UNIVERSITA' DEGLI STUDI DI BRESCIA
Facoltà di Ingegneria
Dipartimento di Elettronica per l'Automazione

XIX CICLO DOTTORATO DI RICERCA IN
INGEGNERIA DELL'INFORMAZIONE
SSD: ING-INF/03

New Techniques for Signal Representation and Coding

Ph.D. Thesis

Ph.D. Advisor:
Prof. **Riccardo Leonardi**

Ph.D. Coordinator:
Prof.ssa **Valeria De Antonellis**

Ph.D. Candidate:
Ing. Marco Dalai

Anno accademico 2005/2006

A Gabriella

"... ma il discorrere è come il correre,
e non come il portare, ..."
– Galileo Galilei –

Abstract

A fundamental problem in the field of signal processing is the necessity of representing and coding signals, this being the key point for the solution of problems arising in well consolidated situations where a signal available as an information source to a given user has to be either exactly or approximately described to another user. This simply described problem led to the development of whole branches of communication theory after the appearance of Shannon's famous paper [88], part of the developed theory being usually known as source coding theory.

Most of the research work on lossy source coding has been focusing on theoretical aspects of bounds on achievable rates and distortions, while from a practical point of view, lossy encoding techniques have been studied principally for the particular case of the squared error distortion criterion. Also, for some years the problems of source coding were only considered for single sources, i.e. for the case where there is only one user who has some information to be transmitted to another user, while, in successive years, more general multiuser situations were considered

In this work we propose a study of particular representation and coding techniques for signals that are of interest in two sense. We first study techniques for signal approximations under the l^∞ norm, as a counterpart to the more consolidated use of the l^2 norm. Rather than focusing on theoretical discussions on the problem of rate distortion theory under the l^∞ norm, we consider the more concrete problem of constructing approximations to given numerical signals. Then we consider the re-emerging field of Distributed Source Coding (DSC) and its application to Distributed Video Coding (DVC), providing an analysis of the relations between DSC and DVC, studying proposed DVC techniques from the literature and developing methods for the practical design of a DVC system. In relation to the DSC paradigm, within this work we develop a theoretical study of uniquely decodable codes for constrained sequences, providing a revisitation of fundamental results on coding and on expected lengths of codes. Focusing then on the topic of DVC, furthermore, we propose a first study of the problem of registering remote images, providing a framework, based on the phase of the Discrete Fourier Transform, for registration of remote images in the case of shift, rotation and scale factors.

Acknowledgments

I would like to thank my supervisor, Prof. Riccardo Leonardi, for his guidance during my PhD studies, for always giving me the possibility of investigating different research directions and for his sincere support when I had whatever type of difficulties. My gratitude also goes to Prof. Pierangelo Migliorati, for many helpful discussions during these three years and to Dr. Pier Luigi Dragotti, for his help and for research insights during the time spent at the Imperial College.

I would also like to thank Prof. Marco Campi, for supervising my minor but, in particular, for always teaching me so much during our discussions.

I am indebted to all colleagues and friends that contributed to make my PhD a great experience at the Department. Particular thanks to Marzia, Nicola and Sergio, I really enjoyed the time spent with you in our old (sigh) office during the first two years. Also many thanks to Fabrizio, Francesca, Manuel, Dario, Livio, Alberto, Luca, Francesco, Michele and, more recently, Claudia and Roberto. As for my period in London, I would like to thank Jesse, Loïc, Nick and in particular Nicolas for their friendly hospitality.

I want to express deep gratitude to my whole family. Thanks to my parents, to my brother and my grandmothers, you were always able to express great interest in my PhD. A special thank to Gabriella, for her invaluable support during the whole three years and in particular during this last one! Thank you so much.

Contents

Introduction

Signal representation and coding is a key problem in the field of signal processing, as it is always important to have both a good way to represent signals in a digital form and to have good techniques for the encoding of these representations so as to efficiently store or communicate them. In this thesis some new aspects of representation and coding are considered and studied, keeping as a main objective the investigation of innovative approaches to some problems that are either not much studied in the signal processing community or either of recently increasing interest.

In signal processing a well consolidated setting for representation and coding problem is the situation where a signal available as an information source to a given user has to be either exactly or approximately described to another user. This simply described problem contains all the elements that lead to the development of a whole branch of communication theory that deals with the lossless and lossy representation of information sources, i.e. source coding theory [28]. In source coding theory, starting from a probabilistic description of the information sources, the bounds on the number of bits required to represent a source within a given precision are studied. This field of research initiated by Shannon in his famous paper [88] has been receiving much attention in the years and many subproblems have been identified. What is important to the present thesis are the following two facts:

1. Most of the research work on lossy source coding in its general form has been focusing on theoretical aspects for the study of bounds on achievable rate-distortion trade-off under certain probabilistic model assumptions. From a practical point of view, lossy encoding techniques have been studied principally only for the case where signals are real functions of time (or space) and the distortion is measured as the quadratic difference between an original signal and its approximation.

2. For some years the problems of source coding were only considered for single sources, i.e. for the case where one user has some information to be transmitted to another user, without any further entity involved in the problem. In successive years more general situations were considered where more information sources may be involved in a multiuser communication scenario [28]. In this new situation, different encoding techniques are possible. They take advantage of the presence of different sources available to different users. Even if from an information theoretic point of view these

multiuser problems have been extensively studied from the late 70's, practical applications of the underlying ideas have been only very recently conceived.

The two above items are the starting points for the definition of the aims of the work presented in this thesis. The main objective is in fact the study of particular representation and coding techniques for signals that are innovative in the sense of the two above mentioned points. In particular, this thesis is evolves in different directions.

As a first topic, we develop a study of techniques for signal approximations under the l^∞ norm, as a counterpart to the more consolidated use of the l^2 norm. Rather than focusing on theoretical discussions on the problem of rate distortion theory under the l^∞ norm, we consider the more concrete problem of constructing approximations to given numerical signals. We then consider the re-emerging fields of Distributed Source Coding (DSC) and Distributed Video Coding (DVC) for the development of research contributions that are related to these topics. In particular we provide an analysis of the relations between DSC and DVC, with the study of proposed DVC techniques from the literature, and the development of new methods for the practical design of a DVC systems. We also take inspiration from DSC and DVC topics for the development of a detailed study of some problems that are of interest on their own. In particular, we develop an information theoretic study of decodable codes for certain sources with memory, and we close the thesis with a study of a problem of remote image registration which is a formalization of a problem frequently encountered in DVC systems.

The main innovative contributions of the thesis are contained in Chapters 1, 4, 5 and 6, whose content has partially been (or is going to be) published in [29, 34], [35], [32] and [36] respectively. Chapters 2 and 3 mainly contain discussions and presentation of research results from the literature.

Structure of the thesis by chapter.

In Chapter 1 the problem of signal approximation in the l^∞ norm is studied from a practical point of view, proposing algorithms and techniques for the concrete construction of piecewise approximation under this distortion criterion, that has not received much attention in the literature. In particular approximations of signals in linear spaces are studied and an efficient algorithm is presented for the particular case of straight line approximations. Piecewise approximations are then studied and algorithms for the construction of minimal and optimal approximations are proposed, with an analysis of the associated computational complexity. The problem of the encoding of the obtained approximations is considered both for the case of approximation in general linear spaces, and for the particular case of straight line approximations.

In Chapter 2 the topic of Distributed Source Coding (DSC) is introduced. The main results are presented and the fundamental ideas underlying DSC are explained using simple examples. The basic ideas of what DSC is are first presented without any theoretical analysis by means of a simple example. Then, the Slepian-Wolf and the Wyner-Ziv theorems, the two fundamental results of DSC, are introduced and explained. The connection between

DSC and channel coding is finally analyzed with some detail in order to provide elements to the discussions presented in the following chapters.

Chapter 3 introduces the recently emerged field of Distributed Video Coding (DVC). DVC is the application of DSC principles to the practical problem of the encoding of video sources. In this chapter the first constructions of DVC systems proposed by Berkeley's and Stanford's groups for the single video source problem are presented in details. Some comments are given on the two proposed approaches and a brief review of the most recent contribution from the literature is provided.

In Chapter 4 a two-fold contribution to the problem of DVC is given. We first propose a structural analysis of the use of DSC principles in video coding problems and we focus the attention on the differences between the theory of DSC and the practical problems encountered in a DVC system. We put primarily the attention on the "correlation issue" and we clarify the connection between requirements and motivations for the use of DVC with some structural constraints to be faced in practical systems.

In Chapter 5 a theoretical study on the use of DSC like codes for the encoding of sources with memory is provided. As a theoretical modelization of a single camera DVC system, we consider the use of particular types of codes for encoding certain memory sources. In particular, in this chapter, the theory of unique decodability is revisited for the case of constrained sources, providing some unexpected results in the field of lossless coding. A detailed analysis of the equivalence between McMillan theorem on unique decodability and a previous channel coding theorem by Shannon is provided.

In Chapter 6 we propose a study of a topic, that we call "remote image registration", that can be considered as a fundamental component in the use of distributed coding techniques for images and video. In this chapter we consider the following problem. Let X and Y be two images that have the same content apart from some shift, rotation or scale factors. A transmitter has access to X while a receiver has access to Y. We study the problem of extracting information from X to be sent to the receiver, so that it can recover the shift, rotation and scale of its own image Y with respect to X. We call this problem "remote image registration" because it is indeed a problem of image registration where the two images are not available at the same point, and only a low rate description of one of them can be used. We show how to solve the registration of shifts by using appropriate sampling and quantization of the phase of the phase of Discrete Fourier Transform, and we then extend the method to handle rotation and scale components.

Chapter 1

Approximations of signals under the l^∞ norm

All exact science is dominated
by the idea of approximation.
– Bertrand Russell –

1.1 Introduction

Approximation of discrete signals by means of continuous time functions has been studied extensively in the literature (see for example [75, 101] and [70] and references therein for an overview). Most of the attention has been dedicated to approximations under the l^2 norm, which means that the goodness of the approximation is established by evaluating the mean square value of the error. In this chapter we study approximations under the l^∞ norm.

Given a discrete set of points $D = \{x_i\}$, $x_i \in \mathbb{R}$, we consider a discrete signal s as a function $s : D \to \mathbb{R}$ that associates a real valued $s(x_i)$ to each value x_i in D. We indicate with l^∞ the set of functions f bounded over D and with $\|\cdot\|_\infty$ the norm defined, for $f \in l^\infty$, by

$$\|f\|_\infty = \sup_{x \in D} |f(x)|.$$

As usual, the distance between two functions f_1 and f_2 is then defined as the norm of the difference function, i.e. $d_\infty(f_1, f_2) = \|f_1 - f_2\|_\infty$. The problem of approximating a signal s under the l^∞ norm consists on finding a function g from a given set of functions G such that the approximation error, that is the distance $d_\infty(g, s)$, satisfies some given constraint. In some cases we will be interested in finding g which approximates s with an error smaller than a given threshold δ, while in other cases we will want g to be the function of G that

[0]This chapter includes research results published in [34].

minimizes this error. In the following sections we will study some of these particular problems analyzing in details some rather interesting special cases. The chapter is organized as follows. In Section 2 we briefly present the important case of approximations in linear spaces; we summarize the current approach for finding the optimal approximation, which reduces to solving a linear program, and we show how to use this technique for the solution of a more general approximation problem. For this section we refer the reader to [101] for a detailed analysis of the approximation problem and to [65] for a general study of linear and non linear programming theory, even if it is not necessary for the understanding of the chapter. In Section 3 we consider the particular case of straight line approximations; we propose an efficient geometric algorithm for finding the optimal solution, showing the computational advantage of this method over the currently best performing linear programming technique. Then, in Section 4, we consider the problem of piecewise approximations in linear spaces. We show how to partition a given signal into a minimum number of segments so as to obtain a piecewise approximation within a given tolerance; we then show how to optimize the partition so as to minimize the error with the same number of segments. Finally, in Section 5, we analyze the case of straight line piecewise approximations presenting a more efficient procedure based on the results of Section 3. For a deeper analysis of computational geometry and optimization techniques used in Sections 3 to 5 we refer the reader to [79, 51] and [26, section VI] but still this is not necessary for the understanding of the proposed methods.

1.2 Linear Spaces and Linear Programming

A very important special case of approximation problems is obtained when the domain D contains only a finite number n of points x_i, $i = 1 \dots n$, and the set G is a linear space generated from a finite set of basis functions. If $m \leq n$ is a fixed integer, we take a set $B = \{b_1, b_2, \dots, b_m\}$ of m linearly independent functions b_j[1], and we consider the set $G = \text{span}(B)$; this means that for every $g \in G$ there exists a sequence of coefficients c_j such that

$$g = \sum_{j=1}^{m} c_j b_j. \tag{1.1}$$

This hypothesis implies that every possible approximation g to the signal s is uniquely identified by a sequence of real coefficients that are its representation in the B basis. Now, for uniformity with the literature, let us map every function g of G in the vector $\mathbf{g}$ of $\mathbb{R}^n$ whose i-th component is the value $g(x_i)$, so as to work in a subspace over $\mathbb{R}^n$ instead of the space G.
If $\mathbf{A} \in \mathbb{R}^{n \times m}$ is the matrix with elements $a_{ij} = b_j(x_i)$, equation (1.1) is mapped to

$$\mathbf{g} = \mathbf{A}\mathbf{c}.$$

[1]Here the term "linearly independent" will mean that any non-trivial linear combination of the b_j functions cannot be null over every point of D.

Thus, if we want to find the optimal l^∞ approximation of a signal s in F, we have to find the coefficient vector $\mathbf{c} \in \mathbb{R}^m$ that minimizes the value

$$\|\mathbf{s} - \mathbf{A}\mathbf{c}\|_\infty \tag{1.2}$$

This problem has been studied extensively in the mathematical and mathematical programming literature (an exhaustive overview can be found in [101]; see [15] for a classic result) and the most recent approach consists in converting it to a linear program in $m + 1$ dimensions. Accordingly, let $\mathbf{u} \in \mathbb{R}^n$ be the vector with all its components equal to 1; then, for every fixed $\mathbf{c}$, the value in equation (1.2) is given by the smallest possible value of e, say $e^*(\mathbf{c})$, that satisfies

$$-e\mathbf{u} \leq \mathbf{s} - \mathbf{A}\mathbf{c} \leq e\mathbf{u},$$

where inequalities between vectors are to be intended, here and in what follows, component by component. Subsequently, minimizing (1.2) is equivalent to minimize $e^*(\mathbf{c})$ as a function of $\mathbf{c}$.

If we set $\mathbf{d} = (\mathbf{c}, e)$ and we call $\mathbf{e}_{m+1}$ the $(m + 1)$-th vector of the canonical base of $\mathbb{R}^{m+1}$ (i.e the vector whose $(m + 1)$-th component is equal to 1 and all other components are zero), we are minimizing the linear function

$$z = \mathbf{e}_{m+1}^T \mathbf{d}$$

under the conditions

$$\begin{bmatrix} \mathbf{A} & \mathbf{u} \\ -\mathbf{A} & \mathbf{u} \end{bmatrix} \mathbf{d} \geq \begin{bmatrix} \mathbf{s} \\ -\mathbf{s} \end{bmatrix}$$

This formulation is exactly the enunciate of a linear programming problem in $m + 1$ dimensions. Thus, for finding the solution of the approximation problem, it is possible to take advantage of the most advanced linear programming techniques that are available in the literature. In our case, however, it is particularly interesting to note that, if the parameter m can be considered fixed and much smaller than n, it is possible to solve the problem in $O(n)$ expected operations, as shown in [87, 69] (see also [81, Ch. 9]). In the following we will always consider the parameter m to be constant, and we will thus assume that in linear spaces it is possible to compute the optimal l^∞ approximation in linear time (with respect to the number n of samples).

It is interesting to note that the idea of l^∞ approximation can be extended to a more general approach. Suppose, indeed, that we are still interested in controlling the approximation error in every point, as in l^∞ approximations, but assigning different weights (or, more precisely, different offsets) to different domain coordinates, i.e. approximate s with a maximum error that differs from point to point. Formally this is expressed by stating that we want to find a function g such that

$$|s(x_i) - g(x_i)| \leq t(x_i), \quad i = 1 \ldots n \tag{1.3}$$

where $t(x_i)$ is the allowed error in the point x_i. Interestingly this problem can be treated in the same way, by using the additional variable e and minimizing e subject to the constraints

$$|s(x_i) - g(x_i)| \leq t(x_i) + e, \quad i = 1 \ldots n \tag{1.4}$$

In this case, clearly, we aim at finding a non-positive value of e, thus verifying if the problem is feasible or not (i.e. a function g satisfying eq. (1.3) exists). If a positive value is obtained, we conclude that the problem is not feasible but having reached the knowledge of how far we are beyond the tolerance t. If, instead, a negative value of e is obtained, we know that the problem is feasible and we find the approximation that maximizes the margin from the threshold. In the latter case, however, it is important to note that the minimum value of e cannot be less than $-\min_i(t(x_i))$, as the values on the right hand side of eq. (1.4) must be non-negative. Thus, by calling x_m the point in which t reaches the minimum, in some cases it is possible to fit s in x_m while still satisfying eq. (1.3) for every other point x_i. In this case, in the linear program, we have a constraint (given by the point x_m, i.e $|s(x_m) - g(x_m)| \leq t(x_m) + e$) that is orthogonal to the minimization vector and thus the solution is not unique. In this situation it could be convenient to project the problem into the hyperplane $s(x_m) - g(x_m) = 0$, so as to optimize the approximation over $x_{i \neq m}$ while imposing exact interpolation in x_m. We conclude by clarifying that a program for the classical l^∞ approximation can be used for this more general type of approximations. In fact, let d be a constant such that $d > \|t\|_\infty$. Thus, if we set $s^+(x) = s(x) - t(x) + d$ and $s^-(x) = s(x) + t(x) - d$, it is easy to see that eq. (1.3) is equivalent to

$$\begin{cases} |s^+(x_i) - g(x_i)| \leq d \\ |s^-(x_i) - g(x_i)| \leq d \end{cases} \quad i = 1 \dots n$$

Thus, the approximation of s with a variable tolerance t can be obtained by approximating s^+ and s^- jointly with the usual l^∞ norm. This idea has been of practical utility, in the field of image coding, for example in [30], where a separable approach has been used for the problem of finding bidimensional l^∞ sub-optimal bilinear approximations. We refer the reader to [30, 31, 33] for more details.

1.3 Linear approximations

In this section we study the particular case of linear approximations, i.e. by means of first order polynomials. In this case the general function g of G is expressed as $g(x) = ax + b$ where a and b are real numbers. It is clear that in this case we can take $B = \{1, x\}$; thus the space G has dimension 2 and the problem of finding the best approximation of a given signal s is equivalent to solve a 3-dimensional linear program. In particular, we have to minimize the linear function

$$z = [0, 0, 1] \begin{bmatrix} a \\ b \\ e \end{bmatrix}$$

under the constraints

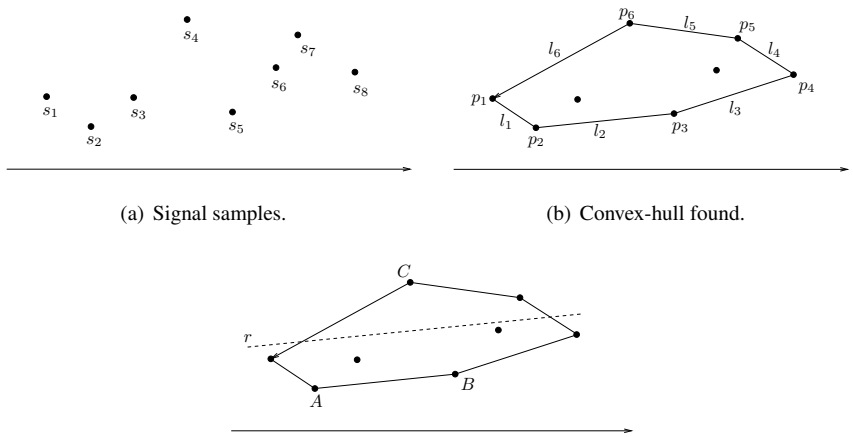
$$\begin{bmatrix} x_1 & 1 & 1 \\ x_2 & 1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & 1 & 1 \\ -x_1 & -1 & 1 \\ -x_2 & -1 & 1 \\ \vdots & \vdots & \vdots \\ -x_n & -1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ e \end{bmatrix} \geq \begin{bmatrix} s(x_1) \\ s(x_2) \\ \vdots \\ s(x_n) \\ -s(x_1) \\ -s(x_2) \\ \vdots \\ -s(x_n) \end{bmatrix}$$

For what has been said at the end of the previous section, very efficient linear programming techniques are available for this problem and the optimal approximation can be found in $O(n)$ expected number of operations. The expectation is due to the fact that such linear programming techniques are based on randomized methods and thus the number of operation used for a fixed signal is a random variable. We propose here a geometric based algorithm which can outperform the linear programming technique, by exploiting the particular nature of the problem. The advantages of this algorithm will be detailedly exposed in the next section; we only remark here that it is deterministic and uses $O(n)$ operations in the worst case, as we will prove in what follows.

1.3.1 Geometric algorithm

Let $D = \{x_i\}_{i=1\ldots n}$ be the domain of n points of $\mathbb{R}$, s be the signal, and let S be the set of n points s_i of the signal samples in the plane, i.e. $s_i = (x_i, s(x_i))$. Let Q be the convex hull of S, that is the smallest convex polygon that contains every point of S. Let us define some notations for clarity. Let k be the number of sides of Q; we indicate with p_i, $i = 1, \ldots, k$, the vertices of Q in counterclockwise order, with p_1 the left most one. For convenience we add a new point $p_{k+1} = p_1$; then we indicate with l_i, $i = 1 \ldots k$, the side $\overline{p_i p_{i+1}}$. Let m be the integer such that p_m is the right most vertex of Q; then we will call lower-hull the polygonal line formed by the sides l_i, $i = 1 \ldots m-1$, and upper-hull the polygonal line formed by the sides l_i, $i = m \ldots k$. We will consider that the vertices p_1 and p_m belong to both the upper- and lower-hull. Finally, given a side l and a point p we will say that p is x-internal to l if the vertical line through p cuts the side l; on the contrary, we will say that p is x-external on the left or on the right, the meaning being obvious.
Now, suppose for a moment, for simplicity of the presentation, that Q has no pair of parallel sides. Then, to each side l of Q it is possible to find a vertex $v(l)$ of Q that is the most distant one from l in the orthogonal direction; we call $v(l)$ *opposite vertex* to the side l.

Proposition 1.3.1 *Under the above hypothesis, there exists one and only one side l of Q such that $v(l)$ is x-internal to l. The optimal linear approximation of the signal s over the domain D is then the line r parallel to l and equidistant from l and $v(l)$. (For a proof see appendix A). We call the extremities A and B, $A < B$, of the side l and the opposite vertex*



(a) Signal samples.

(b) Convex-hull found.

(c) Solution found.

Figure 1.1: Steps of the geometric method for single link optimal solution

$C = v(l)$ pivot points of the set S, so as to identify the three points that determine the optimal linear approximation.

This proposition gives a very useful property of the geometry of the polygons and by using this proposition we can construct a very efficient geometric algorithm to find the l^∞ optimal linear approximation of a signal s (see Fig. 1.1).

Algorithm 1

- *Compute the convex-hull of the set S,*
- *scan the sides of the convex-hull computing their opposite vertex until the pivot points A, B and C are found,*
- *compute the solution line r.*

We now give a detailed explanation of the first two steps of Algorithm 1 (the third step is only a simple computation), for which we propose efficient sub-algorithms showing that the number of operations is $O(n)$.

Computing the convex-hull

Finding the convex-hull of a set of points in the plane is one of the most studied problems of computational geometry and several algorithms are available for this task (see [14] and [79]). One important thing to be considered here is that the points are sorted with respect

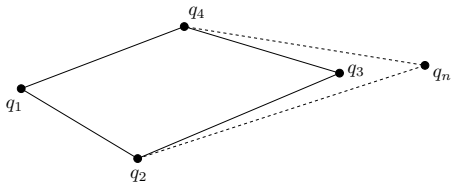


Figure 1.2: Example of construction of the convex hull with Graham's method. The point q_n has to be inserted after q_2 because $\overrightarrow{q_1q_2} \times \overrightarrow{q_2q_n} > 0$ while $\overrightarrow{q_2q_3} \times \overrightarrow{q_3q_n} < 0$. Symmetrically, q_n has to be inserted before q_4 because $\overrightarrow{q_4q_3} \times \overrightarrow{q_3q_n} > 0$ while $\overrightarrow{q_1q_4} \times \overrightarrow{q_4q_n} < 0$

to the x coordinate. Under this hypothesis it is possible to find the convex-hull of the set S in $O(n)$ operation using Graham's algorithm ([52]). Here we recall only that the main idea is to construct the convex-hull by moving from left to right; at every step the polygon is updated by adding a new point and removing the sides of the polygon that are visible by the entering point. The only basic operation that is required for this algorithm is the evaluation of the order of three generic points q_1 q_2 and q_3 in the plane[2], and it is easy to see that this evaluation is nearly equivalent to the evaluation of the vector product $\overrightarrow{q_1q_2} \times \overrightarrow{q_2q_3}$ (see Fig. 1.2 for a graphical explanation. See also [26, Sec. 33.3] for a detailed description of these operations).

Finding the vertices A, B and C

Once we have constructed the convex hull Q of the set S we have to search the side l such that its opposite vertex $v(l)$ is x-internal to l. We now state some simple lemmas that suggest an efficient way to find the searched l and $v(l)$. These lemmas are proved in appendix, where they are also used for the proof of proposition 1.3.1.

Lemma 1.3.2 *Every side of the lower-hull has its opposite vertex in the upper-hull and* viceversa.[3]

Lemma 1.3.3 *If we move from one side of the polygon to its consecutive in counterclockwise (ccw) direction, the respective opposite vertex, if it changes, moves in ccw direction too.*

Lemma 1.3.4 *A Vertex p_j, $1 < j \leq k$, is the opposite vertex of a side l_i, i.e. $p_j = v(l_i)$, if it is more distant from l_i than the vertices p_{j-1} and p_{j+1}.*

These considerations lead to a good algorithm for finding the opposite vertex of each side of the lower-hull, and thus also the searched l and $v(l)$. As a general notation we call j_i the integer such that $p_{j_i} = v(l_i)$.

[2]The order of three points q_1 q_2 and q_3 is defined to take value: 0 if the three points are aligned, 1 if the oriented polygonal $q_1 \rightarrow q_2 \rightarrow q_3$ turns counterclockwise and -1 if it turns clockwise.

[3]Remember that p_1 and p_m belong to both the upper- and lower-hull

Algorithm 2

- *Find the opposite vertex of l_1: starting from p_m scan in ccw direction the vertices of the upper-hull, computing their distances from l_1 until we find a vertex p_{j_1+1} which is less distant from l_1 than p_{j_1}. Thus $v(l_1) = p_{j_1}$.*
- *Continue by considering the sides of the lower-hull to find their opposite vertices. For each side l_i we have to control the vertices of the upper-hull from $v(l_{i-1})$ (in ccw direction) until we find a vertex p_{j_i+1} that is less distant from l_i than p_{j_i}. Then $v(l_i) = p_{j_i}$.*
- *Do the same, symmetrically, for the upper-hull sides.*

Proposition 1.3.5 *Algorithm 2 requires at most $3k$ vector product evaluations[4] of the type $\overrightarrow{p_i p_{i+1}} \times \overrightarrow{p_{i+1} p_j}$ for finding the opposite vertices of all convex-hull sides.*

Proof. Consider for a moment only the number of distance evaluations required for finding the opposite vertices of the lower-hull sides. Consider the generic side l_i and suppose we have found the opposite vertex $p_{j_{i-1}}$ of the previous side l_{i-1}, i.e. $p_{j_{i-1}} = v(l_{i-1})$. It is easy to see that for finding the opposite vertex p_{j_i} of l_i one must compute $2 + (j_i - j_{i-1})$ distances. For example, suppose $v(l_2) = p_8$ and $v(l_3) = p_{10}$; if $p_8 = v(l_2)$, in order to find $v(l_3)$ one has to compute the distances of p_8, p_9, p_{10} and p_{11} from l_3, and thus $4 = 2 + (10 - 8)$ distances. For the first side l_1, the same argument holds setting $j_0 = m$, as for the side l_1 we start check the vertices starting from p_m. This means that the total number of computed distances is $\sum_{i=1}^{m-1}(2 + j_i - j_{i-1}) = 2(m-1) + (j_{m-1} - m)$. But clearly $j_{m-1} \leq k + 1$ and thus the opposite vertices of the sides of the lower hull are found by computing at most $k + m - 1$ distances. Considering the symmetry of the problem we can say that the opposite vertices of the upper-hull sides can be found by computing at most $2k - m + 1$ distances, for a total of at most $3k$ distance evaluations. It is clear, however, that the algorithm will stop, for the problem of interest, when the side l with opposite x-internal vertex has been found. Finally, we now show that in fact one does not need to compute 3k distances but only $3k$ vector products of the type $\overrightarrow{p_i p_{i+1}} \times \overrightarrow{p_{i+1} p_j}$, which represent a smaller computational cost. In fact, when searching the opposite vertex of the generic side l_i, we do not need to really know the distances of the generic point p_j from l_i, but only compare the values for different j. Considering that the distances of p_j from l_i (for varying j but fixed i) are proportional to the areas of the triangles of vertices p_i, p_{i+1}, and p_j, we can compare the value of these areas instead of the distances. Since twice the area of the triangle of vertices p_i, p_{i+1}, and p_j equals the vector product $\overrightarrow{p_i p_{i+1}} \times \overrightarrow{p_{i+1} p_j}$, this implies that the algorithm requires only $3k$ such vector products. ∎

Going back to Algorithm 1, it can be stated that this algorithm requires $O(n)$ operations and that the only required basic function is the evaluation of vector products. Whereas from

[4] Recall that k is the number of sides of the convex hull.

a theoretical point of view Algorithm 2 is quite useful, it can be further improved by using the following property of a convex-hull Q.

Lemma 1.3.6 *Given two consecutive sides l_i and l_{i+1}, their common vertex p_{i+1} is opposite vertex of every side between $v(l_i)$ and $v(l_{i+1})$ (in the path not containing l_i and l_{i+1}, obviously).*

With such a consideration Algorithm 2 can be modified as follows:

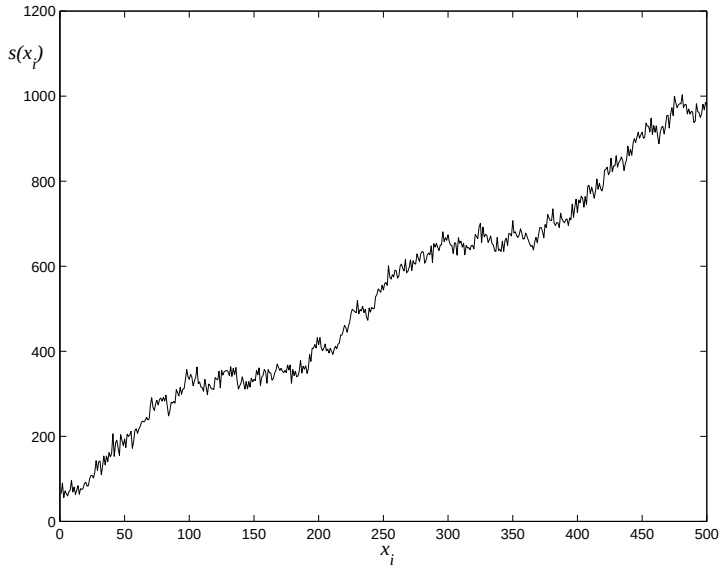
Algorithm 3

- *Find $v(l_1)$, opposite vertex of l_1, as suggested in Algorithm 2;*
- *for $i \geq 1$, once $v(l_i)$ is found check*
  - *a) if $v(l_i)$ is x-external to l_i on the right go on searching $v(l_{i+1})$,*
  - *b) else if $v(l_i)$ is x-internal to l_i terminate the search,*
  - *c) else if $v(l_i)$ is x-external to l_i on the left, search the side of the upper-hull between $v(l_{i-1})$ and $v(l_i)$ such that the common vertex of l_{i-1} and l_i, i.e. p_i, is x-internal to it.*

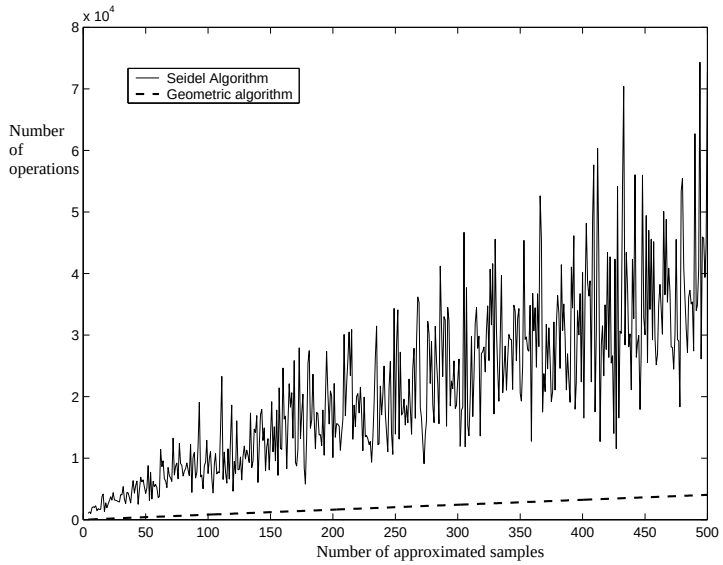
It is important to note that this algorithm is strongly based on the proof of Proposition 1; this ensures that one of the items b) or c) is reached before finishing scanning the sides of the lower-hull and thus the algorithm always finds the solution. With this algorithm the number of computed vector products is reduced by about a factor 2 in the mean case with respect to the performance of Algorithm 2.

1.3.2 Performance comparison

Compared to the linear programming solution the geometric algorithm has many advantages. The first one is that it is very easy to implement and it generates a very compact code. As it has been shown, all the computations in the construction of the convex-hull and the scanning of its side-vertices pair can be reduced to a vector product operation; thus, the implementation requires a few loops calling a simple function for the computation of a vector product. Furthermore, as the vectors are always coplanar, this operation is only a sum of products of the type e=ab+cd, which can be executed very efficiently on many DSP's. Moreover, the memory usage is very limited; the only memory space needed (apart from the input sequence) is a vector containing the indices of the points that are vertices of the convex-hull Q, which represent at most n integers. Furthermore, it is important to note that, if we are working with discrete signals, almost all the computations can be performed using only fixed point arithmetic. The only need for floating point operations is indeed due to the construction of the optimal line from the pivot points and the evaluation of the approximation error, which represent a fixed number of operations. This is an important consideration in case a floating point unit is not available.



(a) Signal used for approximations. The near-linear behaviour has been obtained by summing small sinusoidal functions and white gaussian noise to a straight line.



(b) Number of operations used by the two algorithms.

Figure 1.3: Comparing the number of operations used by the geometric method and Seidel's randomized algorithm for linear programming in small dimensions.

Finally, an important consideration is about the computation time. First of all, from a theoretical point of view, our algorithm has a computation time that is $O(n)$ in the worst case, while the linear programming techniques can only provide a solution in $O(n)$ on average. For practical considerations, then, we have compared our algorithm with an *ad-hoc* implementation of the Seidel randomized algorithm for linear programming in small dimensions [87], which is known to be very fast for this kind of problems. For this purpose we have counted the number of operations used by the two algorithms, so as to remove any dependency on the machine architecture, type of data (we recall that our algorithm can work without using floating point arithmetics) and, mostly important, memory usage[5]. We have taken a signal with near-linear behavior, shown in Fig. 1.3(a), and we have computed the linear approximation of the first n points, with n varying from 4 to 500. The number of operations used by the two methods, as a function of n, are plotted in fig. 1.3(b). As it can be seen, the Seidel algorithm presents an irregular behavior, due to its randomized nature, having linear complexity in the mean. The geometric method, instead, gives a regular increase of the number of operations, which leads to a speed up by a factor ranging from 3 to more than 15 with respect to the Seidel algorithm, with a mean gain of about 8.

1.4 Piecewise approximations with error bound

Often signal approximation in linear spaces is not a practical tool for signal processing and coding due to the fact that the dimension of the approximation space must increase with the number of samples if we want to keep small values of the error. Thus, it is necessary to divide the domain in smaller subdomains (intervals) such that the signal can be approximated with small error in a small dimensional space within every domain. With this idea, we define a more general function space: given a set of functions B (and the induced G) we call G_T the set of functions g over D for which there exists a partition of D in subdomains Δ_k such that the restriction of g to every Δ_k is in G.

In this section and in the next one, we study the problem of optimizing the partition of the domain into connected subdomains while approximating the signal within an error threshold. Consider that, given a partition of the domain, the problem of approximating the signal within each subdomain is only an application of what has been described in the preceeding section 1.2. So, in the following, the emphasis will be addressing mainly the partition of the domains. For the sake of clarity, we suppose that the set S is characterized by $x_i = i$, $i = 1 \dots n$, even though the presented results hold in the case of non uniform samples. Given any piecewise approximation g of the signal, we characterize it with an error $e(g)$, a number of connected subdomains $\nu(g)$ and a *partition set* $P(g) = \{p_i(g)\}_{i=1..\nu(g)-1}$ of values such that $p_i = m + 1/2$ if m is the last point of the i-th interval and $m + 1$ is the

[5]It is also relevant to notice that, for a fast implementation of the Seidel algorithm, one should not make use of dynamic memory allocation; this implies the necessity of allocating more than $30n$ floating point variables, against the n integers of the geometric algorithm. If instead one wants to reduce memory usage (in any case much more than n integers), memory should be allocated dynamically, thus leading to a significant reduction of the computational efficiency.

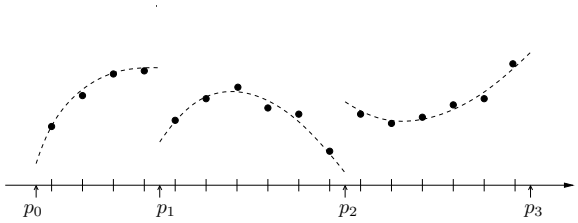


Figure 1.4: Example of piecewise approximation of a 16 sample signal with associated partition points. Here $\nu(g) = 3$, $p_0 = 0.5$, $p_1 = 4.5$, $p_2 = 10.5$ and $p_3 = 16.5$

first point of the $(i+1)$-th interval. Moreover, we set $p_0(g) = 1/2$ and $p_{\nu(g)}(g) = n + 1/2$, and it is implicitly considered that the partition points p_i can only take values of the type $m + 1/2$ with $m \in \mathbb{N}$. See fig. 1.4 for an example of piecewise approximation with the associated partition points. All considered intervals[6] are measured on a discrete half integer value. Thus we will identify the "approximation on the interval [3/2,7/2]" as "the one of locations 2 and 3"; similarly, by stating "the partition point p_i is in $]3/2, 7/2]$" is equivalent to say "$p_i \in \{5/2, 7/2\}$".

We now study the problem of optimally partitioning the domain, by introducing the idea of minimal and optimal approximations for a given error threshold δ.

1.4.1 Minimal Solution

The problem to be solved is the following: given the set S of n samples of signal s, the set B of the basis functions and an error bound δ, we want to find an approximation $g \in G_T$ of s with error $e(g) \leq \delta$ such that the number $\nu(g)$ of intervals is the smallest possible.
In general there are more solutions to this problem, and we aim at finding at least one of them. Interestingly enough, it is possible to find two solutions (not necessarily distinct) with a very simple algorithm, by scanning the signal in a progressive fashion.

