Universal transmission and compression: send the model?

Aline Roumy

ANNÉE 2018

UNIVERSITÉ DE RENNES 1

UNIVERSITE BRETAGNE LOIRE

HDR / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

Domaine : Signal, Image, Vision
Ecole doctorale MathSTIC

présentée par

Aline ROUMY

préparée au centre de recherche de
Inria Rennes-Bretagne Atlantique

Universal transmission and compression: send the model?

soutenue à Rennes
le 26 novembre 2018

devant le jury composé de :

Giuseppe CAIRE
Professeur, TU Berlin / rapporteur

Pierre DUHAMEL
DR, CNRS L2S / rapporteur

Béatrice PESQUET-POPESCU
Directrice Recherche Innovation, Thalès / rapportrice

Inbar FIJALKOW
Professeure, ENSEA / examinatrice

Claude LABIT
DR, Inria Rennes / examinateur

Philippe SALEMBIER
Professeur, UPC Barcelona / examinateur

Vladimir STANKOVIC
Professeur, Univ. Strathclyde, Glasgow / examinateur

Contents

Chapter 1

Introduction

My research lies in the area of signal and image processing, digital communication, information theory, computer vision and machine learning. It tackles the question of **universality in digital communication**: digital communication refers to the problem of a source sending a digital (or digitalized) data over a noisy channel (see Fig. 1.1), and universality means that neither the source statistics, nor the channel statistics are known a priori. Another characteristic of my research is to consider the problem of communicating visual data: **image and video**.

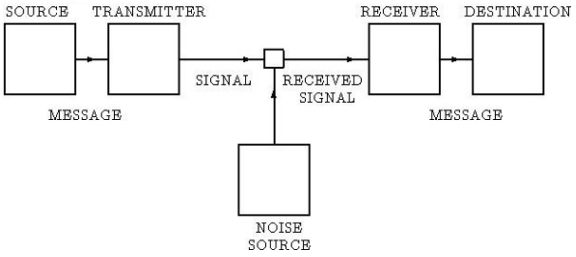

Figure 1.1: Scheme of a digital communication system as drawn in the seminal paper [87, Fig. 1].

Terminology. Different terms are used for the same idea:
- data compression = source coding,
- protection against channel noise = channel coding,
- protection against channel disturbances = protection against noise, but also against any other disturbances such as: multipath in wireless transmissions, or multiuser interferences, ...

Universal communication: a separable problem into source and channel coding. For the classical model (see Fig. 1.1) of a source with a single output and a channel with a single input and a single output, [87] showed that separation between (i) source coding, (where all the redundancy in the source is removed) and (ii) channel coding (where a minimum of redundancy is added in order to remove the effect of the channel noise) is optimal. In other words, optimizing the two problems separately and connecting the two solutions obtained in series (as in Fig. 1.2) is optimal. Moreover, this *separation principle* holds quite generally, even when the source and channel statistics are unknown [46, Th. 3.8.3, Th 3.8.4]. Based on this principle, research subjects can be dealt separately into

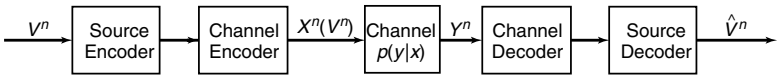

Figure 1.2: Separate source and channel encoding is optimal [26, Fig. 7.15].

source coding and channel coding.

Challenges for the universal *compression* of visual data. The question of universal data compression has been dealt with statistics and information theory. It was shown that, when the source probability distribution is unknown, the data can still be compressed. Moreover, the compression rate is the same with or without the probability distribution knowledge, when the length of the data tends to infinity. This is the case, in particular, with the Lempel-Ziv codes [108, 109]. At finite length, however, universality has a cost of the order of $O(\log(n)/n)$ [27, Th. 7.5], where n is the length of the data sequence to be processed. This is a positive result as this cost is small, and vanishes as the length n tends to infinity. However, this extra amount $O(\log(n)/n)$ of data to be transmitted due to universality holds only for very restricted classes of probability distributions. In particular, there exists no universal code for the class of stationary ergodic processes [89]. Instead, for certain subclasses, such as the identically and independently distributed (i.i.d.), or the order k Markov processes, this cost is $O(\log(n)/n)$, where the multiplicative scaling factor depends on the number of parameters needed to identify a distribution within the class [27, Th. 7.5]. This is a negative result for visual data, which are notoriously neither stationary nor ergodic.

