CS

Theorem 4 ([23, 24])**.** *Let s be an S-sparse signal supported on T and measured by Θ_Ω. Assume that the restricted isometry constant for the matrix $\Theta_{\Omega T}$ is such that $\delta_{2S} < \sqrt{2} - 1$. Then the solution $\hat{s}$*

to

$$\hat{s} = \min_{s^*} \|s^*\|_{l_1} \quad \textit{subject to} \quad \Theta s^* = y$$

is exact, i.e., $\hat{s} = s$.

This result is deterministic, not involving a non-zero probability of failure and is also universal in the sense that all sufficiently sparse vectors are exactly reconstructed from $\Theta_\Omega s$.

We can interpret this result as a slightly stronger condition that is related to the l_1 norm reconstruction strategy. In fact, it can be shown that for

- $\delta_{2S} < 1$ solution to the l_0 norm is unique; and
- $\delta_{2S} < \sqrt{2} - 1$ solution to the l_0 norm and the l_1 are unique and the same.

5.3 Robust CS

Most signals are not usually sparse; they can be approximately sparse or have an exponential decay. Moreover, measurements are not usually perfect and some level of noise is added to them. For CS to be suitable for real application it must be robust to these kinds of inaccuracies. Therefore, a lot of effort was made to set conditions and theorems to expand the CS theory.

In this section, we will present theorems that make CS robust to applications when:

- the signal is not exactly sparse; or
- measurements are corrupted by noise.

5.3.1 Signals that are not Exactly Sparse

In general we cannot assume that images are sparse in a specific domain. However, they are compressible in the sense that, after the DCT or Wavelet transform, the coefficients decay rapidly, typically like a power law.

In this case, if x is an image, $s = \Psi x$ is only approximately sparse, and, therefore, we denote by s_S the best S-sparse approximation of s, i.e., the result obtained when we force the $N - S$ smallest coefficients of s to be zero.

The following theorem evaluates the performance of CS in this scenario.

Theorem 5 ([24]). *Assume that s is approximately sparse and let s_S be as defined above. Then if $\delta_{2S} < \sqrt{2} - 1$, the solution $\hat{s}$ to*

$$\hat{s} = \min_{s^*} \|s^*\|_{l_1} \quad \textit{subject to} \quad \Theta_\Omega s^* = y$$

obeys

$$\|\hat{s} - s\|_{l_1} \leq C \cdot \|\hat{s} - s_S\|_{l_1}$$

and

$$\|\hat{s} - s\|_{l_2} \leq C_0 s^{-1/2} \cdot \|\hat{s} - s_S\|_{l_1}$$

for reasonable values of the constant C_0.

Roughly speaking, the theorem says that CS recovers the S largest entries of s. Notice that, in the particular case when s is S-sparse, $\|\hat{s} - s_S\| = 0$ and the recovery is exact.

This result has the following desired properties:

- it is a deterministic statement and there is no probability of failure;
- it is universal in that it holds for all signals; and
- it holds for a wide range of values of S.

Again, the demonstration of the above theorem is not the objective of this section and, therefore, will not be presented here. For the interested reader, we recommend [24, 25].

5.3.2 Signals that are Corrupted by Noise

Another very import and realistic scenario to consider is when the acquired data is corrupted with noise, i.e.,

$$y = \Phi x + n$$

where n is an unknown noise contribution bounded by a known amount $\|n\|_{l_2} \leq \epsilon$.

The property that will allow the method to be applicable is *stability* [25]: small changes in the observations should result in small changes in recovery. Hence, considering the undersampling problem, the best result we can hope for is a reconstruction error proportional to ϵ.

Demostration: [25] Consider the best possible condition in which we know *a priori* the support T of s_S. In this case, we can reconstruct $\hat{s}$ by a Least-Squares method, for example:

$$\hat{s} = \begin{cases} (\Theta^*_{\Omega T}\Theta_{\Omega T})^{-1}\Theta^*_{\Omega T}y & \text{on } T \\ 0 & \text{elsewhere} \end{cases}$$

and suppose that no other method would exhibit a fundamentally better performance. Therefore,

$$\hat{s} - s_S = (\Theta^*_{\Omega T}\Theta_{\Omega T})^{-1}\Theta^*_{\Omega T}n$$

and if the eigenvalues of $\Theta^*_{\Omega T}\Theta_{\Omega T}$ are well behaved, then

$$\|\hat{s} - s_S\|_{l_2} \approx \|\Theta_{\Omega T}n\|_{l_2} \approx \epsilon.$$

□

Therefore, the result we are searching for is a bound for Θ that guarantees that the reconstructed $\hat{s}$ obeys

$$\|\hat{s} - s_S\|_{l_2} \leq C_1\epsilon \tag{5.1}$$

for a rather small constant C_1.

This can be achieved by minimizing the l_1 norm and considering the constraint $\|\Theta_\Omega s - y\| \leq \epsilon$.

Theorem 6 ([24]). *Assume that* $y = \Theta_\Omega s + n$ *where* $\|n\|_{l_2} \leq \epsilon$. *Then if* $\delta_{2S} < \sqrt{2} - 1$, *the solution* $\hat{s}$ *to*

$$\hat{s} = \min_s \|s\|_{l_1} \quad \textit{subject to} \quad \|\Theta_\Omega s - y\|_{l_2} \leq \epsilon$$

obeys

$$\|\hat{s} - s\|_{l_2} \leq C_0 S^{-1/2} \cdot \|\hat{s} - s_S\|_{l_1} + C_1 \epsilon \tag{5.2}$$

for reasonable values of the constant C_0 *and* C_1.

It is noteworthy that the reconstruction error is a superposition of two factors: the errors that yield from sparsity approximation and the error that results from the additive noise.

For the reader interested in the proofs of Theorems 5 and 6 we recommend [22, 24].

5.4 Design of Efficient Sensing Matrices

It is, of course, of great importance to have matrices that preserve the RIP. Given a sensing matrix Φ, the calculus of the associated restricted isometry constant is NP hard and thus testing this property at each acquisition is unfeasible. We can, however, determine some measurement ensembles where the RIP holds.

The actual problem is to design a fat sensing matrix Θ_Ω, so that any subset of columns of size S be approximately orthogonal. Here, randomness re-enters the picture because setting a deterministic Θ_Ω may be a very difficult task (especially considering large values of S), but it can be easily shown [22] that trivial random structures perform quite well.

Interestingly, the high dimensionality of the usually handled signals also gives a positive contribution. It can be shown [26] that if N is large, a small set of randomly selected vectors in $\mathbb{R}^N$ will be approximately orthogonal.

The following results obtained by [22, 25] provide several examples of matrices that obey RIP.

Theorem 7 (Gaussian Matrices). *Let the entries of* Θ_Ω *be i.i.d., Gaussian with mean zero and variance* $1/M$. *Then the RIP holds with overwhelming probability if*

$$S \leq C \cdot M/\log(N/M)$$

for a relatively small constant C.

Theorem 8 (Random Projections). *Let* Θ_Ω *be a random Gaussian matrix whose rows were orthonormalized. Then the RIP holds with overwhelming probability if*

$$S \leq C \cdot M/\log(N/M)$$

for a relatively small constant C.

A measurement using this matrix involves projecting the signal on an orthogonal subspace which was chosen uniformly at random. Notice that the result of Theorem 7 is the same as Theorem 8 because, essentially, we have the same Gaussian matrix.

Theorem 9 (Binary Matrices). *Let the entries of* Θ_Ω *be independent taking values* $\pm 1/\sqrt{M}$ *with equal probability. Then the RIP holds with overwhelming probability if*

$$S \leq C \cdot M/\log(N/M)$$

for a relatively small constant C.

This case is also very similar to Theorem 7. However, it measures the correlation between the signal and random sign sequences instead of the correlation between the signal and white noise.

Theorems 7, 8 and 9 can be extended to several other distributions, but we will not present them here. Instead, we will focus on a much stronger result.

Theorem 10 (General Orthogonal Measurement Ensembles)**.** *Let* Θ *be an orthogonal matrix and* Θ_Ω *be obtaineed by selecting* M *rows from* Θ *uniformly at random. Then the RIP holds with overwhelming probability if*

$$S \leq C \cdot \frac{1}{\mu^2} \cdot \frac{M}{(\log N)^6}$$

for a relatively small constant C*.*

Theorem 10 is very significant because, as we have mentioned before, in many applications the signal is not sparse in the time domain, but rather in a fixed orthonormal basis Ψ. Therefore, this theorem guaranties that if we can determine an orthogonal matrix Φ such that $\mu(\Phi, \Psi)$ is small[3], then recovery is exact when the measurements are taken with Φ_Ω.

This result is not trivial and certainly not optimal, but researchers have been unable to improve it up until now [18].

[3]This is equivalent to setting $\Theta = \Phi\Psi^*$ and forcing $\mu(\Theta)$ to be small.

Chapter 6

Experiments

In this chapter we will verify CS theory by means of examples in image acquisition.

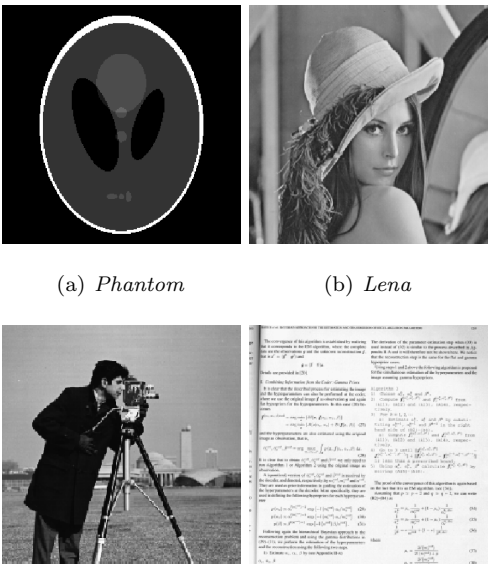
The different acquisition approaches will be evaluated in terms of their peak signal to noise ratios (PSNR) for different amounts of measurements, M.

The source code used to generate the following results is available at www.impa.br/∽aschulz/CS and a detailed explanation on how to reproduce them is given at the end of this chapter.

6.1 Experimental Setup

CS investigations were made on four different images of size $N = 256^2 = 65536$, which differ in terms of both sparsity and high energy coefficient distribution in the frequency domain (see Figure 6.1). Since *Phantom* is piecewise constant and *Lena* is smooth, the energy is mostly concentrated on the high frequency coefficients. On the other hand, since *Text* is an image with abrupt intensity variations, its energy is spread along almost all the DCT basis. Middling, *Camera man* has an intermediate energy spread, displaying strong inten-

sities at some DCT diagonals which correspond to the sharp image lines.



(a) *Phantom* (b) *Lena*

(c) *Camera man* (d) *Text*

Figure 6.1: Test images.

Since the images are stored in the computer as a matrix of pixels, we simulated acquisition by means of measurements that involve linear combinations of these pixels.