For finding these solutions we first need an algorithm that finds, given any point $s_k = (x_k, s(x_k))$, the "longest" possible approximation of s starting from it in one direction, e.g. the maximum value l such that the points $s_i = (x_i, s(x_i))$, $i = k \dots k + l$ can be approximated in B with error smaller than the threshold δ. This leads to an approximation of the l points which is consistent with the error constraint δ. The algorithm is the following:

Algorithm 4

- *Compute the optimal approximations over the intervals* $[k, k+1]$, $[k, k+2], \dots, [k, k+2^j], \dots$ *until an error larger than* δ *is obtained (which happens for* $j = \lceil \log_2 l \rceil$*); call* $a = \lceil \log_2 l \rceil$.
- *Find* l *with a binary search on the interval* $[2^{a-1}, 2^a[$.

[6] We use bracket notation for intervals. So, $[a, b]$ is the interval containing both a and b while $]a, b[$ contains none, $[a, b[$ contains a but not b and $]a, b]$ contains b but not a.

Proposition 1.4.1 *Algorithm 4 requires an expected number of $O(l \log l)$ operations.*

Proof. Consider the first step of the algorithm; the optimal approximation over the generic interval $[k, k + 2^j]$ can be found in $O(2^j)$ expected operations using Seidel randomized algorithm, as explained in section 1.2. Thus, a is found in $\sum_{j=0}^{a} O(2^j) = O(2^{a+1})$ operations. Then, the second step of the algorithm computes at most a approximations of length less than or equal to $2l$. So, the expected number of operations for the second step is $O(a \cdot l) = O(l \log l)$, which is the dominating term. ∎

We now apply the proposed algorithm for the construction of two approximations that we will prove to be minimal. We indicate these approximations with $\overrightarrow{g}$ and $\overleftarrow{g}$ so as to emphasize the fact that they are obtained by scanning the signal respectively from left to right and viceversa. Here we give the algorithm for finding $\overrightarrow{g}$.

Algorithm 5

- *Start by scanning the signal from the first point s_0. Using Algorithm 4, find the first longest possible approximation segment, and thus the partition point $p_1(\overrightarrow{g})$. Set i to 1.*
- *Given $p_i(\overrightarrow{g})$, compute the longest possible approximation segment (using Algorithm 4) starting from $p_i(\overrightarrow{g})$, and thus find $p_{i+1}(\overrightarrow{g})$. Repeat until the end of the signal is reached.*

Proposition 1.4.2 *Algorithm 5 requires an expected number of $O(n \log n)$ operations.*

Proof. Note that, set $l_i = p_i(\overrightarrow{g}) - p_{i-1}(\overrightarrow{g})$, from proposition 1.4.1, we need $O(l_i \log l_i)$ operation for finding $p_i(\overrightarrow{g})$. Thus, we need $\sum_i O(l_i \log l_i)$ operations; considered that $\sum_i l_i = n$, we have $\sum_i O(l_i \log l_i) < \sum_i O(l_i \log n) = O(n \log n)$. ∎

Clearly, the same procedure of Algorithm 5 can be used analyzing the signal from right to left to obtain the approximation that we denote with $\overleftarrow{g}$. It is also clear that both $\overrightarrow{g}$ and $\overleftarrow{g}$ depend on δ; we now show that they are indeed minimal for that given value of δ.

Proposition 1.4.3 *The two approximations $\overrightarrow{g}$ and $\overleftarrow{g}$ lead to the same number of segments $k = \nu(\overrightarrow{g}) = \nu(\overleftarrow{g})$, and every approximation h such that $e(h) \leq \delta$ satisfies $\nu(h) \geq k$.*

Proof. Consider the construction of $\overrightarrow{g}$. The way $p_1(\overrightarrow{g})$ was obtained implicitly says that it is not possible to approximate the interval $[1, p_1(\overrightarrow{g})+1]$ with a single segment (without exceeding the value of δ); h cannot be an exception and thus $P(h)$ must have a partition point $p_1(h)$ in the interval $[1, p_1(\overrightarrow{g})]$. Similarly it is not possible to approximate with a single segment the interval $[p_1(\overrightarrow{g}), p_2(\overrightarrow{g})+1]$, so that $P(h)$ must have at least another point $p_2(h)$ in $]p_1(\overrightarrow{g}), p_2(\overrightarrow{g})]$ as $p_1(h) \leq p_1(\overrightarrow{g})$. By iterating the argument, this proves by induction that for $1 \leq i \leq k-2$ there must exist a point of $P(h)$ in the interval $]p_i(\overrightarrow{g}), p_{i+1}(\overrightarrow{g})]$

and thus $\nu(h) \geq k$. In particular, setting $h = \overleftarrow{g}$, we obtain that $\nu(\overleftarrow{g}) \geq \nu(\overrightarrow{g})$; but, by symmetry of the construction process, in the same way we could prove that $\nu(h) \geq \nu(\overleftarrow{g})$ and thus, setting now $h = \overrightarrow{g}$, $\nu(\overrightarrow{g}) \geq \nu(\overleftarrow{g})$. This means that $\nu(\overrightarrow{g}) = \nu(\overleftarrow{g})$ and that this number k of intervals is minimal. ∎

1.4.2 Optimal Solution

In the preceeding section we have seen how to find a piecewise approximation that uses the minimum number of intervals in $O(n \log n)$ operations. More specifically, we have seen that it is possible to find two solutions $\overrightarrow{g}$ and $\overleftarrow{g}$, each being minimal. Now, given that the number of used intervals cannot be further lowered, we can ask for the minimal approximation that minimizes the approximation error. For this task, we now show that the $\overrightarrow{g}$ and $\overleftarrow{g}$ solutions provide two partition sets that are a sort of extremes of the possible partition sets of any minimal approximation. More precisely, we have the following.

Proposition 1.4.4 *If h satisfies $e(h) \leq \delta$ and $\nu(h) = k$ then, for every $1 \leq i \leq k-1$, we have $p_i(\overleftarrow{g}) \leq p_i(h) \leq p_i(\overrightarrow{g})$.*

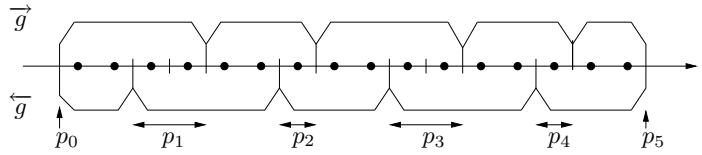
Proof. If $e(h) \leq \delta$, we have already proved (in the proof of Proposition 1.4.3) that there exists a point of $P(h)$ in $]p_i(\overrightarrow{g}), p_{i+1}(\overrightarrow{g})]$ for $0 \leq i \leq k-2$. If $\nu(h) = k$ then in each interval there is exactly one point, which has to be $p_{i+1}(h)$. This holds for $h = \overleftarrow{g}$ so that $p_i(\overrightarrow{g}) < p_{i+1}(\overleftarrow{g}) \leq p_{i+1}(\overrightarrow{g})$. By symmetry we can say that if $e(h) \leq \delta$ and $\nu(h) = k$ there is exactly one point $p_i(h)$ in $[p_i(\overleftarrow{g}), p_{i+1}(\overleftarrow{g})[$ for $1 \leq i \leq k-1$ and, for $h = \overrightarrow{g}$ we obtain $p_i(\overleftarrow{g}) \leq p_i(\overrightarrow{g}) < p_{i+1}(\overleftarrow{g})$. By combining these inequalities we reach the result that if h is a k-link solution then $p_i(\overleftarrow{g}) \leq p_i(h) \leq p_i(\overrightarrow{g})$ for every $1 \leq i \leq k-1$. ∎

From now on we will call $w_i = p_i(\overrightarrow{g}) - p_i(\overleftarrow{g}) + 1$ the number of possible values that p_i can take, using the notation $p_i^j = p_i(\overleftarrow{g}) + j - 1$, $j = 1 \ldots w_i$. Furthermore, we follow the notation of the previous section and set $l_i = p_i(\overrightarrow{g}) - p_{i-1}(\overrightarrow{g})$; thus $w_i \leq l_i$ for every value of i (see Fig. 1.6).

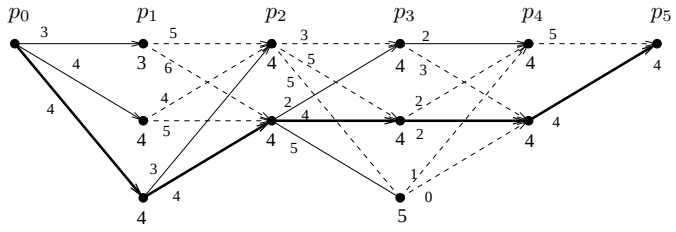
The above consideration provides a very important property of the possible partitions of the domain to obtain a minimal solution, because it reduces enormously the number of possible choices of the partition points and it gives then the possibility of efficiently finding the optimal solution. We show, in fact, that it is possible to reduce the problem of finding the optimal approximation to the problem of finding a *minimum-path* in a graph (see [26, Sec. VI] for a presentation of general graph algorithms. Also note later that our graph problem is not a minimum-path problem in the classic sense, as the metric is usually additive in such problems, but this has no practical implication). Note that every piecewise approximation can be considered as a path between the first point and the last one, passing through some nodes that are represented by the partition points. Every interval is then represented as a link between two nodes, and we can associate to every link a cost given by the l^∞ approximation error over the corresponding interval. Considering that every minimal approximation has

Example of Domain Partition

We show an example of the algorithm for finding the optimal partition set. A signal, which we suppose to have 16 samples, is first approximated as explained in paragraph 1.4.1, by scanning it from left to right and right to left, obtaining the two approximations $\overrightarrow{g}$ and $\overleftarrow{g}$. Suppose that these two partitions result as shown in fig. 1.5(a), so that $P(\overrightarrow{g}) = \{0.5, 4.5, 7.5, 11.5, 14.5, 16.5\}$ and $P(\overleftarrow{g}) = \{0.5, 2.5, 6.5, 9.5, 13.5, 16.5\}$. Thus, we have some restrictions on the possible positions of the optimal p_i, as shown in the figure. Now, we construct the trellis associated with these possible choices. So, we have three states for p_1 and p_3 (i.e. $w_1 = w_3 = 3$) and two states for p_2 and p_4 (i.e. $w_2 = w_4 = 2$). To find the optimal path in the trellis we flag every link with a weight given by the approximation error on the interval relative to that link (small numbers in Fig. 1.5(b)); then, moving from left to right, we flag the nodes with the *accumulated state metric* values (larger numbers on Fig 1.5(b)), as explained in section 1.4.2. In Fig. 1.5(b) the dashed links are those that have been discarded because of non-optimality while the solid ones are the optimal choices for their entering nodes. The bold path is overall the optimal one, as it is the only path from p_0 to p_5 that can be constructed with solid links only.



(a) Example of partitions of a domain



(b) Trellis equivalent to the partition example.

Figure 1.5: Finding the optimal partition set.

exactly one partition point in every interval $[p_i(\overleftarrow{g}), p_i(\overrightarrow{g})]$, we can represent the whole set of these approximations with a trellis graph of the type shown in fig. 1.5(b), in which nodes on the same column represent possible positions of one single partition point. In this graph we have to search the path that goes from the first point to the last one minimizing the greatest value encountered on its links. It is not difficult to note that this problem can be solved with a variant of the well known Viterbi algorithm ([98]); the only difference is that, while the Viterbi algorithm is based on an additive metric, here we have to use the maximum metric. This means that the cost of a path in the trellis is not given by the sum of the costs of the single links, but by the maximum of their values. The idea is to find the optimum path proceeding from left to right. We label every node p_i^j with an *accumulated state metric* $e(p_i^j)$, which represent the cost of the optimal path from the point p_0 to it, and we establish its antecedent $a(p_i^j)$ which is the optimal choice of the point p_{i-1} for reaching p_i^j. In order to be more precise, called $\varepsilon(p_i^j, p_{i+1}^l)$ the cost of the link connecting p_i^j to p_{i+1}^l, we give the detailed algorithm for finding the optimal path in the graph.

Algorithm 6

- *Define, for $r = 1, \ldots, w_1$ the accumulated state metrics of the p_1^r points as $e(p_1^r) = \varepsilon(p_0, p_1^r)$ and the antecedent points as $a(p_1^r) = p_0$.*
- *Given the accumulated state metrics of the points p_i^r, $r = 1, \ldots, w_i$, define, for $l = 1, \ldots, w_{i+1}$, the value $e(p_{i+1}^l)$ as*

$$e(p_{i+1}^l) = \min_r \left(\max(e(p_i^r), \varepsilon(p_i^r, p_{i+1}^l))\right), \tag{1.5}$$

  and $a(p_{i+1}^l) = p_i^{r_{min}}$, where r_{min} is the value of r that gives the minimum in (1.5). Iterate this point until $p_{k+1} = n + 1/2$ is reached.

Consider the behavior of this algorithm. In the first step the accumulated metrics for the points p_1^r are set. Then, for every possible choice p_2^l of p_2 we select the point p_1^r such that the value $\max(e(p_1^r), \varepsilon(p_1^r, p_2^l))$ is the smallest possible. So, for every p_2^l we keep only one point p_1^r and consequently one path, that is the optimal choice for reaching p_2^l. Then we define the new accumulated metric $e(p_2^l) = \max(e(p_1^r), \varepsilon(p_1^r, p_2^l))$ and we repeat iteratively the process, finding the optimal value of p_2 for every possibile choice of p_3 and so on. At the end we will reach p_k establishing the optimal choice of p_{k-1} and then, by backpropagation, the optimal path from p_0 to p_k.

Now we want to study the computational complexity of this procedure; for this purpose consider that the most expensive operations are not due to the Viterbi algorithm, but to the evaluations of the link costs $\varepsilon(p_i^r, p_{i+1}^l)$. Thus, we should reduce as much as possible these evaluations and this can be done by considering some particular relationships between the costs of the links entering or leaving the same node. In otheer words, the minimization in eq. (1.5) can be performed by considering only a subset of the p_i^r as candidates for being $a(p_{i+1}^l)$.

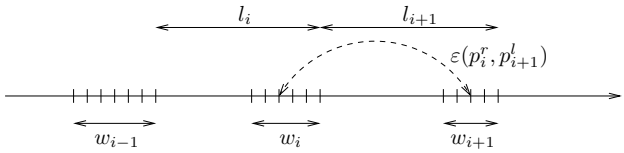


Figure 1.6: Computational complexity of the evaluation the link weights $\varepsilon(p_i^r, p_{i+1}^l)$. For every fixed p_{i+1}^l we can find the optimal p_i^r with a binary search, thus computing $\log_2(w_i)$ approximation. The number of points used in every approximation is at most $w_i + l_{i+1}$; clearly, for every i we have $w_i \leq l_i$ and thus the number of expected operation in finding the optimal p_i^r for every fixed p_{i+1}^l is less than or equal to $(l_i + l_{i+1}) \log_2 l_i$. When all the p_{i+1}^l points are checked the number of expected operation is then at most $(l_i + l_{i+1}) l_{i+1} \log_2 l_i$

Proposition 1.4.5 *For a fixed p_{i+1}^l, $e(p_{i+1}^l)$ in eq. (1.5) (and thus $a(p_{i+1}^l)$) can be found with a binary search on p_i^r, evaluating only $\log_2 w_i$ link costs $\varepsilon(p_i^r, p_{i+1}^l)$.*

Proof. First consider that $\varepsilon(p_i^r, p_{i+1}^a) \geq \varepsilon(p_i^r, p_{i+1}^b)$ if $a > b$ and this implies by induction (as it is obvious) that $e(p_i^a) \geq e(p_i^b)$ if $a > b$. Furthermore, if $a > b$, we clearly have $\varepsilon(p_i^a, p_{i+1}^l) \leq \varepsilon(p_i^b, p_{i+1}^l), \forall l = 1, \ldots, w_{i+1}$. This means that, for every fixed p_{i+1}^l, when r ranges from 1 to w_i the values $e(p_i^r)$ are nondecreasing and the values $\varepsilon(p_i^r, p_{i+1}^l)$ are nonincreasing. Thus, suppose that for a given value of r, say $r = a$, we have $e(p_i^a) > \varepsilon(p_i^a, p_{i+1}^l)$; then, in this point, clearly $\max(e(p_i^r), \varepsilon(p_i^r, p_{i+1}^l)) = e(p_i^r)$ and, in order to lower this value so as to find the minimum in eq. (1.5), we must move p_i^r from p_i^a on the left. On the contrary, if for a value of r, say $r = b$, we have $e(p_i^b) < \varepsilon(p_i^b, p_{i+1}^l)$ then we must move p_i^r on from p_i^b on the right, so as to decrease the value of $\varepsilon(p_i^r, p_{i+1}^l)$. The above argument implies that it is possible to search the optimum p_i^r with a binary search; we check first the point $p_i^{[w_i/2]}$, then $p_i^{[w_i/4]}$ or $p_i^{[3w_i/4]}$, depending on whether $e(p_i^{[w_i/2]}) > \varepsilon(p_i^{[w_i/2]}, p_{i+1}^l)$ or not, and so on, dividing by a factor of 2 the possible positions of p_i^r at every step, and thus finding the minimum in $\log_2 w_i$ steps. Note that if a value of r is found such that $e(p_i^r) = \varepsilon(p_i^r, p_{i+1}^l)$, then this r is optimal. ∎

We now give an upper bound on the number of operations needed for the execution of algorithm 6.

Proposition 1.4.6 *Algorithm 6 needs at most an expected number of $O(n^2 \log n)$ operations, being n the total number of samples.*

Proof. We refer to Fig. 1.6 as a support for the computational complexity analysis. In the whole proof we make use of the fact that for every i we have $w_i \leq l_i$. We first consider the computations of the values $e(p_1^r)$ and $e(p_k)$, and then consider all other partition points. For finding the accumulated state metrics of the points p_1^r we have to compute $\varepsilon(p_0, p_1^1), \varepsilon(p_0, p_1^2), \ldots, \varepsilon(p_0, p_1^{w_1})$, and thus w_1 approximations, each one of length less

than or equal to l_1. So, the total number of expected operations needed for the p_1^r points is at most $O(w_1 \cdot l_1) \leq O(l_1^2)$. For finding $e(p_k)$, instead, from Proposition 1.4.5, we have to compute $\log_2 w_{k-1}$ approximations, each one of length less than or equal to $l_k + w_{k-1}$. So, for p_k, the expected number of operations is at most $O((w_{k-1} + l_k) \cdot \log_2 w_{k-1}) \leq O((l_{k-1} + l_k) \cdot \log_2 l_{k-1})$. For every remaining point[7] p_i, $1 < i < k$, we have to compute $w_i \log_2 w_{i-1}$ approximations of length less than or equal to $w_{i-1} + l_i$, and the expected number of operations is bounded by $O((l_{i-1} + l_i) \cdot l_i \cdot \log_2 l_{i-1})$. Thus the total number of expected operations is at most

$$O\left(l_1^2 + \sum_{i=2}^{k-1} ((l_{i-1} + l_i) l_i \log_2 l_{i-1}) + (l_{k-1} + l_k) \log_2 l_{k-1}\right) \tag{1.6}$$

and, considered that $l_i \leq n$ for all i (and thus that we can bound the logarithms with $\log_2 n$), we have at most

$$O\left(n \log_2 n \left(l_1 + \sum_{i=2}^{k-1} (l_{i-1} + l_i) + (l_{k-1} + l_k)\right)\right) \tag{1.7}$$

operations. Now, as $\sum_i l_i = n$, this quantity is at most $O(2n^2 \log_2 n) = O(n^2 \log n)$. ∎

It is possible to show that this bound cannot be further lowered, because we can construct an example of partition for which the algorithm has complexity of exactly $O(n^2 \log_2 n)$ expected operations. Neverthless, this estimation can be very pessimistic in most practical situations. In many applications, in fact, we can suppose that the maximum length $\max_i(l_i)$ of the approximation intervals is asymptotically bounded[8] with respect to the number of points n. In this case we have the following:

Proposition 1.4.7 *If $\max_i l_i$ is bounded with respect to the number of points n, then algorithm 6 needs an expected number of $O(n)$ operations.*

Proof. Suppose $l_i \leq L, \forall i$, independently of the value of n. Then equation (1.6) can be bounded by

$$O\left(L^2 + \sum_{i=2}^{k-1} 2L^2 \log_2 L + 2L \log_2 L\right) \leq O(kM) = O(k), \tag{1.8}$$

where $M = 2L^2 \log_2 L$. But, clearly, k satisfies $n/L \leq k \leq n$, and the complexity is thus $O(n)$. ∎

[7] i.e. for evaluating the values $e(p_i^j), j = 1, \ldots, w_i$ for a fixed $1 < i < k$.

[8] This assumption is completely natural when the variation of the number of samples n is due to the time windowing of a given signal. Consider for example an audio signal: unless we are talking of silence, it makes sense to suppose that the maximum interval length does not depend on the number of samples we are studying (if we are using a predefined constant sampling frequency).

Table 1.1: Minimum number of intervals and optimal error value obtained for various error threshold δ, when approximating the signal dotted in fig. 1.7(a).

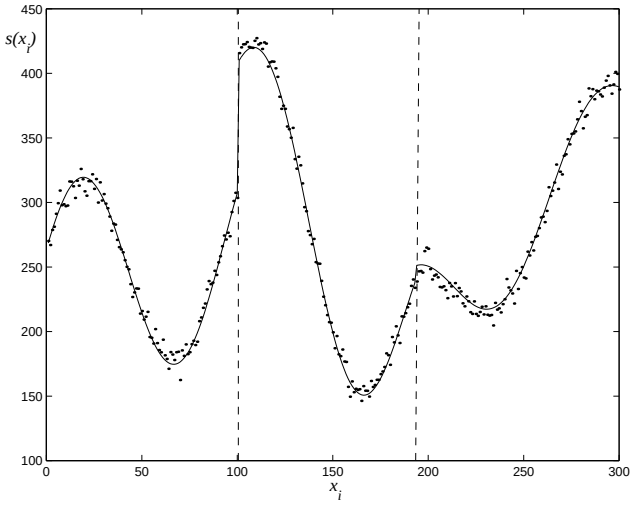
δ	k	opt. er.	δ	k	opt. er.
120	1	101.94	11.55	8	11.29
101.93	2	52.72	11.28	9	10.68
52.71	3	13.91	10.67	10	10.60
13.90	4	13.74	10.59	11	10.42
13.73	5	13.05	10.41	12	10.04
13.04	6	11.58	10.03	13	9.93
11.57	7	11.56	9.92	14	9.43

Now that we have estimated the complexity of the method with respect to the number of points, it is important to clarify that the execution time is very much influenced by the selected error threshold. Even if at a first glance this seems counterintuitive, we have to consider that the dimension of the obtained trellis depend on the error bound δ. Suppose, in fact, that the signal is such that it can be approximated with a number k of intervals if and only if the error threshold δ is in the interval $[\delta_1, \delta_2[$. Then, clearly, the obtained optimal solution has an error equal to δ_1, and does not depend on δ if it is in the considered interval. On the contrary, the two minimal approximations $\overrightarrow{g}$ and $\overleftarrow{g}$ depend on δ and, in particular, the larger the value of δ the more different their partition sets P. As a consequence, the constructed trellis varies from a trivial one for the value $\delta = \delta_1$ to a maximum dimension when δ approaches δ_2. This fact is better shown with an example; in Fig. 1.7(a) we can see the optimal approximation (solid line) obtained when approximating the given signal (dotted one) with an error threshold in the interval $[14, 52]$, using as basis functions[9] $B = \{1, x, \sin(\pi x/50), \cos(\pi x/50)\}$. As we can see from Table 1.1, this interval of values for δ leads to an optimal approximation that uses 3 intervals, (and has a max error of 13.91). In Fig. 1.7(b) we can see how the number of operations used by the algorithm increases significantly with δ, as explained.

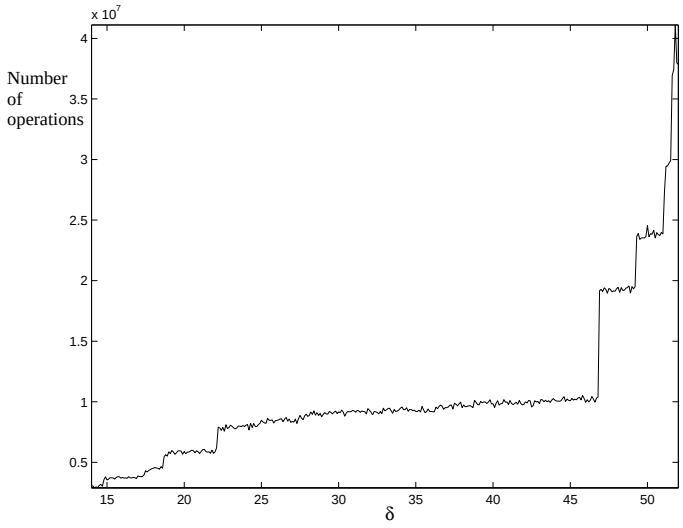
1.4.3 Representation by irregular samples and coding

In the first section of this chapter we have recalled that every l^∞ (1-link) signal approximation problem in an m-dimensional linear space can be reduced to a linear program. Thus, it is not difficult to show that, the solution of the problem leads to the identification of a (not necessarily unique) set of $m + 1$ samples that uniquely specify the optimal approximation. This means that there exists a set of $m + 1$ samples (out of the, say, n) such that the optimal approximation is only due to them, and removing all the other $n - m - 1$ samples does not change the optimality of this approximation. We call these points *pivot points*, coherently

[9] Note that it is often computationally useful, in practice, to use shift-invariant basis.



(a) Signal approximation with $\delta \in [14, 52]$.



(b) Number of executed operations when approximating the signal with respect to the value of $\delta \in [14, 52]$.

Figure 1.7: Example of an optimal l^∞ signal approximation by means of mixed piecewise linear and cosinusoidal expansions. Given the dotted signal shown in Fig 1.7(a), we have computed the optimal approximation obtained choosing a value of δ in the interval $[14, 52]$. As we can see in Fig. 1.7(b), the number of operations increases with δ. In particular, it is interesting to see the increase in the value of the number of multiplications for some specific values of δ. For example, when δ reaches values near 47, we can see that the number of multiplications notably increases. This is due to the fact that when δ changes from 46.7 to 46.8 the point $p_1(\overleftarrow{g})$ goes from 100.5 to 94.5, thus with a consequent increase in the value of w_1 and then in the dimension of the trellis.

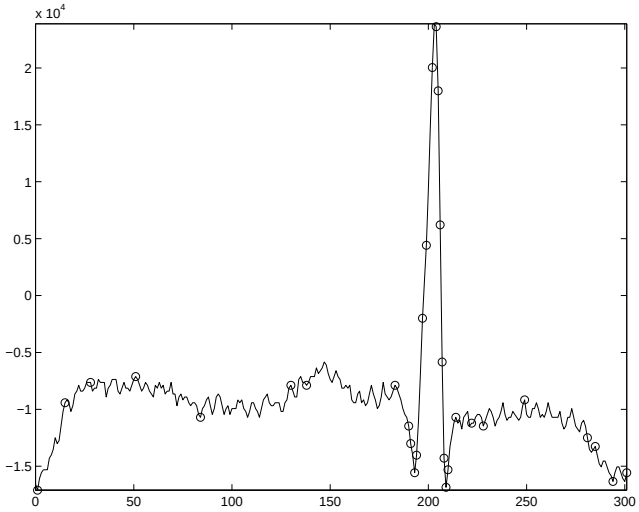


Figure 1.8: Example of signal subsampling by means of l^∞ piecewise approximations. The circles represent the pivots point of the optimal segmentation subdomains. Here we have used the same electrocardiogram signal of Fig. 1.10, with the optimal approximation of Fig. 1.10(d) .

with the fact that they indeed corresponds with the the previously defined *pivot points* for the case of straight line approximations.

Thus, l^∞ approximation can also be seen as a tool for irregular signal subsampling, in the sense that it automatically gives a subset of samples that bring the behavior of the whole signal (with a confidence related to the m and δ values). In the case of piecewise approximations, moreover, the study we have performed leads to the determination of a minimal number of points and a domain partition that optimally describe the whole signal. In Figure 1.8 we show an example of the result of this subsampling procedure when applied to the electrocardiogram signal of Fig. 1.10 of the next section. It is clear however, that the reconstruction of the approximation beyond the pivot points by using only these samples can be performed if the partition is known.

These consideration is also important from the point of view of the problem of encoding the obtained approximations. Consider in fact the problem of encoding the piecewise approximation obtained for a given signal. Then, we first need to encode the partition of the domain, and then we have to encode the shape of the approximation within each subdomain. Note that the encoding of the partition is not complicated, as it can be viewed as the encoding of a sequence of integers. It is instead more interesting to focus on the encoding of the linear approximations within the domains. As the optimal approximation in an m-dimensional space is identified by the m coefficients, the problem of coding the

approximation can be viewed as the problem of coding the m coefficients obtained as the solution of the linear programming problem associated to the approximation problem. It is important however to note that these coefficients are usually real values[10], and thus the encoding of these coefficients requires a quantization. In general, it is not easy to control the additional approximation error introduced in any point when applying a quantization to the coefficients of the optimal approximation, and it is not easy, in particular, to determine (and encode!) the number of required bits for every coefficients, or the optimal rate allocation for these coefficients. So, the encoding of the coefficients as a technique for the encoding of the approximation is not an easy task, and it requires sophisticated techniques if we are interested in keeping the error below a given threshold around the optimal one.

In this context, it is of particular interest to consider again the pivot points. Rather than identifying the approximation by means of the m coefficients, we can also identify it with the $m+1$ pivot points. Note that in practical situations, the pivot points, being samples of the signal, are usually already available in a quantized form. So, if we encode the solution by coding the pivot points, we do not incur any quantization problem, because data are typically already quantized. Of course, for every pivot point we need to encode both the x coordinate and the associated value of the signal. This may lead to poor coding performance in some cases, but it is however important to consider that differential encoding can be used, which at least reduces, from one pivot point to the other, the number of bits required to represent the x coordinate.

A particular special case that should be considered with some care is the case of piecewise linear approximations, which is studied in detail in the next section from the algorithmic point of view. For the case of straight line approximations, it is worth noticing that another possible technique of encoding can be used, which reveals to be of particular efficiency. In order to encode a straight line in an interval, indeed, we can encode the value that the line takes at the extremal positions of the domain. Even in this case, of course, we have to consider the quantization problem. It is not difficult to show, however, that the induced error within every point of the domain is a convex combination of the quantization error introduced at the extremal points of the domain. So, by quantizing the values at the extremal points, we have a precise bound on the error added to every internal point. For this particular case, it is interesting to go a little further in the analysis of the quantization error. Suppose the signal s is available in a quantized form, and suppose without loss of generality that the quantization step is 1. Consider the problem when an approximation $\tilde{s}$ of s with error threshold δ has to be encoded. Suppose an approximation f of s is constructed so that $\|s - f\|_\infty < \delta$. Now let $\tilde{f}$ be the quantized approximation, obtained by quantizing the values of f at the extremal points of a domain. In this case, of course the quantization error at the extremal points if at most 1/2 in absolute value and thus, as the induced error in every point is a convex combination of quantization error at the extremal points, we have that $\|f - \tilde{f}\|_\infty \leq 1/2$. Now consider the reconstruction operation. For every point of the domain, an approximation $\tilde{s}$ of the signal is obtained by quantizing the corresponding value

[10]Unless we restate the approximation problem in an integer-programming setting, which involves more complicated techniques.

of $\tilde{f}$, and again we obtain that $\|\tilde{s} - \tilde{f}\|_\infty \leq 1/2$. So, in conclusion we have

$$\|s - \tilde{s}\|_\infty \leq \|\tilde{s} - f\|_\infty + \|f - \tilde{f}\|_\infty + \|\tilde{f} - \tilde{s}\|_\infty < \delta + 1 \tag{1.9}$$

But of course the difference between the quantized signals s and $\tilde{s}$ is an integer in every point, so that $\|s - \tilde{s}\|_\infty \leq \delta$. This means that with the given strategy, it is only necessary to approximate the signal with error strictly smaller than δ in order to recover in the decoding phase an approximation with error not exceeding δ. Furthermore, for this particular case, the encoding of the values on the extremal points of the domains can be performed using differential encoding. Consider for example the optimal approximation shown in Figure 1.10(d) below. It is clear that there is considerable correlation between the value of the approximating line at the end point of one subdomain with the value of the approximating line in the starting point of the following subdomain. Thus, differential encoding can be used for these extremal values, with some conspicuous saving in the rate spent.

1.5 Piecewise linear approximations

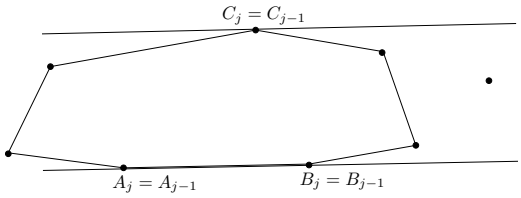
In the preceeding section we have described an algorithm for finding the optimal piecewise approximation when working in general piecewise linear spaces G_T; in that case we considered that the approximation over every interval could be obtained by using the linear programming approach and thus with average time proportional to the number of samples. It is clear that if we are interested in piecewise linear approximations, the geometric method exposed in section 1.3.1 must be preferred, as it gives much better performance.

Anyway, we show here that with a slight different version of the method exposed in section 1.3.1 it is possible to improve the construction algorithms for the minimal and optimal solutions. We have in fact the following result

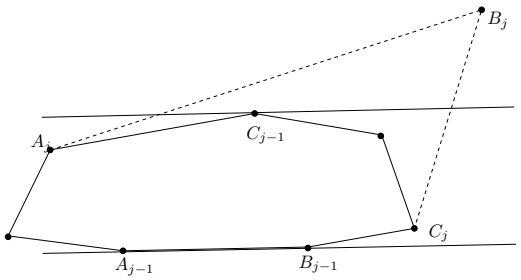
Proposition 1.5.1 *Given n sample of a signal and an $l \leq n$, it is possible to find the optimal straight line approximations of the sets of points $S_j = \{s_i\}_{i=1\ldots j}$ with $j \leq l$ in at most $O(l)$ operations.*

Proof. The proof is constructive, in the sense that we show how to find the approximations of the sets S_j, $j = 1 \ldots l$, in $O(l)$. Considering the approximation of S_l with the geometrical method of section 1.3.1, we note that the convex-hull is constructed in a progressive way, i.e. by adding points from left to right and updating the polygon at every step. This means that before finding the convex-hull of S_l we have found the convex-hull of every S_j with $j < l$. Now, it is possible to see that the pivot points of S_j can be obtained from those of S_{j-1} in a number of operation O_j such that $\sum_j O_j = O(l)$.

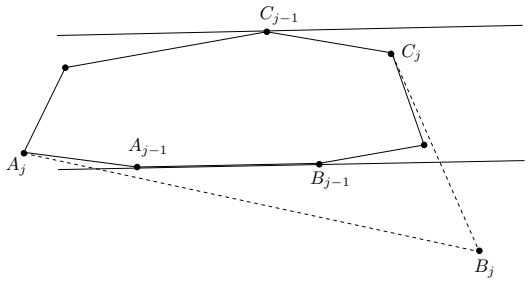
Consider the convex-hull Q_{j-1} with pivot vertices A_{j-1}, B_{j-1} and C_{j-1} and, for a generic point P of a convex-hull Q, let $x(P)$ be the index m such that $P = s_m$. Consider now the new entering point s_j (see Figure 1.9 for a graphical representation). We can distinguish three cases:



(a) Case 1



(b) Case 2



(c) Case 3

Figure 1.9: Finding the pivot points A_j, B_j and C_j of Q_j after the insertion of the new point s_j. We can distinct three cases, every one being solvable with a number of operations that is at most $O(x(C_j) - x(C_{j-2}))$.

1. in the first case s_j lies in the strip of plane delimited by the lines passing through the pivot vertices of Q_{j-1}. In this case the pivot points do not change and thus $O_j = 0$.

2. In the second case s_j is outside the strip from the side determined by C_{j-1}. In this case s_j is the pivot B_j, A_j is the consecutive of s_j in Q_j and C_j has to be searched to the right of B_{j-1} in $O(x(C_j) - x(B_{j-1}))$ operations.

3. In the third case s_j is outside the strip from the side determined by A_{j-1} and B_{j-1}. In this case s_j is still the pivot B_j, A_j is the vertex that precedes s_j in Q_j and C_j has to be searched at the right of C_{j-1} in $O(x(C_j) - x(C_{j-1}))$ operations.

Considering that $x(C_{j-1}) < x(B_{j-1})$, we can see that the worst case is the third. Thus, in the worst case we need a number of operations that is $\sum_j O(x(C_j) - x(C_{j-1}))$ and, using the telescopic property, this is $O(x(C_l)) = O(l)$ operations. ∎

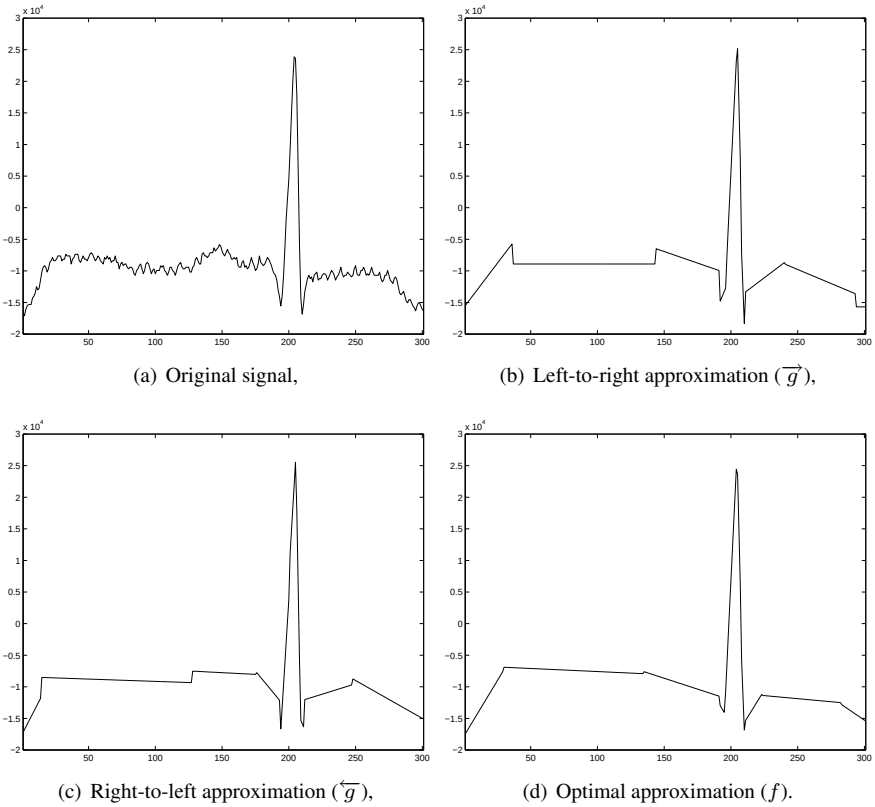
As we have said, this fact is very useful in the study of piecewise linear approximations. In particular, we have the following result.

Proposition 1.5.2 *For the case of piecewise linear approximations, Algorithms 5 and 6 require at most $O(n)$ and $O(n^2)$ expected operations respectively.*

Proof. The result is essentially a consequence of the fact that, from Proposition 1.5.1, it is possible to avoid in Algorithms 5 (or better in Algorithm 4 used in Algorithm 5) and 6 the binary searches. In details, consider the construction of $\overrightarrow{g}$ (the same holds for $\overleftarrow{g}$). Using the progressive approximation construction explained above, we can scan the signal by adding a point at every step, starting a new interval every time the error exceeds the given threshold δ. This way the number of operation for every interval $[p_{i-1}(\overrightarrow{g}), p_i(\overrightarrow{g})]$ is $O(l_i)$ and thus the total number of operation is $O(n)$.