Therefore the challenges in compressing visual data are to efficiently (i) **model** the visual data, (ii) **code the model**, and (iii) **code the data** according to the model. These problems will be discussed in Chap. 2.

Challenges for the universal *transmission* of data. If universality does not incur any loss in data compression (at least asymptotically), this is not the case for the transmission over noisy channels. Indeed, when the channel statistics are not known, the best performance is determined by the worst channel [46, Th. 3.2.1]. To compensate for this negative result, a feedback channel, if possible, is used to inform the encoder of the channel statistics. This method consists in (for the more complex case of multipath channels)
(i) sending a learning sequence known by both the encoder and decoder, (ii) estimating of the channel characteristics (noise level, channel impulse response) at the receiver, and finally (iii) sending information about the channel from the receiver to the encoder through a feedback channel. However, with the advent of turbo-codes [6], this approach had to be revisited. Indeed, turbo-codes were the first practical codes to operate at the optimal level predicted by C. Shannon in [87]. In other words, they were able to operate at a much higher noise level than all its predecessors, such that the whole communication chain, in particular the channel estimation, had to operate at a higher noise level to benefit from the decoding performance of turbo-codes.

Therefore, the challenges for the universal transmission of data are to efficiently
(i) **estimate** the channel even for high noise level and (ii) **dimension** the system. These problems will be discussed in Chap. 3.

Challenges for the universal *transmission* of *visual* data, when the separation principle does not hold. The separation principle holds if the system is unconstrained

in terms of complexity (source and channel code are indeed optimal), or delay (adding the necessary amount of redundancy to combat the channel noise is possible) [29, Sec. 2.2]. For constrained systems, joint source channel coding/decoding must be applied instead. This problems will be discussed in Sec. 3.2.

Notations:

- Upper case letters $X, Y, \ldots$ refer to random variable, random process or a source,
- Calligraphic letters $\mathcal{X}, \mathcal{Y}, \ldots$ refer to alphabet, the subset of $\mathbb{R}$ in which the random variable takes its values
- $|\mathcal{A}|$ is the cardinality of the set $\mathcal{A}$
- $X_i^j = (X_i, X_{i+1}, \ldots, X_j)$ is a (j-i+1)-sequence of random variables or a random vector
  $X^n = (X_1, X_2, \ldots, X_n)$, the index in the above notation is dropped when $i = 1$,
- Lower case $x, y, \ldots$ and $x^n, y^n, \ldots$ mean realization of a random variable, or realization of a random vector.
- $\mathcal{A}^*$ is the set of finite strings over the alphabet $\mathcal{A}$.

Chapter 2

Compression and universality

Universal compression, or universal source coding, is the problem of compressing a data source without knowledge of the source probability distribution. This occurs in particular when images and videos have to be compressed. A natural question to ask is whether universality has a cost compared to the case where the statistics are known. A second question is whether practical codes exist, which achieve the optimal performance of universal source coding. Surprisingly, the latter question is rarely covered in references on source coding for image and video. This text offers a review of these ideas. The presentation does not follow a historical order. It is rather focused on the application to image and video, by first presenting the theory of universal source coding and then describing practical implementations used in video coding. This review constitutes a contribution *per se*.