The following recovery strategies were considered:

DCT-l_1-N Random Noiselet measurements followed by minimization of the l_1-norm of the image's DCT;

B_DCT-l_1-N Random Noiselet measurements followed by minimization of the l_1-norm of the image's block DCT;

DWT-l_1-N Random Noiselet measurements followed by minimization of the l_1-norm of the image's DWT;

SVD-l_1-N Random Noiselet measurements followed by minimization of the l_1-norm of the image's SVD;

TV-N Random Noiselet measurements followed by minimization of the image's TV-norm;

DCT-l_2-L Linear DCT compression scheme;

B_DCT-l_2-L Linear block DCT compression scheme; and

DCT-l_1-LN Linear DCT and random Noiselet measurements followed by minimization of the l_1-norm of the image's DCT

In strategies DCT-l_1-N, B_DCT-l_1-N, DWT-l_1-N, SVD-l_1-N and DCT-l_1-N measurements are taken by choosing at random M waveforms of an $N \times N$ Noiselet transform [27]. Such measurements were chosen because they are highly incoherent with the considered sparse domains and the RIP tends to hold for reasonable values of M. In addition, the matrix created is orthogonal and self-adjoint, thus being easy to manipulate. Below is an illustration of the measurement matrix Φ for $N = 4$.

$$\Phi = \frac{1}{2} \cdot \begin{bmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}$$

Recovery of DCT-l_1-N, B_DCT-l_1-N, DWT-l_1-N and SVD-l_1-N are based on Equation

$$\hat{s} = \min_s \|s\|_{l_1} \quad \text{subject to} \quad \|y - \Phi_\Omega \Psi^* s\|_{l_2} \leq \epsilon, \tag{6.1}$$

where Φ_Ω is the Noiselet matrix Φ restricted to $M = |\Omega|$ randomly selected rows and Ψ is the matrix that transforms the signal into the sparse representation (DCT, block DCT, DWT and SVD).

The efficiency of each strategy is related to how sparse the images are in the considered domain. The DCT and the Wavelet domains were chosen because of their widespread use in image compression standards. We considered taking the DCT transform of the entire image as well partitioning the array of pixels into blocks of size 8×8

(this value was chosen because of its use in the JPEG standard). In addition, since most published theorems relate to orthogonal rather than to the more efficient biorthogonal basis, we used an orthonormal Wavelet basis (Coiflet with 2 vanishing moments).

Since sparsity plays a very significant role in determining CS efficiency, we compare results when considering Ψ the SVD, which gives a very accurate sparse representation. This technique requires knowledge of the SVD basis, that is calculated from the whole image information (not available in CS) and requires a large data rate for transmission (which is not taken into account). Nevertheless we used such results as upper bounds that, although loose, give interesting insights into performance limitations.

In many recent publications [17, 28], CS researchers have used the total variation (TV) norm, which can be interpreted as the l_1-norm of the (appropriately discretized) gradient. Applied to images, the TV-norm minimization favors a certain smoothness that is usually found in natural and manmade pictures without penalizing discontinuous features and is, therefore, very effective. In strategy TV-N, the image is reconstructed by solving the following convex optimization problem:

$$\hat{x} = \min_x \|x\|_{TV} \quad \text{subject to} \quad \|y - \Phi_\Omega x\|_{l_2} \leq \epsilon. \tag{6.2}$$

In strategies DCT-l_2-L and B_DCT-l_2-L, measurements are taken by obtaining the first M DCT coefficients (according to the diagonal zigzag scanning pattern described in Section 2.5.1) and recovery is done by setting to zero the unknown values and then applying the inverse DCT transform. It is relevant to compare these acquisition strategies with CS because they are also nonadaptive, in the sense that the measurement functions are the same for every considered image.

We also evaluated an alternative acquisition scheme suggested in [28] that combines strategies DCT-l_1-N and DCT-l_2-L. In strategy DCT-l_1-LN, we sense the first thousand linear DCT coefficients (i.e., the most important ones according to the zigzag scanning pattern) and $M - 1k$ (where $k = 10^3$) Noiselet waveforms (chosen at random as in strategy DCT-l_1-N). Recovery is based on the minimization of the l_1-norm of the image's DCT.

6.1.1 Implementation Aspects

The experiments were implemented in `MATLAB` and the l_1-`Magic` [29] toolbox was used to solve the optimization problems that recover the sensed image.

The Wavelet basis was generated using the `WAVELAB` [30] package and the Noiselet basis using an algorithm made available by Romberg [28].

Computational Errors

Due to a great number of varying parameters we had trouble configuring the optimization routine that solves Equation

$$\hat{s} = \min_{s} \|s\|_{l_1} \quad \text{subject to} \quad y = \Phi_\Omega \Psi^* s \tag{6.3}$$

in the `l1- Magic` toolbox. Nevertheless, results converge very well using the function that solves Equation 6.1. Since, to the best of our knowledge, there is no significant difference between both procedures, we implemented recovery by solving Equation 6.1 even in the cases where images are strictly sparse.

When measurements are uncorrupted by noise, we assumed $\epsilon = 10^{-3}\|y\|_{l_2}$ and solved the quadratically constrained convex optimization problem. In these examples, results improve (in the sense that higher PSNRs are reached) as we reduce ϵ. Nevertheless, we were able to induce from a few tests that the Measurement $\times$ PSNR curve format is the same for varying values of ϵ. We also observe that, while the improvement is expressive when the images are strictly sparse in the reconstruction domain, reducing ϵ leads to no significant improvement when we can only guarantee approximate sparsity. This phenomenon can be explained by the distortion provoked by the absence of sparsity which overcomes the computational errors, making the adjustment of ϵ ineffective.

In the cases where noise is added to the acquired measurements the value of ϵ must be proportional to a bound on the noise contribution. In these scenarios, the parameter ϵ was chosen according to a series of experiments and varies according to the error size. Calculating the PSNR for varying values of ϵ an optimal value is observed.

Under this value the best solution is outside the convex set bounded by the constraints and above it the solution is less exact.

6.2 Basic CS

To evaluate applications on image compression for Basic CS, it was necessary to force sparsity in the DCT representation of the images. Therefore, for $S = 3.5k$, $6k$, $10k$, and $14k$ (where $k = 10^3$) we selected the $N - S$ smallest DCT coefficients of each image and set them to zero in order to generate the image that would be compressed.

Figure 6.2 shows the results obtained for different sparse representation of *Lena* and acquisition strategies DCT-l_1-N, DCT-l_2-L and DCT-l_1-LN[1]. The first meaningful characteristic that we observe from the results is that compressive sampling routines start to have good performances after a specific number of measurements are taken. This threshold can be associated with the number of samples set by Theorem 2. Notice that this borderline depends linearly on the sparsity of the signal. Comparing the graphs in Figure 6.2 we see that as S (the number of nonzero coefficients) increases, the threshold above which CS is efficient also increases.

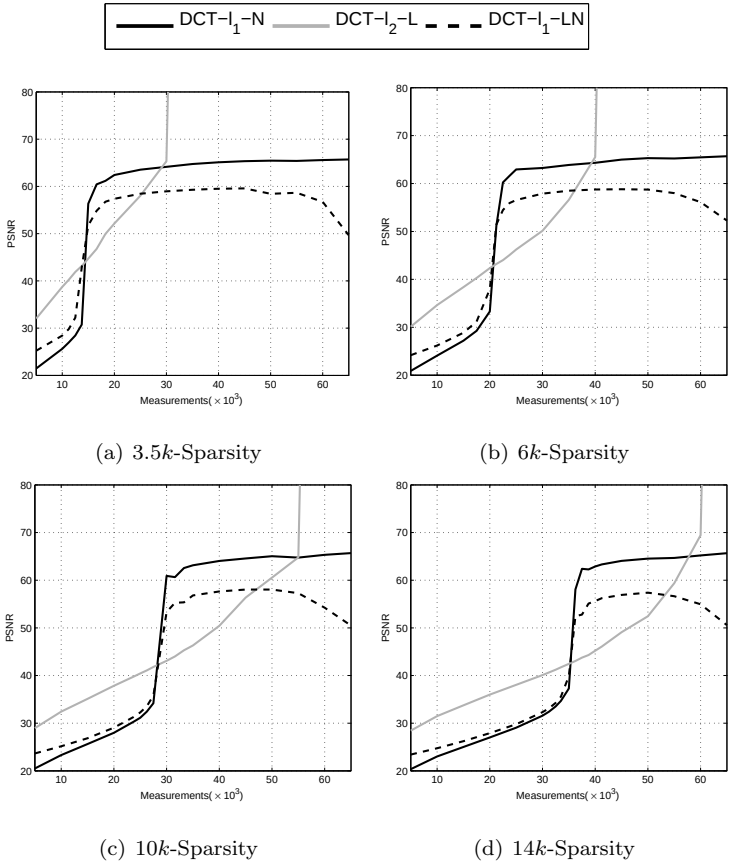
We calculated the coherence by (see Equation 1 for a definition of Θ)

$$\mu(\Theta) = \sqrt{N} \max_{i,j} |\Theta_{i,j}|$$

and obtained $\mu(\Theta) = 2.82$ for strategy DCT-l_1-N, while $\mu(\Theta) = \sqrt{N} = 256$ for strategy DCT-l_2-LN.

Therefore, although the thresholds for both strategies are essentially the same, the coherence of one is almost a hundred times larger than the other. This may strike the reader as a contradiction to the tightness of Theorem 2. Notice, however, that Θ_Ω is not orthogonal in strategy DCT-l_2-LN and thus the theorem cannot be applied in this particular example.

[1] In each graph the PSNR (peak signal-to-noise ratio between the sparse version of the image and the compressed reconstruction) versus the number of measurements is shown.



(a) $3.5k$-Sparsity (b) $6k$-Sparsity

(c) $10k$-Sparsity (d) $14k$-Sparsity

Figure 6.2: Results for different sparse versions of test image *Lena* and CS strategies DCT-l_1-N, DCT-l_2-L and DCT-l_2-LN.

It is also relevant to point out that before the boundary, strategy DCT-l_2-LN performs better than DCT-l_1-N and this tendency is not sustained when CS theory starts to operate. This result can be justified by the fact that, when taking a small number of samples the knowledge of the low frequency coefficients adds more information to

the signal than random measurements. In fact, the best acquisition strategy in this region is the linear DCT.

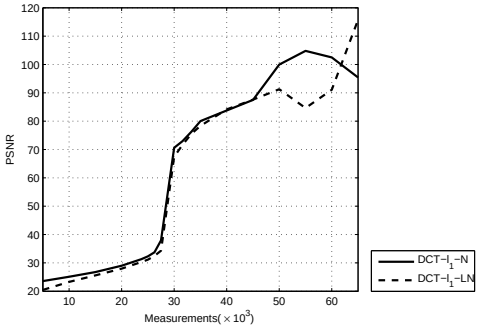
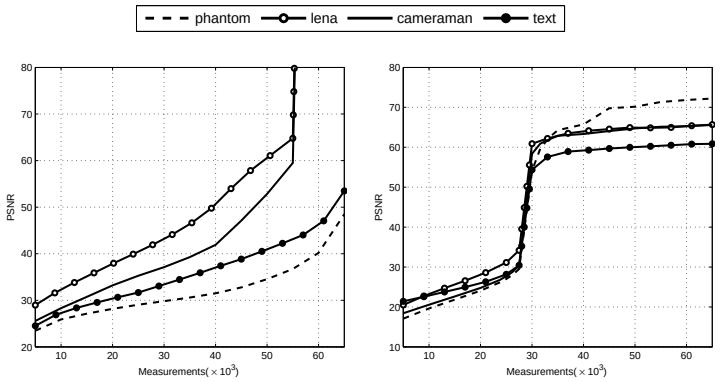


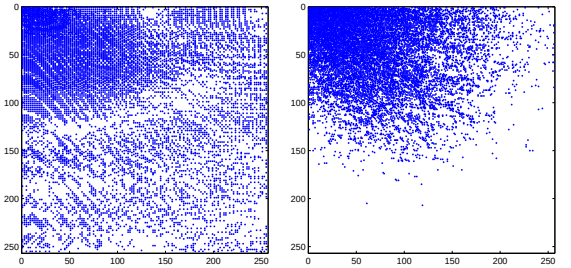
Figure 6.3: Recovery of the $10k$-sparse representation of *Lena* with $\epsilon = 0.001$ for strategy DCT-l_1-N and $\epsilon = 0.1$ for strategy DCT-l_2-LN.