In the same way, consider the evaluation of the generic $e(p_{i+1}^l)$ in the second step of Algorithm 6 (and refer to Fig. 1.6 for a graphical support). Instead of searching the minimum of $\max(e(p_i^r), \varepsilon(p_i^r, p_{i+1}^l))$ over r with a binary search (thus computing $\log_2 w_i$ approximations) we can find all the values $\varepsilon(p_i^r, p_{i+1}^l)$, $r = 1, \ldots, w_i$, in $O(w_i + l_{i+1})$ operations. In fact, we can first compute the approximation on $[p_i^{w_i}, p_{i+1}^l]$ and then add the points $p_i^{w_i-1}, p_i^{w_i-2}, \ldots, p_i^1$ on the left one by one, updating the convex-hull and the optimal solution as explained above, for a total number of operation of $O(w_i + l_{i+1})$. This leads to a gain of a factor $\log w_i$ for every p_{i+1}^l and thus to a gain of $\log n$ in the complexity of the complete algorithm. ∎

As an example of approximations by means of piecewise straight line functions, we show in Fig. 1.10 the results when applying the algorithm to an electrocardiogram signal. The signal samples are 16-bit signed integers (values from -32768 a 32767) and we have set an error threshold $\delta = 2000$. The algorithm has found a minimum necessary number of 10 segments; in Table 1.2 we can see the values of the partition points for the minimal approximations $\overrightarrow{g}$ and $\overleftarrow{g}$ and for the optimal one f, together with the relative approximation



(a) Original signal, (b) Left-to-right approximation ($\overrightarrow{g}$),

(c) Right-to-left approximation ($\overleftarrow{g}$), (d) Optimal approximation (f).

Figure 1.10: Example of approximation of an electrochardiogram signal by means of piecewise straight line approximations.

errors and the associated computational complexity. Fig. 1.10 provides the original signal and its approximations.

1.A Geometric properties of convex hulls

In this section we prove the statements given in section 1.3.1 on the geometrical properties of the convex hull Q of a set of points S. For a better readability we restate lemmas 1 to 4 of section 1.3.1, as they are also necessary for the proof of Proposition 1.3.1. We recall the used nomenclature. Given a set $S = \{s_i\}$ of n points in the plane, we call Q its convex-hull. If k is the number of sides of Q, we call p_i, $i = 1 \dots k$, the vertices of Q in counterclockwise order, with p_1 the left-most one. For clarity, we add a point $p_{k+1} = p_1$ and set m, $m \leq k$, be the integer such that p_m is the right-most vertex. For $i = 1 \dots k$, we call l_i the side $\overline{p_i p_{i+1}}$

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	e
$\overrightarrow{g}$	36.5	143.5	191.5	196.5	204.5	207.5	210.5	240.5	293.5	1984
$\overleftarrow{g}$	15.5	128.5	176.5	194.5	201.5	206.5	209.5	212.5	248.5	1941
f	29.5	134.5	191.5	195.5	204.5	207.5	210.5	223.5	282.5	1631

Table 1.2: Partition points, errors and number of multiplications for the approximations $\overrightarrow{g}$, $\overleftarrow{g}$ and f, when setting $\delta = 2000$ in the approximation of the signal plotted in fig. 1.10(a). The number of operations used for the three approximations is respectively about $6 \cdot 10^3$, $8 \cdot 10^3$ and 10^5

and $v(l_i)$ the opposite vertex to the side l_i, i.e. the most distant vertex of Q from l_i in the direction orthogonal to l_i (distances between vertices and sides will always be considered in this sense in what follow). We say that $v(l_i)$ is x-internal to l_i if the vertical line through $v(l_i)$ cuts l_i.

Lemma 1.A.1 *Every side of the lower-hull has opposite vertex in the upper-hull and* viceversa. ∎

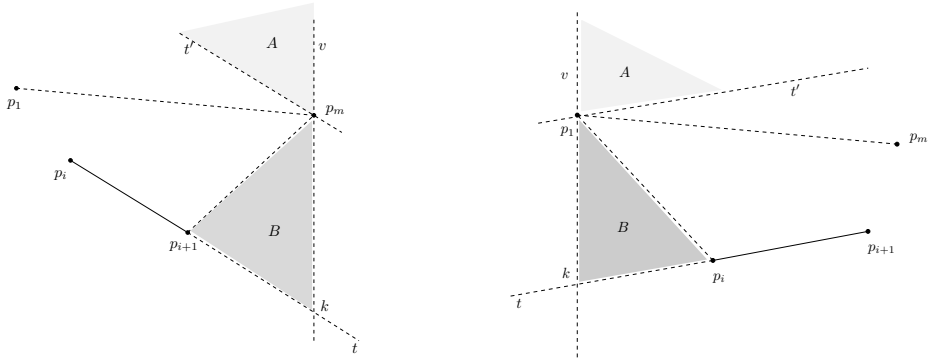


Figure 1.11: Possible region for the opposite vertex of a lower-hull side

Proof. This fact is somehow obvious. Anyway, consider a side l_i of the lower hull, and suppose p_m is more distant from l_i than p_1. This situation is shown in fig. 1.11(a) where t' is the line parallel to line t to which l_i belongs. By the definition of m, $v(l_i)$ must lie at the left of p_m and, by the definition of opposite vertex, $v(l_i)$ must lie above line t'. Thus, it is easy to see that $v(l_i)$ must lie in the portion of plane indicated with A. Any point p_j, $i < j < m$, of the lower hull must instead lie in the B area. Thus the point $v(l_i)$ belongs to

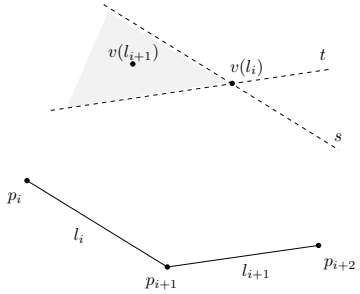


Figure 1.12: Relation between the opposite vertices of two consecutive sides.

the upper-hull[11]. If p_1 is more distant from l_i than p_m we obtain the equivalent symmetric situation shown in Fig. 1.11(b) which leads to the same conclusion. For the upper-hull sides we can operate symmetrically with a vertical flip and thus prove the converse. ∎

Lemma 1.A.2 *If we move from one side of the polygon to its consecutive in counterclockwise (ccw) direction, the respective opposite vertex, if it changes, moves in ccw direction too.* ∎

Proof. We consider the generic sides l_i and l_{i+1} with their opposite vertices as shown in Fig. 1.12; lines s and t are parallel to l_i and l_{i+1} respectively. It is clear that $v(l_{i+1})$ cannot be farther than $v(l_i)$ from l_i and must be at least as distant as $v(l_i)$ from l_{i+1}. Thus $v(l_{i+1})$ must lie in the shaded portion of plane between s and t, and thus it is positioned in ccw direction with respect to $v(l_i)$. Note that this do not mean that $v(l_{i+1})$ is the consecutive vertex of $v(l_i)$ (see Lemma 4). ∎

Lemma 1.A.3 *A vertex p_j, $1 < j \leq k$, is the opposite vertex of a side l_i, i.e. $p_j = v(l_i)$, if it is more distant from l_i than vertices p_{j-1} and p_{j+1}.* ∎

Proof. This follows directly by the convexity of the convex-hull. If p_j is more distant than p_{j-1} and p_{j+1} from l_i and if there were a vertex p_r more distant than p_j, than the segment $\overline{p_j p_r}$ would not be inside the convex-hull, which is absurd. ∎

[11] Again remember that we consider p_1 and p_m to pertain both to upper- and lower-hull.

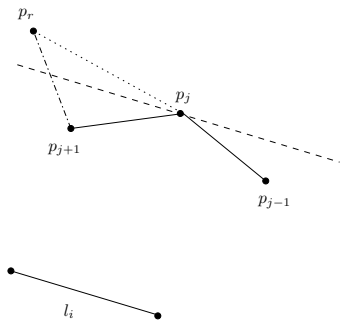


Figure 1.13: The opposite vertex can be recognized considering only its neighbors.

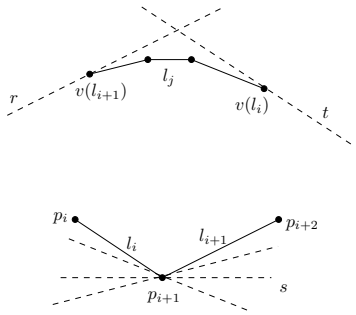


Figure 1.14: A reciprocity property between sides and opposite vertices.

Lemma 1.A.4 *Given two consecutive sides l_i and l_{i+1}, their common vertex p_{i+1} is opposite vertex of every side between $v(l_i)$ and $v(l_{i+1})$ (in the path not containing l_i and l_{i+1}, obviously).* ∎

Proof. Consider Fig. 1.14, where the position of the opposite vertices is justified and imposed by Lemma 2. Lines t and r are parallel to l_i and l_{i+1} respectively. From the fact that $v(l_i)$ and $v(l_{i+1})$ are opposite vertices of l_i and l_{i+1} and from the convexity of the convex-hull, we can see that every side between $v(l_i)$ and $v(l_{i+1})$ has a slope which is "intermediate" between the slopes of t and r. So, the generic side l_j between $v(l_i)$ and $v(l_{i+1})$ has a slope which is intermediate between those of l_i and l_{i+1}; this means that its parallel s through p_{i+1} leaves p_i and p_{i+1} in the same halfplane and thus p_{i+1} is more distant than p_i and p_{i+2} from l_j. From Lemma 3 this implies that p_{i+1} is opposite vertex of l_j. ∎

Proof of Proposition 1.3.1

We start by demonstrating that there exists at least one side l whose opposite vertex $v(l)$ is x-internal to it. Suppose that every side l_i has its opposite vertex $v(l_i)$ which is not x-internal; then, clearly, $v(l_1)$ must be on the right of l_1 and $v(l_{m-1})$ must be on the left of l_{m-1}. So, there must exist an integer $j < m$ such that $v(l_{j-1})$ is on the right of l_{j-1} and $v(l_j)$ is on the left of l_j. Then, from Lemma 4, p_j is the opposite vertex to every side between $v(l_{j-1})$ and $v(l_j)$; the vertical line through p_j must cut one of these sides and so there exists a side whose opposite vertex, p_j, is x-internal to it, so that the initial hypothesis was inconsistent.

Now, suppose we have three points A, B and C of Q such that C is the x-internal opposite vertex to the side $\overline{AB}$. For these three points the optimal linear approximation is easily proved to be the line r parallel to $\overline{AB}$ and equidistant from $\overline{AB}$ and C. The error produced by this line in approximating s at every x coordinate x_i is proportional to the distance of the point $s_i = (x_i, s(x_i))$ from the line; the way r has been selected[12] ensures that A, B and C are the points of S mostly distant from r and so, the l^∞ approximation error of r is due to A, B and C. But for these three points r is optimal and so it is for the whole set S, as A, B, C are peculiar vertices of the convex hull.

Finally we show that there cannot exist another triplet of points A', B' and C' such that C' is x-internal to the side $\overline{A'B'}$. Supposing these three points exist, they should lead to an optimal solution r'. Calling $e(t; q_1, q_2, q_3)$ the error produced by the line t over the points q_1, q_2 and q_3 we should have

$$e(r'; A', B', C') \geq e(r'; A, B, C) \geq e(r; A, B, C) \tag{1.10}$$

since r' reaches its maximum error on A', B' and C', and r is optimum for A, B and C. But symmetrically we have

$$e(r, A, B, C) \geq e(r; A', B', C') \geq e(r'; A', B', C'). \tag{1.11}$$

So the only possibility is that all these $\geq$ must be replaced by $=$ and, consequently, $r = r'$, which means that $\overline{AB}$ is parallel to $\overline{A'B'}$, contrarily to the initial hypothesis that Q has no parallel sides. This argument also proves that if Q has parallel sides[13] the optimal solution is still unique, even if this is not true for the pivot points.

[12] Consider that all the points of S lie in the strip of plane between the the line t passing through A and B and its parallel t' passing through C. The line r is exactly in the medium of this strip and the most distant points are the ones lying on t and t'.

[13] In this case it is necessary to adjust some technical details such as the definition of *opposite vertex*, but the main arguments and their consequences still hold.

Chapter 2

Distributed Source Coding

2.1 Introduction

In the last years an increasing attention has been paid to novel type of encoding techniques that are of interest in the new emerging scenarios of multiuser communication. With the advent of modern technology it is becoming more and more frequent to see different users interested for example in accessing the same information or to have the same user have access to an information source through different channels. This type of problems are studied under the name of *network communication* and the theoretical investigation on channel capacities and rate distortion bounds for this type of scenarios are usually called *multiuser information theory* or *network information theory* [40, 28].

One interesting topic revitalized recently in this field is *Distributed Source Coding* (DSC). In its first and basic version, DSC is the study of the independent encoding of two correlated sources that are to be transmitted to a common receiver. This problem was first studied in a paper by Slepian and Wolf [89] in 1973; their famous result, together with the results obtained in a successive paper by Wyner and Ziv [109], yield the development of DSC as a whole branch of information theory. In this chapter we will introduce the basic knowledge on DSC giving an overview of the underlying ideas and of the most important theoretical results obtained in this field.

2.2 An example

Before discussing about DSC from the information theory point of view, it is interesting to present the idea that is behind the theorems with the use of some simple examples. An easy example of what we want to study is the following. Suppose a transmitter A wants to communicate a number X to a receiver B. Suppose both transmitter and receiver know that the number X is chosen at random uniformly in the range [0,999]. This means that A has to communicate to B three decimal digits in order to send the value of X. Now, suppose the

receiver already has an idea of the value of X in the sense that it knows a value Y with the guarantee that $X - 9 \leq Y \leq X$. If Y was also available to the transmitter A, then it could use this value to simplify the encoding of X. In fact, instead of sending the three required digits for the encoding of X it could simply send one digit, i.e. the value of $X - Y$. This type of encoding method is a predictive one, in the sense that the value of Y is used as a prediction of X, and only the difference is encoded.

Now, suppose Y is not known to A. Then clearly it is not possible for A to send only the one digit of the value of $X - Y$. Anyway, it is interesting to note that A does not need to send all the three digits of X, as the last digit is sufficient. In fact, the decoder, knowing the value of Y, can recover the value of X from its last digit. For example, suppose $Y = 236$ and suppose the last digit of X is 3. Then, clearly, as $Y \leq X \leq Y + 9$, we easily deduce that $X = 243$. With this simple example, we have introduced what is known as the problem of source coding with *side information* at the decoder. Here the side information is Y, the variable which is useful for the encoding of X, the latter being instead the variable we are really interested in. Additionally, with this example we have shown that the side information, even if it is only known to the decoder, allows to reduce the amount of information that the encoder has to transmit. A rigorous and quantitative analysis of this fact is contained in the Slepian-Wolf and Wyner-Ziv theorems that will be presented in the next sections.

Consider now a slightly different setting. Suppose both A and B have to transmit their information to a third user, say C. So, A has to communicate X to C while B has to communicate Y, with the additional constraint that A and B cannot share their information X and Y between them. Then, if A and B act as if they were alone, both of them will need to send three digits to C. If instead they consider the real situation, they can use the above presented method for a more efficient communication: B sends the value of Y using three digits, while A sends only the last digit of X. This way, C can correctly recover both X and Y. With this scheme, the total number of sent digits is four, three from B and one from A. It is clear that the reversed scheme could also be used, with A sending three digits and B only one. The interesting fact is that it is also possible to balance the work and send two digits from both A and B and still allow C to recover both X and Y. This can be done in the following way: B sends the first and the third digits of Y, while A sends the second and the third digits of X. With our example, where $X = 243$ and $Y = 236$, the information sent by B is '2?6' and the information sent by A is '?43'. It is not difficult to realize that for the receiver C this information, together with the constraint $X - 9 \leq Y \leq X$, is sufficient to recover that $X = 243$ and $Y = 236$. So, again, we see in a simple example many interesting things. First of all, A and B are sending their information to C using less digits than what they would need if they were alone, even if they cannot share their information. Additionally, we see that the way the amount of information can be split between the different users is not uniquely determined, as different configurations are possible (we have seen in this example at least 3 different choices). This is another important result which is treated in a rigorous way in the Slepian-Wolf theorem.

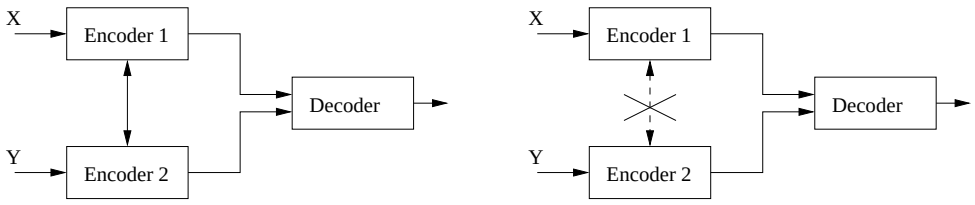
2.3 Information theory

2.3.1 Problem setting and basic results

In this section we are interested in providing an overview of the DSC problem from the information theoretic point of view, thus introducing the problem in a formal way and then presenting the Slepian-Wolf and Wyner-Ziv results.

The first general situation we want to to consider is the situation where two correlated sources X and Y are to be encoded and transmitted to a single receiver. For the sake of simplicity we will consider here the case of discrete memoryless sources with a finite alphabet, and we will specify when necessary the different hypotheses assumed in the stated results.

Consider a situation as shown in Figure 2.1(a), and consider the problem of encoding X and Y so as to send their values to the decoder. What we want to study is the amount of rate that is required in order to have a lossless transmission, i.e., where the decoder can recover without distortion the values of X and Y. Note that in this scheme the two encoders are allowed to communicate between each other, and the hypothesis is that there is no limitation in the amount of information they can share. So, in this case we can consider that both Encoder 1 and 2 know the values of both X and Y. This means that from an information theoretic point of view we have a situation where the two sources are to be encoded jointly and sent to the decoder, the two encoders actually operating as one single encoder. In this case, if we do not consider the channel communication problems, it does not make sense to consider the rates spent individually by each encoder; it is very well known, instead, that the minimum total rate that has to be spent in order to have a lossless encoding of X and Y is their joint entropy $H(X, Y)$.



(a) Joint encoding (b) Distributed encoding

Figure 2.1: Two different scenarios for a two-source problem.

Consider now the problem of encoding X and Y when the situation is as depicted in Figure 2.1(b). In this case the two encoders cannot communicate each other and they have to separately encode X and Y and send their codes to the common decoder. The problem turns out to be what the admissible rates for lossless communication in this case. It is clear that Encoders 1 and 2 can send X and Y using respectively a rate of $H(X)$ and $H(Y)$ bits. The total amount of rate is $H(X) + H(Y)$ which is greater than $H(X, Y)$ under the

hypothesis that X and Y are correlated. In this case, anyway, the decoder would receive part of information in a redundant way. Suppose that the decoder decodes first the value of Y; then, the value of X, being correlated with Y, is already "partially known" and the complete description received by Encoder 1 would be somehow redundant. Considering the example we have presented in Section 2.2 we may argue that it is possible in some way to reduce the rate for X, or even to reduce both the rates for X and Y in some flexible way. The surprising result obtained by Sepian and Wolf [89] is that not only the rate for X and Y can be actually smaller than $H(X)$ and $H(Y)$, but that there is no penalty in this case with respect to the case of Figure 2.1(a) in terms of total required rate. The only additional constraint in this case is that there is a minimum amount of rate $H(X|Y)$ to be spent for X and a minimum amount of rate $H(Y|X)$ for Y, which represent the intuitive idea that every encoder must send at least the amount of information of its own source that is not contained in the other source. In particular, Slepian and Wolf gave the following theorem for the case of memoryless sources.

Theorem 2.3.1 (Slepian-Wolf, 1973, [89]) *Let two sources X and Y be such that (X_1, Y_1), $(X_2, Y_2), \ldots$ are independent drawings of a pair of correlated random variables (X, Y). Then it is possible to independently encode the source X and the source Y at rates R_X and R_Y respectively, so that a common receiver will recover X and Y with arbitrarily small probability of error, if and only if*

$$R_X \geq H(X|Y) \tag{2.1}$$

$$R_Y \geq H(Y|X) \tag{2.2}$$

$$R_X + R_Y \geq H(X,Y) \tag{2.3}$$

∎

The above theorem holds for memoryless sources as considered in the Slepian' and Wolf's paper. Few years later Cover [27] extended the theorem to the more general case of multiple stationary ergodic sources, giving a simple proof based on the asymptotic equipartition property, i.e. the Shannon-McMillan-Breiman theorem [88, 67, 20]. In the case of two sources the theorem is obviously reformulated by substituting entropies with entropy rates in equations (2.1)-(2.3).

The set of all (R_X, R_Y) pairs satisfying equations (2.1)-(2.3) form the so-called *achievable region* which is shown in Figure 2.2. It is important here to clarify that, as stated in the theorem, we are considering rate pairs (R_X, R_Y) such that the decoder will recover X and Y without loss with arbitrarily small probability of error. When we say this, we mean that the encoding is considered to operate on blocks of n symbols, and that for sufficiently large n the probability of having an error in the decoding phase can be made as small as we want. Here it is worth noticing that in this sense there is a penalty in the case of distributed encoding with respect the case of joint encoding. In the latter case, in fact, by using variable length codes it is possible to encode the two sources X and Y to a total rate as close as

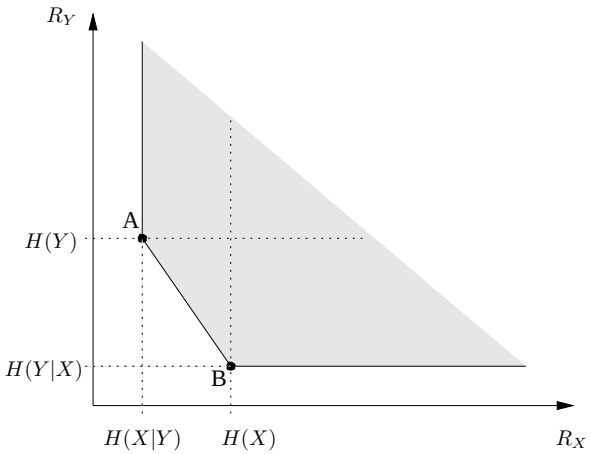


Figure 2.2: Slepian-Wolf region.

desired to the joint entropy $H(X, Y)$ even with a probability of decoding error exactly zero. This case, when the probability of error must be exactly zero, is usually referred to as *zero-error coding*. While for the joint encoding of the sources the zero error region is the same as the achievable region for vanishing error probability, for the distributed coding these two regions are in general different. It is interesting to note that in a remark in [27] the author asserts that the rate region identified by the Slepian-Wolf theorem is also achievable for the zero error coding. It was in the following year that Witsenhausen [104] recognized the different nature of the zero-error problem, and the connection with graph theory, opening the road to further research on zero-error distributed coding [7, 43, 3, 58, 59].

From the example presented in the previous section, it is surely already clear that the two points labeled with A and B in Figure 2.2 are of particular interest. Consider for example point A, which corresponds to the case when $R_Y = H(Y)$ and $R_X = H(X|Y)$. Here the two source X and Y are encoded in a completely different way; the source Y is encoded at a rate equal to its own entropy, and thus it is encoded in a traditional way, while the source X is encoded in a distributed fashion, given that Y is completely available at the decoder. In this case, from the DSC point of view, we can focus on the encoding of X only, and we say that X is encoded with side information at the decoder, the side information being clearly Y. This particular problem is of great importance for different reasons. First of all, from an information theoretic point of view, all the points lying on the segment AB in Figure 2.2 can be obtained by properly switching in time between points A and B, so that the problem of source coding with side information at the decoder is in a sense a building block of the general Slepian-Wolf problem. Second, this problem has its own importance as there are many situations where there is actually only one source to be encoded with the availability of side information at the decoder.

Few years after the publication by Slepian and Wolf [89], Wyner and Ziv [109] obtained an important result for the problem of lossy coding with side information at the decoder, i.e., the case when the source X does not have to be perfectly recovered at the receiver, but with a certain amount of distortion. For lossy source coding, as it is known, the theoretical bounds are described through the computation of the rate distortion function [16, 45, 28]. We do not want to go into the theoretical details of rate distortion here, and we suggest any interested reader to refer to [16] for this. Here we just recall that for the single source problem, supposing that X is an i.i.d. source with marginal p.d.f. $q(x)$ and $d(x, \hat{x})$ is the distortion measure between a reproduction symbol $\hat{x}$ and the original value x, the rate distortion function is given by

$$R(D) = \min_{p \in \mathcal{P}(D)} I(X; \hat{X}) \tag{2.4}$$

where where $\mathcal{P}(D)$ is the set of all conditional probability functions $p(\hat{x}|x)$ such that $E_{x,\hat{x}}[d(x, \hat{x})] \leq D$, i.e. the expected value of the distortion is at most D. In the case when there is side information Y available to both encoder and decoder the rate distortion function simply changes to [16]

$$R(D) = \min_{p \in \mathcal{P}(D)} I(X; \hat{X}|Y) \tag{2.5}$$

where P is now the set of all $p(\hat{x}|x, y)$ such that $E_{x,y,\hat{x}}[d(x, \hat{x})] \leq D$. Wyner and Ziv obtained a characterization of the rate-distortion function when the side information Y is only available at the decoder [109].

Theorem 2.3.2 (Wyner-Ziv, 1976, [109]) *Let two sources X and Y be as in Theorem 2.3.1, and let $q(x, y)$ be their joint distribution. The rate distortion function for the encoding of X with side information Y available to the decoder is*

$$R^{WZ}_{X|Y}(D) = \inf_{p \in \mathcal{P}(D)} [I(X; Z) - I(Y; Z)] \tag{2.6}$$

where Z is an auxiliary variable and $\mathcal{P}(D)$ is the set of all $p(z|x)$ for which there exists a function f such that $E_{x,y,z}[d(x, f(y, z))] \leq D$. ∎

A detailed analysis of the theorem is out of the scope of the present work and we only add some comments that may be interesting for the reader. In addition to prove the above theorem, in [109] the authors observe he following facts:

1. In the general case, for positive distortion values D there is a penalty in the rate distortion bound when the side information is not available to the encoder with respect to the case when it is. This means that the result of Slepian and Wolf does not extend to the lossy case. It has been shown more recently [111], however, that the rate loss is bounded by a quantity that equals half a bit per sample for the case of the quadratic distortion $d(x, \hat{x}) = (x - \hat{x})^2$.

2. Theorem 2.3.2 is valid in a broader setting rather than to the limited case of finite alphabet sources [110]. In particular it is valid if X is a Gaussian source and $Y = X + N$ with N Gaussian and independent of X. In this particular case, under the squared distortion criterion, the rate distortion function can be computed analytically and one has

$$R_{X|Y}^{WZ}(d) = \left(\frac{1}{2}\log\frac{\sigma_N^2\sigma_X^2}{(\sigma_N^2+\sigma_X^2)d}\right)^+, \tag{2.7}$$

where $(\cdot)^+$ is the positive part function, i.e. $(x)^+ = \max\{0, x\}$, σ_X^2 and σ_N^2 being the variance of X and N respectively. In this particular case, the rate distortion function is the same obtained for the case when Y is also available to the encoder. In this case the Slepian-Wolf result does extend to the lossy case.

2.3.2 Additional research results

After the appearance of the papers by Slepian and Wolf and by Wyner and Ziv the field of DSC has received attention by many researchers from the information theory community, and much work still remains to be done. As we have said in the introduction, the two-sources DSC problems specifically studied in [89, 109] can be considered in a more general setting of multiuser information theory [40], so that many connection between this problem and other ones have been studied and different variations on the original two-source problem have been considered. We will only give a brief overview of the results that are more closely related to the setting considered in [89, 109], without pretending to give an exhaustive survey of the whole field of source coding results in multiuser information theory.

The first contribution to the study of the Wyner-Ziv problem was given in [110] by Wyner himself, who provided the extension of the results of [109] to the case of continuous valued sources, as for example the case of Gaussian variables anticipated in [109] as cited above. In [4], Ahlswede and Korner study the problem where X and Y are to be encoded separately and the decoder is interested in the reconstruction of one source only. In [56] the problem with lossy reconstruction of both X and Y is studied where one encoder only has access to Y and the other one has access to X and to a rate constrained description of Y (with possibly null rate), which may or not be also available to the decoder. This problem includes all the previously discussed problems as special cases. Then, a slightly different situation has been studied in [54], where Heegard and Berger address the problem of lossy encoding of the source X for the case where the side information may or may not be available to the decoder or, equivalently, there are two decoders, one with side information Y available and one without. Following the direction of [4], in [17] Berger and Yeung deal with the problem of distributed encoding of X and Y where X is coded losslessly while the encoding for Y is instead lossy.

An important extension of the Wyner-Ziv problem for the case of Gaussian sources has been given in [74]. In this paper, for Gaussian sources X and Y, the author studies the problem of lossy encoding of X with partial side information at the decoder, i.e. a rate constrained description. This can also be seen as a generalization of [4] to the lossy case. Then,

in [112] Zamir and Berger showed that in the low distortion regime there is no loss in the total rate for the separate lossy encoding of both X and Y with respect to the case of joint encoding. In [46] Gastpar proposes a generalization of the Wyner-Ziv problem to the case of multiple sources finding new bounds on the achievable rate regions. Finally, an important result has probably[1] been obtained very recently in [99], where the authors completely establish the achievable rate region for the lossy separate compression of two correlated Gaussian sources. In particular, the authors show that an optimal encoding technique for this problem is obtained by applying a quantization to the sources (on blocks of symbols) and then applying a Slepian-Wolf encoding technique to the obtained quantization indexes. This result, previously obtained in [112] only for the asymptotic low distortion case, is of great importance as it gives an insight into the problem of constructing practical Wyner-Ziv coding techniques that turn out to be optimal at least for the case of Gaussian sources. We will indeed see in the next chapters that practical techniques using DSC principles actually use this approach, i.e. splitting the Wyner-Ziv problem in the cascade of a quantization and a Slepian-Wolf encoder.

The above listed contributions are part of the work done by researchers from the information theory community in the study of achievable rates and rate distortion results for the two terminal source problems initiated by Slepian and Wolf. On the other hand, some interesting contribution have focused on the problem of practical implementation of Slepian-Wolf and Wyner-Ziv coding techniques. One of the first works which has had a great impact in the signal processing community is [76], where Pradhan *et al.* investigate the use of algebraic channel codes for the problem of Slepian-Wolf encoding of correlated sources. The paper is based on the strict relationship between Slepian-Wolf coding and channel coding, first noticed by Wyner in [107], that is of great importance for the understanding of the topics discussed in the next chapters. We thus briefly discuss this relationship with some details, providing meaningful examples, in the next section.

2.4 DSC and channel coding

As we have anticipated in the previous section, there is a close connection between the Slepian-Wolf problem and channel coding. This relation was first noticed by Wyner in [107] where the author uses an interesting example of binary sources to present an intuitive proof of the Slepian-Wolf problem. A more detailed description of this fact can be found in [113], where nested linear/lattice codes are studied in the context of the dual problems of source coding with side information at the decoder and channel coding with side information at the encoder. An interesting study of practical construction of coding techniques for the problem of coding with side information at the decoder using algebraic channel codes can be found in [76]. The use of channel codes has then been studied for the more general Slepian-Wolf problem with two encoders in [85], in [48] and with a deeper analysis in [77]. In this section we assume the reader has familiarity with the basic theory of algebraic channel codes (see

[1] The paper is still under the peer reviewing process at the time this works is written.

[18] for an introduction). Also, in this section we only consider binary codes, which are obviously based on the operations defined on the binary alphabet (for this reason we will write '+' or '-' without any distinction).

2.4.1 Coding with side information

The following example is taken from [76] and is now extensively used in non-information theoretic communities in order to explain the main idea at the base of DSC. Let X be a 3-bits binary word, with every bit independently and equiprobably distributed, in other words X is uniformly distributed in the set $\{0,1\}^3$. Let than Y be a 3-bits word correlated with X so that, given X, Y is uniformly distributed in the set of words whose Hamming distance from X is at most 1. We want to consider the problem of encoding X when Y is available at the decoder. In this simple example it is easy to see that $H(X) = H(Y) = 3$, $H(X|Y) = 2$ and thus $H(X,Y) = 5$. It is also clear that if Y was known at the encoder then the encoding of X could be done with the use of 2 bits; as X and Y differ at most by 1 bit, there are only 4 possible values of X for every value of Y. Now, for the Slepian-Wolf theorem we know that if Y is available only at the decoder, then it is still possible to encode X with a rate $R_X = 2$ bits. The Slepian-Wolf theorem says that this rate is achievable in the sense that, by encoding sufficiently large blocks of source outputs, the average rate can be made as close to 2 bits as we want, with an arbitrarily small probability of error. In this case, anyway, there is a simple solution that allows to encode *every* word X with exactly 2 bits with probability of error exactly equal to zero. The procedure is the following. Partition the set $\mathcal{X} = \{0,1\}^3$ of all X outputs into four disjoint sets in the following way

$$\mathcal{X}_{(0,0)} = \{(0,0,0),(1,1,1)\}, \qquad \mathcal{X}_{(0,1)} = \{(0,0,1),(1,1,0)\}, \tag{2.8}$$

$$\mathcal{X}_{(1,0)} = \{(0,1,0),(1,0,1)\}, \qquad \mathcal{X}_{(1,1)} = \{(1,0,0),(0,1,1)\}. \tag{2.9}$$

Then, the encoding for X is simply done by specifying the index of the set, between the above ones, that contains the actual output x. It is not difficult to verify that, whatever the side information Y is, every subset $\mathcal{X}_s$ contains only one word with hamming distance at most 1 from Y. For example, suppose the outcome of Y is $y = (1,1,0)$ and the code for the outcome x of X is $(1,1)$; then either $x = (1,0,0)$ or $x = (0,1,1)$, but only the first choice is within distance 1 from Y. The trick here is that the partition of the set $\mathcal{X}$ is such that the words in every subset have distance 3, so that at most one of them is within distance 1 from Y.

It is now easier to see the connection with channel coding. Note that the set $\mathcal{X}_{(0,0)}$ is a (3,1) repetition code. The remaining sets are then the *cosets* induced by this code in the space $\{0,1\}^3$, i.e. every set $\mathcal{X}_{s\neq(0,0)}$ is obtained by summing to the words of the set $\mathcal{X}_{(0,0)}$ a word of Hamming weight 1. Thus the sets $\mathcal{X}_s$, $s = (0,1),(1,0),(1,1)$ are all the translated of $\mathcal{X}_{(0,0)}$ and they preserve thus the same distance properties between their words as $\mathcal{X}_{(0,0)}$. So, by using a binary algebraic code we have partitioned the space $\{0,1\}^3$ in four subsets such that the words in every subset are well spaced. We now go back to the encoding of X. Note that the code associated to a particular word $x = (x_0, x_1, x_2)$ can be algebraically

computed as $(x_0 + x_1, x_0 + x_2)$. Thus the code s_x for a word x can be computed simply as $s_x^T = Hx^T$, where H is the matrix

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}. \tag{2.10}$$

This is exactly the parity check matrix of the (3,1) repetition code, whose generating matrix is trivially $G = (1, 1, 1)$. The result encoding of X corresponds thus to simply compute the *syndrome* of the outcome words with respect to the (3,1) repetition code. The decoding operation can then be performed similarly in an algebraic way. At the decoder in fact the syndrome of y can be computed as $s_y = Hy^T$ and, by linearity, we have $s_y + s_x = Hx^T + Hy^T = H(x + y)^T$. So, the sum of s_x and s_y gives the syndrome of the binary difference e between x and y, which we know to have Hamming weight equal to 1. This means that in order to detect the difference e we only need to compute $s_e = s_x + s_y$ and check in the coset specified by s_e for the word of weight 1, known as the *coset leader*. Then easily adding this coset leader to y we obtain x. In the example above where $s_x = (1, 1)$ and $y = (1, 1, 0)$, we have $s_y = (0, 1)$. So, $s_e = (1, 0)$ and the coset leader of $\mathcal{X}_{(1,0)}$ is $(0, 1, 0)$; so, x is correctly decoded as $x = y + (0, 1, 0) = (1, 0, 0)$. Note that in this case where the coset leader is known to have hamming weight equal to one, the syndrome is always one of the columns of the matrix H and the coset leader is just the "indicator" vector of that column.

The procedure that we have here described through the use of a simple example is actually valid in more general situations for the encoding of n-bit vectors when the correlation is expressed in terms of the Hamming distance. The key point is that if a channel code is able to correct d errors, then every two words in a coset have distance at least $2d + 1$ and thus, the syndrome of a word x is sufficient to recover the word if there is, as a side information at the decoder, a word y which has distance at most d from x.

It is important now to point out that computing the syndrome of a word for a code C is the same as computing the parity bits of another code C'. For example, in the case considered above , the parity check matrix H of the (3,1) repetition code used for the computation of the syndrome s_x can be seen as the matrix used for the evaluation of the parity bits in a (5,3) systematic code C' with generating matrix

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{2.11}$$

This code C' is systematic because the first three bits are the information bits, and the remaining two bits, that are the parity bits, are exactly the same as the syndrome of the information word for the (3,1) repetition code (note that the last two columns of G are the row of H above). This gives again a new insight into the use of channel codes for the problem of coding with side information at the decoder. The technique described corresponds to say that instead of sending the whole word x we only send some parity bits of a systematic code. At the decoder the information bits are available as side information

y with at most one error. So, the parity bits are used at the decoder in order to correct y to obtain x.

The above idea of using channel codes for this task can as well be applied in the case when X and Y are binary memoryless sources correlated bit-by-bit in a probabilistic way with $\delta = p(X = Y)$. In this case, the code operates on words of n bits. For every positive ϵ, if n is sufficiently large, the probability that X and Y differ by more than $n(\delta + \epsilon)$ bits can be made arbitrarily small. Thus, using a proper channel code, the encoding of X can be performed using $H(X|Y)$ bits with arbitrarily small probability of error. Note that in this case the probability of error cannot be made exactly zero, as there is always for example a small but positive probability of having two n-bits words from X and Y that differ by all n bits.

2.4.2 Two sources Slepian-Wolf problem

As we have explained in previous sections, the points of the Slepian-Wolf region on the segment between A and B in Figure 2.2 can be reached by properly multiplexing the encoding of X with side information Y with the encoding of Y with side information X. Anyway, it is possible to reach those points also operating a more symmetric encoding of the sources so that both X and Y are encoded in a distributed fashion. The technique explained in the previous section for using channel codes in the problem of coding with side information at the decoder can be extended so as to deal with the more general Slepian-Wolf problem with two sources, so as to balance the rate between the source X and the source Y. Similar approaches have been independently proposed for this problem in [85], in [77] and in [48]. In order to explain this extension we report an example, similar to the one used in the previous paragraph, which has been used for example in [77] and in [48]. In this case we consider two source X and Y that are 7-bits words, again correlated in the sense that their Hamming distance is at most 1.

Asymmetric coding

Consider first the problem of encoding X when Y is available at the decoder, or equivalently, when 7 bits are used for the encoding of Y. In this case, we can use a technique similar to the one proposed in the previous paragraph. We use the systematic Hamming (7,3) code for the generation of the cosets. The generating matrix for this code is

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{2.12}$$

Thus the code for a word x is obtained by computing its syndrome, i.e. $s_x = Hx^T$, where

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{2.13}$$

At the decoder the syndrome s_y of y is computed and used for the evaluation of the syndrome of the difference between x and y, i.e. $s_e = s_x + x_y$. Then the cosed leader of the cosed indicated by s_e is summed to y to obtain x. For example, suppose $x = (1,0,1,0,0,0,1)$ and $y = (1,0,1,1,0,0,1)$. Then the code for x is $s_x^T = Hx^T = (1,0,0)^T$. At the decoder, the syndrome of y is $s_y = (0,1,1)$, so that $s_e = s_x + s_y = (1,1,1)$. This is the fourth column of H, so that $x = y + (0,0,0,1,0,0,0)$ is correctly recovered. So, 3 bits are spent for the encoding of a word x ,which is optimal since in this case since $H(X|Y) = 3$.