This chapter is organized as follows. First, Section 2.1 reviews the problem of lossless data compression under the condition that the probability distribution is perfectly known. Section 2.2 reviews that, in universal lossless data compression, there is an excess rate (in bits per source symbol), which vanishes as the sequence length goes to infinity. This excess rate is of the order of $O(\log(n)/n)$ for a large variety class of stationary ergodic sources, where n is the source input length. Then, two practical implementations are presented which achieve the same excess rate. Section 2.4 presents fundamental principles for image and video coding. Finally, Section 2.5 introduces my contributions, which are detailed in Sections 2.6 to 2.9. For easier reading, each main result is first summarized in a sentence in bold, and then further developed.

2.1 Lossless data compression

Let us consider a source X on the discrete alphabet $\mathcal{X}$. This source generates a random process, which is defined as a sequence of random variables $(X_n)_{n\geq 0}$ with probability distribution P. Let P_n be the marginal distribution on $\mathcal{X}^n$ of the distribution P. Now, a source sequence is encoded with a code C_n defined by an encoding function f_n and a decoding function g_n. In this section, the probability distribution P is known at both the encoder and the decoder such that the code mappings f_n, g_n depend inherently on P.

Different types of codes C_n exist to compress the input data sequence x^n. It can be with fixed (FL) or variable length (VL), and without error for all lengths$n$ (zero error) or only asymptotically (see Table 2.1 for the definitions of the properties of the code).

For a FL code, the encoding function assigns to each source sequence of length n, x^n, a fixed-length *codeword*. In other words, given the source sequence length, all codewords have the same length, such that each codeword can be seen as an index taken from an set.

Property	**Definition**
FL fixed length	the length of the coded sequence is the same for all input sequences
VL variable length	the length of the coded sequence depends on the input sequence
zero error	$\forall n \forall x^n, g_n(f_n(x^n)) = x^n$
asymptotically error free	the probability of error P_e^n tends to zero as the input sequence length $n \to \infty$

Table 2.1: Properties of lossless source codes

The *compression rate* $R \in \mathbb{R}^+$ is the ratio between the codeword length in bits and the source sequence length, and is therefore expressed in *bits per source symbol*:

$$R = \frac{\text{number of bits to represent a codeword}}{n} \quad \text{bits per source symbol.} \tag{2.1}$$

The encoding and decoding functions depend explicitly on the compression rate and are $f_n : \mathcal{X}^n \to \{1, \dots, 2^{nR}\}$, $g_n : \{1, \dots, 2^{nR}\} \to \mathcal{X}^n$, respectively. For ease of presentation, we assume that the entries of the codewords are bits. Generalization to larger alphabets requires a scaling factor to take into account the codeword alphabet size.

For a VL code, the encoding and decoding functions are $f_n : \mathcal{X}^n \to \{0,1\}^*$ and $g_n : \{0,1\}^* \to \mathcal{X}^n$. Indeed, the encoding function assigns to each source sequence x^n, a codeword, whose length depends on the sequence x^n. The compression rate is defined as the ratio between the average length of the codewords and the input sequence length:

$$R = \frac{\log \mathbb{E}_{X^n}[\ell(f_n(X^n))]}{n} \quad \text{bits per source symbol} \tag{2.2}$$

where $\ell(.)$ stands for the function giving the binary length of the codeword.

Finally, for both FL and VL codes, the *probability of error* denoted P_e^n depends on the input length n, and is the probability that the decoded sequence differs from the input sequence:

$$P_e^n = \mathbb{P}\left(g_n(f_n(X^n)) \neq X^n\right). \tag{2.3}$$

The goal of lossless source coding is to find the lowest compression rate R such that the probability of decoding error decays asymptotically to zero with the input length n, or even such that there is no error ($f_n(g_n(.))$ is the identity).

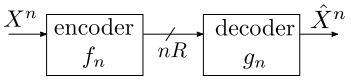

Figure 2.1: Source coding scheme.