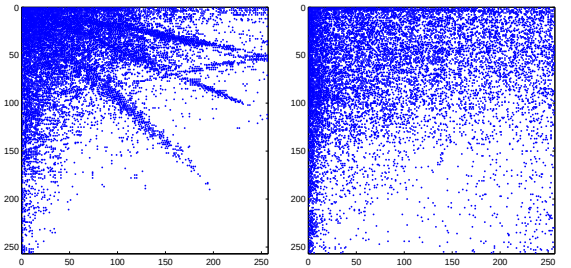
(a) Linear Compression (b) Compressive Sensing

Figure 6.4: Results for strategies DCT-l_2-L (on the left) and (on the right) for the $10k$-sparse representation of the four test images.

A very important comment is that, although it may seem that for M higher than the threshold strategy DCT-l_1-N performs better than DCT-l_2-LN, this is not true. We should consider that after the threshold the signal is perfectly reconstructed and what we see are measurement errors. To illustrate this point, we plotted in Figure 6.3 the recovery of the $10k$-sparse image *Lena* for very small values of ϵ. Notice that the oscillation for high values of M supports the idea of additional computational errors.



(a) *Phantom* (b) *Lena*



(c) *Camera man* (d) *Text*

Figure 6.5: Spectral distrifution of the $10k$-sparse representation of the test images.

Figure 6.4(a) shows results for strategy DCT-l_2-L on the $10k$-sparse representation of the four test images and Figure 6.4(b) shows results for strategies DCT-l_1-N on the $10k$-sparse representation of

the four test images. It is noteworthy (see Figure 6.4(b)) that CS performance does not vary between the tested images. We have discussed that the energy distribution of the considered images are quite different and Figure 6.5 confirms this by illustrating the distribution of the nonzero coefficients in the DCT domain when $10k$-sparsity is forced.

Nevertheless, different spectral distributions do not affect CS. This is because the algorithm depends on how sparse the signal is, but is nonadaptive in the sense that the position of the significant coefficients is irrelevant. Comparing Figures 6.4(a) and 6.4(b) we observe that while performance of linear compression schemes highly depend on the spectral distribution of the images, these differences do not influence CS recovery.

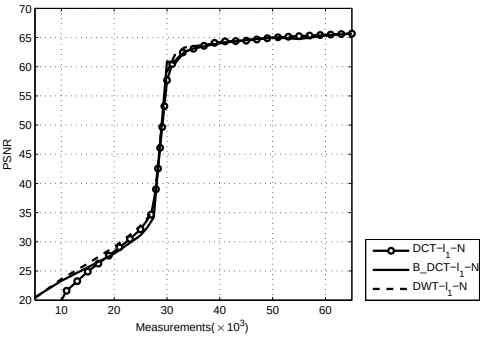


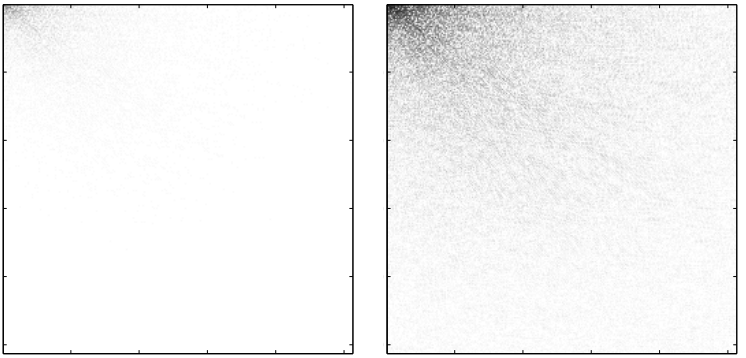
Figure 6.6: Results for the $10k$-sparse representation of *Lena* and strategies DCT-l_1-N, B_DCT-l_1-N DWT-l_1-N and SVD-l_1-N.

From Figure 6.6 we observe that performance is practically the same for the $10k$-sparse representation of *Lena* and strategies DCT-l_1-N, B_DCT-l_1-N, DWT-l_1-N and SVD-l_1-N. This is because in each case, we imposed sparsity in the specific domain in which reconstruction would take place and the incoherence is very similar in all acquisition schemes.

6.3 Sparsity Errors

We have argued that images can be well approximated by a sparse represented with little loss in terms of human perception. Nevertheless, the images are not strictly sparse in these domains. This observation is supported by Figure 6.7, that shows the DCT transform of test image *Lena*. While in 6.7(a) it seems sparse, the method for stretching contrasts used in 6.7(b), enhances the presence of small but nonzero coefficients.

In this section we aim at testing CS when strict sparsity cannot be assumed. We used the original images (without imposing sparsity) and evaluated CS acquisition strategies by measuring the PSNR for different number of measurements.



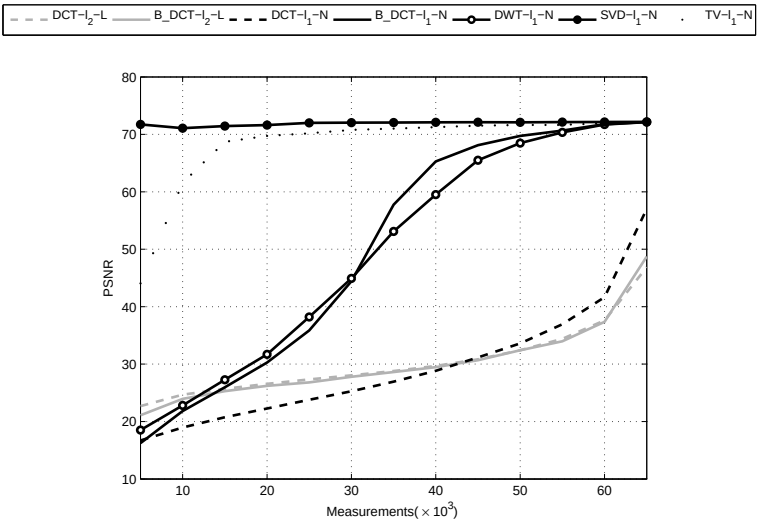
(a) (b)

Figure 6.7: Different visualizations of *Lena*'s DCT.

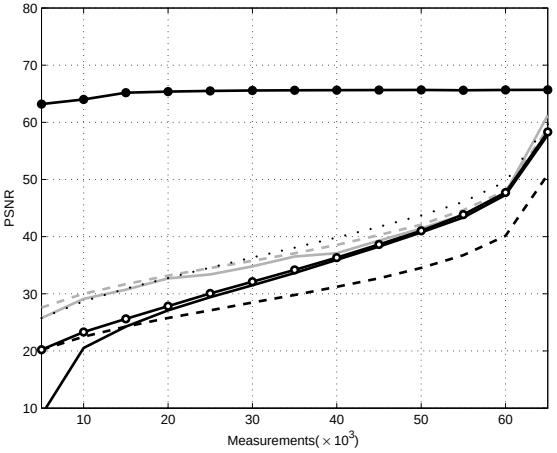
In Figures 6.8 and 6.9 we compare strategies DCT-l_1-N, B_DCT-l_1-N, DWT-l_1-N, SVD-l_1-N, DCT-l_2-L and B_DCT-l_2-L for all four test images.

From the results, we conclude that CS performance depends very strongly on the choice of the basis that best represents the signal. Results improve as we move from DCT to Wavelet transforms and CS is very efficient when the SVD basis is used.

It must be reinforced that the reconstruction method based on the

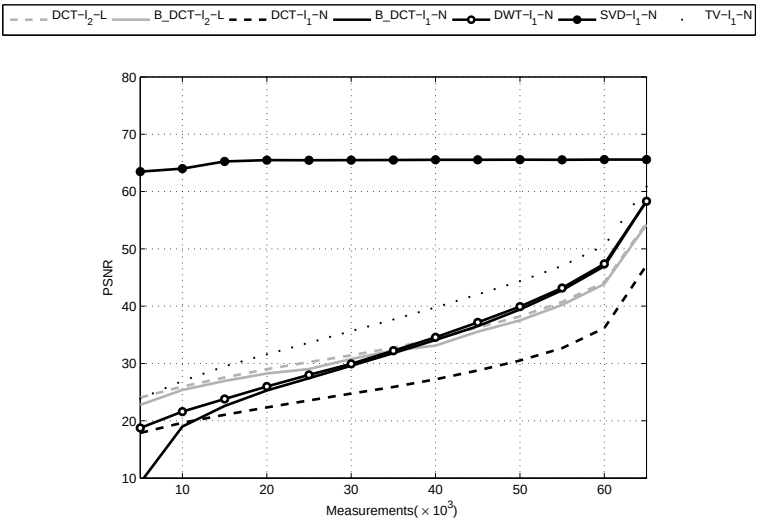


(a) *Phantom*

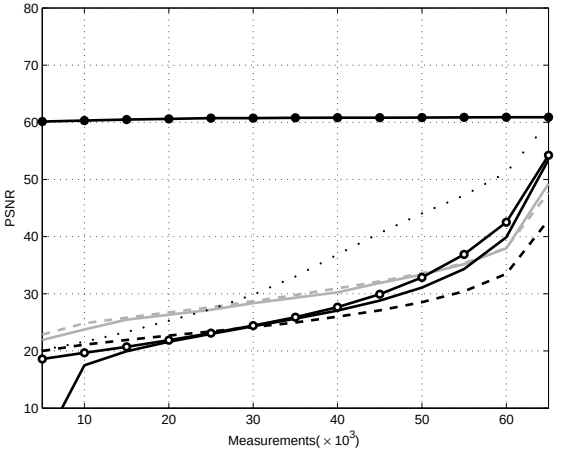


(b) *Lena*

Figure 6.8: Results for CS recovery considering sparsity errors.



(a) *Camera man*



(b) *Text*

Figure 6.9: Results for CS recovery considering sparsity errors.

SVD cannot be used in practice because it requires an *a priori* knowledge of the image's Singular Value Decomposition. Nevertheless, the results for acquisition strategy SVD-$l-1$-N are relevant because they establish that CS performs well when we use a domain in which the signal is very much sparse.

Comparing DCT-l_1-N and DCT-l_2-L we observe that considering the DCT as the sparse domain, CS is worse than the linear compression scheme, even considering large number of measurements. The explanation to this somewhat disappointing result is that the images are not sparse in the DCT domain, as can be seen in Figure 6.7.

Nevertheless, when the image is partitioned into blocks of size 8×8, results improve significantly supporting the observation made in Section 2.2 that block partitioning enhances sparsity (compare strategies DCT-l_1-L and B_DCT-l_1-L).

We will take this opportunity to check Theorem 5, that states that if we need M measurements to recover an S-sparse signal, then if the signal is not sparse, we would recover the S largest coefficients with this number of samples. In other words, we want to confirm that forcing S-sparsity and taking M measurements is approximately the same as taking M measurements on the original image (if M is the number of measurements associated with the sparsity value S by Theorem 2).