Symmetric coding

Now, consider the case where we want to encode both X and Y in a distributed fashion, that is, instead of sending Y with 7 bits and X with 3 bits, we want to share the 10 bits equally with 5 bits each source. Then a similar approach based on channel codes can be used. Instead of using the Hamming code for the generation of cosets, we split the generating matrix G in two submatrices G_1 and G_2 by taking respectively the first two rows and the last two rows of G, that is

$$G_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}, \quad G_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{2.14}$$

This two matrices are used as generating matrices for two codes C_1 and C_2 whose parity check matrices are

$$H_1 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{2.15}$$

Now, the encoding of X and Y is done by using the code C_1 for the generation of the cosets for X and using C_2 for the generation of the cosets for Y. So, for two words x and y we compute the two syndromes $s_x^T = H_1 x^T$ and $s_y^T = H_2 y^T$ and send them to the decoder. Here, given the pair of syndromes s_x and s_y it is possible to uniquely determine the words x and y using the constraint that they differ by at most one bit.

In fact, suppose on the contrary that there are two different pairs of word (x', y') and (x'', y'') satisfying the same syndromes and the same distance constraint. Then, as $s_{x'} = s_{x''}$, $x' + x''$ is a codeword for C_1 and, for similar reasons, $y' + y''$ is a codeword for C_2. Thus, as C_1 and C_2 are subcodes of the Hamming code, $(x' + x'') + (y' + y'')$ is a codeword for the Hamming code. But $(x' + x'') + (y' + y'') = (x' + y') + (x'' + y'')$ has at most weight equal to 2, and as the Hamming code has distance 3, the only word with weight smaller than 3 is the null word. So, $(x' + x'') = (y' + y'')$, but $(x' + x'')$ is in C_1 while $(y' + y'')$ is in C_2. As the rows of G_1 and the rows of G_2 are independent (being G_1 and G_2 submatrices of G), the only intersection of C_1 and C_2 is the null word, that is $x' = x''$ and $y' = y''$. This

proves that the two syndromes s_x and s_y uniquely specify the two words x and y under the given distance constraint. So, the encoding of X and Y can be performed in a symmetric distributed fashion.

In this case the decoding process is more involved as it is necessary to jointly identify the elements of the cosets specified by s_x and s_y. We show here how this can be done in this case. The first important step is to reconstruct the cosets indicated by the syndromes; the main idea is thus to identify two words of 7 bits that gives respectively a syndrome equal to s_x and a syndrome equal to s_y. Let $s_x = (s_{x1}, s_{x2}, \ldots, s_{x5})$ and $s_y = (s_{y_1}, s_{y_2}, \ldots, s_{y_5})$ be the two syndromes. Given that the last five columns of H_1 constitute an identity matrix, consider words of seven bits with the first two components equal to zero. It is then clear that the s_x is the syndrome of the word $\alpha = (0, 0, s_{x1}, s_{x2}, \ldots, s_{x5})$. But s_x is the syndrome of the original word x, and thus x and α are in the same coset for the code C_1. Indeed, it is easy to verify that $H_1\alpha^T = s_x^T = H_1x^T$. Given that words in the same coset always differ for a codeword, we can say that $x = \alpha + w_1$ where w_1 is a word of the code C_1.

Then, with a very similar reasoning, considered that the first two columns of H_2 together with the last three ones constitute an identity matrix, it is easy to see that the s_y is the syndrome of the word $\beta = (s_{y_1}, s_{y_2}, 0, 0, s_{y_3}, s_{y_4}, s_{y_5})$, and thus that y and β are in the same coset for C_2, as $H_2\beta^T = s_y^T = H_2y^T$. So we can write $y = \beta + w_2$ where w_2 is a word of C_2. Now let $\gamma = \alpha + \beta$; we have $\gamma = (x + w_1) + (y + w_2) = (x + y) + (w_1 + w_2) = e + w$ where e has Hamming weight at most 1 and w is a word of the Hamming code. So, as γ is computable at the decoder, we can recover e and w by simply "decoding" γ with the Hamming code, i.e. by using the parity matrix H. So, we compute $H\gamma^T = H(e + w)^T = He^T$, and as e has weight at most 1, $H\gamma^T$ is either zero or one of the columns of H. Thus, e is recovered as the indicator vector of that column. Finally, we can easily compute $H_2x^T = H_2(y + e)^T = s_y + H_2e^T$ and now, considering the equations $H_1x^T = s_x$ and $H_2x^T = s_y + H_2e^T$ as a set of 10 equations in the 7 unknown components of x, we can extract 7 independent equations and thus perfectly recover x. The same can be obviously done for y.

Chapter 3

Distributed Video Coding I

3.1 Introduction

since a few years an increasing interest has been devoted to the application of the DSC principles to different communication problems. One of the most studied applications by researchers all over the world is certainly *Distributed Video Coding* (DVC). DVC is the application of DSC principles to the problem of video coding. It has been proposed independently by two different groups, namely Girod's group, from the University of Stanford [1], and Ramchandran's group from the UC Berkeley [80]. Starting from these pioneering works, DVC has now become an active field of research.

DVC was initially concerned with the application of the DSC ideas to the problem of encoding single video sequences, thus proposing alternative solutions to the traditional video coding techniques mainly centered around the use of motion compensation and transform coding. There are different motivations for the use of DSC techniques in this context. The most important motivations are probably the shift of the computational complexity from the encoder to the decoder. Another argument in favor of a DSC based approach for video coding is the native error robustness in presence of error-prone communications. In short, the classic video coding techniques such as H.264/AVC (see [103, 82] and references therein for details) adopt motion estimation at the encoder for motion compensated prediction encoding of the information contained in the frames of a sequence. This leads to codecs with very good rate distortion performance but at the cost of computationally complex encoders and of fragility with respect to transmission errors over the channel. The computational complexity of the encoder is high due to the motion search that is required in order to properly perform predictive coding from frame to frame. Fragility then is due to insertion of errors in the prediction loop, that causes drift. Therefore, the fragile source coding approach must be followed by powerful channel coding for error resilience. In addition further processing must be designed often at the receiver to adopt the most effective error concealment strategy. DSC techniques are intrinsically based on the idea of exploiting redundancy without

performing prediction in the encoding phase, and leaving to the decoder the problem of deciphering the received codes using the correlation or redundancy between the sources. For these reasons the use of DSC in single source video coding has appeared as a possible solution for a robust encoding with the possibility of flexibly allocating computational complexity between encoder and decoder.

Other than for single source video coding, DVC is now intended as the application of DSC in the more general problem of multi-source (or multi-view) video coding. In many real world applications, such as surveillance networks or acquisition from camera arrays, correlated video sources are to be transmitted from different points to a single receiver, which is interested in recovering the whole set of video sequences. In the classic approach to this type of scenarios, the sources are typically encoded individually without considering the possibility of exploiting the correlation or, alternatively, when there is possibility of communication between the cameras, intercamera prediction can be performed. In some cases, anyway, it would be convenient to exploit the correlation between different views without having to communicate data between cameras, so as to keep the acquisition devices and the encoding process as simple as possible. In these cases, DSC theory comes in as a natural approach to the problem, and DVC is thus in this context a natural application of DSC to sources representing similar video sequences.

Despite this fact, the use of DSC techniques in multi-camera scenarios appears to be much more complex than in the case of single camera systems. We will focus on this point in the next chapter where we provide a high level study of the differences between single-camera and multi-camera DVC systems and its relation with DSC. In this chapter we aim at giving the basic knowledge on DVC, providing a survey of the original schemes proposed at UC Berkeley and Stanford University. We further present more recent contributions made by other research teams.

3.2 Applying DSC to video coding

Before presenting the specific schemes proposed by Stanford University and UC Berkeley for DVC it is necessary to briefly introduce the basic DVC ideas.

Consider a video sequence composed by frames $X_1, X_2, \cdots, X_N$, let R and C be the number of rows in every frame X_i, and let $X_i(r, c)$ represents pixel at location (r, c) in a frame. It is clear that the frames of a video sequence are very redundant, i.e., a video is a source with strong spatio-temporal memory. This happens in two senses that we would like to distinguish. First the content of every frame is redundant in the sense that the pixels $X_i(r, c)$ for a fixed i and varying r and c are strongly correlated, meaning that if we model the frames as stochastic processes, the random variables representing pixel colors that are spatially close in the frame are correlated. Second, the visual content of neighboring frames is very similar, the only difference being usually small movements of the objects, unless a scene change, a flash or some similar "rare" event occurs. In general we will refer to *intra*-correlation the spatial correlation and to *inter*-correlation as the temporal one.

The fact that there is this double-face redundancy in a video is of course of extreme importance to design effective encoders. By properly exploiting the intra- and inter-correlation, it is possible to obtain high compression efficiency in the lossy case. The classic techniques for video coding, starting from H.261 and MPEG1 until the most recent outcome such as H.264/MPEG-4 AVC, exploit the correlation of a video sequence by combining the use of transforms for removing the intra-correlation and the use of motion compensated prediction for dealing with the inter-correlation. We are mostly interested here in this second aspect, i.e. the motion compensated prediction between frames. In the basic situation we can consider the problem of encoding a frame X_i when the previous frame X_{i-1} has already been encoded and it is available in an approximated form, say as $\tilde{X}_{i-1}$, at the decoder. In this case, what a classic video coding technique would do is to estimate the motion field M_i between the frame $\tilde{X}_{i-1}$ and X_i; then, by "applying" this motion to the frame $\tilde{X}_{i-1}$ obtain an approximation of X_i, say $X_i' = M_i(\tilde{X}_{i-1})$. The encoding of X_i is then performed using the prediction, and instead of encoding X_i, the motion field M_i and the prediction error $e_i = X_i - M_i(\tilde{X}_{i-1})$ are sent. The encoding of e_i is usually achieved by transform coding so as to exploit the remaining intra-correlation. This represents only a very coarse prespective of modern video codecs, as an accurate fine-tuning of tools is necessary to achieve optimal Rate-Distortion performance as suggested by different standards (MPEG1/2/4). Nevertheless, the main point for the purpose of the present chapter is sufficiently described in that form: in classic video coding standards a frame is encoded by applying motion compensated prediction from previously encoded frames. This operation is responsible for the exploitation of the source temporal correlation and is usually a somehow complex operation for the encoder. At the decoder, instead, processing is rather simple. The motion field, received from the encoder is applied to the available reference frame (or frames) and used to generate the prediction, which is then successively updated with the received prediction error.

The use of DSC for the problem of video coding is based on the idea that we can consider the frames (or portions of frames) of a video sequence as different correlated sources. So, when we have to encode a frame X_i after having already encoded the frame X_{i-1} as $\tilde{X}_{i-1}$, we can consider that $\tilde{X}_{i-1}$ is already available at the decoder and thus, invoking Slepian-Wolf' and Wyner-Ziv' results, we can consider that we could encode X_i without actually using, and not even knowing, $\tilde{X}_{i-1}$ in the encoding phase. This is the very basic idea under DVC, which has then to be further refined in order to lead to concrete coding schemes. Note that the DSC scenario considered in this case is the problem of source coding with side information at the decoder and, for video sequences, one is usually interested in lossy compression. For this reason DVC is often also referred to as Wyner-Ziv (WZ) coding of video and, more generally, we call Wyner-Ziv coding whatever encoding technique based on the presence of side information at the decoder. By extension, we will often refer to the bits associated to a Wyner-Ziv encoding as the Wyner-Ziv bits and we will often refer to the part of video already available at the decoder as Side Information (SI), in some cases referring to a whole frame or in other cases to portions of frames or even to groups of frames. We will clarify this fact in the next chapter where a detailed study of the correspondences between DSC and DVC is provided. In the mean-time we will make a somehow not rigorous use of

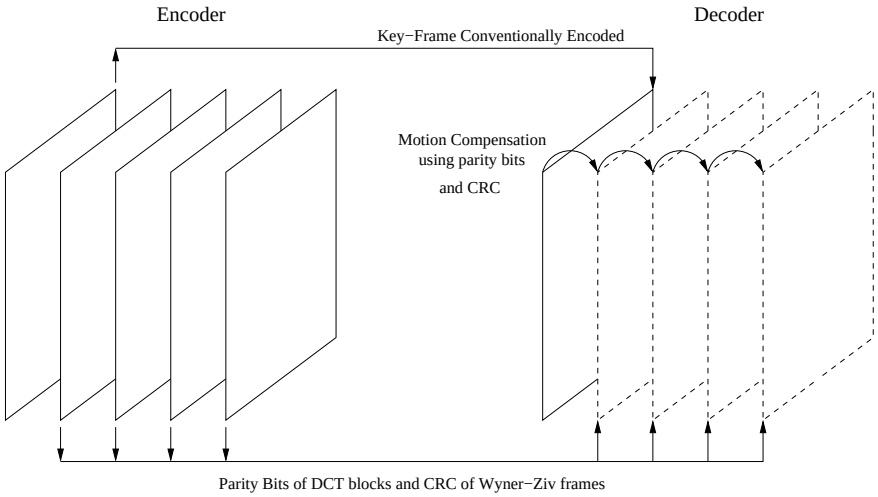


Figure 3.1: Scheme of frame encoding and decoding in PRISM.

these terms, their meaning being clear from the context.

3.3 PRISM codec

In this section we will describe the so called PRISM codec, proposed by Puri and Ramchandran [80] in 2002. We focus only on a small group of frames, the encoding process can then be clearly iterated along a video sequence so as to cope with whatever number of frames. Let again $X_1, X_2, \ldots, X_n$ be the frames. At a very coarse level we can say that the first frame is encoded in a conventional way, using standard techniques for image coding, and then the remaining frames are encoded in a distributed fashion. For every frame, a Wynner-Ziv encoding is applied based on the presence of the previous frame at the decoder. This general scheme is represented in Figure 3.1, where some additional details of the actual technique used for the WZ frames are anticipated. We now proceed to a detailed description of the operations performed by the encoder and by the decoder.

The first frame X_1 is encoded in an intra-mode using for example a block based approach similar to the ones used in JPEG [100]. This means that the frame is divided for example in 8×8 pixel blocks, a DCT is applied on every block and the coefficients are quantized and then entropy encoded using a run-amplitude (RA) code. For the following frames block based process is considered again. The generic frame X_i is divided in 8×8 blocks; let X_i^k be the k-th block, and let $X_i^k(r, c)$ be its pixels. Then every block undergoes the following processing.

1. every block is analyzed so as to estimate its correlation with the content present in

the previous frame: in order to keep the computational complexity of the encoder as low as possible, no motion search is operated; the generic block X_i^k is compared with X_{i-1}^k and the sum of absolute differences is computed, i.e., $\epsilon_i^k = \sum_{r,c} |X_i^k(r,c) - X_{i-1}^k(r,c)|$.

2. Depending on the value of ϵ_i^k every block belongs to either of the three following groups:

   (a) if ϵ_i^k is smaller than a given threshold, say $\epsilon_i^k \leq \epsilon_{\min}$, than block X_i^k is classified as a *SKIP* block;

   (b) if ϵ_i^k is larger than a given threshold, say $\epsilon_i^k \geq \epsilon_{\max}$, than block X_i^k is classified as an *INTRA* block;

   (c) otherwise, block X_i^k is classified as a *WZ* block. WZ blocks are further divided in 16 different classes $C_1, C_2, \ldots, C_{16}$, depending on their ϵ_i^k value, so that the encoder can operate differently on blocks with different level of correlation.

3. A flag is transmitted for conveying the nature of the block (SKIP/INTRA/WZ). If X_i^k is a SKIP block, no further information is encoded. SKIP mode means that the decoder replaces the block with the same position block in the previous frame. If X_i^k is an INTRA block, it is encoded in a traditional way using a DCT based approach with a RA code, as in JPEG. The decoder can thus decode this type of blocks without any reference to other frames. If X_i^k is a WZ block, instead, the index specifying the associated class is added. The block is then encoded as described below (see also Figure 3.2).

Encoding procedure for the WZ blocks

1. The DCT of the block is computed. Let $\hat{X}_i^k(f)$, $f = 1, 2, \ldots, 64$ be the coefficients of the DCT when assuming a zig-zag scan ordering as typically done in JPEG;

2. The high frequency coefficients $\hat{X}_i^k(f)$, $f = 16, \ldots, 64$ are quantized and encoded using a RA code, according to the JPEG standard. The quantization of these coefficients clearly depends on the desired quality for the reconstructed video;

3. The remaining low pass coefficients are actually encoded in a WZ fashion in the following way

   (a) A first quantization is applied to the coefficients depending on the class of the block, i.e. a specific quantization parameter $q(j,f)$ is used for the coefficient $\hat{X}_i^k(f)$ if block X_i^k is in the class C_j. Let ${}_q\hat{X}_i^k(f) = \lfloor \hat{X}_i^k(f)/q(j,f) \rfloor$ be the quantized coefficient. The 16×15 quantization matrix values $q(j,f)$ is fixed at the encoding phase and is constructed a priori by properly training the codec on test sequences as clarified later.

(b) The last 2 bits of the binary representation of the quantized coefficients, for a total of 30 bits, are extracted and fed to a systematic trellis code with rate 2/3. The resulting 15 parity bits are the syndrome of the 15 coefficients, i.e. the actual WZ bits.

(c) A 16-bit CRC is also computed from the 15 quantized coefficients. The resulting 16 bits form the hash of the block and are used in the decoding phase in order to detect the correct WZ decoding of the block.

(d) If the quantization levels $q(j, f)$ used for the low frequency coefficients are too large for the required quality, additional refinement bits will be sent separately. This basically corresponds to quantizing at a finer level the coefficients $\hat{X}_i^k$ and sending the missing bits with a traditional encoding technique.

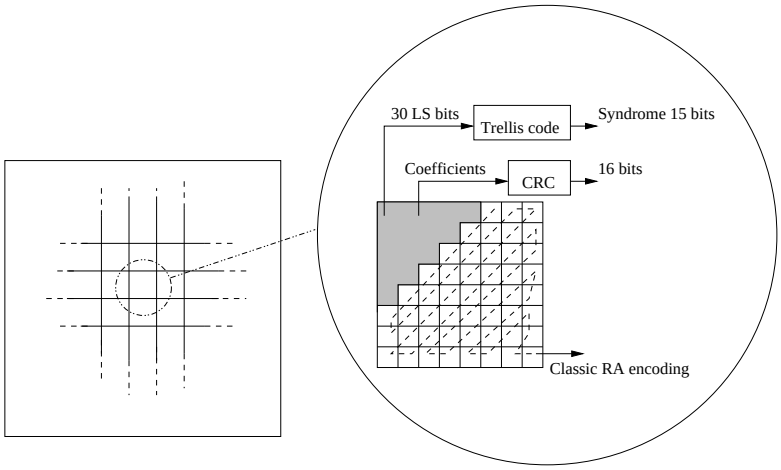


Figure 3.2: Encoding procedure for the WZ blocks in PRISM.

The above explained procedure for the encoding of the WZ frames is not completely specified since it is not yet clear how the the values of the used parameters are established. We refer here to the thresholds $\epsilon_{\min}$ and $\epsilon_{\max}$, to the quantization parameters $q(i, j)$ and even to how the C_j, $j = 1, \ldots, 16$ classes are determined based on the value of ϵ_i^k. As we have anticipated for the quantization levels, all the values of the parameters are established by properly training the codec on test video sequences. This is due to the fact that the level of correlation between the different blocks in the frames is responsible for the success or the failure of the decoding process. It is important to properly set $\epsilon_{\max}$ so that only the blocks that are sufficiently correlated with the previous frame are encoded in a WZ way, and to set $\epsilon_{\min}$ so that blocks very similar to the homologous blocks in the previous frame are not unnecessarily encoded with a complicated WZ procedure. Thus the classes C_j and the quantization parameters $q(j, f)$ are important because the syndrome extracted from the DCT

coefficients depends on the applied quantization. The last two bits of a quantized coefficient, in fact, can only identify the value of the original unquantized coefficient $\hat{X}_i^k(f)$ modulo $4q(j,f)$. In order to have the syndrome uniquely identify the quantized coefficient ${}_q\hat{X}_i^k(f)$ when a side information Y is available at the decoder, it is necessary to have $|Y - \hat{X}_i^k(f)| < 2q(j,f)$. So, the quantization parameter $q(j,f)$ must be chosen so as to be at least half the value of the correlation error between the coefficient to be encoded and the side information available at the decoder.

So, in order to properly set the quantization parameters it is necessary to partially simulate on test sequences the operations performed in the encoding and in the decoding process and find the average correlation error between the WZ coefficients and the side information, depending on the value of ϵ_i^k, which is the only measure of correlation available during the encoding process. For this reason it is now important to present the decoding process, the procedure for the tuning of the parameters being easily understood later. Of course it is not necessary to describe here the decoding operations for the SKIP and the INTRA blocks, so that we only expose the decoding for the WZ blocks.

Decoding procedure for WZ blocks

The decoding for a block X_i^k is performed by combining a sort of motion estimation and a WZ decoding in the following way:

1. For a WZ block many different blocks are tested as side information. A sliding window selects candidates SI blocks in positions around the position of the given block but in the previous frame. Let us call $Y_i^k(n)$, $n = 1, 2, \ldots$ the candidate predictors for the block X_i^k.

2. Every candidate block is used as SI; it is transformed and quantized using the specific quantizer for the class containing X_i^k (that is written in the header of the code for that block)

3. The two least significant bits (the modulo-4 value) of the first 15 low pass coefficients are extracted and used as side information for a WZ decoding, where the channel code is the trellis code used in the encoding phase and the parity bits are the syndrome bits received from the encoder. A least square decoding if performed so as to minimize the squared error between the decoded sequence of modulo-4 values and the modulo-4 values of the SI coefficients. These 15 modulo-4 values are then used to replace the two least significant bits of the SI coefficients.

4. The CRC-16 is computed on the so obtained "corrected" side information coefficients. If the CRC matches the decoding is considered correct and the procedure stops, otherwise another block is selected from the previous frame and the process is repeated from step 2. If no available SI allows to match the CRC there is no way to reconstruct the 15 missing coefficients and a concealment strategy must be adopted.

5. When the procedure for low frequency coefficients is terminated, the high frequency coefficients are decoder in a traditional mode and inserted to fill the DCT transform of block. The inverse transform is then applied to obtain the pixel values of the block.

The procedure is thus based on the idea of trying different possible blocks as SI for the current block to be decoded. The channel decoding operation is performed on every possible candidate SI block and the CRC is used as a hard decision for the effective correctness of the decoding process. If the quantization level and the threshold are well tuned, the hope is that at least a good predictor should be available and that the process will correctly recover the quantized WZ coefficients when this predictor is picked up. When wrong predictors are tested, instead, after the channel decoding, the probability that the CRC will match with the one received by the encoder if the reconstructed quantized coefficients are wrong is very small. This idea of testing different predictors is an operation very similar to the motion estimation usually performed at the encoder, with the difference that here the task is performed by the decoder.

Now, it is easier to clarify how the training stage for the tuning of the parameters operates. Different video sequences are tested and the statistical correlation between the value of ϵ_i^k for a given block and the optimal prediction error after motion compensation is studied. This way, good choices for $\epsilon_{\min}$ and $\epsilon_{\max}$ can be set and a map from the value of ϵ_i^k to values of the quantization parameters $q(j, f)$ can be constructed so as to ensure that the decoding process will be succesful with high probability.

3.4 Stanford solution

In this section we consider the architecture proposed by the Stanford group in [1]. With respect to the PRISM codec considered in the previous section the Stanford architecture adopts different choices for the application of WZ principles to the case of video sequences. The main difference is that the frames of the sequence are considered as a whole, and the WZ coding is applied to a whole frame and not to single blocks. So we can actually identify some WZ frames that are completely encoded in a WZ way, without differentiating the processing on a block to block basis. A basic idea is in fact to estimate the motion at the decoder and create an entire side information frame, even before considering the WZ decoding, and thus the use of the parity bits. So, the technique used by PRISM of embedding the motion estimation for every block with its own WZ decoding is changed here to the creation of side information for the whole frame, using motion, and then WZ decoding of this frame.

The coarse idea is to split the frames of the sequence at the encoder dividing them in two groups. Let again $X_1, X_2, \ldots$ be the frames; in the simpler version of the codec, odd-indexed frames $X_1, X_3 \ldots$ are encoded in an intra-mode conventional way, that is as a sequence of images, while even indexed frames $X_2, X_4 \ldots$ are encoded in a WZ fashion. At the decoder, the intra-coded frames are used in order to create an approximation for the WZ frames by motion compensated interpolation. Then, the parity bits are used in order to "correct" these predictions and recover the frames. This idea is graphically represented in

Figure 3.3.

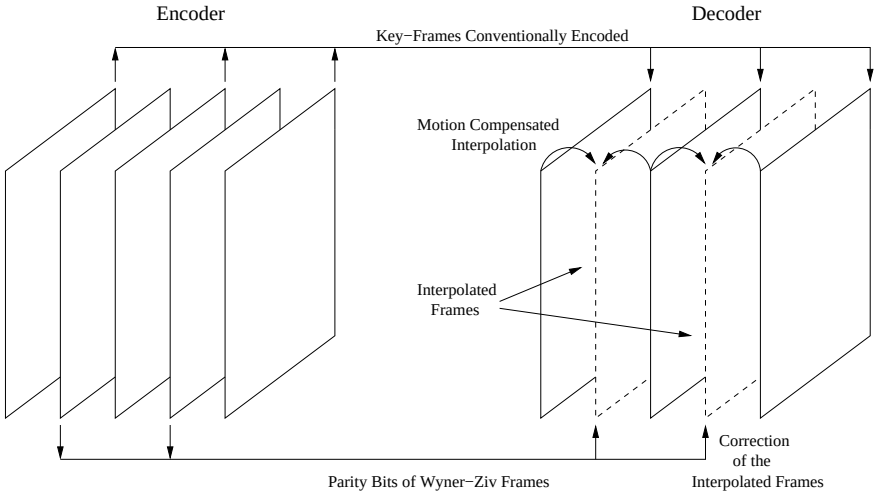


Figure 3.3: Scheme of frame encoding and decoding in the Stanford approach.

It is important to consider that this is only a very general description of the general idea and that the actual details for the design of such a system are not well described even in [1]. Many research papers have been devoted to the study of this DVC system and the description we give in the following part of this section is obtained by combining interpretations of details from different authors. Also, it is necessary to clarify in advance one particular characteristic of this architecture, which is the need of a feedback channel from the decoder to the encoder. This feedback channel is used in the process of WZ decoding in order to request more parity bits from the encoder if the received ones are not sufficient to properly decode the source. Even if from a theoretical point of view this feedback channel could be eventually removed by introducing higher functionalities at the encoder side, up to now no clear and simple solution is available from the literature for this task. We will discuss later some related problems and we will consider in the next chapter the use of feedback channels in general DVC problems.

We thus proceed to a more specific description of the encoding operations for the WZ frames and of the associated decoding process.

Encoding of WZ frames

The encoding of a WZ frame, say X_{2n}, is performed following list of operations reported below:

1. The pixels of the frames are properly quantized to say 2^M levels. Let ${}_qX_{2n}$ be the quantized frame and let ${}_qX^i_{2n}(r,c)$ be the i-th bit of the binary representation of ${}_qX_{2n}(r,c)$.

2. The bits of the binary representations of the quantized values ${}_qX_{2n}(r,c)$ are juxtaposed to form in bit-planes; let ${}_qX^i_{2n}$ be the i-th bitplane. Every bitplane is fed into a turbo encoder and its resulting parity bits S^i are stored in a buffer.

The encoding procedure as shown above is notably simple. It is worth noticing here that this scheme is usually referred to as "Pixel Domain" WZ coding, because of the fact that the quantization and WZ encoding is applied directly to the pixel values. The same scheme can be applied however with the variation of having a transform applied before quantization.

We now present the decoding operation for the WZ frames

Decoding of WZ frames

The decoding process for a WZ frame X_{2n} is as follows

1. Let X'_{2n-1} and X'_{2n+1} be the two reconstructed key frames adjacent to X_{2n}. By applying a motion compensated interpolation, X'_{2n-1} and X'_{2n+1} are used for the construction of an approximation Y_{2n} of X_{2n}, which is the Side Information for the WZ decoding.

2. The SI is assumed to be a noisy version of the original frame, i.e., for every pixel it is assumed that $Y_{2n}(r,c) = X_{2n}(r,c) + N(r,c)$ where N is a white noise. This noise is assumed to have a Laplacian distribution with parameter α which has to be estimated or to be somehow known to the decoder. With this model, given Y only at the decoder, it is possible to assign a probability to the values $X_{2n}(r,c)$ as

$$P[X_{2n}(r,c) = x] = \frac{1}{2S}\alpha \exp\left(-\alpha|x - Y_{2n}(r,c)|\right) \tag{3.1}$$

   where S is a rescaling factor due to the fact that the pixel values are always clipped in some range (and thus the Laplacian is in reality a clipped Laplacian).

3. The actual WZ decoding operates bitplane-by-bitplane. Consider the first bitplane, i.e., the most significant bit. With the probabilistic model described above it is possible to compute the probability that the first bit of every quantized coefficient is 0 or 1.

4. These probabilities are fed to the turbo decoder as "channel values" of the information bits. The turbo decoder, using a feedback channel asks for parity bits from the encoder. The encoder applies a puncturing on the parity bits and sends a first set of them. The turbo decoder tries to decode the channel values with these parity bits to recover the bitplane ${}_qX^i_{2n}$.

5. If the turbo decoding process fails, more parity bits are requested through the feedback channel, and the process repeat until the turbo decoder is able to correctly recover the bitplane.

6. When the first $g \geq 1$ bitplanes have been reconstructed, the conditional probabilities $p[X_{2n}(r,c) = x|_q X_{2n}^i, \ldots,_q X_{2n}^g]$ are used to compute the probabilities for the $g+1$-th bitplane, and the process is repeated from step 4.

7. After all bitplanes of ${}_qX_{2n}$ have been recovered, the best estimate X'_{2n} of X_{2n} is constructed by taking for every pixel the expected value

$$X'_{2n}(r,c) = E[X_{2n}(r,c)|Y_{2n}(r,c),_q X_{2n}(r,c)] \tag{3.2}$$

under the probabilistic model introduced above.

3.5 Additional developments

In the previous sections the first proposed architectures for practical DVC systems have been discussed. As we have anticipated, after the appearance of the works by Stanford University and UC Berkeley many researchers have put efforts in the development of new ideas that would allow a practical system to reach better performance. It is important to clarify that the approaches in [1, 80] have substantially dominated the DVC panorama, as they are still considered the starting point for many works in the field. So, many of the new ideas that have been proposed in the last four years have focused on the development of new techniques and architectural blocks for the improvement of the systems in [1, 80] and on possible variations or combinations of structural blocks from these two main schemes. In this section an overview of the new development obtained for the case of single camera systems is given. These latest developments are clustered in different sets in order to point out the different aspects of DVC that are being studied by the scientific community.

3.5.1 Side Information quality improvements

A great deal of work has been devoted to the improvement in the generation of the Side Information. Obviously the quality of the Side Information is strongly related with the quality of the motion estimation that is available to the decoder. This two aspects are strongly related. So, in order to improve the quality of the side information different methods have been proposed in the literature that are based on improving the quality of the motion estimation at the decoder. For the Stanford approach, for example, one of the first proposals came for the Stanford group itself, with the publication of [2], where the use of hashes is proposed for facilitating the motion estimation task performed at the decoder. The main idea is to send, for the WZ frames, not only parity bits but also a low-rate coarse description of the blocks of the frames. This description can be used by the decoder in order to find a better estimate of the motion, or even to extract a motion field and a prediction for the WZ frame only from past frames, without any need to wait for a successive intra coded frame. Successive contributions in the direction of better SI constructions in the Stanford scheme was proposed in [11]. In this paper a smoothing post-processing operation on the motion field extracted from two consecutive key frames is shown to improve the quality of

the estimation for the WZ frame in between them. Another approach along this line was presented in [61] where the use of mesh-based motion compensated interpolation is shown to give better results than simple block matching. Finally, in [57] a sub-pel motion estimation technique has been used to provide slightly better performance than integer-pel one. In [12] and [9] the authors present two techniques based on the idea of iterating the Wyner-Ziv decoding operation with successive refinement of the Side Information. In a sense, the idea is to take advantage of the increased information available on a WZ frame after a first stage WZ decoding in order to create a better estimation of the motion.

3.5.2 Correlation Noise Modeling and Rate Allocation

In the decoding process of DVC systems, an important role is played by the assumed model of correlation between the available SI and the original WZ frame. In [1] the difference between the SI frame and the WZ one is assumed to have a Laplacian distribution with zero mean and a fixed variance. It was not clear however how the variance of the Laplacian should be set so as to reach the best performance, and in practice fixed values optimally computed off-line for the sequences were used in the first DVC implementations. As the variance of the noise affect the probabilities that are fed into the turbo decoder, it is of great importance to set its value in a proper way. This means that an accurate study of the statistical modeling of the correlation noise is important in order to achieve good performance. In [96] a non-Laplacian distribution is used and the effect of quantization on the key frames on the noise statistics is studied. In [102] and [71] a non-stationary model of the noise is proposed that consider the combination of Laplacian and other particular ditributions. In [35], instead, a non stationary Laplacian model for the noise is proposed, where the variance of the distribution is tuned for every point of the WZ frame depending on a measure of confidence extracted by the motion compensated difference between the key frames. This contribution is described in detail in Section 4.3 of this work. Further developments in this direction were provided in [21] and [23] where a detailed study of adaptive models at different resolutions is proposed. An interesting alternative idea to the use of a model for the correlation error between SI and original WZ frame has been adopted in [73]. Here there is no construction of SI from the key frames and both frames are used as references with a multi-hypothesis technique in order to take better advantage of the contained information. In particular this approach leads to better performance in handling covered and uncovered regions. For the approach proposed by Berkeley a similar problem must be considered. In this case, having a good model for the correlation noise is important in order to properly quantize the blocks in the encoding phase. In the original scheme proposed in [80] a training stage was used in order to create a map of the quantizers to be used for every block of the frame depending on a coarse measure of correlation extracted at the encoder. In [10] a first study of the statistics of the correlation noise is proposed in order to suppress the training stage. Note that in the case of the PRISM codec, the modeling of the noise actually directly impact the rate to be spent in the encoding phase. This relates the modeling of the correlation noise to the allocation of the rate in the encoding phase. In fact, in a DVC system

one of the most difficult problems to be solved is the allocation of the required rate for a given aimed quality of the reconstructed video sequence. In fact, the use of channel codes for the correction of the SI makes it difficult to freely tune the quality of the decoded frame depending on the rate spent in the Wyner-Ziv code. The typical problem is that there is a required amount of parity bits that are necessary for the correction of the SI; if fewer parity bits are allocated by the encoder, then the decoder cannot recover the original data. The particularly annoying thing is that the degradation is not graceful. If the parity bits are not sufficient they are almost useless and a very poor reconstruction is obtained at the decoder. So, it is very important to correctly estimate at the encoder the amount of rate required. In the Stanford approach this problem is masked by the presence of the feedback channel. But in case a feedback channel is not available, in order to have the scheme in [1] work, it is necessary to allocate the rate at the encoder. Very few works have been done along this line. A simple idea for the rate allocation is proposed in [8] while in [22] a detailed study of the use of the feedback channel in the Stanford scheme is provided.

3.5.3 Architectural Developments

Finally, a set of developments in the field of single camera DVC schemes have been proposed in the literature that are not strictly based on the improvement of the performance of the codecs proposed by Stanford and Berkeley but better on variations on these schemes. In [93] Tagliasacchi *et al.* proposed the use of intra-coded blocks in a Stanford based approach. In a few words, the idea is to encode in a conventional way the blocks of the WZ frames for which a good prediction is not possible at the decoder due, for example, to occlusions. Two approaches for the selection of the blocks to be encoded in a conventional way are presented, one with decision at the encoder side and one with decision at the decoder side, the latter being possible only when there is a feedback channel of course. In [44] Fowler *et al.* propose a variation to the PRISM codec based on a wavelet decomposition of the video sequence rather than a jpeg-like block partitioning. In [94] then, a new proposal for exploiting the spatial redundancy is presented that does not use transform coding but exploit the correlation between neighboring pixels in the WZ decoding phase, modifying the statistical model used for the bit probabilities by using the intra-frame memory. In [66] the idea of sending hashes for a better motion estimation at the decoder in the Stanford scheme is further developed by sending a low resolution of the video encoded with a zero-motion H.264 codec. This is also related to the use of DVC techniques for scalable video coding, which has also been discussed, for example in [92] and in [86].

Chapter 4

Distributed Video Coding II

4.1 Introduction

After having described the first schemes for DVC proposed in the literature, we here develop an in-depth analysis of the problem of DVC from a structural point of view. In particular, we are interested in providing here an analysis of the underlying problems in DVC and of the fundamental structure of DVC systems. A detailed study of the relation and differences in the hypothesis between DSC and DVC is proposed, with the identification of an important component that we call *correlation issue*. We will show that this correlation issue is strongly related to the motivations for the use of DVC for single- and multi-camera systems, and to the architectural constraints that need to be considered for both cases.

As detailed in the previous chapter, in the case of single camera systems we consider a sequence as the composition of different sources, i.e., every frame is considered as a different source. We are then interested in encoding the frames independently in a distributed fashion while decoding them jointly in order to take advantage of the existing correlation between frames. In a multi-camera system, instead, we actually have different sources and we want to encode them independently, by taking advantage of the correlation between the source, in order to improve the compression performance, but without having any communication link between the cameras. In the case of multi-camera systems, we may have some cameras operating in a distributed fashion while some other cameras operate with a classic approach.

4.2 From theory to practice

In this section we want to point out some important base-level considerations that are essential in order to discuss about structural issues in the design of a distributed video coding

[0]This chapter includes research results published in [35].

system. As already clarified, Distributed Video Coding is the application of Distributed Source Coding (DSC) theory to the problem of coding video sources. It is clear that whenever a theory is applied to a concrete setting there are issues to be considered, and this is the case for DVC too. Without going too much into details here, we want to clarify some of the most important differences between the underlying hypothesis under which DSC theory was developed and the real situations where we want to study the use of DVC as a new framework to the problem of video coding. Primarily we are interested in applying DVC to different settings that we can divide in two main families, namely single-camera systems and multi-camera systems. The application of DVC to these two scenarios has different motivations and the problems encountered in the design of such systems are different though strongly related to the motivations for which DVC is used. So, it is essential to first focus on the motivations for the use of DVC techniques in the case of single-camera and multi-camera systems.

4.2.1 Motivations for DVC

There are different reasons why distributed source coding techniques are of interest in the case of video coding. In order to have these motivations clear, it is necessary to consider separately the different scenarios, namely single-camera systems and multi-camera systems. The motivations are mainly the following:

Single-camera systems:

- Reduced complexity encoders. The use of distributed source coding techniques allows to consider different frames of a video sequence as different correlated sources and thus encode them separately. This way, instead of encoding the video sequence using motion compensated prediction, like in the classic video coding techniques, no prediction is performed at the encoder. This reduces the complexity of the encoder itself, as it does not operate any motion search.

- Error resilience: The distributed approach to video coding leads to a scheme that is more robust to transmission errors with respect to classic prediction based techniques. In the case of predictive coding, in fact, the presence of an error in a given frame of the video sequence leads to a propagation of the error in all successive frames, as they that are encoded based upon this corrupted frame. In the case of DVC instead, the absence of a prediction loop at the encoder leads to a higher level of robustness, as an error in a given frame does not propagate to the next frames.

So, for the case of single-camera systems we have the above two main motivations for the use of DVC techniques.

For the case of multi-camera systems we have instead the following obvious other motivation for using DVC.

Multi-camera systems:

- Take advantage of the correlation between different views for reducing the total rate transmitted from the cameras to the receiver, without having to share information between the cameras.

The motivations above explained for the adoption of a DVC approach for single and multi-camera systems are to be considered very carefully as they determine the main issues in the application of DSC to video coding. We will in fact now clarify in the next section that one of the problems to be solved in the development of DVC systems is the correlation issue. This problem is related to the motivations for the use of DVC; we will thus clarify this connection after explaining what we mean by correlation issue.