In the class of stationary ergodic sources defined on a discrete alphabet, the optimal compression rate for lossless encoding while knowing the probability distribution, is the entropy rate. This holds for FL and VL codes. If the source X is stationary and ergodic, defined on a discrete alphabet $\mathcal{X}$, with probability distribution P known at both encoder and decoder, then the *optimal compression rate* for FL codes [46, Th 1.3.1, Ex. 1.3.2] and VL codes [46, Th 1.7.1, Rk. 1.7.3.] are both equal to:

$$\mathcal{H}(P) = \lim_{n\to\infty} \frac{1}{n} H(X^n) = \lim_{n\to\infty} \frac{1}{n} H(X_n | X^{n-1}) \quad \text{bits per source symbol,} \tag{2.4}$$

$$\text{with } H(X^n) = -\sum_{X^n \in \mathcal{X}^n} P_n(X^n) \log_2 P_n(X^n). \tag{2.5}$$

$H(X^n)$ is called the entropy of the random vector X^n with joint distribution P_n and $\mathcal{H}(P)$ is the entropy rate. The second equality in (2.4) follows because the source is stationary. Optimality here means that not only there exists a code that achieves this rate, but also that there is no code that can achieve a smaller rate.

For instance, in the special case of a an i.i.d. source, the compression rate in (2.4) reduces to the entropy of the source

$$H(X) = -\sum_{x\in\mathcal{X}} P(x)\log_2 P(x) \qquad \text{bits per source symbol,} \tag{2.6}$$

and can be achieved by either:

- a FL code [42, Th. 3.4. p 55], where each typical sequence is assigned a distinct codeword, and all non-typical sequences are assigned the same codeword,
- a VL [26, Th. 3.2.1], where the typical sequences are encoded with $nH(X^n)$ bits, the non typical sequences with length $n\log_2|\mathcal{X}|$ bits, and a flag bit is used to indicate if the sequence is typical or not,
- a VL code, where more probable symbols (or input sequences) are encoded with shorter codewords (like Huffman, Shannon [26, Sec 5.4.- 5.8] or Arithmetic code [26, Sec 13.3.]).

These optimal VL codes are zero error, whereas the optimal FL code is only asymptotically error free.

2.2 Universal source coding and its theoretical cost

This section tackles the more realistic and interesting setup, where the distribution of the source is known neither at the encoder nor at the decoder. This problem is referred to as *universal source coding*. The goal of the section is to review the optimal compression rate that can be achieved for universal source coding, and compare it to the case where the probability distribution is known. If the probability distribution is available at both encoder and decoder, the code will be called *distribution-aware code*, or *aware code*. Similarly, when the distribution is neither available at the encoder nor at the decoder, we will call the code a *distribution-unaware code*, or an *unaware code*. In this context, a distribution unaware code is said to be *universal* if it achieves the same asymptotic compression rate as a distribution aware code.

The discussion concerns stationary ergodic sources as in the previous section, where the probability distribution of the source is known (see Sec. 2.1). However, two restrictions to this distribution ensemble will be made. First, the alphabet is not only discrete as in the previous section, but it even needs to be finite. Indeed, the necessary and sufficient condition [51] for approaching a distribution, and therefore for the existence of a universal code, is not satisfied for discrete distributions even with finite entropy [43, Th. 3.2.1]. Second, the class of distributions can not be too large. Indeed, there exists no universal code for the class of stationary ergodic processes [89] or for the class of Markov processes of an unknown finite order [88].

Formally, let X be a source on the *finite* alphabet $\mathcal{X}$ with stationary ergodic distribution P. Let P_n be the marginal distribution on $\mathcal{X}^n$ of the distribution P. In the following, we will consider subclasses $\mathcal{P}$ of the class of stationary ergodic distributions. In the context of universal coding, it is clearer to write the entropy (2.6) of a source as a function of the distribution:

$$H(P) = -\sum_{x\in\mathcal{X}} P(x)\log_2 P(x) \tag{2.7}$$

FL unaware codes of memoryless sources with finite alphabet experience an excess compression rate, compared to *distribution aware* codes. Therefore, universal coding focuses on VL codes. Consider a memoryless source with distribution $P \in \mathcal{P}$. Here, $\mathcal{P}$ is the subclass of i.i.d. sources. Further assume that the set $\mathcal{Q} \subseteq \mathcal{P}$ to which P belongs, is known. With a FL code, the optimal compression rate is the worst (highest) compression rate $\sup_{Q\in\mathcal{Q}} H(Q)$ [26, Th. 11.3.1] [46, Th. 1.3.1]. This leads to an excess rate with respect to optimal distribution aware coding:

$$\text{Excess Rate} = \sup_{Q\in\mathcal{Q}} H(Q) - H(P) \qquad \text{bits per source symbol.} \tag{2.8}$$

As was seen above, even for the simple case of i.i.d. sources, there exists no FL universal codes. Therefore, universal coding focuses on VL codes.

At this stage of the development, it is important to be precise about the notion of *distribution knowledge*. The statement [26, Th. 11.3.1] [46, Th. 1.3.1] is given a different interpretation in [47, Th. 3.21] from the one formulated above. More precisely, [47, Th. 3.21] concludes the existence of FL universal codes. This is due to the fact that in the latter Theorem, it is assumed that the probability distribution is not known but that its entropy is known, which allows to determine the compression rate. Then, since the coding scheme is FL and does not depend on the probability distribution but on its rate only, it is concluded that there exists universal FL code. Here instead, it is assumed that when the distribution is not known, neither the distribution nor any statistics are known. In particular, the entropy of the true distribution is not known.

Optimizing a VL code is equivalent to optimizing a probability distribution for decoding. VL coding can lead to ambiguity while decoding a sequence of codewords, since the parsing may not be unique. Since the goal is lossless compression, only injective encoding functions are considered, which are referred to as *uniquely decodable VL codes*. Then, [26, Th. 5.5.1.] states that there is a one-to-one correspondence between a *uniquely decodable VL code* with codeword set $\mathcal{C}$ and the set of codeword lengths $\{\ell(c), c \in \mathcal{C}\}$ satisfying the Kraft inequality

$$\sum_{c\in\mathcal{C}} 2^{-\ell(c)} \leq 1. \tag{2.9}$$

Formally,

- For any uniquely decodable code $\mathcal{C}$, the set of codeword lengths satisfies (2.9)
- Conversely, if the mapping $\ell : \mathcal{C} \to \mathbb{N}$ satisfies (2.9), then there exists a uniquely decodable code with length mapping $\ell : \mathcal{C} \to \mathbb{N}$.

Moreover, this length mapping defines a probability distribution. Indeed,

- if the mapping $\ell : \mathcal{C} \to \mathbb{N}$ satisfies (2.9), then $Q(c) = 2^{-\ell(c)}$ is a quasi-probability distribution since $\sum_{c\in\mathcal{C}} Q(c) = \sum_{c\in\mathcal{C}} 2^{-\ell(c)} \leq 1$. One only needs to add a codeword s.t. $\sum_c Q(c) = 1$ on the extended set to build a probability distribution.
- Conversely, if Q is a probability distribution on $\mathcal{C}$, then $\ell(c) = \lceil -\log_2(Q(c)) \rceil$ is a length mapping that satisfies (2.9), since $\sum_{c\in\mathcal{C}} 2^{-\ell(c)} \leq \sum_{c\in\mathcal{C}} Q(c) = 1$.

Therefore, there is a one-to-one correspondence between probability distributions and uniquely decodable codes. Optimizing a code is thus equivalent to finding the best probability distribution for decoding.