Notice that to generate Figure 6.2 we calculated PSNR by comparing the recovered data to the *sparse representation* of the original image. Therefore in Figure 6.10 we compared results from Section 6.2 with the *original* test images.

We can make out that $20k$ measurements are needed to recover the $3.5k$-sparse representation of *Lena*[2] and, therefore, Theorem 5 guarantees that $20k$ measurements recover the $3.5k$ most significant coefficients of the original image. Notice that, compared to the original image, the reconstruction of the $3.5k$-sparse representation results in PSNR = 28.8 and the reconstruction of the original image, when $20k$ measurements are taken, results in PSNR = 26.6, as shown in Figure 6.10. The same analysis can be made on the other graphs and Table 6.1 compares the different PSNR calculated when we compare, to the original image, the results obtained when sparsity is or not

[2]To compute this value we also considered Figure 6.2.

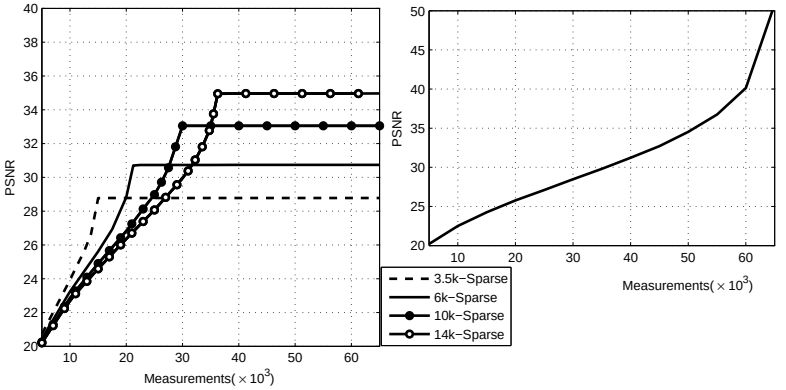


Figure 6.10: On the left, results for strategy DCT-l_1-N on test image *Lena* when different levels of sparsity are forced and PSNR is measured by comparing with the original (only approximately sparse) image. On the right, results for the same image and recovery strategy when no sparsity is forced.

forced before CS measurements are applied. The variations can be associated with the constant C_0 of Theorem 5.

6.4 Measurement Errors

As mentioned in the previous chapter, acquired data is usually corrupted by noise and a very important property of CS that allows it to be used in practice is its robustness to this kind of inaccuracy.

In this section, we will evaluate CS performance when noise is added to the acquired measurements. We consider both Gaussian and quantization noise.

Table 6.1: Different PSNR calculated when we compare, to the original image, the results obtained when sparsity is or not forced before CS measurements are applied.

Test image *Lena*			
Measurements	Sparsity is forced		Sparsity is **not** forced
$M = 20k$	$S = 3.5k$	PSNR = 28.8	PSNR = 26.6
$M = 25k$	$S = 6k$	PSNR = 30.7	PSNR = 27.8
$M = 35k$	$S = 10k$	PSNR = 33.0	PSNR = 30.2
$M = 40k$	$S = 14k$	PSNR = 34.9	PSNR = 31.5

6.4.1 Gaussian Errors

We consider test image *Lena* and strategy DWT-l_1N. Since the acquired measurements are corrupted by an independent white Gaussian noise, we have

$$y = \Phi_\Omega x_0 + n$$

where n is a random variable with normal distribution and variance σ^2.

Figure 6.11 shows the result obtained for $\sigma^2 = 0.1, 1, 3, 5, 10$

It is interesting to observe that the curve format stays the same when errors are added and that, since we are considering a sparse signal, there is a characteristic threshold above which CS is effective. Nevertheless, even considering the number of measurements high enough so that efficiency is reached, PSNRs are smaller for higher values of σ. This confirms that the reconstruction error is proportional to the measurement error, a very important result already stated in Chapter 5.

6.4.2 Quantization

In general, measurements cannot be taken with arbitrary large precision, and a round-off error is added to the acquired data. This quantization process is very important to our study because we are interested in compressing the signal. As seen in Chapter 2, the size

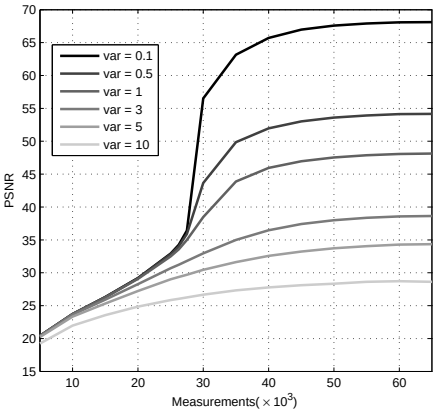


Figure 6.11: Results for applications of scheme DWT-l_1-N to noisy versions of the $10k$-sparse representation of image *Lena*.

of the quantization step is extremely relevant to determine the compression rate which, in turn, is used to evaluate compression efficiency based on the rate-distortion criteria.

Unlike the Gaussian noise, the quantization error is deterministic and signal-dependent. Therefore, a relevant contribution to CS theory consists in verifying how it performs in the presence of quantization errors and, then, plot the Rate $\times$ Distortion curve.

A Sparse Example

We consider Strategy DWT-l_1-N and the $10k$-sparse representation of image *Lena*, when a scalar uniform quantizer of varying step sizes is applied to the measurements.

As we have previously discussed, the parameter ϵ used to solve Equation 6.1 was chosen according to a series of experiments. To illustrate the calculus of the optimal value for ϵ, we present in Table 6.2 the variations of the PSNR according to parameter ϵ for different quantization steps and a fixed number of measurements, $M = 45k$. We highlighted the chosen ϵ for each quantization step.

Table 6.2: PSNR values (dB) for $M = 45k$ and several values of ϵ and q_s (quantization step).

$\mathbf{q_s = 0.01}$		$\mathbf{q_s = 0.1}$		$\mathbf{q_s = 0.5}$	
$\boldsymbol{\epsilon = 0.001}$	**87.85**	$\epsilon = 0.001$	76.78	$\epsilon = 0.001$	63.88
$\epsilon = 0.005$	87.84	$\epsilon = 0.005$	76.78	$\epsilon = 0.005$	63.88
$\epsilon = 0.010$	87.83	$\boldsymbol{\epsilon = 0.01}$	**76.78**	$\epsilon = 0.010$	63.87
$\epsilon = 0.050$	86.60	$\epsilon = 0.050$	76.78	$\boldsymbol{\epsilon = 0.050}$	**63.87**
$\epsilon = 0.100$	86.48	$\epsilon = 0.100$	76.78	$\epsilon = 0.100$	63.87
$\epsilon = 0.500$	84.21	$\epsilon = 0.500$	76.07	$\epsilon = 1.000$	63.83
$\epsilon = 1.000$	83.14	$\epsilon = 1.000$	76.78	$\epsilon = 5.000$	63.42
$\mathbf{q_s = 3}$		$\mathbf{q_s = 10}$		$\mathbf{q_s = 50}$	
$\epsilon = 0.001$	48.25	$\epsilon = 0.500$	37.79	$\epsilon = 10.00$	24.79
$\epsilon = 0.010$	48.25	$\epsilon = 1.000$	37.79	$\epsilon = 50.00$	24.87
$\epsilon = 0.100$	48.25	$\epsilon = 5.000$	37.80	$\epsilon = 200.0$	25.06
$\boldsymbol{\epsilon = 0.500}$	**48.27**	$\boldsymbol{\epsilon = 10.00}$	**37.80**	$\epsilon = 500.0$	25.34
$\epsilon = 1.000$	48.26	$\epsilon = 50.00$	37.78	$\boldsymbol{\epsilon = 800.0}$	**25.56**
$\epsilon = 5.000$	48.24	$\epsilon = 100.0$	37.72	$\epsilon = 1000$	25.52
$\epsilon = 10.00$	48.18	$\epsilon = 250.0$	37.19	$\epsilon = 2000$	24.59
$\epsilon = 20.00$	48.08	$\epsilon = 500.0$	35.90	$\epsilon = 5000$	19.95

Notice that the optimal ϵ increases and diminishes proportionally to the quantization step (that reflects the error size) and that there is an optimal value for each step size, as explained in Section 6.1.1. From Table 6.2, however, we observe that both of these behaviors are not exact. This is also due to computational errors that are noticeable since the PSNR variations are small.

For each fixed quantization step we varied the number of measurements and plotted the Rate $\times$ PSNR curve, as shown in Figure 6.12.

The rate was calculated, as follows

$$Rate = \frac{M}{N} \cdot H_y$$

where H_y is the entropy of the measured data y and $N = 256^2$ is the image size.

To calculate H_y we built an histogram based on the minimum and maximum values assumed by y ($y_{\min}, y_{\max}$) and the quantization step, q_s. Hence, we obtain a vector v_y of size

$$K = \frac{y_{\max} - y_{\min}}{q_s}$$

where $v_y(k)$ indicates the number of coefficients of y that range between $\{y_{\min} + (k-1)q_s, y_{\min} + kq_s\}$. The problem of unused quantization values is resolved by considering each of them to have occurred once,

$$v'_y(k) = v_y(k) + 1, \forall k \in \{1, 2, \ldots, K\}$$

Hence, the probability of occurrence of each symbol is given by

$$p_y(k) = \frac{v'_y(k)}{\sum_{i=1}^{K} v'_y(i)}$$

and H_y is calculated as in Equation 2.2.

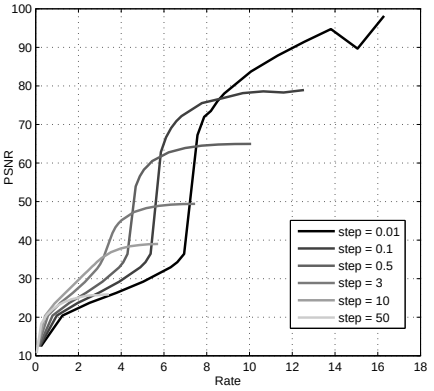


Figure 6.12: Results for applications of strategy DWT-l_1-N to quantized versions of the $10k$-sparse representation of image *Lena*.

We observe a threshold, related to the transition point, where CS theory starts to operate efficiently. As we increase the size of the

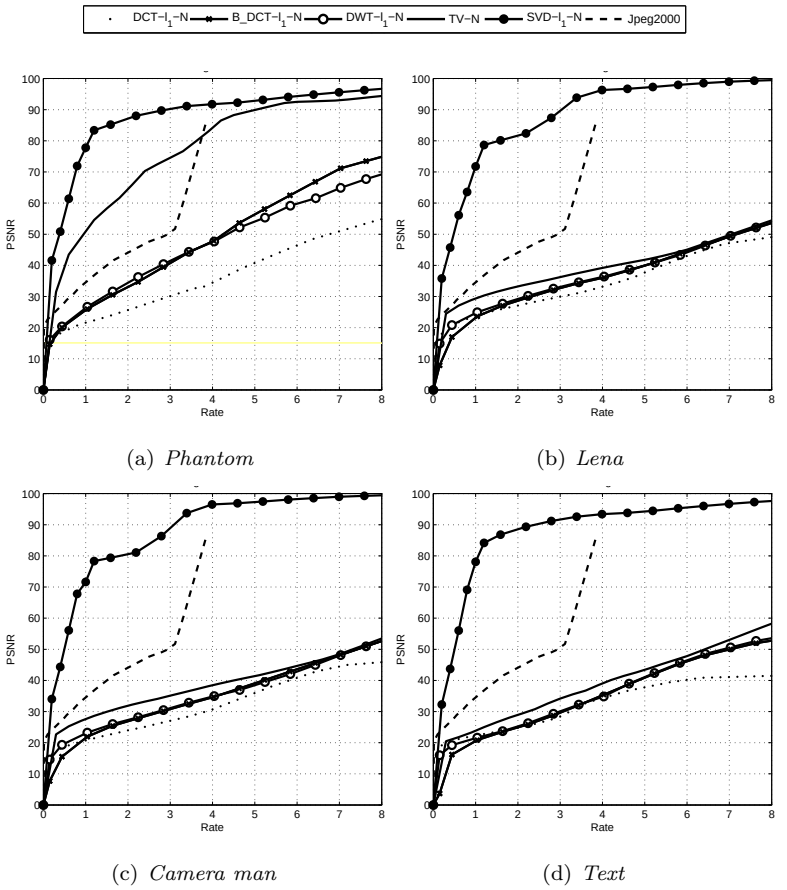
quantization step, the curve approaches the y axis but the PSNR diminishes. This was expected because, by boosting quantization effects, we minimize rate but create higher distortions.