4.2.2 Correlation issues

As it was previously anticipated, in this section we want to clarify the difference between the hypothetical setting under which DSC theory was developed and the practical situation that is encountered in the case of DVC. The first thing to say is that that in DSC some strong hypotheses of stationarity and ergodicity of the sources are made, while it is clear that video sources are strongly non stationary nor ergodic (or, better said, they are not adequately represented by such type of models). More precisely, there is a substantial difference in the hypothesis on *a priori* information of the encoder and the decoder. In DSC theory one assumes that both encoder and decoder a priori know the joint statistical properties of the sources to be encoded, while in the case of DVC one does not know a priori the joint statistical characteristics of the involved sources, or more precisely one can only have a partial knowledge of their joint statistics. We have not yet specified what the involved sources for DVC are. We can refer to the signals of different cameras in a multi-camera system or to the different frames of a single camera system that we consider as separate sources. In order to simplify the discussion, in this section we only focus on those cases where DVC is intended as a source coding problem with Side Information (SI) at the decoder, i.e. when we are interested in operating on the corner points of the Slepian Wolf region or, for the case of lossy source coding, in a Wyner-Ziv setting. In this case we can rephrase the previous ideas by saying that in DSC the joint distribution between the SI and the data to be encoded is known at both encoder and decoder, while in the case of DVC there is no such knowledge. This is usually expressed by simply saying that the correlation between the SI and the original data is not known a priori. In order to properly discuss the problem it is now necessary to clarify what we mean when we say "SI" and "correlation" in a DVC problem. In the case of DSC, the terms "SI" and "correlation" are very precisely described in terms of random variables; the SI is a random variable that is "correlated" with the original data to be sent, in the sense that they are not statistically independent. In the case of DVC, instead, the terms "SI" and "correlation" have often different meanings. We try to propose a formalism to the underlying ideas so as to have some clear description of the entities involved in a DVC system. In a DVC system we typically have to encode a frame of a video sequence, say X,

and we suppose that different frames of the same sequence, or from different sequences, are available to the decoder. The information theoretic SI in our setting is represented by the whole set of frames available at the decoder. This is in fact the whole data available at the decoder that is correlated with the frame to be encoded. In a practical context anyway, it is very difficult to directly handle the correlation between this set of frames and the frame to be encoded X. The duality between coding with side information at the decoder and channel coding suggests the use of channel codes for this problem. In general, channel codes are developed for recovering an original signal from a noisy version of it. So, in order to use channel coding techniques within DVC we have to reduce the problem to a situation where we have at the decoder a noisy version of the original data. This means that the whole set of SI frames cannot be used in a channel decoder, but we have first to extract from this set of frames an approximation Y of the original signal. This approximation Y is then the "new" side information, which is actually used for recovering the original data X, and many times we refer to Y itself as the side information. The difference between Y and X represents the *virtual noise* that the channel decoder has to remove, and thus the similarity between X and Y is often again called "correlation" between the side information and the original data. Note that with this scheme we have moved the problem from a situation where we only have an imprecise idea of correlation to a situation where we can use a difference as measure of similarity, and we have thus introduced a normed space. In order to formalize the above discussion and clarify the description of the entities and of the operations involved in a DVC scheme we summarize the situation as follows:

1. A frame X of a sequence has to be encoded in a distributed fashion. The encoder sends a Wyner-Ziv code $S(X)$ of the frame (for example some parity bits) to the decoder;

2. The decoder has access to different decoded frames from the same sequence or from different sequences. We call these data Prior Side Information (PSI), and we indicate it with Y_p;

3. The decoder extracts from Y_p an approximation Y_e of X which we call Extracted Side Information (ESI);

4. The decoder obtains a decoded version of X, say X_d, by "correcting" Y_e with the use of Wyner-Ziv bits received from the encoder.

In the above scheme we can now identify the points where the correlation issue arises. There are two such points which are related to the two faces of the correlation problem. The first point is the extraction of Y_e from Y_p. In order to construct the estimation Y_e of the original data X, it is in fact necessary to know what type of processing has to be performed on the PSI Y_p in order to construct a good approximation of X. The second point is the correction of Y_e to obtain X, where the statistical properties of the difference between X and Y_e are important in order to perform the correction of Y_e to obtain $E[X|Y_e]$. In order to deal with the first point of the correlation issue, namely with the construction of Y_e, we can

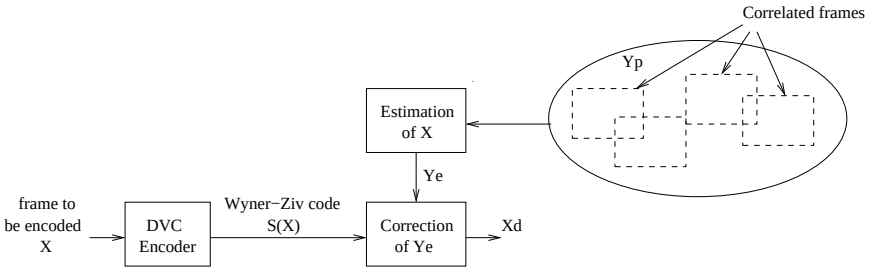


Figure 4.1: Graphical representation of prior and extracted side information and correction of the latter.

introduce the following formalism. We define on Y_p a set F of transforms and we say that Y_e is obtained by choosing a particular f_e in F so that $Y_e = f_e(Y_p)$. The first problem is thus to choose a proper f_e in order to obtain a good Y_e. In theory, we should choose the optimal f in F, which is $f_{opt} = \arg\min_f(\|X - f(Y_p)\|)$. The choice of the optimal function f_{opt} requires anyway in general some information on the data X, and it is clear that X is not known at the decoder (our objective is indeed to transmit X). This is the first correlation issue, that is in order to extract the optimal Y_e we would have to know the original data X at the decoder. If we analyze deeper the situation we realize that in fact the problem is that we need both Y_p and X in order to construct the optimal Y_e. But X is the data to be sent from encoder to decoder, while Y_p is the side information available at the decoder and not at the encoder. Now we note that it is not possible to keep a general approach to the problem, as there is a fundamental difference between the single camera systems and the multicamera ones. In a single camera system, in fact, Y_p is actually available at the encoder, but we do not want to use it for several reasons. In particular, if we want to use DVC in order to have a low-complexity encoder we cannot or we should not make use of Y_p at the encoder in order to identify the best function f_{opt}. On another hand, if we are only interested in using DVC for error robustness concerns, we can find the optimal f_{opt} at the encoder and then communicate this choice to the decoder (assuming such side information will not be significantly affected by transmission errors). In a multi-camera system, instead, we are only interested in the case where the cameras cannot communicate between each other, and this implies that the encoder, who knows X, does not know Y_p. So, the problem of choosing a good f_e must be solved at the decoder. If no information is available on the data X we should construct the side information Y_e in the best possible way given that there is no information on X. In a more general case, we can anyway assume that some description $D(X)$ of X is sent by the encoder in order to help the decoder in the choice of a proper function f_e for the extraction of Y_e. This description could be a low resolution version of X or some contour information or high level description or any other appropriate information, such as the geometry of the acquisition system (in Chapter 6 we consider and study with some detail an example of such descriptions that is useful in order to perform registration at the decoder side). The decoder can use $D(X)$ in order to extract from Y_p a

good estimate of X. Note that if $D(X)$ is not a complete description of X, in the general case the decoder cannot find f_{opt} because it cannot evaluate the difference between $f(Y_p)$ and X. So, we have to assume that the decoder estimates the difference between $f(Y_p)$ and X, and then chooses as f_e the function that minimizes this estimate. We can thus consider the more general encoding/decoding scheme as follows:

1. The encoder sends a description $D(X)$ of the frame X and a Wyner-Ziv code $S(X)$;
2. The decoder extracts Y_e based on Y_p and $D(X)$. In order to do this, the decoder sets $Y_e = f_e(Y_p)$ with $f_e = \arg\min_f(Est(\|X - f(Y_p)\|))$ where $Est(\|X - f(Y_p)\|)$ is an estimate of $\|X - f(Y_p)\|$ based on the knowledge of $D(X)$.
3. The decoder constructs X_d by correcting Y_e with the WZ bits $S(X)$. So, X_d is a function of Y_e and $S(X)$, say $X_d = g(Y_e, S(X))$. The decoder can possibly go back to step on step 2, using all the information available at this point in order to extract a better Y_e and repeat the WZ decoding again, and then reiterate the process.

In this scheme a second correlation issue arises in the last step, as we said above, where we have to recover X_d from Y_e and $S(X)$. It is important in fact to clarify that the decoding of X_d from Y_e and $S(X)$ is based on channel coding techniques, as explained in Chapter 2. The main idea, in fact, is that the side information Y_e is very similar to X and thus, if we have some parity bits of the original X we can recover it by "correcting" Y_e. So, in this step, channel codes are used in order to correct Y_e to obtain X_d. In this phase, in order to have soft channel code working properly, it is important to have some information on the statistics of the difference between Y_e and X when the WZ encoding is performed. In particular, as the WZ code $S(X)$ is created at the encoder side, it is in theory necessary to know at the encoder the amount of bits needed to correct Y_e. In DSC it is assumed that the encoder knows the correlation with the side information, while in our situation the difference between X and Y_e is not a priori known. Furthermore, as Y_e is usually created at the decoder, the encoder does not know it. It is thus necessary to find a solution to the problem of allocating the rate for $S(X)$. This point is of great importance in DVC and again in order to tackle the problem it is necessary to consider the different scenarios, namely single and multiple camera systems.

In a single camera system, as the encoder has access to Y_p it is possible in principle to evaluate the distortion between Y_e and X. If the motivation for the use of DVC is the light complexity of the encoder we should not compute Y_e at the encoder side. In many cases it is however possible to estimate the difference between Y_e and X without completely constructing Y_e. The better we want to estimate this difference and the more complex the encoder must be. So, in the single camera system there is a trade-off between the computational complexity we are allowed to use at the encoder and the efficiency in the rate allocation for $S(X)$. If we do not want to perform any estimation at the encoder, we have to allocate the rate for $S(X)$ basing this only on a priori assumptions and we must therefore be either very conservative or very little robust. We have anyway a very different situation if we consider the possibility of having a feedback channel from decoder to encoder, as in he case for example of the Stanford codec. In this case, it is possible to simply computer at

the encoder much more bits than required for $S(X)$ and then use the feedback channel to let the decoder ask bits to the encoder until it has enough to decode the sequence X_d form Y_e. It is important to note that the availability of this feedback channel significantly changes the problem from a structural point of view. We postpone a detailed discussion of this topic to the next section.

In the multiple camera system the situation is very different. In this case in fact, the encoder does not have access to the intercamera side information Y_p, so that there is no trade off between the computational complexity of the encoder and the efficiency in the rate allocation for the WZ code. In the case of multicamera systems the encoder can only guess the amount of bits needed by the decoder based on some a priori knowledge. Even in this case, the presence of a feedback channel from decoder to encoder completely changes the perspective, and allows for a correct allocation of bits to the WZ code $S(X)$. It is clear that the use of feedback channels in an encoding scheme has to be considered with much care, and so we leave to the following section a detailed discussion on this topic.

4.2.3 Feedback channels

In this section we aim at clarifying an important topic in DVC, namely the use of feedback channels. With the term feedback channels we refer to a channel that can be used in order to transmit information from the decoder to an encoder (or between encoders in a multi-camera systems). The first thing to clarify is that the presence of feedback channels is usually not considered in the theory of DSC and in particular there is no assumed feedback channel in the Slepian-Wolf and in the Wyner-Ziv theory. So, it is important to clarify that the use of feedback channels in DVC is not motivated by any theoretical results in DSC. Instead, the use of feedback channels in DVC has to be considered as a new element with respect to DSC, and the aim of this section is to clarify why the introduction of this channel has some motivations that are structurally due to the differences between DVC and DSC. In the previous section we have presented the "correlation issue" in DVC and we have clarified that the most important difference between DVC and DSC is indeed that in DVC there is no knowledge on the correlation between the involved sources. As we said, the correlation issue is two sided, in the sense that there are two main problems; the first is that the decoder has to construct an approximation of X from the prior side information Y_p and the second problem is that in order to allocate the rate for the WZ code the encoder must know the correlation between the original data and the extracted side information Y_e. As we have clarified in the previous section the first of these two points have an impact in the encoding of the data X and led to the conclusion that it may be necessary in some cases to help de decoder in the extraction of Y_e by letting the encoder send some description $D(X)$ of the data X to the decoder. The second correlation issue, instead, is due to the fact that the encoder needs to know something that is in most cases only known at the decoder. This clearly motivates the fact that in DVC we are interested in considering what could be done with and without the presence of a feedback channel from decoder to encoder. Again we want to clarify that in the case of DSC there is no such problem as the encoder completely

knows the statistics of the sources and it is thus able to allocate the proper amount of bit rate for the WZ code. In the case of DVC we could see the use of feedback channel as a penalty to pay for the fact that we have not a complete knowledge of the sources. In order to have a clear understanding of the problems, we need to consider again the importance and the real need of a feedback channel by separating the discussion for different scenarios, namely single-camera systems and multi-camera systems.

Consider first the single camera scenario. In this case the distributed source coding approach is adopted in order either to keep a light computational complexity of the encoder or to ensure error robustness. The single source is seen as a composition of different sources and we use a distributed approach to decouple the encoding of different frames. From a *topological* point of view, there is just a single source to be encoded by one encoder and all the information descriptive about the source could be generated at the encoder. So, the correlation between X and Y_e can in principle be computed at the encoder without any need of feedback channels from decoder to encoder. However, as long as we are interested in using DVC techniques in order to have a light encoder, we do not want to construct Y_e at the encoder because this is an expensive operation. Instead of constructing Y_e, we can try to estimate the correlation between X and Y_e by performing some computationally simple operations on Y_p. In the general case we would need to balance the computational complexity of the encoder with the ability to allocate the optimal rate for the WZ code of X. So, if in a single camera system there is no available feedback channel, we will need to find a trade-off between the requirement of having a light complexity encoder and the requirement of having good compression performance. If a feedback channel is available, instead, then we can leave to the decoder all computationally expensive operations and use the feedback channel to ask the right amount of bits to the encoder. From a structural point of view this discussion suggests that in order to have good performance for a DVC technique in a single source coding a feedback channel is probably important, but it is not strictly necessary in order to solve the problem if we can accept a trade-off in our requirements.

For the case of multi-camera sources, instead, the problem is really different. In fact, if there is no feedback channel, in a multi-camera scenario the encoder does not have access to the intercamera side information available at the decoder. In this case, there is less a discussion on the complexity of the encoder. However, as it will not be possible to estimate the correlation between X and Y_e (unless some limited direct communication would be allowed between the different cameras), the only possibility is that we have some a priori knowledge on such level of correlation. In the general case, this restriction leads to a very poor robustness of the system and it is necessary to be very conservative if the correlation level is not well known. If instead a feedback channel from the decoder to the encoder (or alternatively between coders) is available, then there is a structural difference in the architecture of the system, as it is possible to send from the decoder to (between) the encoders important information that would not be known otherwise by the encoders. This simple observation clarifies that with respect to the single camera scenario, the feedback channel in a multi-camera scenario is of much higher importance. Depending on the way we want to use the feedback channel, in fact, we can construct different encoding-decoding schemes

and different uses of the feedback channel lead to different implications.

As a final discussion on this section we want to clarify how we have to reconsider a DVC scheme when we are allowed to use feedback channels. In our discussion we have motivated the fact that there are some strong reasons for considering the use of a feedback channel in certain situations. Namely we have explained the problems that arise due to the correlation issue. Once the benefits of having a feedback channel has been motivated, one still needs to define a way to use this channel, and the rate that can be sent through the channel. Consider in fact that when we add a feedback channel, from a topological point of view we have completely changed the situation. If no constraints are set on the use of this channel we can easily fall in unrealistic conditions. Take as an example a multicamera system. We are there interested in using DVC in order to take advantage of the correlation between different views for compression performance, but without having communication between the cameras. If we then allow the presence of a feedback channel from decoder to encoders, and if we do not put any restriction on this channel, we have then reduced the problem to a multicamera system with communication between cameras, as we can send the data of a camera to other cameras passing through the decoder and then through the feedback channels. So, we have reduced the DVC problem to a multiple view coding problem where the "distributed" part disappears. In the case of a single camera system there is a similar consideration that should be kept in mind, namely if there is a feedback channel and we do not put the appropriate constraints on it we can transmit the side information Y_e to the encoder which can use it in order to perfectly estimate the difference between X and Y_e with no computational burden. In both cases there are in reality some synchronization issues, that will not be taken into account in this work. So, we conclude that the rate sent over the feedback channels should have some strong constraint in order to still talk about DVC. Finally, we briefly comment here that the use of feedback channels is strongly related to the working condition and applications of DVC. The first obvious but important point to clarify is that the possibility of using a feedback channel strongly depends on whether the system has to work in real-time or not, for example, for archival purposes. It is clear that in the first case it is not possible to use feedback channels. In the same way, the use of feedback channels may be possible in real-time contexts but in this case it will be necessary to impose some additional constraint in terms of latency.

4.3 Improving turbo codec integration in Stanford codec

In Section 4.2.2 we have presented an analysis of what we called the correlation issue, and we have clarified that there are two meaning for the word "correlation" in a distributed video coding system, one of them being related to the statistical properties of the prediction error between the extracted side information Y_e and the original data X, i.e. the model of the *correlation noise*. As we have anticipated, this model is important in order to properly use channel codes for the WZ decoding of data; in this section we aim at proposing an improvement of the correlation noise model in the DVC scheme proposed by the Stanford group, presented in Section 3.4, in order to improve the integration of turbo codecs in a DVC scheme. With respect to the Stanford scheme, some contributions have been made in the literature that focus on improving the performance of the Wyner-Ziv coding by improving the quality of the constructed side information (see for example [11, 12]) as explained in Section 3.5.1. Only a few attention (see [96]) has been paid instead to the problem of better modeling the correlation between side information and the original data in order to improve the channel code performance. Here, limitedly to the Stanford approach, the problem of finding a good model for the correlation between the side information and the original data is considered. In particular, the main objective of this section is to propose a good model for the correlation noise between an original video frame and a prediction obtained by motion compensated interpolation between adjacent frames. So, as in [1], we are here only interested to the very basic situation where every odd-indexed frame is supposed to be available at the decoder while even-indexed frames are Wyner-Ziv encoded.

The starting point for this work is an implementation of the Stanford architecture, provided by the IST[1] group within the DISCOVER project [39], which incorporates into the basic codec structure some important modifications performed by IST group researchers [11, 12]. For ease of description of the proposed contribution, we clarify here the working hypothesis and we briefly recall the basic scheme of the Stanford architecture, putting the attention on the details of the correlation model that were not deeply analyzed in the previous chapter.

For the present study, we do not consider quantization of the key frame, so that every odd-indexed frame X_{2n+1} of the video sequence is supposed to be available at the decoder, and we focus on the problem of Wyner-Ziv encoding of even indexed frames X_{2n}. For these frames a bit-plane based encoding approach is considered (see Fig. 4.2); the gray level values are uniformly quantized and the bitplanes are fed one by one into a turbo encoder. A systematic turbo code is used in order to extract from each bitplane some parity bits to be passed to the decoder. In the decoding phase, for every even frame X_{2n} an estimation Y_{2n} is constructed by applying motion compensated interpolation between the two adjacent frames X_{2n-1} and X_{2n+1}. The parity bits output from the turbo encoder are then used in order to correct the estimation Y_{2n} and extract a better reconstruction X'_{2n} of the original frame. The corresponding codec architecture is shown in more details in Fig. 4.2.

[1]We would like to thank Catarina Brites, João Ascenso and Fernando Pereira, Instituto Superior Técnico, Instituto de Telecomunicações, Lisbon, Portugal, for providing the initial version of the used software.

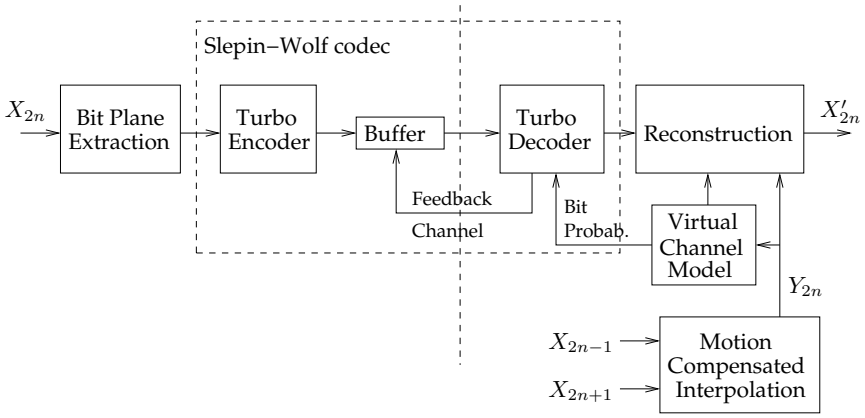


Figure 4.2: Architecture of the considered codec.

We clarify again that here we assume, as in [1, 11], that the key frame are losslessly available at the decoder. This hypothesis is not admissible in a practical video coder but the effects of quantization on the key frames can be considered of secondary importance for the study presented in this section.

Of the whole architecture shown in Figure 4.2 it is important to consider here the virtual channel model block and the turbo codec part. In the virtual channel block the correlation model between the side information Y_{2n} and the original frame X_{2n} is used in order to compute bit probabilities to be fed into the turbo decoder where Soft-Input Soft-Output decoders are used (see Fig. 4.4). This bit probabilities computation is detailed in the next section where we describe both the classic method used in the Stanford scheme and a non-stationary model that we propose as an improvement.

4.3.1 Virtual Channel Model

Let us call $X_{2n}(r, c)$ the pixel value in the r-th row and c-th column of the $2n$-d frame. The side information Y_{2n} is constructed at the decoder by motion compensated interpolation between frames X_{2n-1} and X_{2n+1}. Assuming constancy of the motion between X_{2n-1} and X_{2n+1}, this means that for every (r, c) point an estimation $Y_{2n}(r, c)$ of the value $X_{2n}(r, c)$ is computed as

$$Y_{2n}(r,c) = \frac{X_{2n-1}(r - v_x, c - v_y) + X_{2n+1}(r + v_x, c + v_y)}{2}, \tag{4.1}$$

where v_x and v_y are (halves of) the estimated motion vector components. We are not interested here in how v_x and v_y may be computed, we refer to [11] for an important contribution in this direction. Once the whole side information frame Y_{2n} has been constructed, it is

used for the bit probabilities evaluation. This means that for every point (r, c) the value of $Y_{2n}(r, c)$ is used in order to evaluate the probability of every bit of $X_{2n}(r, c)$ being 1 or 0. What is done in the literature (see [1, 11]) is to consider that the virtual noise between X_{2n} and Y_{2n} has a Laplacian distribution with zero mean and estimated standard deviation $1/\alpha$. Hence, for every possible value of the amplitude x, the probability that $X_{2n}(r, c)$ is equal to x is given by[2]

$$p[X_{2n}(r, c) = x] = \frac{1}{2}\alpha \exp\left(-\alpha|x - Y_{2n}(r, c)|\right) \tag{4.2}$$

Let then $X^i_{2n}(r, c)$ be the i-th bit of the value $X_{2n}(r, c)$ and let Z_i be the set of x values that have i-th bit equal to zero; then for every i we compute

$$p[X^i_{2n}(r, c) = 0] = \sum_{x \in Z_i} p[X_{2n}(r, c) = x] \tag{4.3}$$

This way, for a given bitplane i we can compute the probabilities $p^i_0(r, c) = p[X^i_{2n}(r, c) = 0]$ for all the values of r and c, and we consider these probabilities as channel probabilities to be input to the turbo decoder.

It is important to note that in the above presentation the α parameter is assumed to be fixed for all (r, c) positions. If we look at bit probabilities as "confidence levels" assigned to the bits, the fact that the α parameter is fixed for different positions means that the side information Y_{2n} is considered to have the same confidence in all points. In other words there is an implicit assumption that the quality of the side information is constant across the frame, without considering the quality of the motion estimation.

So, using a Laplacian distribution model with a fixed parameter corresponds to giving the same confidence to the side information in every point of the frame. It is not difficult to realize that the quality of the side information is very different from point to point depending on the quality of the motion, on the presence of occlusions, lighting changes, ... So, a good model for the noise between X_{2n} and Y_{2n} should take into account this space varying nature. A possible adjustment to the model consists in considering the virtual noise to have nonstationary Laplacian distribution. In other words we let the α parameter vary from point to point and we thus indicate it with $\alpha(r, c)$. The effect of this choice during the turbo decoding process is that well predicted values are considered to be more reliable by the decoder and it is then easier to correct errors where the discrepancy between X_{2n} and Y_{2n} is actually higher.

In this setting, the important point is then how to set the value of $\alpha(r, c)$ depending on the information we have on the side information confidence in the (r, c) point. A simple yet effective approach we have considered in this work consists on using expression (4.1) in order to set the value of $\alpha(r, c)$. In fact, for every (r, c) point, in addition to the value of the obtained side information $Y_{2n}(r, c)$, it is very important to consider the two values

[2] Actually the amplitude of the Laplacian must be rescaled in order to have total probability be equal to 1, as the amplitude values x are typically clipped between 0 and 255.

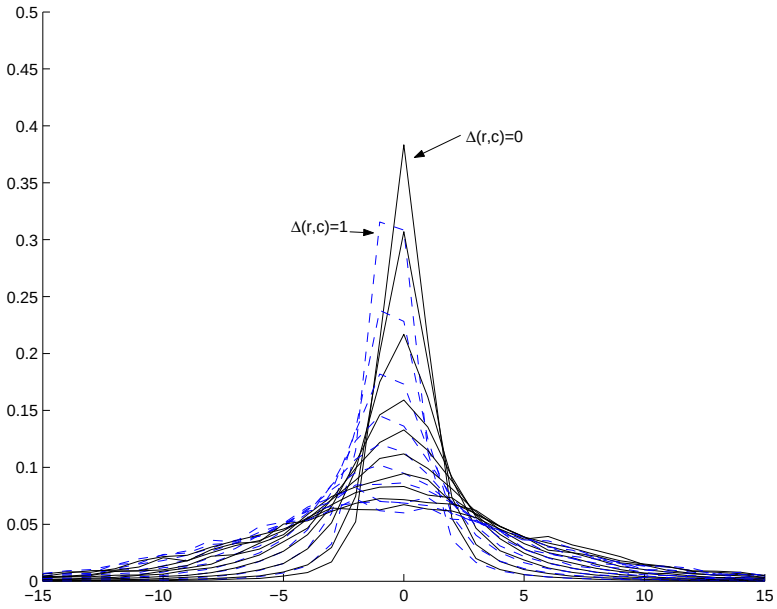


Figure 4.3: Histogram of the prediction error conditioned to the value of Δ, for $\Delta = 0, \ldots, 20$. In this example we have used the first 100 frames of the *foreman* sequence, QCIF resolution at 30fps. The solid and dashed lines represent even and odd values of Δ, so as to make it easy to distinguish the curves.

$X_{2n-1}(r - v_x, c - v_y)$ and $X_{2n+1}(r + v_x, c + v_y)$ from which $Y_{2n}(r, c)$ is obtained as an average. It is clear in fact that, in a typical sequence, the more those two values differ the less confidence we should give to their average. So, we should use an expression for $\alpha(r, c)$ so that it decreases when the value

$$\Delta(r, c) = |X_{2n-1}(r - v_x, c - v_y) - X_{2n+1}(r + v_x, c + v_y)| \tag{4.4}$$

increases and viceversa[3]. In Figure 4.3 we see an example of the prediction error statistic once it is conditioned to the value of $\Delta(r, c)$.

A possible expression for the $\alpha(r, c)$ values, which has shown to give good empirical results, is the following:

$$\alpha(r, c) = \frac{\beta}{\gamma + \Delta(r, c)}. \tag{4.5}$$

where β and γ are estimated parameters constant along every frame. Like for the α parameter, the optimal choice for the values of β and γ depends on the sequence and it is

[3]Remember that the standard deviation is $1/\alpha$.

thus necessary to estimate them from the key frames and from the previously decoded WZ frames. The main point here, apart from considering particular expressions for the $\alpha(r, c)$ values, is to have a nonstationary model of the noise. In the above expression (4.5) for $\alpha(r, c)$ we have only used the value of $\Delta(r, c)$ but it is clear that further information may be used, as for example the value of the motion vector of the block containing the point (r, c). Moreover the best choice for the $\alpha(r, c)$ parameter may also depend on the values of $\Delta(i, j)$ for (i, j) in a neighborhood of (r, c) and not just in that exact location.

4.3.2 Pre-Interleaving

In the previous section we have presented a possible way of handling the nonstationary nature of the correlation noise. As we said, the main benefit obtained from such an approach is that it improves the turbo decoding process by providing more reliability to better predicted pixels and low reliability to badly predicted ones.

Another important characteristic of the correlation noise is the memory property. In fact, as the quality of the side information in some areas is higher than in some other ones, we conclude that $\alpha(r, c)$ parameter will have most of the times high values on pixels that are placed close together. This implies that, for a generic bitplane, many consecutive bits are affected by high values of virtual noise, as in the case of typical burst errors. So, if the turbo encoder is fed with bits read row by row from the frame bitplane, the first of the two SISO decoders inside the turbo decoder (see Fig. 4.4) is faced with the problem of correcting sequences of consecutive very noisy bits. So, due to the fact that the used codes are recursive convolutional codes, in order to correct these noisy areas a high number of parity bits is required from the first decoder. But in the considered scheme the bitrate is managed by using rate compatible puncturing, as explained in [1]. It is then not possible to simply increase the number of parity bits associated to noisy areas, and additional requested parity bits are "spread" all over the frame. This implies that in order to have a sufficient number of parity bits for noisy areas we must have more parity bits, most of which are wasted in low noise areas.

Note that this problem does not affect the second SISO decoder inside the turbodecoder, as the interleaver positioned before this second decoder cancels the burst effect spreading noisy bits far apart in the bitstream. So, a simple but important benefit for the use of turbo codes in this framework results from placing an interleaver also before the first encoder. This provides substantial improvements in terms of rate distortion performance.

4.3.3 Experimental results

In this section some experimental results are shown. In Figure 4.5 the rate distortion performance comparison for the foreman sequence is shown, where the improvements given by non stationary model and by the pre-interleaver are visible. For this sequence we have set $\alpha = 0.37$ for the stationary model. For the non stationary model we experimentally set $\gamma = 10$ and we set β so as to have an average standard deviation equal to the stationary model. This way we are sure that the shown results are only due to the non stationary model

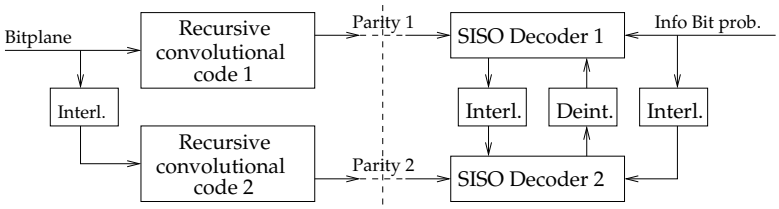


Figure 4.4: Turbo codec scheme.

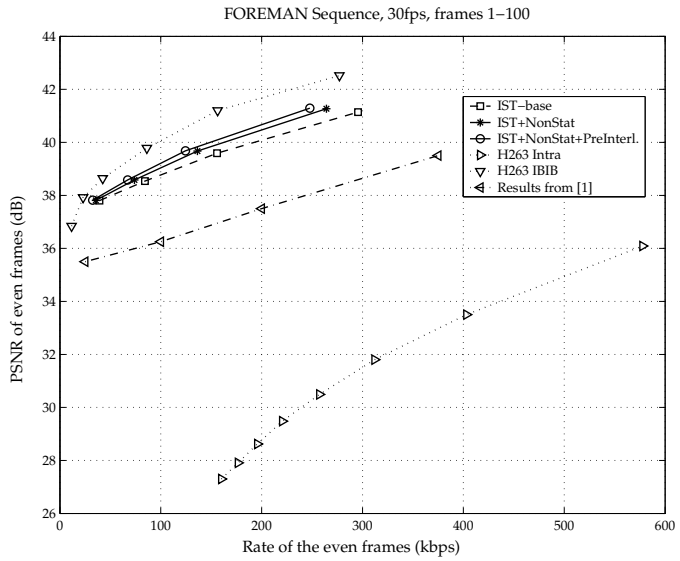


Figure 4.5: Rate-distortion performance for the first 100 frames of foreman, QCIF, 30fps.

and not from some different a priori assumptions. In Fig. 4.6 the empirical standard deviation of the correlation noise conditioned to the value of $\Delta(r, c)$ is shown. It can clearly be observed that the value of the correlation noise (represented by α) is strongly correlated with the value of $\Delta(r, c)$. It can be noticed the standard deviation of the noise (even when conditioned to $\Delta(r, c)$) depends on the motion level in the sequence, since the first 100 frames of the tested sequence exhibit less motion with respect to the next 100 frames.

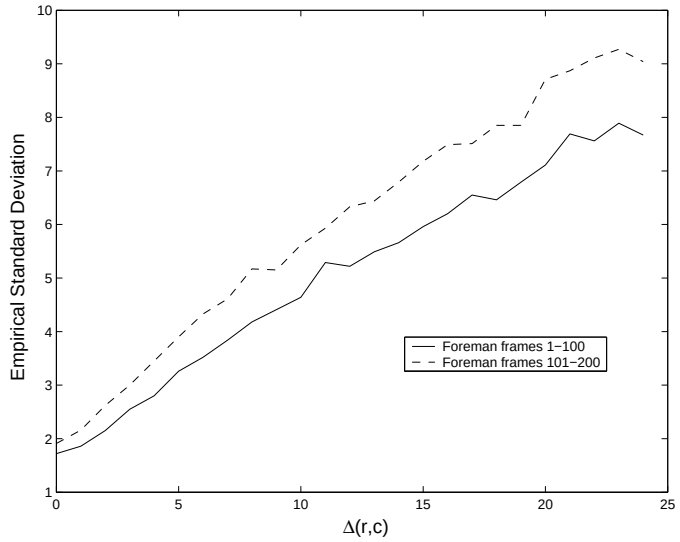


Figure 4.6: Empirical standard deviation conditioned to $\Delta(r, c)$ on foreman.

Chapter 5

Coding Constrained Sequences

Heavier than air flying machines are impossible.
– Lord Kelvin –

5.1 Introduction

In this chapter we want to consider a problem which is related to the approach, considered in the previous chapters for the case of video sequences, of using Distributed Source Coding techniques to encode a single source. If we abstract the idea to a general level, without considering the particular case of video coding, we realize that what we are doing, from a source coding point of view, is to use a DSC approach to exploit the memory of a source. In this sense, instead of using predictive encoding techniques, we use a distributed approach and we let to the decoder the task of exploiting the memory of the source in order to decode the received strings of code symbols.

Consider for a moment the problem of source coding with side information at the decoder. From the point of view of source coding as the construction of mappings from source symbols to codewords, the side information problem actually reduces to the use of mappings that are possibly not invertible by themselves, but are invertible when a side information is present. So, embedded again into the application of coding single sources with memory, the use of DSC techniques actually corresponds to the use of codes that are not invertible by themselves, but are invertible once the memory properties of the sources are taken into account.

In this chapter we want to further investigate this problem from an information theoretic point of view. So, we consider the problem of encoding a source with memory with codes that are not necessarily *decodable* in the classic sense (as it will be clarified later), but that are decodable under the constraint of a memory property. The model we consider here is the case of first order Markov sources, and we consider thus codes as fixed mappings from the

[0]This chapter includes research results published in [32].

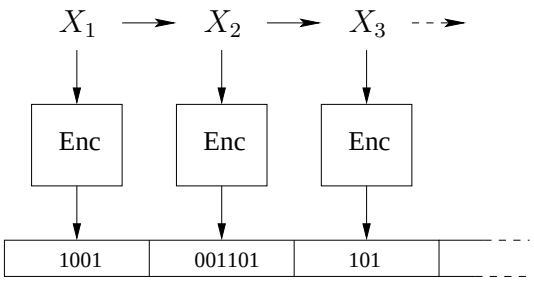


Figure 5.1: Encoding of a Markov source by fixed mapping from symbols to bits.

alphabet symbols to strings of bits as show in Figure 5.1. Here we are only concerned with lossless coding of discrete sources, and we will always assume such a condition in every discussion, without explicitly recalling it every time.

The above setting leads to a problem which has very interesting connections with a couple of very basic concepts in the field of source coding, namely unique decodability and expected lengths of codes. In his famous work [88], Shannon showed that the entropy of a source is the fundamental quantity governing the rate required for a lossless representation of its sequences of symbols. Shannon's work was "only" focused on asymptotic rates and showed that, asymptotically, it is not possible to encode a source at rates below the entropy. No assertion was done on the minimum rate required for the representation of a finite number of symbols. In the succeeding years, attention was also paid to the minimum average length of a lossless code for a finite number of symbols. The key result in this direction, due to McMillan [68], is that every "*uniquely decipherable*" code must satisfy the Kraft inequality [60]. From this fact, it is easy to derive that the average length for symbol of a "uniquely decipherable" code is at least the entropy of the source. Building upon this result, in the information theory community it is usually asserted that for every source the average code length for a block of n symbols is greater than or equal to the entropy of those symbols.

In this chapter we want to analyze in great detail the above sketched situation. We show that the deduced properties of codes actually hold only under certain hypotheses, that are less general than what is usually considered in the information theory community. In particular, we show that if one considers sources with memory, then there exist stationary and ergodic sources and associated lossless block codes such that the average length of the code for n symbols is always strictly smaller than their entropy. Motivated by this fact, we revise the idea of unique decodability and related issues with an eye to the case of constrained sequences. We propose (or enforce) a source-specific definition of unique decodability and we derive a weak Kraft inequality which represents a really necessary condition for unique decodability. In accordance, we propose a variation of the Sardinas-Patterson test for testing the unique decodability of a given code for a constrained source. In addition, we propose an analysis of the proofs of McMillan's theorem showing that a proof

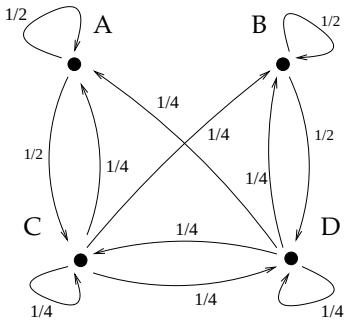


Figure 5.2: Graph, with transition probabilities, for a Markov Chain.

essentially equivalent to the ones in [68, 55] was almost already present in Shannon's work [88] in a much more general form, hidden in a formula for the evaluation of the capacity of certain channels.

5.2 A preview example

Consider a source X generating symbols $X_1, X_2, X_3, \ldots$ extracted from the set $\mathcal{X} = \{A, B, C, D\}$ following the Markov chain rule graphically shown in figure 5.2. The labels on the arrows indicate the transition probabilities from one symbol to the other. The transition matrix associated with this source is thus

$$\mathbf{P} = \begin{bmatrix} 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}. \tag{5.1}$$

Let $\mathbf{S}_i$ be the probability distribution row vector on $\mathcal{X}$ at step i and let the initial state be uniformly distributed over the four symbols, i.e., $\mathbf{S}_1 = [1/4, 1/4, 1/4, 1/4]$. The distribution at successive instants can be computed using the recursive relation $\mathbf{S}_{i+1} = \mathbf{S}_i\mathbf{P}, \forall i \geq 1$. It is easy to verify that the uniform distribution is the stationary distribution of the transition matrix, so that with our hypothesis we have $\mathbf{S}_{i+1} = \mathbf{S}_i\mathbf{P} = \mathbf{S}_1$. So, the considered source is stationary and, given that the matrix $\mathbf{P}$ is irreducible, it is also ergodic.