There exists a practically implementable VL code, called arithmetic code, that is optimal. Indeed this code belongs to the the set of prefix codes, a subset

of uniquely decodable codes, but still achieves the same compression gain as uniquely decodable codes. Moreover, it is a practical scheme as it allows to encode a sequence sequentially and use a distribution learned on the fly. A *prefix code* is such that no codeword is a prefix of any other codeword. Consequently, the code allows instantaneous or more precisely sequential decoding, and the set of prefix codes is a subset of the uniquely decodable code set. Interestingly, prefix codes and uniquely decodable codes achieve the same optimum compression gains. Indeed, prefix codes and uniquely decodable codes satisfy the Kraft inequality [26, Th. 5.2.1, Th. 5.5.1]. So, without loss of optimality, we can consider prefix codes.

One possible prefix code is the *arithmetic code* [26, Sec. 13.3] that encodes a sequence using fixed-precision arithmetic. This method is appreciated for its complexity that is linear in the length of the sequence, but also because it allows to encode sequentially and use a source distribution that is learned on the fly. Therefore arithmetic coding is naturally used in universal compression.

Criteria for optimal VL universal code. To evaluate the quality of a code built from an arbitrary distribution Q_n, we compute the difference between the codeword lengths obtained with the encoding distribution Q_n, and the optimal code (built from the true distribution P_n). This leads to the *loss*:

$$\mathcal{L}(x^n) = -\log Q_n(x^n) + \log P_n(x^n) \qquad \text{bits per source sequence of length } n. \tag{2.10}$$

This loss (2.10) is expressed in *bits per source sequence of length* n, as it computes the number of additional bits needed to compress a whole sequence of length n. From this loss function (2.10), different criteria can be derived to optimize the code distribution Q_n. Indeed, we can either optimize for the worst input sequence (regret) or for a criterion averaged over all input sequences (redundancy). Loss, regret and redundancy are the terms used in statistics. [27, Sec. 6.1] uses a different terminology. The Loss is called Pointwise redundancy, the Redundancy is called Expected redundancy, whereas the regret is called the Maximum redundancy.

– **The universal code optimal according to the *Minimax regret* (worst case over the sequence and over the distribution) is given by the Normalized Maximum Likelihood (NML) distribution.**
First, the loss is computed for the worst input sequence, which is called the *regret* (see Table 2.2). Then, this regret is evaluated for the worst distribution of the class. The reason for computing this *Maximum regret* is that it allows to analyze uniform convergence over the distribution class $\mathcal{P}$. Finally, the code distribution that minimizes the Maximum regret, is called the optimal distribution for the *Minimax regret*, which is defined in Table 2.2.

Criterion	**Definition**
Regret	$R^*(Q_n\|P_n) = \max_{x^n} \log \dfrac{P_n(x^n)}{Q_n(x^n)}$
Maximum regret	$R^*(Q_n\|\mathcal{P}) = \sup_{P\in\mathcal{P}} \max_{x^n} \log \dfrac{P_n(x^n)}{Q_n(x^n)}$
Minimax regret	$R^*(\mathcal{P}) = \inf_{Q_n} \sup_{P\in\mathcal{P}} \max_{x^n} \log \dfrac{P_n(x^n)}{Q_n(x^n)}$

Table 2.2: Definition of the Minimax regret

For any class of distributions $\mathcal{P}$ over a finite alphabet $\mathcal{X}$, the minimum is attained [27, Th. 6.2] when Q_n is the *normalized maximum likelihood distribution* (NML_n):

$$\arg\inf_{Q_n}\sup_{P\in\mathcal{P}}\max_{x^n}\log\frac{P_n(x^n)}{Q_n(x^n)} = Q_n^{\mathrm{NML}} : z^n \mapsto \frac{\sup_{P\in\mathcal{P}} P_n(z^n)}{\sum_{y^n}\sup_{P\in\mathcal{P}} P_n(y^n)} \tag{2.11}$$

So, in practice, given a set of distributions $\mathcal{P}$, the optimal code that achieves Minimax regret (uniformly over the class $\mathcal{P}$) can be computed by:

1. for each input sequence z^n, compute the length of the coded sequence $-\log Q_n^{\mathrm{NML}}(z^n)$
2. find the codewords as in [26, Th. 5.5.1.] (Kraft inequality)

– **The universal code optimal according to the *Minimax redundancy* (average case over the sequence and worst case over the distribution) is given by a mixture distribution.**
The regret computes the worst case loss. One can instead compute the average loss, which is also called the *redundancy*. Then, as for the regret, the Minimax redundancy is derived, see Table 2.3, where $D(P_n||Q_n)$ stands for the Kullback-Leibler divergence between the probability distributions P_n and Q_n.