Both Sparsity and Quantization Errors

To formally evaluate the performance of CS, we have to consider the real case, where quantization errors are added and the images are only approximately sparse. We can then calculate Rate $\times$ PSNR for varying quantization steps and plot the rate-distortion curve by selecting the most efficient quantization step ate each point (approximately the convex hull of all Rate $\times$ PSNR curves).

In Figure 6.13 the rate-distortion curve was plotted for all tested images and strategies DCT-l_1-N, B_DCT-l_1-N, DWT-l_1-N, SVD-l_1-N and TV-N. We can observe that CS recovery schemes that perform the l_1-norm minimization in the Wavelet domain are far less efficient than the JPEG2000 standard. However, by analyzing the results for strategy SVD-l_1-N and for the test image *Phantom* on strategy TV-N, we can see that there is room for improvement; in both cases one gets better results than with JPEG2000. The *Phantom* image in the frequency domain and the SVD transform are both very sparse. This indicates that, by choosing representations that strengthen sparsity, one can reduce not only the number of measurements needed to reconstruct the signal but also the approximation error.

It is important to mention that, though strategy SVD-l_1-N presents an upper bound to CS performance, it is not really practical because it requires an *a priori* knowledge of the image's SVD. Figure 6.14 highlights this argument by contrasting recovery of the image *Camera man* using as a basis *Camera man*'s SVD and *Lena*'s SVD. In Figure 6.15 the Rate $\times$ PSNR curve was plotted for all test images and strategies DCT-l_1-N, DWT-l_1-N, TV-N and SVD-l_1-N using varying quantization steps. It can be observed that, for a particular compression rate, each image and recovery strategy has an optimal quantization step that produces the highest PSNR. If the image is not sparse in the considered domain, the curves show that it is more efficient to take a large number of measurements and compensate for the potential rate increase by enlarging the quantization



(a) *Phantom* (b) *Lena*

(c) *Camera man* (d) *Text*

Figure 6.13: Rate-Distortion curves for compression standard JPEG2000 and CS acquisition strategies DCT-l_1-N, B_DCT-l_1-N, DWT-l_1-N, SVD-l_1-N and TV-N.

step.

We have studied in the previous chapter that the recovery error is bounded by the sum of the measurement error and the error due

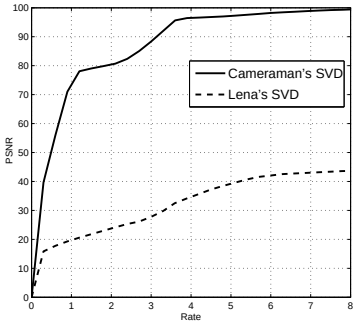


Figure 6.14: Rate × PSNR curve for test image *Camera man* and two versions of strategy SVD-l_1-N. One of them uses the natural (Camera man's) SVD basis and the other one uses *Lena*'s SVD to reconstruct the signal.

to the fact that the signal is not strictly sparse. In the context of quantization errors we can rewrite Equation 5.2 as

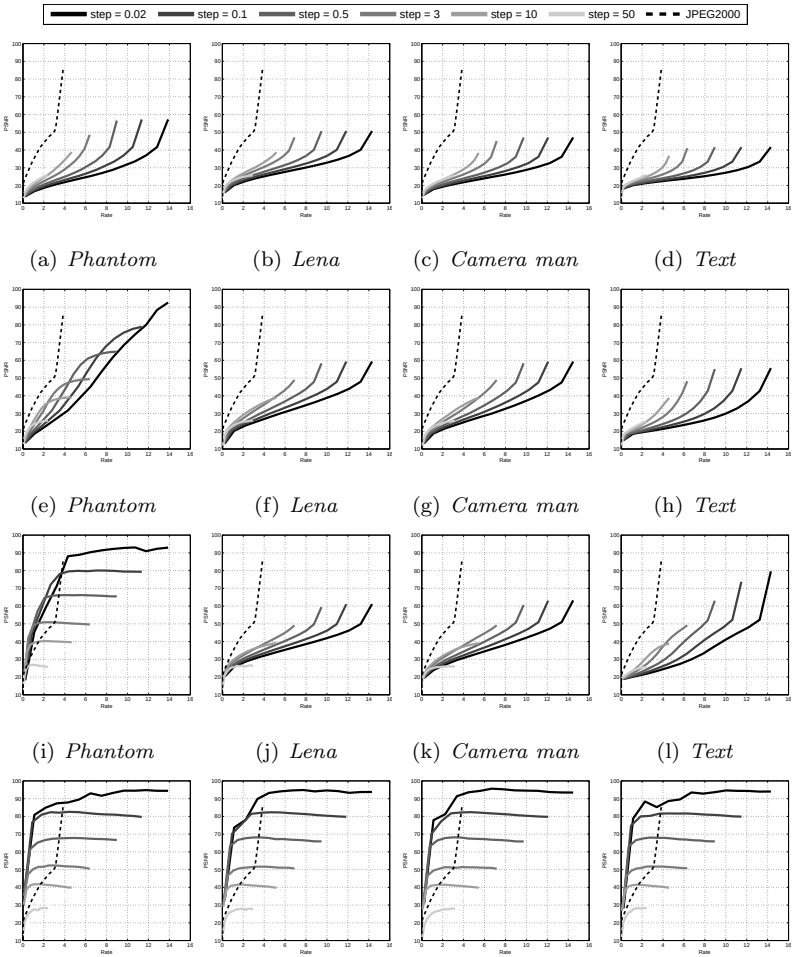
$$\|y - \Phi x\|_{l_2} \leq C \cdot \left(\epsilon_q + \underbrace{S^{-1/2}\|x_S - x\|_{l_1}}_{\epsilon_s} \right), \tag{6.4}$$

where C is relatively small and x_S is an approximation of x where the S largest coefficients in the Ψ domain are observed.

This implies that the reconstruction error in CS is of the order of the *maximum* of the quantization (ϵ_q) and sparsity errors (ϵ_s) [31].

This result is closely related to the fact that, for a fixed PSNR, the ideal quantization step is approximately the same in all evaluated scenarios (see Figure 6.15). The PSNR determines the acceptable distortion and, therefore, the values of ϵ_q and ϵ_s. Since ϵ_q only depends on the quantization step, the fixed PSNR determines the optimal quantization step.

The value ϵ_s, on the other hand, depends on the sparsity distribution and, hence, on the number of measurements. Therefore we



(a) *Phantom* (b) *Lena* (c) *Camera man* (d) *Text*

(e) *Phantom* (f) *Lena* (g) *Camera man* (h) *Text*

(i) *Phantom* (j) *Lena* (k) *Camera man* (l) *Text*

(m) *Phantom* (n) *Lena* (o) *Camera man* (p) *Text*

Figure 6.15: Rate $\times$ PSNR for varying quantization steps: (a-d) shows results for strategy DCT-l_1-N, (e-h) shows results for strategy DWT-l_1-N, (i-l) shows results for strategy TV-N and (m-p) shows results for strategy SVD-l_1-N.

can relate to Equation 6.4 by observing Figure 6.16, which shows the results in terms of Number of Measurements × PSNR. For each strategy, the number of measurements determines ϵ_s; in addition, all quantization steps that make ϵ_q of the order of ϵ_s (or smaller) result in the same PSNR. Therefore, all curves overlap until the number of measurements is large enough so that ϵ_s exceeds ϵ_q (see Figure 6.16(b)). In Figure 6.16(a)), it is noteworthy that for quantization steps smaller than 3, the curves overlap completely. This is so because as the errors due to sparsity are very large, reducing the quantization step is ineffective in increasing PSNR. In contrast, in Figure 6.16(d), where the image is strongly sparse in the considered domain (SVD), ϵ_s tends to be much smaller, and therefore such behavior is not observed.

6.5 Software

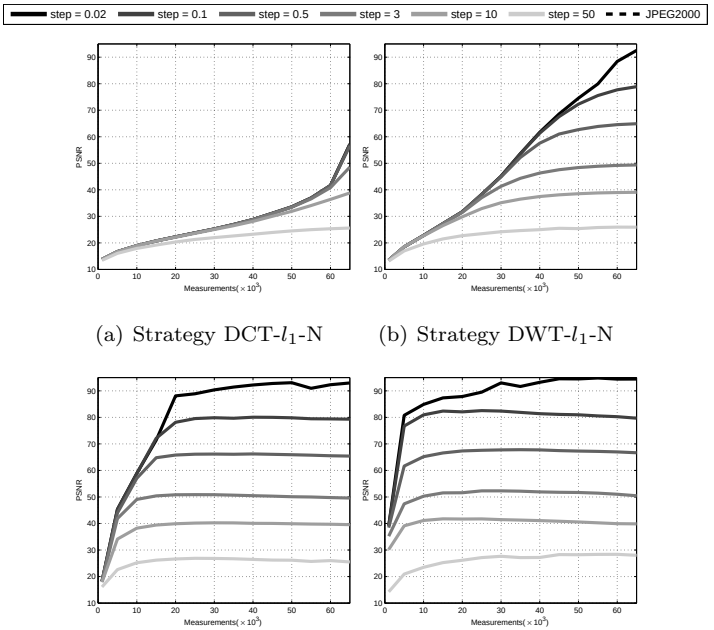
The MATLAB scripts that reproduce all the above results are available at www.impa.br/∽aschulz/CS. The zip file already includes the test images, the optimization functions from L1-Magic, and the algorithm for generating Noiselets made available by Justin Romberg. The recovery strategies that make use of Wavelets require the WAVELAB toolbox, that can be downloaded from [30].

Before starting, it is necessary to compile the mex code that generates Noiselets. To do so, simply open file CS-codes/Measurements in MATLAB and run:

```
>> mex realnoiselet.c
```

Since the optimization algorithm is computationally expensive, it may take a while to run it. For simpler tests, we recommend using smaller images, such as lena64.pgm (available at CS-codes/Data).

Below, we specify the 8 main functions (in folder CS-codes/CS) that implement the acquisition strategies described in Section 6.1. Examples of how to use these functions to generate graphs, such as the ones shown in this chapter, can be found in CS-codes/Demos.