We want to consider possible encoding techniques for this source. In order to evaluate their performance we first compute the entropy of the source. For every $n \geq 1$ we clearly have

$$
\begin{aligned}
H(X_1, X_2, \ldots, X_n) &\overset{(a)}{=} H(X_1) + H(X_2|X_1) + \cdots + H(X_n|X_{n-1}) && (5.2)\\
&\overset{(b)}{=} H(X_1) + H(X_i|X_{i-1})(n-1), \quad \forall i > 1 && (5.3)\\
&\overset{(c)}{=} 2 + \frac{3}{2}(n-1)
\end{aligned}
$$

where we have used in (a) the Markov property of the source, in (b) the stationarity, and in (c) the given probability assignment.

We now consider two different binary codes for this source.

Classic code

We call this code "classic" as it is the most natural way to encode the source given the particular structure. For the first symbol we have four equiprobable choices, so that we use 2 bits for it, in the obvious way. For the next symbols we note that we always have dyadic conditional probabilities. So, we apply a state-dependent code. For encoding symbol k we use, again in an obvious way, 1 bit if symbol $k-1$ was an A or a B, and we use 2 bits if symbol $k-1$ was a C or a D. This code seems to perfectly fulfill the source as the number of used bits always corresponds to the uncertainty. Indeed, if we compute the average length of the code for the first n symbols we have

$$
\begin{aligned}
E[l(X_1, X_2, \ldots, X_n)] &= E[l(X_1)] + \sum_{i=2}^{n} E[l(X_i)] && (5.4)\\
&\overset{(a)}{=} 2 + \frac{3}{2}(n-1) && (5.5)
\end{aligned}
$$

where in (a) we have used twice the fact that the distribution is always uniform. So, the expected number of bits used for the first n symbols is exactly the same as their entropy, so we would say with some certainty that the code has optimal performance.

Alternative code

Let us consider a different code, obtained by applying the following fixed map from symbols to bits: $A \to 0$, $B \to 1$, $C \to 01$, $D \to 10$. It is easy to see that this code is not *uniquely decodable* in the classic sense, defined for example as in [28] and discussed in details in the next section. This is because different sequences of symbols are mapped into the same codeword, for example AB and C are both coded to 01. It is also easy to see that, for the particular source we are considering in our example, the code does not introduce ambiguity, because different sequences that are producible by the source are mapped into different codes, so that it is possible to "decode" any sequence of bits without ambiguity. For example the code 01 can only be produced by C and not by AB because our source cannot produce such sequence (the transition from A to B is impossible in our source). It is not difficult to verify that it is indeed possible to decode every sequence of bits by considering two bits at a time. If a 00 (or a 11) is found then clearly there is an A symbol

followed by a code starting with a 0 (or a B symbol followed by a code starting with a 1). If, instead, a 01 pair is found (or a 10) then a C is decoded (or a D). The coding technique is thus defined by the following shemes for the encoding and decoding operations

$$\begin{array}{cccc} \text{Encoding} & & & \\ A & \to & 0 \\ B & \to & 1 \\ C & \to & 01 \\ D & \to & 10 \end{array} \qquad \begin{array}{ccc} \text{Decoding} & & \\ 00\ldots & \to & A+0\ldots \\ 01\ldots & \to & C+\ldots \\ 10\ldots & \to & D+\ldots \\ 11\ldots & \to & B+0\ldots \end{array} \tag{5.6}$$

Now that we have shown that the code can indeed be used to represent without loss our source, we evaluate its performance. The expected number of bits in coding the first n symbols is easily computed as

$$\begin{aligned} E[l(X_1X_2X_3\cdots X_n)] &= \sum_{i=1}^{n} E[l(X_i)] \\ &= \frac{3}{2}n \end{aligned} \tag{5.7}$$

Unexpectedly, the average number of bits used by the code is strictly smaller than the entropy of the symbols. So the performance of this code is even better than what we would have considered to be the "optimal" one obtained with the classic coding technique.

It is important to point out here that we have just shown a code for a stationary ergodic source that maps sequences of n symbols into strings of bits such that the average code length is smaller than the entropy of those n symbols, and this happens for every n. In source coding, the expected difference between the code length and the entropy is usually called *redundancy* and is usually supposed to be a nonnegative quantity. Thus, in a sense we could say that our code is affected by *antiredundancy* instead of *redundancy*. Note that there is a huge difference in our situation with respect to that of the so called *one-to-one codes* (see [6] for details). In that case, it is assumed that only one symbol must be coded, and one is interested in studying codes as maps from symbols to binary strings without any need to study the decodability of concatenation of codewords. Under those hypotheses, Wyner [106] first pointed out that the average codeword length can always be made lower than the entropy, and different authors have studied bounds on the expected code length over the years [19, 84]. Here instead, we are using a classic block code and we are applying this code to compress sequences of symbols of whatever length, concatenating the code for the symbols, one by one as in the classic scenario.

A number of questions arise after considering the above simple example. We should consider with some further detail what happens with our coding procedure and what happens when the number of symbols goes to infinity. The first thing to point out is that the average gain per symbol goes to zero as n increases – as it must be, being an immediate consequence of the Asymptotic Equipartition Property for ergodic sources [67]. Looking carefully at our example, we note that our coding strategy uses $3/2$ bits on average for coding the first

symbol, while the entropy associated with the random variable X_1 is 2. For the following symbols, in turn, the entropies $H(X_i|X_{i-1})$ equal $3/2$ bits, and thus they have exactly the same value as the number of bits used by our code. So, we can say that our code only gains in the first symbol. But this fact is somehow interesting; our code assigns to the first symbol a number of bits smaller than its entropy, using the memory properties of the source, without affecting unique decodability. Thus, given that we are usually interested in coding a finite number of symbols, the problem of finding the optimal coding strategy arises.

It is also interesting to consider in this specific example the difference between the two above proposed coding technique from the point of view of computational complexity, and relate this discussion to what was said about using DSC techniques instead of predictive coding for exploiting the memory of sources. In particular, we know that the compressibility of the source is due to the fact that the conditional entropy $H(X_i|X_{i-1})$ is smaller than $H(X_i)$. Note now that the "Classic code", which essentially is a state-dependent Huffman code, actually fits with the "conditional entropy idea" in the sense that it really encodes each symbol given the preceding one. This implies that the encoder must trace the state of the source and choose the code for the new symbol, exactly in the same way as it done in predictive coding. On the contrary, the non prefix-free codeword assignment of our alternative code allows a very simple encoding phase, as there is a fixed mapping from symbols to code bits, with the same (even better, but not asymptotically) compression performance. The point is that we are making a different use of the decoder knowledge about possible transitions. Note that, even for the Huffman code, we are supposing that the decoder exactly knows which transitions are possible and which are not, as impossible transitions are not associated to any code. The difference is that with the alternative code we are making the decoder more active.

We would like to point out here that in practice the proposed approach was already used in other contexts that are also related to the use of DSC in coding sources with memory. One of the oldest examples seems to be that of modulo-PCM codes ([42]) for numerical sequences. In that case, given a numerical source with certain memory properties, only the modulo-4 value of every sample is encoded. The task of understanding the original value using the memory of the source is left to the decoder.

5.3 Unique decodability for constrained sequences

In this section we briefly survey the main definitions and theorems on uniquely decodable codes as presented in the literature and we then propose an adequate treatment of the case of constrained sequences by introducing a generalized Kraft inequality and a generalized Sardinas-Patterson test.

5.3.1 Classic results

We will consider [45] and [28] as representative references for what can be viewed as the classic approach to lossless source coding. In the above cited references the main steps in the

study of data compression can be considered to be approximatively the same. We summarize here the main classic definitions and theorems, adding comments and introducing the main variation on unique decodability definition, so as to give a precise idea of the collocation of the work presented in the next sections. We restrict our attention to finite alphabet sources in order to avoid unnecessary complication in the formulation, and we use the term *random variables* in a somehow improper way to indicate a finite set of symbols with associated probabilities. With this assumption in mind we can give the following definition of source.

Definition 5.3.1 *An information source X is a one-sided infinite sequence of random variables $X_1, X_2, X_3 \ldots$ taking values in a finite alphabet $\mathcal{X}$. The source X is said to be memoryless if $X_1, X_2, \cdots$ are independent and identically distributed (i.i.d.). Furthermore, we say that the source has memory if the random variables $X_1, \ldots, X_n$ are not independent.*

The following definitions are essentially taken from Cover [28].

Definition 5.3.2 *A variable-length code for a random variable X is a map from the (source) alphabet $\mathcal{X}$ to $\mathcal{D}^*$, the set of finite length sequences of symbols from a $\mathcal{D}$-ary (code) alphabet. For every $x \in \mathcal{X}$ let $C(x)$ be the codeword associated to x and let $l(x)$ be the length of $C(x)$.*

In order to have a code represent a random variable in a lossless way, it is necessary that different values of $\mathcal{X}$ are mapped to different codewords.

Definition 5.3.3 *A code is said to be non-singular if, for $x_i, x_j \in \mathcal{X}$,*

$$x_i \neq x_j \implies C(x_i) \neq C(x_j). \tag{5.8}$$

With this definition a non-singular code maps different values of $\mathcal{X}$ to different sequences of code symbols. We have defined an information source as a sequence of random variables, so that in general we are interested in coding sequences of source symbols, where the code for a sequence of symbols is generated by simply concatenating the code of the symbols one after the other. We need thus some more definitions. Cover [28] proceeds thus with the following definitions.

Definition 5.3.4 *The extension C^* of a code C is the mapping from finite length strings of $\mathcal{X}$ to finite length strings of $\mathcal{D}$ defined by*

$$C(x_1 x_2 \cdots x_n) = C(x_1)C(x_2)\cdots C(x_n), \tag{5.9}$$

where $x_1 x_2 \cdots x_n$ and $C(x_1)C(x_2)\cdots C(x_n)$ indicate respectively concatenation of source alphabet symbols and corresponding concatenation of the associated codes.

Definition 5.3.5 *A code is uniquely decodable if its extension is non-singular.*

We note at this point that the definition of unique decodability hides a very subtle issue. In fact, the given unique decodability definition requires every sequence of symbols from the alphabet $\mathcal{X}$ to be associated with a different sequence of symbols of the code alphabet $\mathcal{D}$. This is thus a definition of unique decodability for the code C without any reference to the particular source for which this code is used, but only to the alphabet of the source. In a sense, with this definition we are requiring the code to be uniquely decodable for the worst case scenario where the source at hand can indeed produce sequences that contain any possible combination of symbols from $\mathcal{X}$. It is instead clear from the example shown in Section 5.2 that there are sources, that we call *constrained sources*, that can produce only a proper subset of sequences from the set of all combinations of symbols of their alphabets. With respect to this point Gallager [45] gives a definition of unique decodability without explicitly using extensions of codes, and referring to *source sequences*:

Definition 5.3.6 *A code is uniquely decodable if for each source sequence of finite length, the sequence of code letters corresponding to that source sequence is different from the sequence of code letters corresponding to any other source sequence.*

Note that, as it is stated, this definition is fundamentally different from the definition given by Cover, because this one actually associates the unique decodability property with the particular source at hand. Unfortunately, Gallager gives this definition in a context where the sources are implicitly assumed to be memoryless. In this case, obviously, the source can produce any sequence of symbols and the definition reduces to be equivalent to the definition given by Cover.

At this point we consider important to introduce the first variation to the classic literature on unique decodability with the following two definitions.

Definition 5.3.7 *Let X be a discrete information source on the alphabet $\mathcal{X}$. We say that X is a constrained source if for at least one finite k there exists an element of $\mathcal{X}^k$ that cannot be obtained as outcome of the first k symbols of the source. Otherwise we say that the source is unconstrained.*

Definition 5.3.8 *Let X be an information source with alphabet $\mathcal{X}$. A code C is said to be uniquely decodable for the source X if no two different finite sequences of source symbols producible by X have the same codeword.*

With these definitions every code that is uniquely decodable in the classic sense is uniquely decodable for every source and every uniquely decodable code for an unconstrained source is uniquely decodable in the classic sense. Finally, in general, uniquely decodable codes for a constrained source are not uniquely decodable if we adopt the classic definition.

A particular class of codes that are uniquely decodable in the classic sense - and thus for any source - are the well known *prefix codes*. We sat that a word w_1 is a prefix of another word w_2 if w_2 is obtained by concatenating w_1 with an appropriate string of code symbols. For example, for binary codes, the word '011' is a prefix of '01101'.

Definition 5.3.9 *A code is called a prefix-code if no codeword is a prefix of any other codeword.*

These codes are also called *instantaneous* because, under the prefix condition, it is possible to decode a sequence of codewords one by one as soon as we receive them without having to wait for the end of the message. An example of a variable length prefix code was used by Shannon in [88] in order to prove the direct part of Theorem 9 on the average rate required to encode information sources. All prefix codes satisfy the so called Kraft inequality [60]:

Theorem 5.3.1 (Kraft Inequality) *Let $l_i, i = 1, \ldots, n$, be the lengths of the codewords of a prefix code and let D be the size of the code alphabet. Then*

$$\sum_{i=1}^{n} D^{-l_i} \leq 1 \tag{5.10}$$

Conversely, if a set of integers $l_i, i = 1, \ldots, n$, satisfies this inequality, the l_i are necessarily codeword lengths of a prefix code. ∎

This inequality plays an important part in the study of the average codeword length of codes. Using inequality (5.10), in fact, it is not difficult to show that if a prefix code is used for encoding a random variable X, then the average codeword length is not smaller than the base-D entropy $H_D(X)$, i.e. $E[l(X)] \geq H_D(X)$. If p_k are the probabilities of the symbols in $\mathcal{X}$ we have in fact [68]

$$\begin{aligned} H_D(X) - E[l(X)] &= \sum_k p_k \log_D \frac{1}{p_k} - \sum_k p_k l_k && (5.11)\\ &= \sum_k p_k \log_D \frac{D^{-l_k}}{p_k} && (5.12)\\ &\overset{(a)}{\leq} \sum_k p_k \left(\frac{D^{-l_k}}{p_k} - 1 \right) \log_D e && (5.13)\\ &\leq 0 && (5.14) \end{aligned}$$

where (a) is justified by the inequality $\log_D(x) \leq (x-1)\log_D e$. Consider now the use of prefix codes for the sequences of symbols of a source X. Take the first n symbols of the sequence; every possible outcome of these n symbols can be viewed as a single supersymbol in the alphabet $\mathcal{X}^n$. So, the average code length of a prefix code for the first n symbols of the sequence must be at least the entropy of the symbols. In other words we have the following result [28].

Theorem 5.3.2 *For every prefix code the average length for the first n symbols of a source X satisfies*

$$E[l(X_1, X_2, \ldots, X_n)] \geq H(X_1, X_2, \ldots, X_n) \tag{5.15}$$

This is a very strong result on the average code length of a prefix code, which strengthen the asymptotic results of Shannon to the case of a finite number of symbols. On the other hand, we know that there are codes that are uniquely decodable (in the classic sense) but are not prefix codes. For example, a binary code composed with the words '01' and '011' is not a prefix code, but the concatenation of words is always decodable, as a new word always start with a 0. So, the question of what is the minimum code length for a uniquely decodable code remains. Here a key role is played by McMillan's theorem [68].

Theorem 5.3.3 (McMillan [68]) *If a code C is uniquely decodable in the classic sense then the codewords length satisfy the Kraft inequality*

$$\sum_{i=1}^{n} D^{-l_i} \leq 1 \tag{5.16}$$

Commenting this result Elias says [41]"*There is, therefore, no advantage in either average codeword length or effective decipherability to be gained by using a uniquely decipherable set that is not a prefix set*". Both Gallager and Cover essentially conclude the same and use this theorem to state that every result on codeword lengths obtained for prefix codes hold true for uniquely decodable codes. In particular, it is asserted that, for every source, any uniquely decodable code must satisfy inequality (5.15).

It is important here to point out that with the definition of unique decodability given by Cover one actually have inequality (5.15) satisfied. The most important point, however, is that the set of uniquely decodable codes defined this way does not correspond to the idea of unique decodability one would expect, i.e. the fact that it is possible to decode every message. In this sense, Definition 5.3.8 makes more sense. Unfortunately, or surprisingly, using this definition we find that inequality (5.15) is no longer guaranteed for certain sources with memory, as shown by the example of Section 5.2. We propose thus the following theorem.

Theorem 5.3.4 *There exists at least one source X and a uniquely decodable code for X such that, for every $n \geq 1$,*

$$E[l(X_1, X_2, \ldots, X_n)] < H(X_1, X_2, \ldots, X_n). \tag{5.17}$$

It is clear at this point that this result is due to the fact that for certain constrained sources the Kraft inequality is not a necessary condition for a code to be uniquely decodabile for that given source. Indeed, the code used in the example of Section 5.2, which is composed of the words '0', '1', '01' and '10', clearly does not satisfy the Kraft inequality for binary alphabets. In conclusion, as the Kraft inequality is not a necessary condition for a code to be uniquely decodable for a constrained source, it is important to find a different necessary condition. In the next section we propose a modified Kraft inequality that gives a necessary condition for the case of constrained Markov sources.

5.3.2 Modified Kraft inequality

As we have already discussed in the preceding Section, the important result obtained by McMillan is that a necessary condition for the unique decodability (in the classic sense) of a set of n codewords is that their lengths $l_1, l_2, \ldots, l_n$ satisfy the Kraft inequality. Also, we have clarified that this condition is not necessary for a code to be uniquely decodable for a constrained source. In this section we proposed a modified Kraft inequality which constitutes a necessary condition for the case of first order constrained Markov sources, i.e. Markov sources that have the peculiarity of having some impossible transitions between symbols. Note that in the example of Section 5.2 the source is a constrained Markov source. We put the focus on Markov chains where the source symbols are associated to states, i.e. in the Moore form. We then show that the result is easily extended to the case of Markov sources in the Mealy form, i.e. when every transition is associated to an output symbol, with in general more than one possible transitions between every couple of states.

In order to give an easy presentation of our result consider first Karush's proof of Mcmillan theorem [55]. In his proof Karush uses an elegant trick. Consider the expression

$$\left(\sum_i D^{-l_i} \right)^k . \tag{5.18}$$

If we expand this power of a sum, we obtain a sum of n^k terms each of them being a product of factors D^{-l_i} in a different combination. The way the possible combinations of products are constructed is exactly the same as the way the symbols of the source are concatenated in all possible combinations to obtain sequences of k symbols. Every term in the expansion of (5.18) can thus be associated with a sequence of symbols. For example a sequence starting with $x_1, x_3, x_2, \ldots$ is associated to a term $D^{-l_1} D^{-l_3} D^{-l_2} \cdots$. Now, consider for a given r, all sequences of k symbols giving a total codeword of length r. These words are associated to terms of equal value D^{-r} in the expansion of (5.18). But if the code is uniquely decodable, there are at most D^r sequences giving a code of length r. So, the total contribution of those words in the expansion of (5.18) is at most 1. This fact holds true for every value of r up to the maximum possible code length obtainable with k symbols, that is for $r \leq k l_{\max}$, where $l_{\max}$ is the largest of the l_i. Summing up for all values of r from 1 to

$kl_{\max}$ we cover all possible terms in the expansion, so that we have

$$\left(\sum_i D^{-l_i}\right)^k \leq k\, l_{\max}. \tag{5.19}$$

This inequality must hold for every k. But the right hand side is a linear function of k, while the left hand side is an exponential function and thus, for large enough k, it would exceed any linear function if the base was larger than 1. This leads to the conclusion that $\sum_i D^{-l_i} \leq 1$.

In this proof of McMillan result, by considering expression (5.18) we have implicitly assumed that we are required to distinguish between every possible combination of symbols or, in other words, that the code is uniquely decodable in the classic sense. If the source is constrained, instead, we should only consider the possible combination output by the source.

Let us consider again as an example the source of Figure 5.2, with the binary alphabed (i.e., $\mathcal{D} = \{0, 1\}$) and with l_1, l_2, l_3 and l_4 as the lengths assigned respectively to A, B, C and D. In this case, the terms in the expansion of the left hand side of (5.19) that contains for example $\cdots 2^{-l_1} 2^{-l_2} \cdots$ or $\cdots 2^{-l_1} 2^{-l_4} \cdots$ should not be considered, as A is never followed by B nor D in a source sequence. Let us consider now the matrix

$$\mathbf{Q} = \begin{bmatrix} 2^{-l_1} & 0 & 2^{-l_1} & 0 \\ 0 & 2^{-l_2} & 0 & 2^{-l_2} \\ 2^{-l_3} & 2^{-l_3} & 2^{-l_3} & 2^{-l_3} \\ 2^{-l_4} & 2^{-l_4} & 2^{-l_4} & 2^{-l_4} \end{bmatrix}, \tag{5.20}$$

which is obtained from the transition probability matrix of the source by replacing every non-zero term in the i-th row with 2^{-l_i}. It is not difficult to verify that the really necessary correspondent of eq. (5.19) for our source should be written, for $k > 0$, as

$$\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \mathbf{Q}^{k-1} \begin{bmatrix} 2^{-l_1} \\ 2^{-l_2} \\ 2^{-l_3} \\ 2^{-l_4} \end{bmatrix} \leq k\, l_{\max} \tag{5.21}$$

It is possible to show (see hereafter) that a necessary condition for this inequality to be satisfied for every k is that the matrix $\mathbf{Q}$ has spectral radius[1] at most equal to 1. We present the result in the general form.

Theorem 5.3.5 *Let* $\mathbf{P}$ *be an irreducible* $n \times n$ *stochastic matrix representing the transition probabilities of a Markov chain and* $\mathbf{l} = [l_1, l_2, \ldots, l_n]$ *a vector of* n *integers. Let* $\mathbf{Q}$ *be the* $n \times n$ *matrix such that*

$$\mathbf{Q}_{ij} = \begin{cases} 0 & \text{if} \quad P_{ij} = 0 \\ D^{-l_i} & \text{if} \quad P_{ij} > 0 \end{cases} \tag{5.22}$$

[1] The spectral radius of a matrix is defined as the greatest modulus of its eigenvalues.

Then, a necessary condition for the codeword lengths l_1, l_2, ..., l_n to be lengths of a uniquely decodable D-ary code for a Markov source with transition probability matrix $\mathbf{P}$ is that $\rho(\mathbf{Q}) \leq 1$, where $\rho(\mathbf{Q})$ is the spectral radius of $\mathbf{Q}$. ∎

Proof. We follow Karush's proof of McMillan theorem. Suppose without loss of generality that the set of our source symbols is $\mathcal{X} = \{1, 2, \ldots, n\}$, and call $\mathcal{X}^{(k)}$ the set of all sequences of k symbols that can be produced by the source. Let us set, for convenience of notation, $\mathbf{L} = [D^{-l_1}, D^{-l_2}, \ldots, D^{-l_n}]$ and define, for $k > 0$,

$$\mathbf{V}_k^T = \mathbf{Q}^{k-1}\mathbf{L}^T. \tag{5.23}$$

Then it is easy to see by induction that the i-th component of $\mathbf{V}_k$ is written as

$$\mathbf{V}_k^i = \sum_{x_1, x_2, \ldots, x_k} D^{-l_{x_1} - l_{x_2} \cdots - l_{x_k}} \tag{5.24}$$

where the sum runs over all elements $(x_1, x_2, \ldots, x_k)$ of $\mathcal{X}^{(k)}$ with varying $x_2, x_3, \ldots, x_k$ and $x_1 = i$. So, if we call $\mathbf{1}_n$ the row vector composed of n 1's, we have

$$\mathbf{1}_n \mathbf{Q}^{k-1}\mathbf{L}^T = \sum_{x_1, x_2, \ldots, x_k} D^{-l_{x_1} - l_{x_2} \cdots - l_{x_k}} \tag{5.25}$$

where the sum now runs over all elements of $\mathcal{X}^{(k)}$. Thus, reindexing the sum with respect to the total length $r = l_{x_1} + l_{x_2} + \cdots + l_{x_k}$ and calling $N(r)$ the number of sequences of $\mathcal{X}^{(k)}$ to which a code of length r is assigned, we have

$$\mathbf{1}_n \mathbf{Q}^{k-1}\mathbf{L}^T = \sum_{r=kl_{\min}}^{kl_{\max}} N(r) D^{-r} \tag{5.26}$$

where $l_{\min}$ and $l_{\max}$ are respectively the minimum and the maximum of the values $l_i, i = 1, 2, \ldots, n$. Since the code is uniquely decodable, there are at most D^r sequences with a code of length r. This implies that, for every $k > 0$, we must have

$$\mathbf{1}_n \mathbf{Q}^{k-1}\mathbf{L}^T \leq \sum_{r=kl_{\min}}^{kl_{\max}} D^r D^{-r} = k(l_{\max} - l_{\min} + 1) \tag{5.27}$$

Now, note that the irreducible matrix $\mathbf{Q}$ is also nonnegative. Thus, by the Perron-Frobenius theorem (see [72] for details), its spectral radius $\rho(\mathbf{Q})$ is also an eigenvalue[2], with algebraic multiplicity 1 and with positive associated eigenvector. Let $\mathbf{w}^T$ be this eigenvector; then,

[2]Note that in general the spectral radius is not an eigenvalue as it is defined as the maximum of $|\lambda|$ over all eigenvalues λ.

as $\mathbf{L}^T$ is positive, there exists a maximal positive constant α such that $\mathbf{L}^T = \alpha \mathbf{w}^T + \mathbf{z}^T$, where $\mathbf{z}^T$ is a nonnegative vector. Thus, we can write the left hand side of (5.27) as

$$\begin{aligned} \mathbf{1}_n \mathbf{Q}^{k-1} \mathbf{L}^T &= \mathbf{1}_n \mathbf{Q}^{k-1} \alpha \mathbf{w}^T + \mathbf{1}_n \mathbf{Q}^{k-1} \mathbf{z}^T \\ &= \alpha \rho(\mathbf{Q})^{k-1} \mathbf{1}_n \mathbf{w}^T + \mathbf{1}_n \mathbf{Q}^{k-1} \mathbf{z}^T \\ &= \beta \rho(\mathbf{Q})^{k-1} + \gamma \end{aligned}$$

where $\beta = \alpha \mathbf{1}_n \mathbf{w}^T$ is positive and γ is nonnegative. So, if $\rho(\mathbf{Q}) > 1$, the term on the left hand side of eq. (5.27) asymptotically grows at least as $\rho(\mathbf{Q})^{k-1}$. On the contrary, the right hand side term only grows linearly with k and for large enough k equation (5.27) could not be verified. We conclude that $\rho(\mathbf{Q}) \leq 1$. ∎

We note that if the $\mathbf{P}$ matrix has all strictly positive entries, the matrix $\mathbf{Q}$ is positive with all equal columns. It is known (see again [72]) that the spectral radius of such a matrix is given by the sum of the elements in a column, which in this case is $\sum D^{-l_i}$. Thus, for non-constrained sequences, we obtain the classic Kraft inequality. Furthermore, as the spectral radius of a nonnegative positive matrix increases if any of the elements increases, we deduce that the case when $\rho(\mathbf{Q}) = 1$ correspond to an extreme situation in terms of $\mathbf{P}$ and $\mathbf{l}$. In the sense that if for a given matrix $\mathbf{P}$ there is a decodable code with codeword lengths $l_i, i = 1, \ldots, n$ such that $\rho(\mathbf{Q}) = 1$, then there is no decodable code with lengths l'_i if $l'_i \leq l_i$ for all i with strict inequality for some i. Also, for the same codeword lengths, it is not possible to remove constraints from the Markov chain while keeping unique decodability property.

The most important remark, however, concerns the non sufficiency of the stated condition. In fact, while the classic Kraft inequality is a necessary and sufficient condition for the existence of a uniquely decodable code for an unconstrained sequence, the found inequality $\rho(\mathbf{Q}) \leq 1$ is unfortunately only necessary, and not sufficient. We discuss this point in the next section, where we propose an extension of the Sardinas Patterson test for testing the unique decodability of a code for a constrained sequence.

The above presented discussion is focused on the case of constrained sources that are modeled with Markov chains in the Moore form, as considered for example in [28]. In other words, we have modeled information sources as Markov chains by assigning an output source symbol to every state. This way we have considered only sources that have a memory of one symbol, because transitions in the Markov chains are always considered to be independent. In order to deal with more general sources one can consider memory of higher order and model the source with higher order Markov chains. An elegant treatment of finite memory sources can be constructed by considering a source as a finite state non-deterministic machine where output symbols are associated to transitions between states rather than to states. This corresponds to the Markov source model as described for example in [45]. As in this case the output symbols are associated to transitions, we can informally refer this model as a Markov model in the Mealy form.

As an example, consider the source used in the preview example, whose Moore form is represented in Figure 5.2. We can model this source with only three states using a Mealy

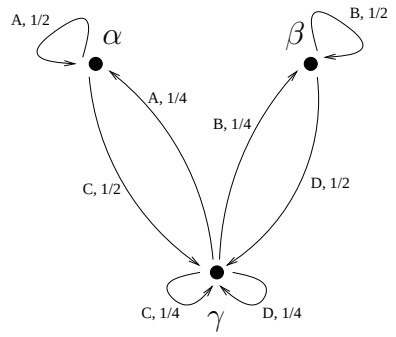


Figure 5.3: Markov chain, in the Mealy form, associated to the source of figure 5.2. Here every arc is labeled with the associated output symbol and the probability of the transition.

representation (see Figure 5.3); the source is in state α if the last output symbol is an A, it is in state β if the last output symbol is a B, and it is in state γ if the last output symbol is a C or a D. Then, symbols are output at the transitions from one state to the other.

Now note that once the source is represented in the Mealy form, it may be interesting to consider coding techniques that associate different codewords to the same symbol depending on the state of the source. In our toy example this would include as a particular case the encoding technique that we have indicated as "Classic Code" in Section 5.2. In our study, anyway, we are not really interested in finding such "adaptive" codes, but rather on the use of "memoryless" codes for coding sources with memory. Thus, even if the Mealy form has the advantage of allowing an easy representation of state-dependent codes, we are primarily interested in studying the bounds obtainable when the code associated to a given symbol is kept fixed, regardless of the particular state of the source. This is coherent with the aim of modeling simple encoders that use singular codes in order to compress a source with memory.

Keeping in mind the explained setting, as we have anticipated at the beginning of the Section, Theorem 5.3.5 can easily be adapted to the case of sources modeled as Markov chains in the Mealy form.

Theorem 5.3.6 (Mealy form) *Let $S_1, S_2, \ldots, S_s$ be s possible states of a source X with alphabet $\mathcal{X} = \{1, 2, \ldots, n\}$, and let $O_{i,j}$ be the subsets of $\mathcal{X}$ of possible symbols output by the source when transiting from state S_i to state S_j. Then, a necessary condition for the set of integers $l_1, l_2, \ldots, l_n$ to be lengths of a uniquely decodable code for the source X is that $\rho(\mathbf{Q}) \leq 1$, where $\mathbf{Q}$ is the matrix defined by*

$$\mathbf{Q}_{i,j} = \begin{cases} 0 & \text{if} \quad P_{ij} = 0 \\ \sum_{v \in O_{i,j}} D^{-l_v} & \text{if} \quad P_{ij} > 0. \end{cases} \tag{5.28}$$

Here $P_{ij} = 0$ means that a transition from state S_i to state S_j is impossible, and $P_{ij} > 0$ means that there is at least one possible transition from S_i to state S_j, i.e. the set $O_{i,j}$ is

non-empty. ∎

Proof. The proof is essentially equivalent to the proof of Theorem 5.3.5 with only small changes for the new setting. ∎

In order to make clear the statement of the theorem, it is interesting to note how it is applied to the particular case of the source used in the preview example. Note that, in the case of a binary code, the Moore form show graphically in Figure 5.2 lead to the matrix $\mathbf{Q}$ as defined in equation (5.20). If we use instead the Mealy representation of Figure 5.3, the matrix $\mathbf{Q}$ defined in Theorem 5.3.6, for the case of a binary code, is a 3×3 matrix, namely

$$\mathbf{Q} = \begin{bmatrix} 2^{-l_1} & 0 & 2^{-l_3} \\ 0 & 2^{-l_2} & 2^{-l_4} \\ 2^{-l_1} & 2^{-l_2} & 2^{-l_3} + 2^{-l_4} \end{bmatrix}. \tag{5.29}$$

The theorem says that the spectral radius of this matrix has to be not larger than 1 in order to have a decodable code for our source. Noticeably, this matrix has not only the same spectral radius, but the same whole spectrum[3] of the matrix defined in (5.20).

As a further example, we can consider what happens in the case of unconstrained sequences. Consider for example an unconstrained source with an alphabet of n symbols. In this case, from the point of view of the possible combination of output symbols, we can model the source with only one state S_1, every symbol being a possible output when moving form state S_1 to itself. The matrix $\mathbf{Q}$ defined in Theorem 5.3.6, for a D-ary code, is in this case a 1×1 matrix, i.e. a scalar value, which equals $\sum_i D^{-l_i}$. So again we obtain the classic Kraft inequality.

5.3.3 Extended Sardinas-Patterson test

In the preceding sections we have shown that the classic Kraft inequality is not, in general, a necessary condition for the unique decodability of a constrained sequence, and we have found a necessary condition under this hypothesis. Unfortunately, the found condition is not sufficient as can be easily shown by means of trivial examples. We consider here only the case of Markov sources modeled in the Moore form for simplicity. Note that the only entities determining the matrix $\mathbf{Q}$ are the length vector $\mathbf{l}$ and the graph associated to the Markov chain, i.e. the state pairs with positive transition probability. Thus, we only consider the transition graphs of the sources without taking into account the value of the transition probabilities. From now on, furthermore, we only consider binary codes for simplicity, without any loss of generality with respect to the general case of D-ary sources.

Consider a source with three symbols A, B and C with transition graph as shown in fig. 5.4(a). It is easy to see that if $\mathbf{l} = [1, 1, 1]$ then $\rho(\mathbf{Q}) = 1$; anyway, it is clearly impossible

[3]The same spectrum intended as the set of eigenvalues. The only difference between the matrix defined in (5.20) and the one in 5.29 is that in the first one the null eigenvalue has multiplicity 2 rather than 1.

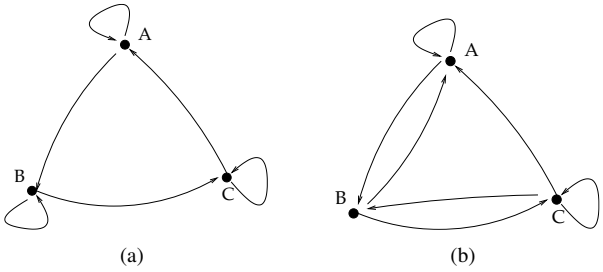


Figure 5.4: Two examples of transition graphs for which $\rho(\mathbf{Q}) \leq 1$ is not a sufficient condition.

to decode the sequences of the source if we assign only one bit to every symbol. In general, we may consider that if the initial state is distributed with positive probability on every symbol, it is not possible to have a decodable code with more than 2^i codewords of length i, since otherwise even the initial state cannot be recovered. Anyway, imposing this additional condition is not sufficient. Take for example a code with $\mathbf{l} = [1, 1, 2]$ for a source with transition graph as shown in fig. 5.4(b); we have $\rho(\mathbf{Q}) < 1$, only two codewords of 1 bit and one codeword of 2 bits, but still a decodable code with those lenghts does not exist. In fact, if we assign for example $A \rightarrow 0$, then we must assign $B \rightarrow 1$ and consequently $C \rightarrow 11$. But so, BCB and CC have the same code.

The above examples show that the question of finding a sufficient condition on the word lengths for the existence of a uniquely decodable code for a constrained sequence appears to be more complicated than with unconstrained sequences. A positive fact is that it is possible to extend the Sardinas Patterson (SP) test [83], originally developed for unconstrained sequences, to the case of our interest of constrained ones. Given a set of codewords, the SP test allows to establish in a finite number of steps if the code is uniquely decodable in the classic sense. Here we modify the classic algorithm for the case of constrained sequences. The generalization is straightforward and we do not give here a formal proof of the correctness, as it would merely be a rewriting of that for the classic SP test, for which we refer the reader to [13, th. 2.2.1].

Suppose our source symbol set is $\mathcal{X} = \{1, 2, \ldots, n\}$ and let us call $W = \{W_i\}_{i=1,\ldots,n}$ the set of associated codewords. For $i = 1, 2, \ldots, n$ we call $F_i = \{W_j | P_{ij} > 0\}$ the subset of W containing all codewords that can follow W_i in a source sequence. We construct a sequence of sets $S_1, S_2, \ldots$ in the following way. To form S_1 we consider all pairs of codewords of W; if a codeword W_i is a prefix of another codeword W_j, i.e. $W_j = W_i A$ we put the suffix A into S_1. In order to consider only the possible sequences, we have to keep trace of the codewords that have generated every suffix; thus, let us say that we mark the obtained suffix A with the two labels i and j, and we thus write it as ${}_iA_j$. We do this operation for every pair of words W_i and W_j from W, i.e. for $i, j = 1, \ldots, n$. Then, for $n > 1$, S_n is constructed by comparing elements of S_{n-1} and elements of W. For a generic

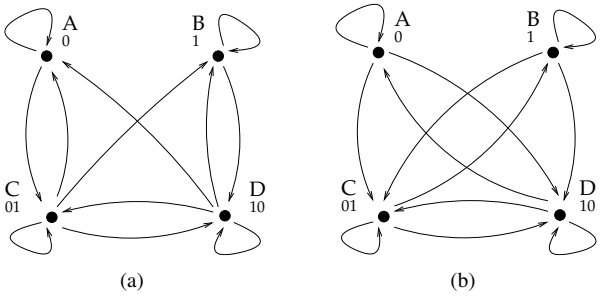


Figure 5.5: Two examples of transition graphs for which $\rho(\mathbf{Q}) \leq 1$ is a sufficient condition.

element ${}_lB_m$ of S_{n-1} we consider the subset F_l of W:

a) If a codeword $W_k \in F_l$ is equal to ${}_lB_m$ the algorithm stops and the code is not decodable;

b) if ${}_lB_m$ is a prefix of a codeword $W_r = {}_lB_mC$ we put the labelled ${}_mC_r$ suffix into S_n;

c) if instead a codeword W_s is prefix of ${}_lB_m = W_sD$, we place the labelled suffix ${}_sD_m$ into S_n.

The code is uniquely decodable if and only if item a) is never reached.

Note that the algorithm can be stopped after a finite number of steps; there are in fact only a finite number of possible different sets S_i and so the sequence $S_i, i = 1, 2, \ldots$ is either finite (i.e., the S_i are empty sets from sufficiently high i) or periodic. We note that the code is uniquely decodable with *finite delay* if the sequence S_i is finite and uniquely decodable with *infinite delay* if the sequence is periodic. In this case the code is still decodable, since finite strings of code symbols can always be uniquely decoded, but the required delay is not bounded. This means that, for any positive n, there are at least two source sequences that produce codes that require more than n symbols delay in order to be disambiguated.

As an example of SP test for constrained sequences we consider the transition graphs shown in fig. 5.5. For both cases we use codewords 0, 1, 01 and 10 for A, B, C and D respectively. For the graph of fig. 5.5(a) we obtain $S_1 = \{{}_A1_C, {}_B0_D\}$, $S_2 = \emptyset$. Thus the code is finite delay uniquely decodable and we can indeed verify that we need to wait at most two bits for decoding a symbol (this code is indeed the code used for the example of Section 5.2). For the graph of fig. 5.5(b), instead, we have $S_1 = \{{}_A1_C, {}_B0_D\}$, $S_2 = \{{}_C0_D, {}_D1_C\}$ and then $S_i = S_2$ for every other $i \geq 3$. So, the code is still uniquely decodable but with infinite delay; in fact it is not possible to distinguish the sequences $ADDD\cdots$ and $CCC\cdots$ until they are finished, so that the delay may be as long as we want.