(a) Strategy DCT-l_1-N (b) Strategy DWT-l_1-N

(c) Strategy TV-N (d) Strategy SVD-l_1-N

Figure 6.16: Number of Measurements × PSNR for different quantization steps and test image *Phantom*.

Strategy DCT-l_1-N

Syntax[3] :

[3]All functions have required arguments and a number of pairs of optional arguments (*options*). The first element of each pair is a string (the keyword that defines the parameter), and the second is the value. Some examples on how to use function DCT_l1_N are:

DCT_l1_N('qc', 'lena.pgm', 10000);

DCT_l1_N('qc', 'lena.pgm', 10000, 's', 35000);

DCT_l1_N('qc', 'lena.pgm', 10000, 'e', 50, 'q', 0.2);

`DCT_l1_N(constraint, ImgName, M,` *options*`)`
The necessary inputs are:
`constraint` - specifies if we are using equation 6.3 with equality constraints in which case the value is `'eq'` or equation 6.1 with quadratic constraints in which case the value is `'qc'`;
`ImgName` - the name of the file where the original input image is stored;
`M` - the number of measurements that will be taken.

The optional inputs are:
`<'e' , epsilon>` - specifies the value of the variable ϵ (the default value is 10^-3y);
`<'q', q_step>` - in the case of add quantization error, it specifies the value of the quantization step;
`<'s', Sparsity>` - in the case of imposing sparsity, it specifies the number of nonzero coefficients.

Strategy B_DCT-l_1-N

Syntax:
`BDCT_l1_N(constraint, ImgName, M, blocksize,` *options*`)`
The necessary inputs are:
`constraint` - specifies if we are using equation 6.3 with equality constraints in which case the value is `'eq'` or equation 6.1 with quadratic constraints in which case the value is `'qc'`;
`ImgName` - the name of the file where the original input image is stored;
`M` - the number of measurements that will be taken.
`blocksize` - specifies the image partitioning, the size of the block is `blocksize` $\times$ `blocksize`

The optional inputs are:
`<'e' , epsilon>` - specifies the value of the variable ϵ (the default value is 10^-3y);
`<'q', q_step>` - in the case of add quantization error, it specifies the value of the quantization step;

<‘s’, Sparsity> - in the case of imposing sparsity, it specifies the number of nonzero coefficients.

Strategy DWT-l_1-N

Syntax:
DWT_l1_N(constraint, ImgName, M, *options*)
The necessary inputs are:
constraint - specifies if we are using equation 6.3 with equality constraints in which case the value is ‘eq’ or equation 6.1 with quadratic constraints in which case the value is ‘qc’;
ImgName - the name of the file where the original input image is stored;
M - the number of measurements that will be taken.

The optional inputs are:
<‘e’ , epsilon> - specifies the value of the variable ϵ (the default value is $10^{-}3y$);
<‘q’, q_step> - in the case of add quantization error, it specifies the value of the quantization step;
<‘s’, Sparsity> - in the case of imposing sparsity, it specifies the number of nonzero coefficients.

Strategy SVD-l_1-N

Syntax:
SVD_l1_N(constraint, ImgName, M, *options*)
The necessary inputs are:
constraint - specifies if we are using equation 6.3 with equality constraints in which case the value is ‘eq’ or equation 6.1 with quadratic constraints in which case the value is ‘qc’;
ImgName - the name of the file where the original input image is stored;
M - the number of measurements that will be taken.

The optional inputs are:
<‘e’ , epsilon> - specifies the value of the variable ϵ (the default

value is 10^-3y);
`<'q', q_step>` - in the case of add quantization error, it specifies the value of the quantization step;
`<'s', Sparsity>` - in the case of imposing sparsity, it specifies the number of nonzero coefficients.

Strategy TV-N

Syntax:
`TV_N(constraint, ImgName, M,` *options*`)`
The necessary inputs are:
`constraint` - specifies if we are using equation 6.3 with equality constraints in which case the value is `'eq'` or equation 6.1 with quadratic constraints in which case the value is `'qc'`;
`ImgName` - the name of the file where the original input image is stored;
`M` - the number of measurements that will be taken.

The optional inputs are:
`<'e' , epsilon>` - specifies the value of the variable ϵ (the default value is 10^-3y);
`<'q', q_step>` - in the case of add quantization error, it specifies the value of the quantization step.

Strategy DCT-l_2-N

Syntax:
`DCT_l2_N(ImgName, M,` *options*`)`
The necessary inputs are:
`ImgName` - the name of the file where the original input image is stored;
`M` - the number of measurements that will be taken.

The optional inputs are:
`<'q', q_step>` - in the case of add quantization error, it specifies the value of the quantization step;

`<'s', Sparsity>` - in the case of imposing sparsity, it specifies the number of nonzero coefficients.

Strategy B_DCT-l_2-N

Syntax:
`BDCT_l2_N(constraint, ImgName, M, blocksize,` *options*`)`
The necessary inputs are:
`ImgName` - the name of the file where the original input image is stored;
`M` - the number of measurements that will be taken.
`blocksize` - specifies the image partitioning, the size of the block is `blocksize` $\times$ `blocksize`

The optional inputs are:
`<'q', q_step>` - in the case of add quantization error, it specifies the value of the quantization step;
`<'s', Sparsity>` - in the case of imposing sparsity, it specifies the number of nonzero coefficients.

Strategy DCT-l_1-LN

Syntax:
`DCT_l1_LN(constraint, ImgName, M, M0,` *options*`)`
The necessary inputs are:
`constraint` - specifies if we are using equation 6.3 with equality constraints in which case the value is `'eq'` or equation 6.1 with quadratic constraints in which case the value is `'qc'`;
`ImgName` - the name of the file where the original input image is stored;
`M` - the total number of measurements that will be taken.
`M0` - the number of linear DCT measurements.

The optional inputs are:
`<'e' , epsilon>` - specifies the value of the variable ϵ (the default value is 10^-3y);
`<'q', q_step>` - in the case of add quantization error, it specifies

the value of the quantization step;
`<'s', Sparsity>` - in the case of imposing sparsity, it specifies the number of nonzero coefficients.

Chapter 7

Applications in Graphics and Vision

The change of paradigm suggested by compressive sensing has drawn great attention from both the scientific and engineering communities. The large body of research already created, as well as, the consolidation of the theoretical foundations in the field established solid grounds for the development of emerging applications in various areas of science and technology.

In this chapter we will give an overview of new applications in *Computer Graphics*, *Vision* and related fields. Due to the vast scope of application areas it would not be possible to review all recent activity in all areas. Therefore, we will leave out applications in other important areas, such as: *Coding and Information Theory*, *Statistical Signal Processing*, *Machine Learning*, *Biosensing*, *Hyperspectral Imaging*, *Radar*, *Astronomy*, *Communications*, and *Circuit Analysis*.

Our choice to focus in Graphics and Vision is clearly motivated by our specific research agenda. However, the interested reader can find a broad coverage of the developments in all the above mentioned application areas in the Rice CS portal [32].

7.1 Overview of Applications

The very nature of Computer Graphics and Vision makes these two areas particularly suited to exploit the compressive sensing results in the development of new techniques for various applications.

Graphics and Vision deal primarily with images and video, which by themselves represent large amounts of raw data. Computer Graphics is concerned with *image synthesis*, while Computer Vision with *image analysis*.

A recent trend promoted the integration of Graphics and Vision based on images. In this way, the subareas of "image-Based Modeling", "Image-Based Rendering", "Model-Based Image Analysis" appeared to take advantage of the synergy of methods that perform image analysis and synthesis in a unified fashion.

The scenario described above creates many opportunities for the use of compressive sensing in new data acquisition hardware as well as in new methods for model inference from reduced datasets.

In the sequence, we will discuss recent results in Graphics Hardware, Imaging Methods, Video, Image Analysis, Medical Imaging, Geophysics, and Image Synthesis.

7.1.1 Hardware

One of the keys for the realization of the full potential of compressive sensing in Graphics and Vision is the design of suitable data acquisition devices. In that context, there are two distinct research avenues: On one hand, it is necessary to develop a method that can capture data related to physical attributes of three-dimensional scenes, such as visual information. Note that, the device must be able to measure the inner-products between the scenes and a set of CS test functions; On the other hand, it would be desirable to encourage architectures

that take advantage of parallelism and distributed processing, such as in a wireless network of sensors. Some steps have already been taken in these two directions.

A landmark in image acquisition for compressive sensing was the development of the *one-pixel camera*, by the Digital Signal Processing group at Rice University [33].

The design of this new digital image/video camera inverts the traditional paradigm for imaging capture. It directly acquires random projections of a scene without first collecting the pixels. The camera architecture employs a digital micromirror array to optically calculate linear projections of the scene onto pseudorandom binary patterns. This information is collected by a single photo-detector (hence a "single pixel"). By the virtue of compressive sensing, it measures the scene fewer times than the number of pixels in the the reconstructed image.

One advantage of this design is that since the camera employs only a single detector, it can also be adapted to imaging at specific wavelengths, even outside the visible range – which would be difficult to do with conventional images.

Figure 7.1 shows a laboratory prototype of the single-pixel camera. In this figure it can be seen the scene (a patch with the letter R), the lens / micromirror assembly and the photo-sensor.

A schematic diagram of the components of the device is shown in Figure 7.2.

Figure 7.3 shows the results of the reconstruction of a simple scene, consisting of a rectangular patch with the letter R, using this hardware. A conventional picture of the scene is shown in Figure 7.3(a) and an image reconstructed from 1300 measurements is shown in Figure 7.3(b). Both images have a resolution of 256 x 256 pixels.

As indicated in Figure 7.2, devices such as the single pixel camera can benefit from a setting that incorporates wireless data transmission and distributed sensing. The challenge is to integrate the measures from many such simple sensors while exploiting the framework of compressive sensing. Some works that address these issues are Bajwa et al.[34] and Baron et al. [35].

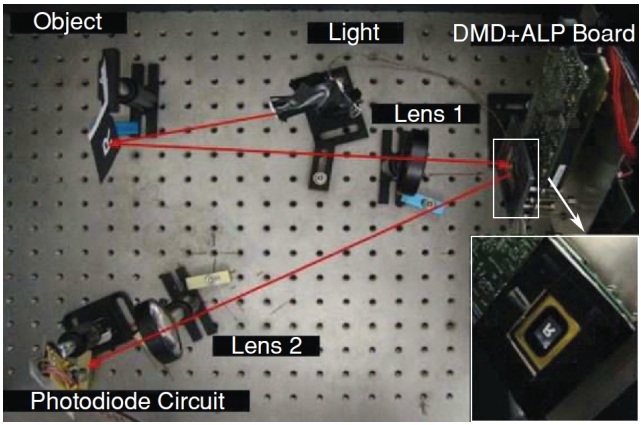


Figure 7.1: Laboratory prototype of the single-pixel camera. (Extracted from [33].)

7.1.2 Imaging

Image Processing is arguably one of the areas that adopted compressive sensing most vigorously.

Note that in this context, the goal is to process the information in various ways using the compressive sensing framework. Such methods and techniques constitute a natural follow-up for compressive sensing devices, such as the single-pixel camera.

Some practical works in this area include: compressive image fusion by Wan et al. [36]; compressive sensing for background subtraction by Cevher et al. [37]; multiscale sparse image representation with learned dictionaries by Mairal et al. [38]; manifold lifting for multiview compressive imaging by Wakin [39]; and image super-resolution using Sparse Representations by Yang et al. [40].

On a different track, a relevant theoretical work in imaging is the one by Hennenfent and Herrmann [41] that discusses irregular sampling and its relations to aliasing and noise.

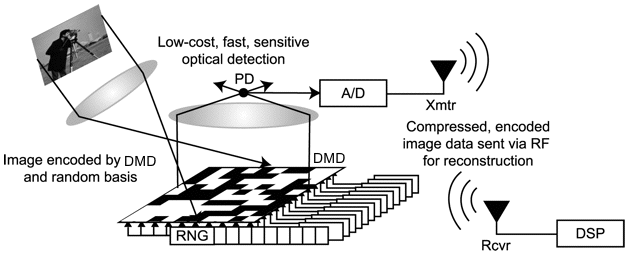


Figure 7.2: Diagram of the single-pixel camera components. (Extracted from [33].)

(a) (b)

Figure 7.3: Example of image captured by the single pixel camera (a) 256x256 conventional image of a black-and-white letter R, (b) Reconstructed image from 1300 random measurements. (Extracted from [33].)

7.1.3 Video

Video Processing extends Imaging in the temporal dimension. In this area the requirements for data representation, processing and transmission are even more demanding.

Some works in this area include: compressive coded aperture video reconstruction by Roummel et al. [42]; distributed compres-

sive video sensing by Kang and Lu. [43]; compressive imaging for video representation and coding by Wakin et al. [44]; compressive video sampling by Stankovic et al. [45]; and multiscale methods for compressive sensing of video by Park and Wakin. [46].

7.1.4 Medical Imaging and Geophysics

Medical Imaging and Geophysics deal with volumetric data, which can be viewed as "3D images". In a sense, this area has similar characteristics of Video Processing. Nonetheless, the type of data has different properties due to the particular nature of the time dimension in video.

Some representative works in this area are: compressed sensing based interior tomography by Yu and Wang [47]; and optimized compressed sensing for seismic data reconstruction by Tang et al. [48].

7.1.5 Vision

The problems in Computer Vision are related to inference of models from images and video. There are two general approaches for the solution of such problems. One approach assumes contextual knowledge and the main task is fitting the data to known models. The other approach attempts to learn the models from the data by extracting the underlying structure. In both cases, the compressive sensing framework helps due to the fact that although the data, in general, has very high dimensionality the model structure has much less degrees of freedom due to data coherence.

Some works in computer vision are: compressive sensing of parameterized shapes in images by Gurbuz et al. [49]; smashed filters for compressive classification and target recognition by Davenport et al. [50]; feature selection in face recognition by Yang et al. [51]; simultaneous sensing matrix and sparsifying dictionary optimization by Duarte-Carvajalino and Sapiro [52]; and sparse representations for image classification by Rodriguez and Sapiro [53].

7.1.6 Computer Graphics

The problems in Computer Graphics range from generating representations of objects in three dimensional scenes to image synthesis of the scenes.

In the image-based approach to modeling and rendering, the solution of these problems is devised using image data.

In this context, a fundamentally difficult problem due to the sheer amount of data involved is the acquisition of samples of the *plenoptic function*, which describes the visual information in a scene.

Some recent works in applying compressive sensing to capture such light fields are: compressive light transport sensing by Peers et al. [54]; compressive dual photography by Sen and Darabi [55]; and compressive structured light for recovering inhomogeneous participating media by Gu et al. [56];

7.2 Case Study

In this section, we will discuss the application of compressive sensing to dual photography proposed by Sen et al. [57]. Our interest in this work results from the fact that it is one of the first applications of CS in computer graphics and that it deals with the fundamental problem of capturing the light transport characteristics of a scene.

7.2.1 Dual Photography

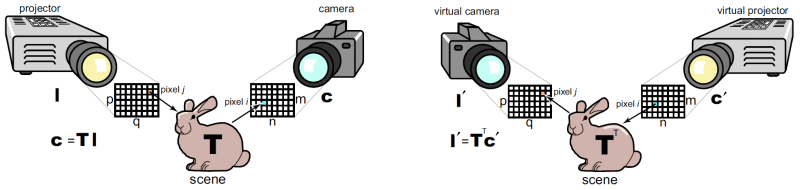
Dual photography is a technique that enables the interchange of cameras and light sources of a scene, therefore allowing us to take pictures from the point of view of the projector.

Fixed the position of the camera and the projector, we can calculate the light field from the projector through the scene and onto the camera. Since the light transport is linear, we can represent it by a matrix $\mathbf{T}$ and the reflectance function can be written as

$$c = \mathbf{T}l \tag{7.1}$$

where c (size $N_c \times 1$) and l (size $N_l \times 1$) are column vectors that represent the projected pattern and the image taken by the camera,

respectively. Figure 7.4(a) shows what will be referred as the primal configuration, where the light is emitted by a real projector and acquired by a real camera.



(a) Primal Configuration (b) Dual Configuration.

Figure 7.4: Diagram of dual photography.

In [57] Helmholtz reciprocity is used to establish that, since the light transport is the same along a light path regardless of the direction of the flow of light, measuring the light that starts from the projector pixel j and arrives at the camera pixel i is equivalent to measuring the transport of energy that starts from the camera pixel i and arrives at the projector pixel j. This means that we can replace the projector for a virtual camera and the camera for a virtual projector (see Figure 7.4(b)) and represent the *dual* of Equation 7.1 as:

$$l^{'} = \mathbf{T}^{T} c^{'} \tag{7.2}$$

Figure 7.5 shows results for this technique. Notice that, once the transport matrix $\mathbf{T}$ must be acquired, we can use this information to relight the scene. Figure 7.5(c) shows a projective pattern that has been virtually projected onto the image when an illumination vector $\bar{l}$ was multiplied by $\mathbf{T}$.

It is also important to mention that this technique can still be used in scenes where diffuse inter-reflections or surface scatterings dominate the appearance. In Figure 7.6 dual photography reveals the front of the card that could not be seen from the point of view of the camera. Notice that in this experiment the light undergoes two diffuse bounces before reaching the camera.

An important aspect to consider in this procedure is the acquisition of the light transport matrix $\mathbf{T}$. A simple way to do this is what

(a) Primal image. (b) Dual image. (c) Image relighting.

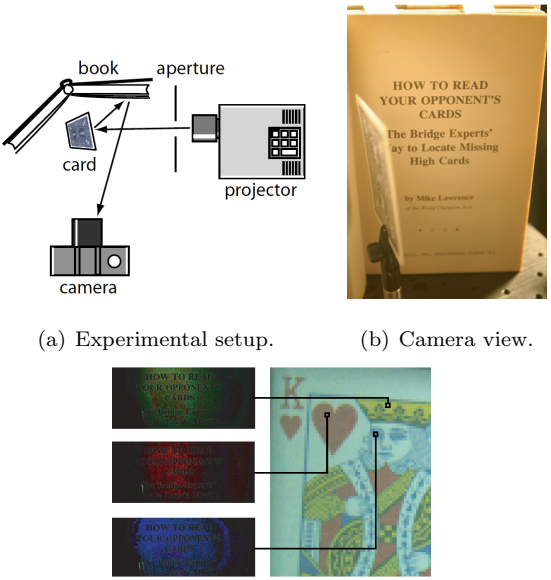
Figure 7.5: Results of dual photography. (Extracted from [57].)

is referred to as "brute force" scan and consists in scanning the projector pixel by pixel (i.e, displaying N_l different patterns each with a single pixel lit up at a time) and acquiring the resulting image with the camera.

This algorithm requires however a very large number of measurements. Consider for example an image of size 512×512 and a projector of resolution 512×512. Assuming that the capture rate is approximately 25 patterns/minute and that data is stored as three 32-bit floats for each matrix element, it would take 7.3 days to acquire the light field and it would require 3.3TB to store the data.

Hence, the challenge in this area is to capture $\mathbf{T}$ as efficiently as possible using algorithms that explore the compressibility of light fields. The problems concerning most of the existing algorithms that explore redundancy to speed up the acquisition of light fields are that most strategies are quite complex to implement and usually it is difficult to estimate bounds on the adaptiveness of the solutions.

We can understand that $\mathbf{T}$ is compressible by observing that, if a scene does not have a lot of global illumination effects, the contribution of a single projector pixel will be concentrated in a small region of the image acquired by the camera. This means that the transport matrix is *sparse*.



(a) Experimental setup. (b) Camera view.

(c) Dual reconstruction.

Figure 7.6: Results of dual photography with indirect light transport. (Extracted from [57].)

In this context, CS comes out a very useful tool that, by exploiting sparsity in the transport matrix, accelerates its acquisition very efficiently.

7.2.2 Compressive Sensing

In CS schemes the idea is to take $M \ll N_l$ measurements, i.e, to generate $M \ll N_l$ approximately random illumination patterns and reconstruct $\mathbf{T}$ from the acquired images.

Let $\mathbf{C}$ be the matrix created by stacking the column vectors c_i for $i = 1, \ldots, M$ and $\mathbf{L}$ the matrix created by stacking the column

vectors l_i (see Equation 7.1). Then

$$\mathbf{C} = \mathbf{TL} \tag{7.3}$$

and therefore

$$c_i^T = \mathbf{L}^T t_i^T \quad \text{for } i = 1, \ldots, N_c \tag{7.4}$$

where c_i^T [1] (size $M \times 1$) and t_i^T (size $N_l \times 1$) are the i-th rows of $\mathbf{C}$ and $\mathbf{T}$ respectively (see Figure 7.7).

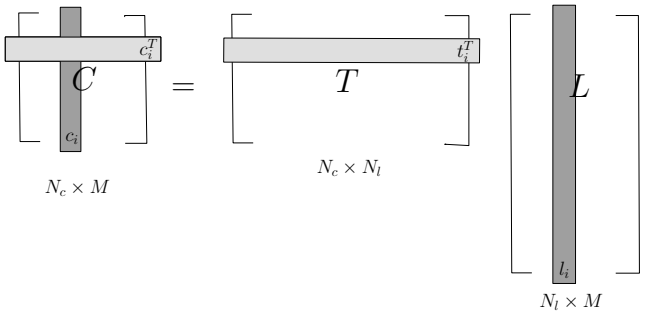


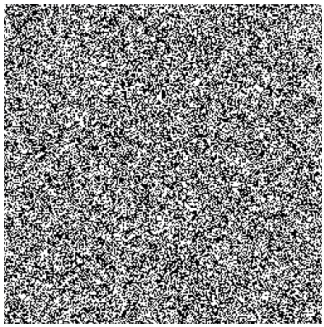
Figure 7.7: Sensing equation.

We have already argued that $\mathbf{T}$ is sparse (t_i is parse $\forall i$) and therefore CS theory states we can recover t_i based on Equation 7.4 using a convex optimization algorithm as long as $\mathbf{L}^T$ preserves the RIP.

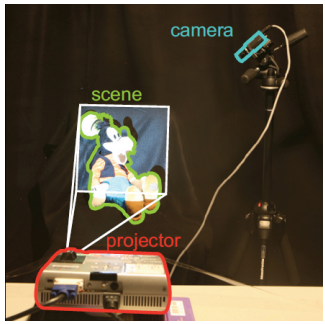
In [55] $\mathbf{L}$ is generated using Bernoulli patterns (the matrix is composed by 1's and -1's randomly selected with equal probability). The experimental setup is shown in Figure 7.8.