5.4 On unique decodability and related topics

5.4.1 Counting methods, McMillan's theorem and a proof by Shannon

In this section we want provide an analysis of McMillan's theorem from an historical point of view, comparing different proofs and in particular by showing that both the original proof by McMillan [68] and Karush's one [55] are essentially equivalent to a proof by Shannon [88] for the evaluation of the capacity of certain channels. In a sense, we can say that McMillan theorem was "almost" already proved in Shannon's paper. Even more interestingly, also our modified Kraft inequality was almost already present in Shannon's paper hidden in the evaluation of the capacity of finite state channels such as the telegraph [88].

Consider first the original proof by McMillan of his own theorem [68]. Let $l_{\max}$ be the maximum of the lengths $l_1, l_2, \ldots, l_n$ and let $w(r)$ the number of words of length r; the Kraft inequality can thus be written as

$$\sum_{r=1}^{l_{\max}} w(r) D^{-r} \leq 1. \tag{5.30}$$

Let then $Q(x)$ be the polynomial defined by

$$Q(x) = \sum_{r=1}^{l_{\max}} w(r) x^r. \tag{5.31}$$

The proof is based on the study of $Q(x)$ as a function of a complex variable x and leads to a stronger result than the Kraft inequality, namely to the result that $Q(x) - 1$ has no zeros in the circle $|xD| < 1$ of the complex plane. As $Q(x)$ is continue and monotone for real $x \geq 0$ the Kfraft inequality easily follows. By removing from the original proof the parts that are not strictly important for the proof of the simple Kraft inequality, we obtain approximately the following flow. Let $N(k)$ be, as before, the number of sequences of source symbols whose code has total length k. As the code is uniquely decodable, there are at most D^k such sequences, i.e., $N(k) \leq D^k$. It is thus clear that the series $1 + N(1)x + N(2)x^2 + \cdots$ converges for values of $x < D$; let $F(x)$ be the sum of the series. Now, the fundamental step in the proof is to consider how the possible $N(k)$ sequences of k letters are obtained. McMillan uses the following reasoning. Let C_r be the set of sequences of length k with a first word of length r; these sets are disjoint because of the unique decodability. For the first r letters of $C(r)$ there are exactly $w(r)$ different possibilities, the number of words or r letters, while for the remaining $k - r$ letters there are exactly $N(k - r)$ different combinations. So, we have

$$N(k) = w(1)N(k-1) + w(2)N(k-2) + \cdots + w(l_{\max})N(k - l_{\max}) \tag{5.32}$$

The above equation holds for every k if one defines $N(r) = 0$ for negative r. Now, take $x < 1/D$, multiply the above equation by x^k and sum for $k = 1$ to infinity. We have

$$F(x) - 1 = F(x)Q(x). \tag{5.33}$$

But as $F(x)$ is positive, $Q(x)$ must be smaller than one. By continuity one clearly see that $Q(1/D)$ is at most 1, which is Kraft inequaliy.

It is interesting to focus the attention on the key point of this proof, which is essentially the combination of eq. (5.32) with the requirement that $N(k) \leq D^k$. In particular it is implicitely established that the value of $Q(1/D)$ determines how fast $N(k)$ grows, and thus if it is possible to have $N(k) \leq D^k$ asymptotically or not.

This basic idea is also used in the proof given by Karush, but in an easier way. Instead of considering the set of code strings of length k, Karush considers the sequences of k symbols of the source as explained in the previous section. After an accurate analysis it is not difficult to realize that the proof given by Karush "only" has the advantage of relating the asymptotic behavior[4] of the sum $1 + N(1)D^{-1} + N(2)D^{-2} + ..N(kl_{\max})D^{-kl_{\max}}$ to the value of $Q(1/D)$ in a more direct way. Thus, the two proofs both use the convergence or the order of magnitude of the sum $1 + N(1)D^{-1} + N(2)D^{-2} + \cdots$ in order to study the asymptotic behaviour of $N(k)$. We could then say that both proofs are based on a combinatorial *counting method* for the evaluation of $N(k)$ and by imposing the constraint that $N(k) \leq D^k$

It is interesting to find that the very same technique had already been used by Shannon in Part I, Section 1 of [88] while computing the capacity of discrete noisless channels. Shannon considers a device which is used to communicate symbols over a channel and wants to study the number of messages that can be communicated per instant time. He says: "*Suppose all sequences of the symbols $S_1, \ldots, S_n$ are allowed and these symbols have durations $t_1, \ldots, t_n$. What is the channel capacity? If $N(t)$ represents the number of sequences of duration t we have*

$$N(t) = N(t - t_1) + N(t - t_2) + \cdots + N(t - t_n). \tag{5.34}$$

The total number is the sum of the numbers of sequences ending in $S_1, S_2, \ldots, S_n$ and these are $N(t - t_1), N(t - t_2), \ldots, N(t - t_n)$, respectively. According to a well known result in finite differences, $N(t)$ is then asymptotic for large t to X_0^t where X_0 is the largest real solution of the characteristic equation:

$$X^{-t_1} + X^{-t_2} + \cdots + X^{-t_n} = 1 \tag{5.35}$$

and therefore

$$C = \log X_0 \tag{5.36}$$

It is not difficult to note that the result obtained by Shannon, if reinterpreted in a source coding setting, is essentially equivalent to McMillan theorem. Indeed, suppose the device considered by Shannon is a discrete time device, emitting a symbol from a D-ary alphabet in every time instant, so that the symbols $S_1, S_2, \ldots, S_n$ are just D-ary words. First note that Shannon's tacit assumption is that the device produces messages that can be decoded at the receiving point. We can then thus rewrite this implicit assumption by saying that symbols

[4]More precisely, in the expansion of (5.18) the coefficient of D^{-r} is, in general, smaller than $N(r)$ for values of r larger than $r/l_{\min}$, but this does not affect the asymptotic behavior of the sum for large k.

$S_1, S_2, \ldots, S_n$ form a uniquely decipherable code. Let us now focus on the capacity of the considered device. As the device sends one symbol from a D-ary alphabet at every instant, it is clear, and it was surely obvious for Shannon, that the channel capacity is in this case at most $\log D$. This means that the obtained value of X_0 above satisfies $X_0 \leq D$. But X_0 is a solution to (5.35), and the left hand side of (5.35) is nonincreasing in X. So, setting $X = D$ in (5.35), the Kraft inequality is easily deduced.

In other words, McMillan's theorem was already "proved" in the Shannon paper, but it was not explicitly stated in the source coding formulation. It is clear that the formulation in the source coding setting, rather than in the channel coding one, is of great importance by its own from an information theoretic point of view. From the mathematical point of view, instead, it is very interesting to note that MacMillan proof is only a more rigorous and detailed description of the counting argument used by Shannon. Mathematically speaking, we can say that not only Shannon had already proved McMillan result, but that he had proved it in few lines, in a simple and elegant way, using exactly the same technique used by McMillan.

Now, note that Shannon did not state the above result as a theorem. In fact, he considered the result only as a particular case, used as an example. He indeed started the discussion with the clarification *Suppose all sequences of the symbols $S_1, \ldots, S_n$ are allowed*, because his main interest was in the general case where the sequences of symbols are produced with some given constraints, as for example in the case of the detailed study of the telegraph in that Section of his paper. The model used by Shannon for constraints is the following. *"We imagine a number of possible states $a_1, a_2, \ldots, a_m$. For each state only certain symbols from the set $S_1, S_2, \ldots, S_n$ can be transmitted [...]. When one of these has been transmitted the state changes to a new state depending both on the old state and the particular symbol transmitted"*. Note that this is exactly the type of constraint that we have indicated as a Markov model in the Mealy form, earlier in this chapter. The general result obtained by Shannon and stated as Theorem 1 in [88] is the following

Theorem 5.4.1 (Shannon) *Let $b_{ij}^{(s)}$ be the duration of the s^{th} symbol which is allowable in state i and leads to state j. Then the channel capacity C is equal to $\log W_0$ where W_0 is the largest real root of the determinant equation:*

$$\left| \sum_s W^{-b_{ij}^{(s)}} - \delta_{ij} \right| = 0. \tag{5.37}$$

∎

This theorem is well known in the field of coding for constrained systems (see for example [63]) and can be considered as the channel coding precursor of the Mealy-form of Theorem 5.3.6 exactly in the same way as the result obtained by eqs. (5.35) and (5.36) is the precursor of McMillan theorem. We now prove that Theorem 5.4.1 can indeed be used to deduce Theorem 5.3.6. We prove this fact by showing that, if the matrix $\mathbf{Q}$ defined in Theorem 5.3.6 has spectral radius larger than 1, then the associated code cannot be

uniquely decodable. In order to do that, we show that if such a code was decodable, then we could construct a channel using a D-ary alphabet with a capacity larger than $\log D$, which is clearly impossible.

Let $\mathbf{Q}(1/W) = \sum_s W^{-b_{ij}^{(s)}}$ be the matrix considered in the determinant equation (5.37). Note that this matrix $\mathbf{Q}(1/W)$ reduces to be the matrix $\mathbf{Q}$ introduced in Theorem 5.3.6 if we set $W = D$, i.e. $\mathbf{Q}(1/D) = \mathbf{Q}$. Suppose now that there exists a uniquely decodable code for a constrained source such that the spectral radius of the matrix $\mathbf{Q}$ in Theorem 5.3.6 is larger than 1. Then, as the code is uniquely decodable, we can construct a discrete-time D-ary channel with channel symbols exactly equal to the codewords of the given code. Then for this channel, with the above definitions, we have $\rho(\mathbf{Q}(1/D)) > 1$. Consider now the capacity of such a channel. The largest solution W_0 of the determinant equation (5.37) can also be considered as the largest positive value of W such that $\mathbf{Q}(1/W)$ has an eigenvalue equal to 1. Consider thus the largest eigenvalue of $\mathbf{Q}(1/W)$, i.e. the spectral radius $\rho(\mathbf{Q}(1/W))$. As the spectral radius of a nonnegative matrix decreases if any of the elements of the matrix decreases, $\rho(\mathbf{Q}(1/W))$ is a decreasing function of W. Then clearly, as $\rho(\mathbf{Q}(1/D)) > 1$, there exists a $W > D$ such that $\rho(\mathbf{Q}(1/W)) = 1$. But this means that we have constructed a D-ary channel with capacity larger than $\log D$, which is clearly impossible. So, the initial hypothesis was wrong, and thus any decodable code for a constrained source is such that the spectral radius of the matrix $\mathbf{Q}$ in Theorem 5.3.6 is not larger than 1.

This shows that the results obtained by Shannon for the channel capacity evaluation in his paper [88], actually correspond to very interesting results in the source coding setting, which hide a generalized form of Kraft-McMillan theorem.

Chapter 6

Remote Image Registration

Change the viewpoint.
Look at it from every possible angle.
– Claude Shannon –

6.1 Introduction

In Chapters 2-4 we have presented the DSC paradigm and its application to video coding in what is known as DVC. As we have clarified, most of the work has been done on the problem of single source video coding, but even the problem of multiple video sources has been recently considered as a relevant application (see for example [91]). For the problem of multiple sources, the case of distributed coding of still images has probably received more attention, and interesting contribution can be found in [95] and [114] as extensions of the PRISM and Stanford architectures. A different perspective has been adopted instead in [47, 49], where the the idea of coding in a distributed fashion the positions of objects in images taken by different views, or the structure of the quadtree decomposition of those images, has been investigated.

In Chapter 5, by taking into account the case of single source DVC, a study of an abstract model for the use of DSC-like codes for sources with memory has been proposed. In this chapter, instead, we want to consider another aspect of DVC which is more related with the problem of the creation of side information at the decoder or, more precisely, with the estimation at the decoder of the correspondences between frames or portions of frames. A fundamental problem encountered in both fields of distributed image and video coding, in fact, is the need of performing compensations at the decoder. In the case of single camera video sequences, for example, a classic non-distributed coding technique consists in applying motion compensation first and then encoding the resulting prediction error. In a distributed system the motion compensation must be performed at the decoder, without

[0]This chapter includes research results to appear in [36].

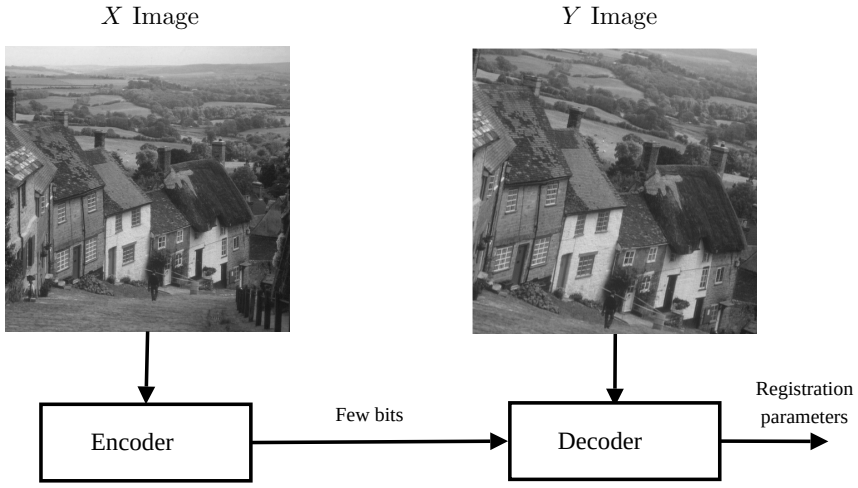


Figure 6.1: Basic idea of Remote Image Registration.

having access to an optimally predicted frame. A similar fact holds in the case of multiple sources, where the problem of disparity compensation at the decoder is the equivalent of the motion compensation of the single source case. This problem has been identified at an abstract level in Section 4.2.2, where the general structure of DVC schemes has been considered. There, the idea of sending some description $D(X)$ of a frame X that would allow the decoder to extract an approximation Y_e of X from the prior side information Y_p has been proposed. In this chapter we study and design an example of such descriptions for the case where the approximation Y_e of X must be created at the decoder side by properly applying shift, rotation, and scale operations to a prior side information frame Y, i.e., by applying a (similarity) registration operation.

So, in this chapter, we do not focus on the problem of correcting the extracted side information Y_e to obtain the original frame X, but we focus instead on the problem of finding a proper description of the frame X (at a very low rate) that can be used to perform a registration at the decoder between the two images Y and X. We call this problem "Remote Image Registration", as we consider it to be a self contained problem, which is clearly motivated by DVC techniques, but may find interesting applications in other different types of distributed problems.

The problem that we consider can thus be summarized in the following way (see Figure 6.1). Two images X and Y are obtained by cropping a common scene, with possible relative shift, rotation and scale between the two cropping operations. Supposing that the Y image is available at the decoder, we want to find an efficient strategy for communicating to the decoder the shift, rotation and scale parameters of the image X with respect to Y. This problem is decomposed in several phases using a Discrete Fourier Transform (DFT)

representation. The simpler sub-problem of shift compensation is first considered, and a detailed technique for the "distributed coding of shifts" is designed. Then, the problems of distributed coding of rotation and scale are reduced to the previously studied shift problem by proper transform operations, and the technique developed for the shift coding can thus be applied one again. This shows in a way that the distributed coding of shift is the really fundamental problem. Furthermore, we would like to clarify here that the shift registration is probably of much higher importance than the compensation of rotation and scale also with respect to a number of applications. For example, in a video coding system, the motion compensation operation can be seen as an extreme application of a shift registration technique at the block level, while rotation and scale are usually not considered even in predictive techniques. So, in a practical DVC systems, we may try to use block-level distributed coding of shifts to operate motion compensation at the decoder, but it would be much more difficult to imagine of rotations and scale to be of some interest in that case. For this reason, we will analyze with a greater detail the problem of shift coding and we will devote a less detailed study to the rotation and scale extension.

In the whole chapter we use the following notations: '$\log(\cdot)$' is the base-2 logarithm; for an integer m, '$\{\cdot\}_m$' indicates the modulo-m operation; the symbol '$\overset{2\pi}{=}$' indicates a modulo-2π congruence and we consider phases always to take values on the interval $[-\pi, \pi]$.

6.2 Distributed coding of shifts

In this section we develop a technique for the distributed coding of shifts in order to solve the target application of detecting, at the decoder, the relative shift of a remote image X with respect to the available side information image Y. We first study an "ideal" one-dimensional problem, where shifts are circular and noiseless signals are used, introducing the main idea of extraction of meaningful information from the DFT phase. Then, we extend the study to the case of 2-dimensional signals and we consider more concrete scenarios where shifts are not circular and images are affected by noise.

6.2.1 One-dimensional problem

Suppose we have two N-point signals $X(\cdot)$ and $Y(\cdot)$ which differ only by a circular shift s, with $0 \leq s < S$, $S < N$, i.e.:

$$X(n) = Y(\{n - s\}_N), \quad n = 0, 1, \ldots, N - 1. \tag{6.1}$$

For the sake of simplicity, let us consider the case when both N and S are powers of 2. Suppose an encoder has to communicate X to a decoder, using Y as side information. If Y is available to both encoder and decoder, and if s is uniformly distributed between 0 and $S - 1$, then $\log(S)$ bits are sufficient for encoding X, as it is actually only necessary to specify the value of s. Suppose now Y is only available to the decoder and not to the encoder. Supported by distributed source coding theory, one may wonder whether it is still

possible to encode X - or equivalently s - using only $\log(S)$ bits. We now prove that this is indeed possible and, in addition, that this can be done in different ways.

First note that if the shape of X and Y is *a priori* known to both encoder and decoder, then the problem is quite trivial. It is only necessary that the encoder and the decoder agree on one particular point p of the shape and use the following strategy. Let p_X and p_Y be the position of p in X and Y respectively; the encoder sends the value of $\{p_X\}_S$ and the decoder estimates s as $\tilde{s} = \{p_Y - \{p_X\}_S\}_S$. The obtained result satisfies $0 \leq \tilde{s} < S$ and it is congruent to s modulo S, so that we necessarily have $\tilde{s} = s$.

On the contrary, if the shape of X is not known *a priori*, the problem becomes more interesting and it must be treated in a different way. An immediate idea is to work in the DFT phase domain. Let $\hat{X}(\cdot)$ be the DFT of X defined by

$$\hat{X}(k) = \sum_{n=0}^{N-1} X(n) e^{-j\frac{2\pi kn}{N}}, \quad k = 0, \ldots, N-1. \tag{6.2}$$

Let $\hat{Y}(\cdot)$ be accordingly the DFT of Y and, for every k, let $\Phi_{\hat{X}}(k)$ and $\Phi_{\hat{Y}}(k)$ be the phase of the coefficient $\hat{X}(k)$ and $\hat{Y}(k)$ respectively. From the relative shift hypothesis in equation (6.1), the phases of the DFT are related by the following equation

$$\Phi_{\hat{X}}(k) \stackrel{2\pi}{=} -\frac{2\pi sk}{N} + \Phi_{\hat{Y}}(k). \tag{6.3}$$

We now show how it is possible to extract few bits from the DFT phase so as to communicate the shift from encoder to decoder. First note that, if we take $k = 1$, we have

$$\Phi_{\hat{X}}(1) \stackrel{2\pi}{=} -2\pi\frac{s}{N} + \Phi_{\hat{Y}}(1). \tag{6.4}$$

Now, given that $s < N$, for every value of s the value on the right hand side of the eq. (6.4) determines a different point in the range $[-\pi, \pi]$, and the phases obtained for different values of s differ by integer multiples of $2\pi/N$. So, in theory, if a quantization ${}_q\Phi_{\hat{X}}(1)$ of $\Phi_{\hat{X}}(1)$ into $2\pi/N$-width intervals is known at the decoder, then by using the value of $\Phi_{\hat{Y}}(1)$ it is possible to recover the value of s. Of course, in this case, ${}_q\Phi_{\hat{X}}(1)$ takes on N different values; anyway, given that $s < S$, only the value of $\{{}_q\Phi_{\hat{X}}(1)\}_S$ is really needed at the decoder. So, only $\log(S)$ bits are required in order to quantize $\Phi_{\hat{X}}(1)$ so that the decoder can recover the value of s. A careful analysis shows that this method is not substantially different, from a theoretical point of view, from the previously mentioned technique involving the use of p_x and p_y. The advantage is that this second method can be used unaltered independently from the shape of X.[1]

This strategy based on the quantization of $\Phi_{\hat{X}}(1)$, even if it is theoretically valid under the assumed ideal hypothesis, has some disadvantages in terms of robustness, because it is

[1] Actually this is not true in some pathological cases. Indeed, if $\hat{X}(1)$ is exactly zero this method cannot be applied. We do not consider this case, since this rarely occurs for practical sequence of signals we are interested in.

based on an arbitrarily precise evaluation of the phase of one coefficient. In the presence of noise, or in the more concrete case where "non-circular" shifts are involved, some phase "errors" are usually introduced, and in the above scheme even a small error can cause a wrong extraction of the value of s.

Here we propose a different method to extract the shift value, which is based on a coarse quantization of more coefficients, rather than on a fine quantization of only one coefficient. The main idea is that if we take a signal and we shift it by one pixel, then by two and so on, the phases of the coefficients of the DFT at different frequencies vary in different ways. For example, the phase of $\hat{X}(N/2)$ changes by π radiants for every pixel shift. Said in other way, the sign of $\Phi_{\hat{X}}(N/2)$ is kept unchanged if X is shifted by an even number of pixels and it changes if X is shifted by an odd number of pixels. Thus we can use the sign of $\Phi_{\hat{X}}(N/2)$ to detect if s is even or odd, i.e., to detect $\{s\}_2$. Once we know the value $\{s\}_2$ we would need to know the value of $\{s\}_4$ and in order to do this we could use the phase of $\Phi_{\hat{X}}(N/4)$, which has periodicity 4. The idea can then be iterated with the same logic for the complete detection of s.

We give now a rigorous explanation of the proposed procedure. Let us consider the phase of DFT coefficients taken at exponentially spaced positions, i.e.

$$\Phi_{\hat{X}}(N/2), \Phi_{\hat{X}}(N/4), \Phi_{\hat{X}}(N/8), \ldots, \Phi_{\hat{X}}(N/S). \tag{6.5}$$

We show that a 1-bit quantization, namely quantizing the sign, of the above phases is sufficient to recover the value of s at the decoder. Note that the total amount of required bits is again $\log(S)$.

Let us write the binary representation of s as $s_\sigma s_{\sigma-1} \cdots s_1 s_0$, $s_i \in \{0,1\}, i = 0, \ldots, \sigma$. First consider the $N/2$-th DFT coefficient. For this coefficient eq. (6.3) becomes

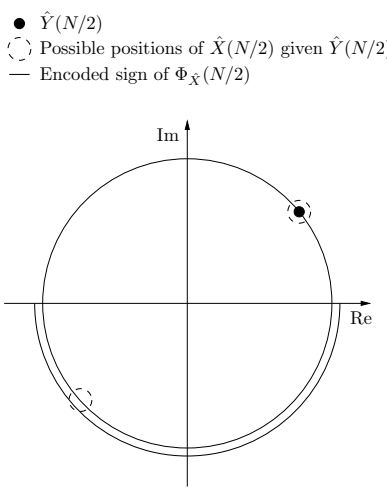
$$\Phi_{\hat{X}}(N/2) \stackrel{2\pi}{=} -\pi s + \Phi_{\hat{Y}}(N/2) \tag{6.6}$$

$$\stackrel{2\pi}{=} -\pi s_0 + \Phi_{\hat{Y}}(N/2). \tag{6.7}$$

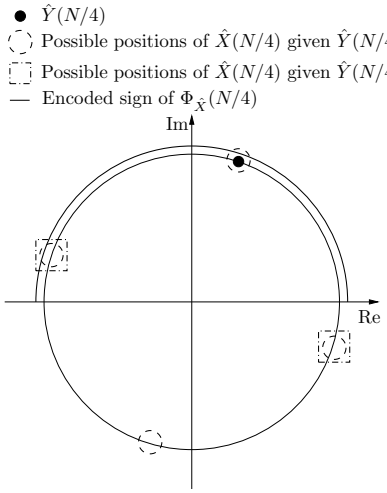
It is clear that when $\Phi_{\hat{Y}}(N/2)$ is known, the sign of $\Phi_{\hat{X}}(N/2)$ uniquely determines the value of s_0. So, one bit extracted from $\Phi_{\hat{X}}(N/2)$ (i.e., the sign) allows to determine the least significative bit of s at the decoder (see fig. 6.2(a) for an example). Now, by using an iterated procedure, we show by induction that the binary representation of s can be reconstructed from the signs of the considered coefficients (see Figures 6.2(b) and 6.2(c) for graphical examples). In fact, supposing that the bits $s_0, s_1, \ldots, s_{h-1}$ has been determined using the signs of $\Phi_{\hat{X}}(N/2), \Phi_{\hat{X}}(N/2^2), \ldots, \Phi_{\hat{X}}(N/2^h)$, and consider the coefficient $\hat{X}(N/2^{h+1})$, we have that

$$\begin{aligned} \Phi_{\hat{X}}(N/2^{h+1}) &\stackrel{2\pi}{=} -\pi \frac{s}{2^h} + \Phi_{\hat{Y}}(N/2^{h+1}) \\ &\stackrel{2\pi}{=} -\pi s_h - \frac{\pi}{2^h}\{s\}_{2^h} + \Phi_{\hat{Y}}(N/2^{h+1}). \end{aligned}$$

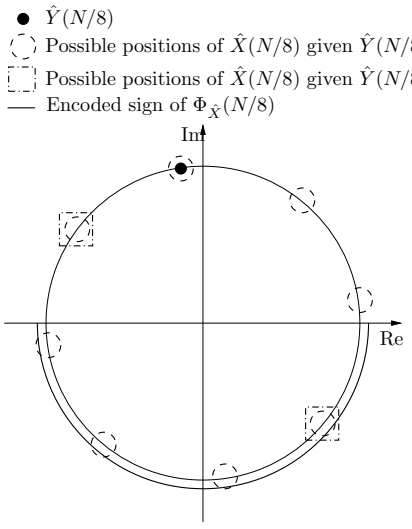
Now, clearly $\{s\}_{2^h} = s_{h-1} \cdots s_1 s_0$ is known to the decoder, so that the only unknown term in the right hand side of the above equation is s_h. So, again, the sign of $\Phi_{\hat{X}}(N/2^{h+1})$



(a) Decoding of bit $s_0 = 1$. Here $\Phi_{\hat{X}}(N/2) = \Phi_{\hat{Y}}(N/2) - \pi$.



(b) Decoding of bit $s_1 = 1$. Here $\Phi_{\hat{X}}(N/4) = \Phi_{\hat{Y}}(N/4) + \pi/4$.



(c) Decoding of bit $s_2 = 0$. Here $\Phi_{\hat{X}}(N/8) = \Phi_{\hat{Y}}(N/8) - 3\pi/8$.

Figure 6.2: Procedure for the decoding of the first three bits of s. In this case we had $s = 3$. Here, without loss of generality, we have represented the values of the DFT coefficients as complex numbers with equal absolute value, so that they all lies on a circle and the phase are easily studied.

uniquely determines s_h. This proves that the $\log(S)$ bits that represent the signs of the phases $\Phi_{\hat{X}}(N/2^i), i = 1, \ldots, \log(S)$, allow the decoder to reconstruct the value of s.

6.2.2 Two-dimensional problem

In this section we apply the theoretical development presented in the previous section to the practical problem of encoding the relative shift between images. We first consider the ideal case where an image X is obtained by applying a 2-dimensional circular shift to an N by N image Y. If $\mathbf{v} = (r, c)$ is the shift vector, where $0 \leq r < R$ and $0 \leq c < C$, with $R < N$ and $C < N$, the relation between the images is

$$X(n, m) = Y(\{n - r\}_N, \{m - c\}_N), \quad n, m = 0, 1, \ldots, N - 1, \tag{6.8}$$

and the relation between the 2-dimensional DFT's is now given by

$$\Phi_{\hat{X}}(k, l) = -j\frac{2\pi kr}{N} - j\frac{2\pi lc}{N} + \Phi_{\hat{Y}}(k, l). \tag{6.9}$$

It is easy to see from the above equation that the problems of determining r and c can be solved in a separable fashion. In fact, by taking for example $l = 0$, we cancel the term including c, and we reduce eq. (6.9) to an equivalent of eq. (6.3), where r plays the role of s. So, by taking respectively $l = 0$ and $k = 0$, we can solve the problem of encoding/decoding r and c independently. By applying the technique explained in the previous section, the only required bits can thus be extracted from the DFT of X as the signs of the phases of pure vertical and horizontal frequencies, i.e.,

$$\Phi_{\hat{X}}(N/2, 0), \Phi_{\hat{X}}(N/4, 0), \ldots, \Phi_{\hat{X}}(N/R, 0), \tag{6.10}$$

$$\Phi_{\hat{X}}(0, N/2), \Phi_{\hat{X}}(0, N/4), \ldots, \Phi_{\hat{X}}(0, N/C). \tag{6.11}$$

In this case, the total amount of required bits is $\log(R) + \log(C)$. So, the 2-dimensional problem in the ideal situation of noiseless circular shifts is solved exactly in the same way as in the 1-D case.

6.2.3 A more realistic scenario: adding redundancy

Now we apply the above approach to a more realistic situation where the two images X and Y are obtained by cropping a common scene from two shifted positions. In this case, with respect to the ideal setting considered before, the shift between X and Y is not a circular one; moreover, the two images are affected by noise. We model this fact by saying that there is a scene $z(n, m)$ and independent noises n_x, n_y such that

$$Y(n, m) = z(n, m) + n_y(n, m), \tag{6.12}$$

$$X(n, m) = z(n - r, m - c) + n_x(n, m). \tag{6.13}$$

An important element to clarify is that, under these different assumptions, we are not anymore interested in using exactly $\log(R) + \log(C)$ bits in order to encode the shift. In fact, due to the noise and to boundary effects, it is reasonable to consider more bits in order to robustly encode the shift. Moreover, in this case, the values of R and C are assumed to be much smaller than N, because when R and C get comparable to N, the overlap between the X and Y image gets smaller and smaller. Finally, it is reasonable to assume that the number of required bits to encode the shift may depend on the strength of the additive noise on the X and Y images. So, for this practical situation, we relax the problem to more informal constraints and we aim at finding a robust strategy in order to use a small number of bits to encode the shift between the images.[2]

The main idea for encoding the shift in this practical situation, then, is to use the insight given by the theoretical development proposed for the ideal case and "extend" the technique by increasing its robustness. In order to do this, it is necessary to add redundancy to the encoded data, as it is usually done in channel coding. In our scheme, when we considered the phase relation expressed by eq. (6.9), we noted that it is possible to solve the problem separately for r and c by putting $l = 0$ and $k = 0$ respectively, so as to use a minimum number of bits. Now, given that we are looking for robustness, it is very useful to go in the opposite direction and note the fact that when l and k are both different from zero the value of the resulting phase is affected by both r and c. So, if instead of using only the coefficients associated to vertical and horizontal frequencies, as in eq.'s (6.10), (6.11), we also consider "diagonal" frequency phases of the form $\Phi_{\hat{X}}(N/2^i, N/2^j)$, we actually add some sort of "parity-check" to the code.

We need to extend the initial idea and to consider the general case where we encode the sign of the phases of coefficients $\Phi_{\hat{X}}(k, l)$ for values of k and l that are either 0 or powers of 2. In this case the procedure for the decoding of the bits of the shift becomes much more involved and it is not possible to use a decoding technique as the one described for the ideal case. Here we actually find that the performance of the coding technique is strongly related to the computational complexity of the decoder.

In our problem, we consider full search methods where all possible values of r and c are tested so as to find the most plausible shift, doing the equivalent operation of a minimum distance decoding in channel coding. Here we propose two different full-search methods which have two different computational complexities for the decoder, the more complex method having of course better performance.

The main idea, which is common to both decoding techniques, is that, theoretical discussions apart, we can see the bits extracted from the phase of the X image as a hash of the image. At the decoder, what we want to do is to estimated the shift that, applied to Y, gives an image with a hash similar to that of X. Actually, the image X and the shift-compensated Y will always coincide only in the central part, as we cannot recreate at the decoder the portion of the X image located on the disappeared boundary. So, in order to smooth the boundary effects we can apply smoothing windows to the X and Y images. For the X

[2] We point out that, for e.g. a 256x256 image, reasonable values of R and C would need to be much smaller than 128, and thus $\log(R) + \log(C)$ bits would mean less than 14 bits.

image, the way the windowing operation is performed is not an issue; we simply multiply the X image by the window before performing the DFT operation. The way this smoothing window is used at the decoder, instead, makes the difference between the complex and the light registration methods proposed here.

We start by describing the optimal more complex technique, which is somehow also the most obvious one. The decoder consider all possible pairs of (r, c) values; for every one of them a circular shift by a (r, c) vector is applied to Y. The resulting image is multiplied by the window so at to remove the boundary effects, it is transformed, and the signs of the phase of the DFT coefficients are extracted. The Hamming distance of the obtained code from the one extracted from X is then computed[3], and the values of r and c that minimize this distance are kept as best estimate of the true shift components. Note that with this technique, when the correct value of r and c are tested, the shifted and windowed image Y differs from the windowed X mainly only for the noise, the border effects being smoothed by the window. This gives to the technique a great robustness. On the other hand, the main disadvantage is that for every (r, c) pair a DFT must be computed for the Y image. This lead to a very high computational complexity that may be considered as an intolerable drawback of this method.

A different choice is to consider a method which has a much lower computational complexity but, on the other hand, cannot reach the same performance of the previous one. In this second scheme, the Y image is multiplied by the window only once, at the beginning of the process, it is transformed, and the submatrix of meaningful coefficients is extracted. Then, for every (r, c) pair, a circular shift on Y by a (r, c) vector is implemented in this subfrequency domain by multiplying the coefficients by appropriate exponential factors. The phase sign are then extracted and again the Hamming distance from the code of X is computed. Again, the (r, c) pair that gives the minimum distance from the X code is kept as estimate of the shift vector. Note that in this case only one DFT is computed, and the operations required for every (r, c) pair have a much lower computational complexity with respect to the previous method.

6.2.4 Experimental results

In order to show the effectiveness of the proposed method and to evaluate the performance in a practical situation, we have run some experiments on test images, and we report here one of these tests. We have only performed extensive simulations using the computationally-light proposed scheme, as the computationally complex scheme requires too many operation to extensively study the performance for different noise strength and shift amplitudes (see Fig. 6.3 for an example of difference of performance of the two methods).

[3]From a theoretical point of view, the use of the Hamming distance is motivated if the phase noise associated to the non-ideal scenario can be considered as an independent noise which induces a binary symmetric channel on the space of the signs of the phases of the DFT coefficients. Note that using a Hamming code correspond to performing a minimum distance hard-decoding of a binary codeword. Other decoding techniques, based for example on soft decoding ideas or exploiting particular structures of the phase noise, are currently under investigation.

In the experiment, we have taken the 512x512 "goldhill" image, and we have constructed the 256x256 X and Y images by cropping portions of goldhill and by adding independent white gaussian noise to them. We have then adopted the computationally simplified method and we have checked whether it gave the right result or not. The experiment was performed by testing, for different number of bits used for the code, various shift vector lengths and increasing noise amplitudes. The results are shown in Fig. 6.4, where we can see that by increasing the number of bits of the code progressively from 25 to 81 we are able to correctly detect shift vectors with increasing amplitudes and for increasing strength of the noise.

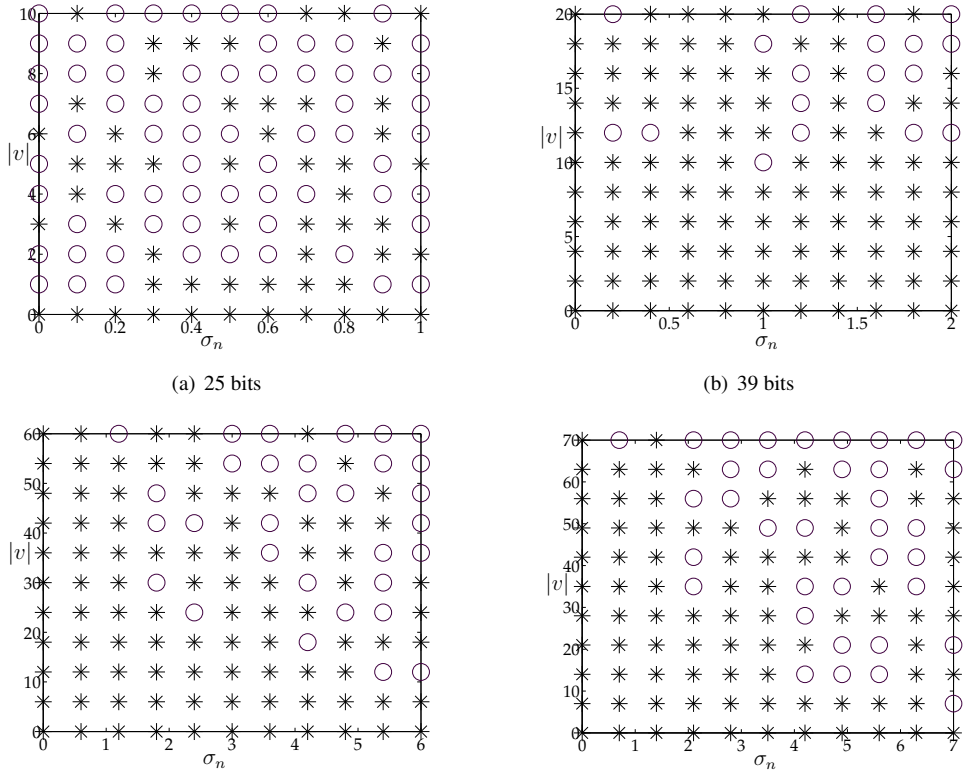
(a) Image X (b) Image Y

Figure 6.3: Example of 256x256 X and Y images cropped from the 512x512 Goldhill image. Here we have $r = 21$, $c = 36$ and n_x and n_y are independent white gaussian noises with $\sigma_{n_x} = \sigma_{n_y} = 2$. In this case 69 bits suffice to correctly detect the shift with the computationally light decoder (in less than 1 second), while 39 bits suffice in the case of the complex decoder (in more than 300 seconds).

6.3 Rotation and scale detection using the Fourier-Mellin transform

6.3.1 From shift to rotation and scale

The ideas presented in the previous section for the distributed coding of relative shifts between images can be further investigated in the direction of an extension for the more general problem where two images do not only differ for a relative shift, but also for rotations and/or a scale factor. By properly operating on the DFT of the images, it is possible to reduce scale and rotations to a shift problem in a proper non-linearly transformed domain. In the field of image registration this idea has been initially proposed in [37], [5] and [38] for the problem of combined translations and rotations. The extension to the case of scale between images has then been studied in [25] and in [90] with the use of what is actually



(a) 25 bits

(b) 39 bits

(c) 61 bits

(d) 81 bits

Figure 6.4: Successes ($*$)/failures ($\circ$) using the computationally light decoding of v, depending on the amplitude of the shift $|\mathbf{v}|$ and on the noise strength (measured with σ_n) , for different number of used bits. Images X and Y were obtained here by cropping the image "Goldhill" at random positions. An asterisk indicates a success while a circle indicates a failure. Note that for visual clarity the axis scale is different for different number of bits used.

known as a Fourier-Mellin transform. We refer to [24] for a survey of image registration techniques.

The important fact of the use of the DFT for dealing with rotations and scale is that it is possible to reduce these two operations to a shift operation in a proper transformed domain, which is the log-polar domain we will now describe. Thus, the procedure described in the previous section for the distributed encoding of shift can in principle be applied also to the problem of distributed encoding of relative rotations and scale of an image X with respect to a side information image Y available at the decoder.

Consider the case where the transformation from the images X to Y is a combination of translation, rotation and scale. Consider for simplicity the case of noiseless images, where images are considered now as continuous domain signals. Thus, for n, m real numbers we can write the relation between the two images as

$$X(m,n) = Y(\lambda(m\cos\theta_0 + n\sin\theta_0) - r, \lambda(-m\sin\theta_0 + n\cos\theta_0) - c) \tag{6.14}$$

If we compute the Fourier transform of these two signals in the continuous frequency domain (i.e., for real k, l)we have

$$\hat{X}(k,l) = \frac{e^{-j\phi_{r,c}(k,l)}}{\lambda^2}\hat{Y}(\lambda^{-1}(k\cos\theta_0 + l\sin\theta_0), \lambda^{-1}(-k\sin\theta_0 + l\cos\theta_0)) \tag{6.15}$$

where $\phi_{r,c}(k,l)$ is the phase term due to the translation by the vector $\mathbf{v} = (r,c)$. If we take the modulus of both sides of the above equation, we can remove this term and we obtain

$$|\hat{X}(k,l)| = \frac{1}{\lambda^2}|\hat{Y}(\lambda^{-1}(k\cos\theta_0 + l\sin\theta_0), \lambda^{-1}(-k\sin\theta_0 + l\cos\theta_0))| \tag{6.16}$$

So, by keeping only the modulus of the transforms, we have preserved the rotation and scale relations between the images, momentarily disregarding any translational mismatch. Now, by changing to polar coordinates, let us set $\tilde{X}(\rho,\theta) = |\hat{X}(\rho\cos\theta, \rho\sin\theta)|$ and similarly $\tilde{Y}(\rho,\theta) = |\hat{Y}(\rho\cos\theta, \rho\sin\theta)|$. Then, with some simple algebraic manipulations we have

$$\tilde{X}(\rho,\theta) = \frac{1}{\lambda^2}\tilde{Y}(\rho/\lambda, \theta - \theta_0) \tag{6.17}$$

Thus, the relative rotation between the images is now reduced to a shift in the second variable in the polar domain. The only non-translational deformation between the two obtained signals is now due to the scaling. Now, by using a logarithmic scale for the radial coordinate in the polar domain, once set $\bar{X}(\rho,\theta) = \tilde{X}(e^\rho,\theta)$ and $\bar{Y}(\rho,\theta) = \tilde{Y}(e^\rho,\theta)$ we obtain

$$\bar{X}(\rho,\theta) = \frac{1}{\lambda^2}\bar{Y}(\rho - \log\lambda, \theta - \theta_0) \tag{6.18}$$

So, the rotation and scale between the two images X and Y are reduced to a shift between the two signals $\bar{X}$ and $\bar{Y}$. Note that all the operations we have applied to the two images X and Y can be performed separately on each one; thus the operation on the image

X can be performed at the encoder without any need to know Y. So, we have reduced the problem of distributed rotation and scale factor coding to a distributed coding of shifts and, for this problem, we can use the same technique described in the previous section. Once the decoder has recovered the rotation and the scale factor, by rotating and rescaling the image Y it can obtain an image Y' that only differ from X by a shift, apart from the noise inevitably due to resampling operations. As last step, then the decoder can recover also the shift if it has been encoded with the distributed shift coding technique as described.

6.3.2 From the ideal case to the concrete problem

The above discussion, anyway, only holds rigorously in continuous space and frequency domains and, moreover, under the hypothesis of noiseless images defined on an infinite domain. For discrete images with limited support there are some issues to address in order to implement an algorithm that might work.

The first point is that if the images X and Y are cropped versions from a same scene, their spectrum is distorted by the effects of the window. The most relevant effect is that some false vertical and horizontal frequencies appear in the spectrum, which are due to the discontinuity that appear in the periodic replication of the image. In this case, if the two images have a relative rotation with respect to the other in the sense that they are cropped with a relative rotation from an infinitely wide scene, then those false vertical and horizontal frequencies do not rotate with the remaining part of the spectrum, but instead stay in the vertical and horizontal direction. So, detecting rotations in the spectrum domain is difficult unless these frequencies are removed. In order to do that it is necessary to use a smoothing window to the two images X and Y (such as for example a Tukey window) so that their periodic replication do not contain significant false vertical and horizontal discontinuities. This operation was already suggested for the coding of shifts presented in the previous section in order to remove the boundary effects. Thus it can remain for the detection of the rotational component.

Another problem is found in the change from Cartesian coordinates to Log-Polar coordinates. In this case, we must consider that the resolution used in the resampling is very important in order to preserve the information about rotation and scale. In particular, the resolution used for the angular coordinate must be sufficient to detect the angular rotation with the required resolution. This is not a great problem, however. A more important problem is the logarithmically spaced resampling in the radial direction. Due to the fact that we can only use integer coordinates in a digital representation of the image, the values of ρ in eq. (6.18) are in practice always integers; given that our algorithm for distributed coding of shifts is studied for the detection of integer shifts, we find that using that algorithm we can only estimate $\log \lambda$ to the nearest integer value. If the logarithms are taken to the base e, the resolution for the scale factor is defined by the interval $[e^{\lfloor \log \lambda \rfloor}, e^{\lceil \log \lambda \rceil}]$ or, in other words, the scale factors that can be detected by the algorithm are the values $\ldots, e^{-2}, e^{-1}, 1, e, e^2, \ldots$. It is important to consider that typically interesting values for λ are values close to 1, and thus, using the base e cannot provide a good resolution for λ. In

order to achieve such a good resolution it is necessary to use a small base μ for the logarithmic resampling, in particular a base μ sufficiently small so as to have two consecutive powers of μ sufficiently tight around λ. As a first approximation, we can consider that if we take a base $\mu = 1 + \epsilon$, where ϵ is much smaller than 1, the detectable scale factors are the values $(1 + \epsilon)^k$ with integer k, which can be approximated at the first order as $1 + k\epsilon$. So, in order to have a resolution ϵ on the detectable scale factor we have to use a base $1 + \epsilon$ in the logarithmic scale of the radial coordinate.

The last point that is important to clarify is that once the spectrum of the images are represented in the Log-Polar domain, in order to apply the distributed shift coding algorithm it is again necessary to apply a windowing in order to reduce the boundary effects. In this case, it is particularly important to perform this operation because the spectrum of natural images in the Log-Polar domain is mostly concentrated around the axis $\rho = 0$, and rapidly vanish for high values of ρ. This causes strong boundary effects because of the fact that ideal shifts for the DFT phase are circular shifts, while here we have a non-circular shift. This is the same problem already explained in the previous section where the windows were applied directly on the images; here we only want to clarify that the boundary effects are much more critical.

In the next section we give an example of how the proposed approach operate in order to recover the rotation and scale of a couple of images. For the problem of distributed coding of relative rotation and scale, with respect to the shift only problem, we found more difficult to properly test the proposed approach with noisy images. Even for noiseless images, which means images created by applying a rotation and a scale factor to a single initial image,[4] the choice of some parameters such as the resolution in the resampling operations, the type of window applied to the image and to the spectrum in the Log-Polar domain, seem to have much more impact on the results. On the other hand, this is no surprise as the transformations applied to the images lead to an accumulation of boundary effects and sampling artefacts that make the algorithm less robust. So, for this problem it is necessary to consider possible ways of increasing the robustness by using more bits for the representation of the phase informations (consider that by applying the algorithm as presented for the shift problem, encoding rotation and scale requires only about 100 bits for 512x512 images). This aspect remains object of on-going research.

6.3.3 Experimental simulation

In this section we use a simulation example to give a step by step description of the operations involved in the distributed coding of rotation and scale.

Consider two images X and Y as shown in Figure 6.5. The Image Y is obtained by rotating X of 10 degrees in the clockwise direction, and applying a scale factor of 1.25. The spectrum of the two images in the Log-Polar domain is shown in Figure 6.6.

[4]Note that in this case there is actually at least a small noise due to the fact that images are resampled on different grids. It is clear that a good interpolation technique must be used to resample the images during the rotation and scaling operations in order to reduce this error.

It is interesting to see that the spectrum of the X image presents two important peaks in the directions of the vertical frequency. This is due to the fact that the sky in the top of the X image and the ground on the bottom have different grey levels, and thus vertical frequencies are generated in the periodic replication of the image. In the Y image this frequency components are not as important as the sky is not present in the image. This shows that the spectrum of the X and Y images in the Log-Polar domain do not simply differ for a shift, but in fact different components appear. In order to remove these false frequencies, a windowing has to be applied to both images. In Figure 6.7 the two images windowed with a Tukey window are shown, and in Figure 6.8 their Log-Polar domain spectrum is shown.

In this case the vertical frequencies of image X have been removed and thus the two spectrum are more similar, shift components apart. There is a problem here due to the fact that the a great portion of the spectrum is concentrated at very low values of the ρ coordinate. Due to this fact, as the scaling factor reduces to a shift along the ρ coordinate, there is in this case a strong boundary effect. This means that the DFT's of the Log-Polar spectra of the two images do not differ only for a linear phase component. This is clearly visible in Figure 6.10(a). So, before applying to the spectrum the distributed shift coding technique, it is necessary to apply a further windowing operation. The so obtained spectrum can then be considered mainly differing for a shift, and this is clearly visible in the phase difference of the DFT as shown in Figure 6.10(b).

The distributed shift coding technique is thus applied to the spectrum signal shown in Figure 6.9(a) which means that only the signs of the phase of certain coefficients of its DFT are extracted at the encoder and sent to the decoder. In this case, we have used a total amount of 100 bits for the encoding of the relative rotation and scale. The real rotation between the images is 10°, while the scale factor is 1.25. We have sampled both the θ and ρ coordinates with 512 samples. This means that the interval between two samples has length $\theta_{\min} = 0.7031^\circ$. The integer multiples of $\theta_{\min}$ that are closest to the real values of the rotation factor are $14\theta_{\min} = 9.8438$ and $15\theta_{\min} = 10.5469$. For the radial coordinate, then, we have chosen $\mu = 1.0086$. The integer powers of μ that are closest to the true scale factor are $\mu^{25} = 1.2393$, $\mu^{26} = 1.25$ and $\mu^{27} = 1.2608$. We remark here that once the base μ is chosen, one still has a choice in how to sample the ρ coordinate; more precisely, the 512 samples can be taken at values $\rho = \mu^{k+i}$, $i = 0, \ldots, 511$, where k is any relative integer. In order to have an effective algorithm it is convenient to choose k so as to spread the 512 samples in appropriate positions in the ρ axis. Here, for example we have chosen $k = 130$, which spreads the samples from $\mu^{130} = 3.0521$ to $\mu^{641} = 245.14$, which is an appropriate range given that the original image is a 512×512 image.

Based on the signs received from the encoder as code of the rotation and scale factors, the decoder can recover the linear component in the difference of the DFT's phases of the Log-Polar spectra, which correspond to detecting the linear component in the phase shown in Figure 6.10(b) using only a small number of bits of the phase associated to 6.9(a). In this experiment, the decoded value of the rotation is $\theta' = 15\theta_{\min} = 10.5469$, which is very satisfactory, even if it is not optimal in terms of distance from the real rotation factor, as $14\theta_{\min} = 9.8438$ would be a better estimate. The scale factor is in this case "correctly"

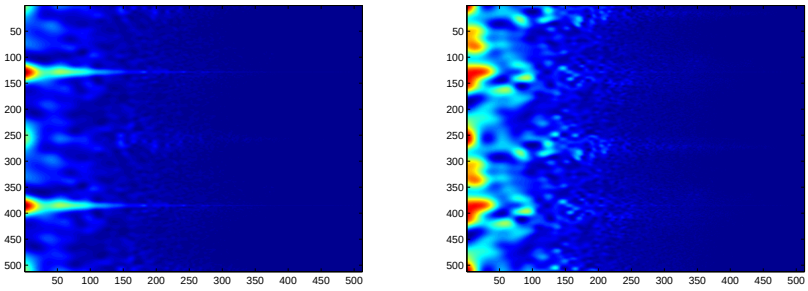
recovered as $\lambda' = \mu^{26} = 1.25$.

Thus, the rotation and scale factors are extracted at the decoder, and the inverse operations can be applied to the Y image in order to register the rotation and scale. The obtained image is shown in Figure 6.11(a), where only the shift component with respect to the X image is present. This shift component is then detected at the decoder using the procedure proposed in the Section 6.2, using the phase hash of the X image sent by the encoder. The obtained shift-compensated image Y is shown in Figure 6.11(b).

In Figure 6.12(b) we can see the "prediction" error when the registered image Y is used to predict at the decoder the X image. We can consider this signal as the actual innovation of the image X with respect to the registered Y image, which corresponds to the idea of conditional entropy $H(X|Y)$ in the information theoretic model of information sources. So, if we consider the problem of communicating X from encoder to decoder, and we consider the Slepian-Wolf setting of distributed coding, once the registration of Y has been performed at the decoder as explained in this section, the amount of information to be transmitted reduces from what is shown in Figure 6.12(a) to what is shown in Figure 6.12(b), which is a much smaller amount of information.

(a) Image X. (b) Image Y.

Figure 6.5: X and Y images used in this example. Here the relative rotation is 10 degrees, and the relative scale is 1.25.

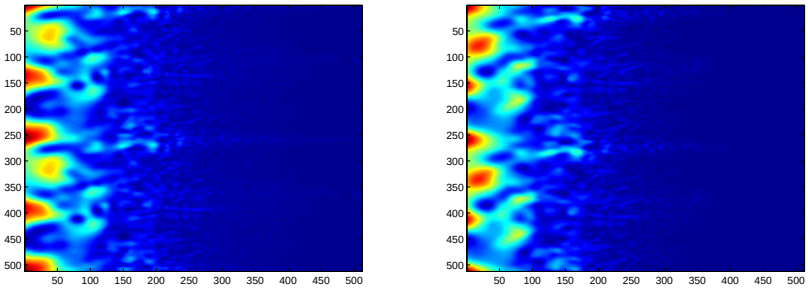


(a) Log-Polar spectrum of X. (b) Log-Polar spectrum of Y.

Figure 6.6: Spectrum of the two images in the Log-Polar domain; here ρ is the horizontal component (the spectrum within every image is renormalized to its peak value).

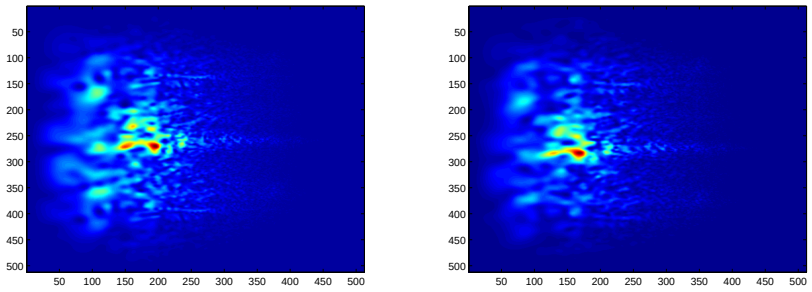
(a) Windowed image X. (b) Windowed image Y.

Figure 6.7: Images after windowing operation. Here we have used a Takey window.



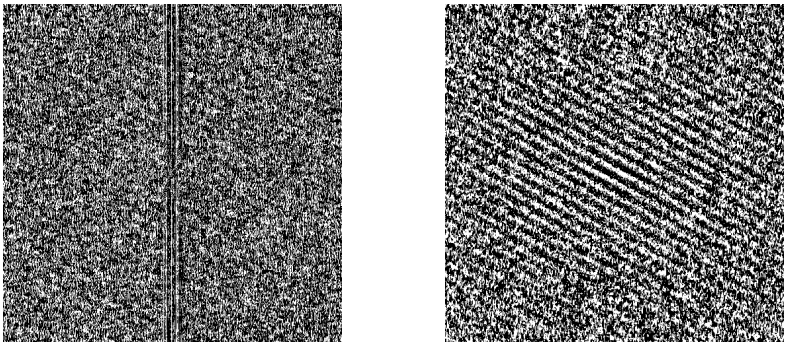
(a) Log-Polar spectrum of windowed X. (b) Log-Polar spectrum of windowed Y.

Figure 6.8: Spectrum of the two windowed images in the Log-Polar domain (the spectrum within every image is renormalized to its peak value).



(a) Windowed Log-Polar spectrum for X. (b) Windowed Log-Polar spectrum for Y.

Figure 6.9: Windowed spectrum of the two windowed images in the Log-Polar domain. The signals now really differ mainly by a shift component.



(a) Phase difference without windowing. (b) Phase difference with windowing.

Figure 6.10: Phase difference of the Fourier transform of the Log-Polar spectrum of the two images. In Figure 6.10(a) the phase difference of the DFT's of the spectrum signals of Figure 6.8 is shown, while 6.10(b) refers to the spectrum signals of Figure 6.9

(a) Rotation and scale registered image Y.

(b) Completely registered image Y.

Figure 6.11: Rotation and scale registered image Y, and successive shift aligned image.

(a) Image X.

(b) Prediction error after registration of Y.

Figure 6.12: Comparison between original image X and prediction error after compensation at the decoder.

Conclusions and Perspectives

Signal representation and coding techniques have been developing in many directions in the last years. In this work we have studied some developments of topics that have not received yet the appropriate attention in the community or that have only emerged recently and are thus still in a very early stage of research.

In particular, we have studied the problem of signal approximations under the l^∞ norm, focusing on the construction of algorithms and methods for the approximation in linear spaces and in piecewise linear spaces. In detail, for the case of first order approximations, we have proposed an efficient algorithm based on geometric considerations that gives much efficient procedure for the construction of optimal approximations. Furthermore, we have developed algorithms for the construction of minimal and optimal approximation to a given signal within the space of piecewise approximations in linear spaces. We have addressed the problem of the encoding of the obtained approximations and we have shown that the so called pivot points represent a useful tool for this aim.

Within this topic, an important future topic of research is the development of theoretical studies of the performance of lossy coding techniques in a rate distortion setting under the l^∞ norm. We point out that rate distortion theory for this particular type of distortion has received almost no interest to the present time. Only first attempts to the understanding of the basic notions have been considered, for example in [97], where the rate for the lossy representation of a memoryless source within a given constraint is considered. However, no study of the complete rate distortion function is available, neither the more interesting case of sources with memory has been considered up to now. Note that the use of approximations in linear spaces we have considered in our study is motivated by the implicit assumption that we are working with signals with memory. For memoryless source, indeed, the use of piecewise approximations is not suitable. Within this context it is interesting to consider the use of approximations in linear spaces as a case of *transform coding*. So, while the use of transform coding techniques has been widely investigated, from the point of view of the rate distortion analysis, for the case of the l^2 norm, no such theoretical studies has been considered for the case of the l^∞ norm.

In this work, after the introduction of DSC and DVC has been given, a study of problems related to the DSC paradigm and its use in the video coding have been provided in different directions. For these problems, also, we have identified some underlying aspects

that we have independently studied as proper topics in image and video processing or from and information theoretic point of view. In particular, we have provided a detailed description of the relationship between the hypothesis considered for the development of DSC and the concrete operational situation where one wants to apply DSC to video coding. The understanding of the problems of correlation estimation and of the use of feedback channels in different situations, such as in single camera systems or in multicamera systems, is important for the future development of this field.

A theoretical study of the use of the DSC approach to the problem of coding sources with memory has been provided. Interestingly, from the information theoretic point of view, the use of DSC for this kind of problems reduces to the use of codes that are not necessarily decodable for every type of sources, but are instead decodable when the memory of the source is used as *a priori* information at the decoder. So, the concept of unique decodability has been revisited and interesting consequences has been pointed out in the case when unique decodability of codes is considered with respect to *constrained* sources. In particular, we have shown that the conditions for optimality of a code have not still been well clarified up to now in the information theoretic literature, as we have provided examples of sources with associated codes such that the code for any finite number of symbols requires a number of bits strictly smaller than the entropy of those symbols. We have also provided a modified Kraft inequality which represents a necessary condition for a set of integers to be word lengths of a uniquely decodable code. This condition is not in general sufficient for the construction of a uniquely decodable code with such integers as code word lengths and we have thus developed a modified Sardinas-Patterson test for testing the unique decodability of a given set of codewords. Further work in this direction, however, remains to be done. The more general problem of finding the optimal code for a given constrained source is indeed not solved, and it would be interesting to further investigate the relation between the developed study of coding constrained sequences with the more consolidated field of coding for constrained systems [63].

Another contribution of this work has been focusing on the topic of remote image registration. This problem arises as a key problem in the field of DVC, and it is related to the idea of correlation as discussed in Chapter 4. In particular, it is important to investigate efficient encoding techniques that allow the decoder to recover the correspondences between the image X to be transmitted and the side information Y. We would like to underline that, even if we have introduced this topic as a problem related to the DVC field, it can be considered to be of significant importance by its own, as an interesting research field. In this work we have provided a first study based on the insight given by theoretical arguments, which reveal promising techniques for the remote registration of shift, rotation and scale factors between images. The study can however be further developed in order to consider more general registration problems, from non linear deformations to the more complicated problem of "local registrations", that is encountered for example in a motion compensation operation. Consider, as a further investigation in this direction, the problem of finding correspondence between images that represent a 3-dimensional scene taken from different positions ad with different directions. In this case, the matching between the views, in a classic setting where

both images are available in one point, is based on the extraction and use of feature points that are used for a first matching in order to find correspondences. Further refinement is then performed based on the texture of the images. In a distributed setting, where images are not available at the same point, this approach can be considered as an interesting starting point. In order to perform the first matching operations, only a subset of the whole image information is used, which means that few bits have to be sent for this first step. So, even this problem can be considered in the setting of remote image registration. In general, what is important is to study the most effective technique for extracting the meaningful information used for finding the matching. With respect to distributed video coding, furthermore, it is necessary to incorporate the use of this techniques in a complete coding setting. This means that after the registration information has been sent to the decoder, some Wyner-Ziv information, generated as parity bits at the encoder, has to be sent in order to correct the "compensated" side information and obtain a better reconstruction of the original image available at the encoder.

References

[1] A. Aaron, R. Zhang and B. Girod. Wyner-Ziv coding for motion video. *Asilomar Conference on Signals, Systems and Computers*, Pacific Groove, USA, 2002.

[2] A. Aaron, S. Rane and B. Girod. Wyner-Ziv video coding with hash-based motion-compensation at the receiver. In *Proc. IEEE Int. Conf. on Image Proc.*, Singapore, October 2004.

[3] R. Ahlswede. Coloring hypergraphs: A new approach to multi-user source coding-I. *J. Combinatorics, Inform. Syst. Sci.*, 4(1): 76–115, 1979.

[4] R. Ahlswede and J. Korner. Source coding with side information and a converse for the degraded broadcast channel. *IEEE Trans. Inform. Theory* 21: 629–637, 1975.

[5] S. Alliney, C. Morandi. Digital Image Registration Using Projections. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(2): 222-233, 1986.

[6] N. Alon and A. Orlitsky. A lower bound on the expected length of one-to-one codes. *IEEE Trans. Inform. Theory*, 40: 1670–1672, 1994.

[7] N. Alon and A. Orlitsky. Source coding and graph entropies. *IEEE Trans. Inform. Theory*, 42: 1329–1339, 1996.

[8] X. Artigas and L. Torres. Improved signal reconstruction and return channel suppression in Distributed Video Coding systems. *47th International Symposium ELMAR-2005 focused on Multimedia Systems and Applications*, Zadar, 2005.

[9] X. Artigas and L. Torres. Iterative Generation of Motion-Compensated Side Information for Distributed Video Coding. In *Proc. IEEE Int. Conf. on Image Proc.*, Genova, 2005.

[10] X. Artigas, M. Tagliasacchi, L. Torres and S. Tubaro. A Proposal to Suppress the Training Stage in a Coset-Based Distributed Video Codec. In *IEEE Int. Conf. on Acoust., Speech, and Signal Proc.*, Toulouse, 2006.

[11] J. Ascenso, C. Brites and F. Pereira. Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Smolenice, 2005.

[12] J. Ascenso, C. Brites and F. Pereira. Motion Compensated Refinement for Low Complexity Pixel Based Distributed Video Coding. In *IEEE Int. Conf. on Advanced Video and Signal-Based Surveillance*, Como, 2005.

[13] R.B. Ash. *Information Theory*. Interscience, New York, 1965.

[14] D. Avis and D. Bremner. How good are convex hull algorithms?. *Symposium on Computational Geometry*, Vancouver, 1995.

[15] I. Barrodale and C. Phillips. Solution of an overdetermined system of linear equations in the Chebyshev norm [F4] (algorithm 495). *ACM Trans. Math. Software*, 1(3): 264–270, 1975.

[16] T. Berger. *Rate-distortion theory: A mathematical basis for data compression.* Prentice-Hall, Englewood Cliffs, 1971.

[17] T. Berger and R. W. Yeung. Multiterminal source encoding with one distortion criterion. *IEEE Trans. Inform. Theory*, 35(2): 228–236 , 1989.

[18] R. E. Blahut, *Theory and Practice of Error-Control Codes*. Addison-Wesley, Massachusetts, 1983.

[19] C. Blundo and R. De Prisco. New bounds on the expected length of one-to-one codes. *IEEE Trans. Inform. Theory*, 42(1): 246–250.

[20] L. Breiman. The individual ergodic theorem of information theory. *Ann. Math. Statist.*, 28: 809–811, 1957; correction, *Ann. Math. Statist.*, 31: 809–810, 1960.

[21] C. Brites, J. Ascenso and F. Pereira. Modeling correlation noise statistics at decoder for pixel based Wyner-Ziv video coding. *Picture Coding Symposium*, Beijing, 2006.

[22] C. Brites, J. Ascenso and F. Pereira. Feedback channel in pixel domain Wyner-Ziv video coding: myths and realities. *14th European Signal Processing Conference*, Florence, 2006.

[23] C. Brites, J. Ascenso and F. Pereira. Studying temporal correlation noise modeling for pixel based Wyner-Ziv video coding. In *Proc. IEEE Int. Conf. on Image Proc.*, Atlanta, 2006.

[24] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[25] Q. Chen, M. Defrise and F. Deconinck. Symmetric phase-only matched filtering of Fourier-Mellin transform for image registration and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16: 1156–1168, 1994.

[26] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*, 2nd ed. MIT Press, Cambridge, 2001.

[27] T. M. Cover. A proof of the data compression theorem of Slepian and Wolf for ergodic sources. *IEEE Trans. Inform. Theory*, 22: 226–228, 1975.

[28] T. M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley, New York, 1990.

[29] M. Dalai and R. Leonardi. Efficient (piecewise) linear minmax approximation of digital signals. In *Proc. Int. Conf. on Acoust., Speech, and Signal Proc.*, Montreal, 2004.

[30] M. Dalai and R. Leonardi. l^∞ norm based second generation image coding. *Proc. IEEE Int. Conf. on Image Proc.*, pp. 3193–3196, Singapore, 2004.

[31] M. Dalai and R. Leonardi. Segmentation based image coding with l-infinity norm error control. *Picture Coding Symposium*, San Francisco, 2004.

[32] M. Dalai and R. Leonardi. Non prefix-free codes for constrained sequences. In *Proc. of Int. Symp. Information Theory*, pp. 1534-1538, Adelaide, 2005.

[33] M. Dalai and R. Leonardi. L-infinity Constrained Approximations for Image and Video Compression. *Picture Coding Symposium*, Beijing, 2006.

[34] M. Dalai and R. Leonardi. Approximations of One-Dimensional Digital Signals under the l-infinity Norm. *IEEE Trans. on Signal Processing*, 54(8): 3111–3124, Aug. 2006.

[35] M. Dalai, R. Leonardi and F. Pereira. Improving Turbo Codec Integration In Pixel-Domain Distributed Video Coding. In *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Proc.*, Toulouse, 2006.

[36] M. Dalai, R. Leonardi and P. L. Dragotti. Distributed coding of shifts using the DFT phase. Accepted at *IEEE Int. Conf. on Acoust., Speech, and Signal Proc.*, Honolulu, 2007.

[37] E. De Castro, C. Morandi. Tracking di immagini in movimento rototraslatorio mediante trasformate di Fourier. *Atti della Accademia delle Scienze dell' Istituto di Bologna, Classe di Scienze Fisiche*, anno 272, Rendiconti, serie XIV, Tomo I, 1983/84, 1984, pp.1-8.

[38] E. De Castro, C. Morandi. Registration of rotated and translated images using finite Fourier transforms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5): 700–703, 1987.

[39] DISCOVER, DIStributed COding for Video SERvices, FET open project FP6-IST-015314. Web page: http://www.discoverdvc.org.

[40] A. El Gamal and T. Cover. Multiple User Information Theory. *Proc. IEEE*, 68(12): 1466–1483, 1980.

[41] P. Elias. Universal codeword sets and representations of the integers. *IEEE Trans. Inform. Theory*, 21(2): 194–203, 1975.

[42] T. Ericson and V. Ramamoorthy. Modulo-PCM: A new source coding scheme. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, Washington, pp. 419-422, 1979.

[43] M. J. Ferguson and D. W. Bailey. Zero-error coding for correlated sources. Unpublished manuscript, 1975.

[44] J. Fowler, M. Tagliasacchi, B. Pesquet Popescu. Wavelet-Based Distributed Source Coding of Video. *European Signal Processing Conference*, Antalya, 2005.

[45] R. G. Gallager. *Information Theory and Reliable Communication*. Wiley, New York, 1968.

[46] M. Gastpar. The Wyner-Ziv problem with multiple sources. *IEEE Trans. Inform. Theory*, 50(11): 2762–2768, 2004.

[47] N. Gehrig and P. L. Dragotti. Distributed Compression of the Plenoptic Function. In *Proc. IEEE Int. Conf. on Image Proc.*, Singapore, 2004.

[48] N. Gehrig and P. L. Dragotti. Symmetric and a-symmetric Slepian-Wolf codes with systematic and non-systematic linear codes. *IEEE Comm. Letters*, 9(1): 61–63, 2005.

[49] N. Gehrig and P. L. Dragotti. DIFFERENT-DIstributed and Fully Flexible image EncodeRs for camEra sensor NeTworks. In *Proc. of IEEE Int. Conf. on Image Proc.*, Genoa, 2005.

[50] B. Girod, A. Aaron, S. Rane and D. Rebollo-Monedero. Distributed Video Coding. *Proc. IEEE*, 93(1): 71-83, 2005.

[51] J. E. Goodman and J. O'Roarke. *Handbook of Discrete and Computational Geometry*. CRC Press, Boca Raton, 1997.

[52] R. Graham. An efficient algorithm for determinimg the convex hull of a finite point set. *Info. Proc. Letters*, 1: 132–133, 1972.

[53] Hanying Feng, Qian Zhao. On the rate loss of multiresolution source codes in the Wyner-Ziv setting *IEEE Trans. Inform. Theory*, 52(3): 1164–1171, 2006.

[54] C. Heegard and T. Berger. Rate distortion when side information may be absent. *IEEE Trans. Inform. Theory*, 31(6): 727–734, 1985.

[55] J. Karush. A simple proof of an inequality of McMillan. *IRE Trans. Inform. Theory*, 7: 118, 1961.

[56] A. Kaspi and T. Berger. Rate-distortion for correlated sources with partially separated encoders. *IEEE Trans. Inform. Theory*, 28(6): 828–840, 1982.

[57] S. Klomp, Y. Vatis and J. Ostermann. Side Information Interpolation with Sub-pel Motion Compensation for Wyner-Ziv Decoder. In *Int. Conf. on Signal Proc. and Multim. Applic.*, Setúbal, 2006.

[58] P. Koulgi, E. Tuncel, S. Regunathan, and K. Rose. On zero-error source coding with decoder side information. *IEEE Trans. Inform. Theory*, 49: 99–111, 2003.

[59] P. Koulgi, E. Tuncel, S. Regunathan, and K. Rose. On zero-error coding of correlated sources. *IEEE Trans. Inform. Theory*, 49: 2856–2873, 2003.

[60] L. G. Kraft. A device for quantizing, grouping and coding amplitude modulated pulses. Master's thesis, Dept. of Electrical Eng., MIT, Cambridge, Mass., 1949.

[61] D. Kubasov and C. Guillemot. Mesh-based motion-compensated interpolation for side information extraction in distributed video coding. In *Proc. IEEE Int. Conf. on Image Proc.*, Atlanta, 2006.

[62] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. In *IEEE 1975 Conference on Cybernetics and Society*, pp. 163–165, 1975.

[63] D. Lind and B. Marcus. *An introduction to Symbolic Dynamics and Coding*. Cambridge University Press, Cambridge, 1996.

[64] A. Liveris, Z. Xiong and C. Georghiades. A distributed source coding technique for highly correlated images using Turbo codes. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, Orlando, FL, 2002.

[65] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison Wesley, Amsterdam, 1984.

[66] E. Martinian, A. Vetro, J. Ascenso, A. Khisti and D. Malioutov. Hybrid Distributed Video Coding Using SCA Codes. *IEEE International Workshop on Multimedia Signal Processing*, Victoria, 2006.

[67] B. McMillan. The basic theorems of information theory. *Ann. Math. Stat.*, 24: 196–219, 1953.

[68] B. McMillan. Two inequalities implied by unique decipherability. *IEEE Trans. Inform. Theory*, 2: 115–116, 1956.

[69] N. Megiddo. Linear programming in linear time when the dimension is fixed. *J. ACM*, 12: 114–127, 1984.

[70] E. Meijering. A chronology of interpolation: From ancient astronomy to modern signal and image processing. *Proc. IEEE*, 90(3): 319–342, March 2002.

[71] P. F. A. Meyer, R. P. Westerlaken, R. Klein Gunnewiek and R. L. Lagendijk. Distributed Source Coding of Video with Non-Stationary Side-Information. *Visual Communications and Image Processing*, Beijing, 2005.

[72] H. Minc. *Nonnegative Matrices*. Wiley, New York, 1988.

[73] K. M. Misra, S. Karande, and H. Radha. Multi-Hypothesis Based Distributed Video Coding using LDPC Codes. *Allerton Conf. on Comm., Control and Comp.*, Monticello, 2005.

[74] Y. Oohama. Gaussian multiterminal source coding. *IEEE Trans. Inform. Theory*, 43(6): 1912–1923, 1997.

[75] M. Powell. *Approximation theory and methods*. Cambridge University Press, Cambridge, 1981.

[76] S.S Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): design and construction. *IEEE Trans. Inform. Theory*, 49(3): 626–643, 2003.

[77] S.S Pradhan and K. Ramchandran. Generalized coset codes for distributed binning. *IEEE Trans. Inform. Theory*, 51(10): 3457–3474, 2005.

[78] S.S Pradhan, J. Kusuma and K. Ramchandran. Distributed compression in a dense microsensor network. *IEEE Signal Processing Magazine*, 19(2): 51–60, 2002.

[79] F. Preparata and M. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New-York, 1985.

[80] R. Puri, and K. Ramchandran. PRISM: A new robust video coding architecture based on distributed compression principles. In *Proc. Of 40th Allerton Conf. on Comm., Control and Comp.*, Monticello, 2002.

[81] M. Rajeev and R. Prabhakar. *Randomized Algorithms*. Cambridge University Press, 1995.

[82] Iain E. G. Richardson. *H.264 and MPEG-4 Video Compression*. Wiley & Sons, UK, 2003.

[83] A. A. Sardinas and G.W. Patterson. A necessary and sufficient condition for the unique decomposition of coded messages. In *IRE Convention Record, Part 8*, pp. 104–108, 1953.

[84] S. A. Savari and A. Naheta. Bounds on the expected cost of one-to-one codes. In *Proc. Intern. Symposium on Inform. Theory*, p. 92, 2004.

[85] D. Schonberg, S. S. Pradhan, and K. Ramchandran. Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources. *Data Compression Conference*, Snowbird, 2004.

[86] A. Sehgal, A. Jagmohan and N. Ahuja. Scalable video coding using Wyner-Ziv codes. In *Proc. Int. Picture Coding Symp.*, San Francisco, 2004.

[87] R. Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete Comput. Geom.*, 6: 593–613, 1991.

[88] C. E. Shannon. A mathematical theory of communication. *Bell Sys. Tech. Journal*, 27: 379–423, 623–656, 1948.

[89] D. Slepian and J.K. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Inform. Theory*, 19: 471–480, 1973.

[90] B. Srinivasa Reddy and B. N. Chatterji. An FFT-based technique for translation, rotation and scale-invariant image registration. *IEEE Trans. Image Processing*, 5: 1266-1271, 1996.

[91] B. Song, O. Bursalioglu, A.K. Roy-Chowdhury and E. Tuncel. Towards a Multi-Terminal video compression algorithm using epipolar geometry. *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, Tolouse, 2006.

[92] M. Tagliasacchi, A. Majumdar and K. Ramchandran. A Distributed Source Coding based Spatio-Temporal Scalable Video Codec. *Picture Coding Symposium*, San Francisco, 2004.

[93] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites and F. Pereira. Intra mode decision based on spatio-temporal cues in pixel domain Wyner-Ziv video coding. In *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Proc.*, Toulouse, 2006.

[94] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites and F. Pereira. Exploiting spatial redundancy in pixel domain Wyner-Ziv video coding. In *Proc. IEEE Int. Conf. on Image Proc.*, Atlanta, 2006.

[95] G. Toffetti, M. Tagliasacchi, M. Marcon, A. Sarti, S. Tubaro and K. Ramchandran. Image Compression in a Multi-camera System based on a Distributed Source Coding Approach. In *Proc. Europ. Signal Proc. Conf.*, Antalya, 2005.

[96] A. Trapanese, M. Tagliasacchi, S. Tubaro, J. Ascenso, C. Brites and F. Pereira. Improved Correlation Noise Statistics Modeling in Frame-based Pixel Domain Wyner-Ziv Video Coding. International Workshop VLBV, Sardinia, 2005.

[97] E. Tuncel, P. Koulgi, S. Regunathan and Kenneth Rose. Zero-error Source Coding with Maximum Distortion Criterion. *Data Compression Conference*, Snowbird, 2002.

[98] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theory*, 13: 260–269, 1967.

[99] A. B. Wagner, S. Tavildar and P. Viswanath. Rate region of the Quadratic Gaussian Two-Terminal source-coding problem. submitted to *IEEE Trans. Inform. Theory* (Feb. 2006).

[100] G. K. Wallace. The JPEG still picture compression standard. *Comm. ACM*, 34(4): 30–44, 1991.

[101] G. A. Watson. Approximation in normed linear spaces. *J. Comput. Appl. Math.*, 121: 1–36, 2000.

[102] R. P. Westerlaken, R. Klein Gunnewiek and R. L. Lagendijk. The role of the virtual channel in distributed source coding of video. In *Int. Conf. on Image Proc.*, Genova, 2005.

[103] T. Wiegand, G. J. Sullivan, G. Bjøntegaard and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7): 560–576, 2003.

[104] H. S. Witsenhausen. The zero-error side information problem and chromatic numbers. *IEEE Trans. Inform. Theory*, 22: 592–593, 1976.

[105] H. Witsenhausen and A.D. Wyner. Interframe coder for video signals. United States Patent Number 4.191.970, 1980.

[106] A. D. Wyner. An upper bound on the entropy series. *Inform. Control*, 20: 176–181, 1972.

[107] A. D. Wyner. Recent results in the Shannon theory. *IEEE Trans. Inform. Theory*, 20(1): 2–10, 1974.

[108] A. D. Wyner. On source coding with side information at the decoder. *IEEE Trans. Inform. Theory*, 21(3): 294–300, 1975.

[109] A. D. Wyner and J. Ziv. The rate distortion function for source coding with side information at the receiver. *IEEE Trans. Inform. Theory*, 22: 1–11, 1976.

[110] A. D. Wyner. The rate-distortion function for source coding with side information at the decoder-II: General sources. *Inform. Contr.*, 38: 60–80, 1978.

[111] R. Zamir. The rate loss in the Wyner-Ziv problem. *IEEE Trans. Inform. Theory*, 42(6): 2073–2084, 1996.

[112] R. Zamir and T. Berger. Multiterminal source coding with high resolution. *IEEE Trans. Inform. Theory*, 45(1): 106–117, 1999.

[113] R. Zamir, S. Shamai and U. Erez. Nested linear/Lattice codes for structured multiterminal binning. *IEEE Trans. Inform. Theory*, 48(6): 1250-1276, 2002.

[114] X. Zhu, A. Aaron and B. Girod. Distributed compression for large camera arrays. In *Proc. IEEE Workshop on Statistical Signal Processing*, St Louis, 2003.

“I would have seen farther,
but giants were standing on my shoulders.”
– Anonymous –