It is noteworthy that, though we can usually assume that $\mathbf{T}$ is sparse, in cases when pixels in the camera get contributions from many pixel in the projector (e.g. due to significant global effects, such as defocussing of the camera), it is necessary to represent t_i in a basis in which it is sparse by applying a transform matrix Ψ.

[1] Notice that c_i^T is a vector that stores a sequence of M measurements of a single pixel.



(a) 256×256 Bernoulli pattern



(b) Experimental setup.

Figure 7.8: Compressive dual photography. (Extracted from [55].)

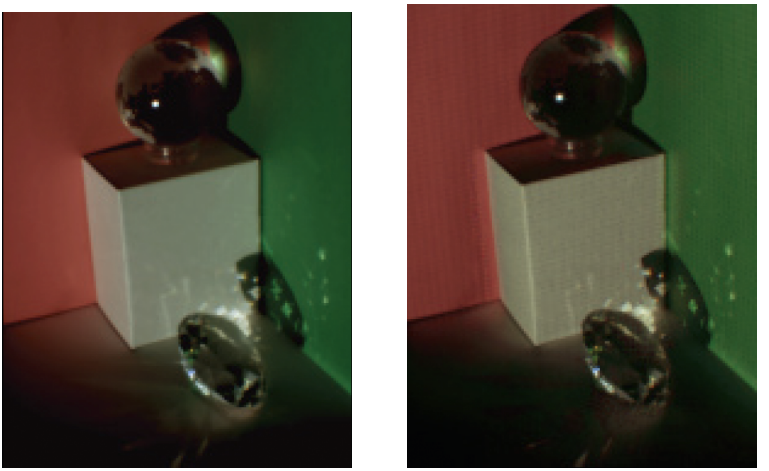
We have already enforced that one of the most significant advantages of CS is that it is nonadaptive. In this scenario, this implies that the procedure does not require real time processing during acquisition as in [57], where an estimation of the energy distribution has to be made prior to sensing. Since the patterns are all pre-computed, they can be displayed at an extremely fast framerate, without the need of any computational power for run-time processing.

Moreover, the illumination patterns are chosen regardless of the scene. This is true even in the cases when we must consider a different basis in which the signal is sparse, once the knowledge of the Ψ basis is only used for reconstruction and not for sensing[2] These simple binary patterns are easy to implement (compared e.g. to a basis of Daubechies Wavelets) and make good use of the limited dynamic range and quantization of the projector, thereby improving the SNR of the results.

Figure 7.9 shows a result obtained by [55]. We observe that the technique is able to capture global illumination effects such as diffuse-diffuse inter-reflections. However, in this more extreme case, they tend to fall off quicker than the ground truth image. The authors associated this difference with the limitations of the HDR capture configuration. Notice that, since the contrast between brightest and

[2]It is of course essential that $\mathbf{L}^T$ still meets the RIP when combined with Ψ.

dimmest entries in this matrix can be large, these limitations can lead to significant inaccuracies.



(a) Ground truth. (b) Rendered image.

Figure 7.9: Results extracted from [55].

Bibliography

[1] Kirk L. Kroeker. Rethinking signal processing. *Commun. ACM*, 52(5):13–15, 2009.

[2] Eduardo da Silva and Lisandro Lovisolo. *TV Digital - Notas de aula*. DEL/Poli/UFRJ, 2007.

[3] Khalid Sayood. *Introduction to data compression*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[4] Anil K. Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[5] Jonas Gomes and Luiz Velho. *From Fourier Analysis to Wavelets*. SIGGRAPH'99 Course Notes 5, SIGGRAPH-ACM publication, Los Angeles, California, USA, August 1999.

[6] Eduardo A. B. da Silva. *Wavelet Transforms for Image Coding*. PhD thesis, Essex University - Colshester, UK, June 1995.

[7] Eduardo da Silva and Gelson Mendonca. *Digital Image Processing*, chapter VII.4, pages 891–910. The Electrical Engineering Handbook. Wai-Kai Chen, Elsevier - Academic Press, 2005.

[8] Simon Haykin. *Sistemas de comunicações analógicas e digitais*. Bookman, São Paulo, SP, Brasil, 2004.

[9] Majid Rabbani and Rajan Joshi. An overview of the jpeg 2000 still image compression standard. *Signal Processing: Image Communication*, 17:3–48, 2002.

[10] Jonas Gomes and Luiz Velho. *Image Processing for Computer Graphics*. Springer Verlag, 1997.

[11] Stèphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, CA, USA, second edition edition, 1999.

[12] Paulo A. R. Diniz, Eduardo A. B. da Silva, and Sergio L. Netto. *Processamento Digital de Sinais - Projeto e Análise de Sistemas*. Bookman, Porto Alegre, 2004.

[13] Rogério Caetano. *Video Coding using Generalized Bit-planes*. PhD thesis, COPPE/UFRJ, March 2004.

[14] Stèphane Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. Technical report, New York, NY, USA, 1993.

[15] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.

[16] Richard Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, 24(4), July 2007.

[17] Emmanuel Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. on Information Theory*, 52(2), February 2006.

[18] The Institute for Mathematics and its Applications (IMA). *Lectures on compressive sampling and frontiers in signal processing*, University of Minnesota, June 2007.

[19] David Donoho and Philip Stark. Uncertainty principles and signal recovery. *SIAM Journal on Applied Mathematics*, 49(3):906–931, 1989.

[20] Emmanuel Candès and Justin Romberg. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23(3):969–985, 2007.

[21] Emmanuel Candès and Michael Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2), March 2008.

[22] Emmanuel Candès and Terence Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. on Information Theory*, 52(12), December 2006.

[23] Emmanuel Candès and Terence Tao. Decoding by linear programming. *IEEE Trans. on Information Theory*, 51(12), December 2005.

[24] Emmanuel Candès. The restricted isometry property and its implications for compressed sensing. *Compte Rendus de l'Academie des Sciences, Series*, 346:589–590, 2008.

[25] Emmanuel Candès, Justin Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8), August 2006.

[26] Emmanuel Candès. Compressive sampling. *Int. Congress of Mathematics*, 3:1433–1452, 2006.

[27] R.Coifman, F. Geshwind, and Yves Meyer. Noiselets. *Appl. Comp. Harmon.Anal.*, 10(1):27–44, 2001.

[28] Justin Romberg. Imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2), March 2008.

[29] Emmanuel Candès and Justin Romberg. L1–magic. www.l1-magic.org.

[30] David L. Donoho, Arian Maleki, and Morteza Shahram. Wavelab. http://www-stat.stanford.edu/ wavelab.

[31] Emmanuel Candès and Justin Romberg. Encoding the l_p ball from limited measurements. *DCC*, 2006.

[32] Rice University. Compressive sensing resources. http://www-dsp.rice.edu/cs.

[33] Marco F. Duarte, Mark A. Davenport, Dharmpal Takhar, Jason N. Laska, Ting Sun, Kevin F. Kelly, and Richard G. Baraniuk. Single-pixel imaging via compressive sampling [building simpler, smaller, and less-expensive digital cameras]. *IEEE Signal Processing Magazine*, 25(2):83–91, March 2008.

[34] Waheed Bajwa, Jarvis Haupt, Akbar Sayeed, and Robert Nowak. Compressive wireless sensing. In *IPSN '06: Proceedings of the 5th international conference on Information processing in sensor networks*, pages 134–142, New York, NY, USA, 2006. ACM.

[35] Dror Baron, Marco F. Duarte, Michael B. Wakin, Shriram Sarvotham, and Richard G. Baraniuk. Distributed compressive sensing. In *Preprint RICE*, 2009.

[36] Tao Wan, Nishan Canagarajah, and Alin Achim. Compressive image fusion. In *EEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008.

[37] Volkan Cevher, Aswin Sankaranarayanan, Marco F. Duarte, Dikpal Reddy, Richard G. Baraniuk, and Rama Chellappa. Compressive sensing for background subtraction. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 155–168, Berlin, Heidelberg, 2008. Springer-Verlag.

[38] Julien Mairal, Guillermo Sapiro, and Michael Elad. Multiscale sparse image representation with learned dictionaries. In *Preprint*, 2007.

[39] Michael B. Wakin. A manifold lifting algorithm for multi-view compressive imaging. In *Picture Coding Symposium (PCS)*, 2009.

[40] Jianchao Yang, John Wright, Yi Ma, and Thomas Huang. Image super-resolution as sparse representation of raw image patches. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.*

[41] G. Hennenfent and F. J. Herrmann. Irregular sampling: from aliasing to noise. In *EAGE 69th Conference and Exhibition*, 2007.

[42] Roummel Marcia and Rebecca Willett. Compressive coded aperture video reconstruction. In *European Signal Processing Conf. (EUSIPCO)*, 2008.

[43] Li-Wei Kang and Chun-Shien Lu. Distributed compressive video sensing. In *EEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009.

[44] Michael Wakin, Jason Laska, Marco Duarte, Dror Baron, Shriram Sarvotham, Dharmpal Takhar, Kevin Kelly, and Richard Baraniuk. Compressive imaging for video representation and coding. In *Proc. Picture Coding Symposium*, 2006.

[45] V. Stankovic, L. Stankovic, , and S. Cheng. Compressive video sampling. In *European Signal Processing Conf. (EUSIPCO)*, 2008.

[46] Jae Young Park and Michael B. Wakin. Multiscale framework for compressive sensing of video. In *Picture Coding Symposium (PCS)*, 2009.

[47] Hengyong Yu and Ge Wang. Compressed sensing based interior tomography. In *Physics in Medicine and Biology*, 2009.

[48] Wen Tang, Jianwei Ma, and Felix J. Herrmann. Optimized compressed sensing for curvelet-based seismic data reconstruction. In *Preprint*, 2009.

[49] Ali Cafer Gurbuz, James H. McClellan, Justin Romberg, and Jr. Waymond R. Scott. Compressive sensing of parameterized shapes in images. In *EEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008.

[50] Mark A. Davenport, R Marco F. Duarte, R Michael B. Wakin, C Jason N. Laska R, Dharmpal Takhar, R Kevin F. Kelly, and R Richard G. Baraniuk R. The smashed filter for compressive classification and target recognition. 2008.

[51] Allen Yang, John Wright, Yi Ma, and Shankar Sastry. Feature selection in face recognition: A sparse representation perspective. In *Preprint*, 2007.

[52] Julio Martin Duarte-Carvajalino and Guillermo Sapiro. Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization. In *Preprint*, 2008.

[53] Fernando Rodriguez and Guillermo Sapiro. Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries. In *Preprint*, 2008.

[54] Pieter Peers, Dhruv K. Mahajan, Bruce Lamond, Abhijeet Ghosh, Wojciech Matusik, Ravi Ramamoorthi, and Paul Debevec. Compressive light transport sensing. *ACM Trans. Graph.*, 28(1):1–18, 2009.

[55] Pradeep Sen and Soheil Darabi. Compressive Dual Photography. *Computer Graphics Forum*, 28(2):609 – 618, 2009.

[56] J. Gu, S. K. Nayar, E. Grinspun, P. N. Belhumeur, and R. Ramamoorthi. Compressive Structured Light for Recovering Inhomogeneous Participating Media. In *Europian Conference on Computer Vision (ECCV)*, Oct 2008.

[57] Pradeep Sen, Billy Chen, Gaurav Garg, Stephen R. Marschner, Mark Horowitz, Marc Levoy, and Hendrik Lensch. Dual Photography. *ACM Transactions on Graphics*, 24(3):745–755, 2005.