Criterion	**Definition**
Redundancy	$\bar{R}(Q_n\|P_n) = \mathbb{E}_{X^n\sim P_n}\log\frac{P_n(X^n)}{Q_n(X^n)} = D(P_n\|\|Q_n)$
Maximum Redundancy	$\bar{R}(Q_n\|\mathcal{P}) = \sup_{P\in\mathcal{P}} D(P_n\|\|Q_n)$
Minimax Redundancy	$\bar{R}(\mathcal{P}) = \inf_{Q_n}\sup_{P\in\mathcal{P}} D(P_n\|\|Q_n)$

Table 2.3: Definition of the Minimax redundancy.

The Maximin theorem allows to compute the optimal distribution [27, Sec 7.2]. If the set of probability distribution is equipped with a probability measure μ, the inf and sup can be exchanged (second equality in (2.12)). Then, the infimum is attained for the mixing measure (2.14). This estimate is analog to the Bayesian method in statistics. Finally, the optimal coefficients $\tilde{\mu}$ for the mixture are obtained by solving the supremum (2.13).

$$\bar{R}(\mathcal{P}) = \inf_{Q_n}\sup_{P\in\mathcal{P}} D(P_n||Q_n) \qquad = \sup_{\mu}\inf_{Q_n}\int_{\mathcal{P}} D(P_n||Q_n)d\mu(P_n) \tag{2.12}$$

$$= \sup_{\mu} D(P_n||Q_n^*(\mu)) \qquad = D(P_n||Q_n^*(\tilde{\mu})) \tag{2.13}$$

where

$$Q_n^*(\mu) = \int_{\mathcal{P}} P_n d\mu(P_n) \tag{2.14}$$

When the class of distributions is parametrized by a parameter θ, the optimal mixing measure $\tilde{\mu}$ in (2.13) is the input distribution achieving the capacity of a channel with input θ and output x^n, the sequence to be compressed [26, Th 13.1.1] [27, Sec. 7.2].

For a large variety of stationary ergodic processes, Minimax regret and minimax redundancy grow as $\frac{\#\text{ of parameters}}{2}\log n$. Therefore, for these classes, VL

universal coding does not incur any asymptotic loss with respect to distribution aware coding. In this paragraph, a function which is asymptotic equivalent to the minimax regret and to the minimax redundancy is reviewed. To do so, we can first notice that minimax redundancy $\bar{R}_n(\mathcal{P})$ and regret $R_n^*(\mathcal{P})$ satisfy $\forall n, \forall \mathcal{P}$

$$\bar{R}_n(\mathcal{P}) \leq R_n^*(\mathcal{P}) \tag{2.15}$$

which follows from the definition of the redundancy (average case), and of the regret (worst case). Then, the strategy will be to derive a lower bound for the minimax redundancy, and then an upper bound for the regret, by mean of a specific code construction. However, to do so, we need to restrict to a subclass of probability distributions $\mathcal{P}$ within the class of stationary ergodic processes, since the redundancy does not exist [89] for the whole class of stationary ergodic sources.

A lower bound for minimax redundancy. For the parametric class of processes $\mathcal{P} = \{P_\theta, \theta \in \Theta \subseteq \mathbb{R}^k\}$ [27, Th. 7.4], the minimax redundancy is bounded below by

$$\frac{k}{2} \log n - K \leq \bar{R}_n(\mathcal{P}) \tag{2.16}$$

where K is a constant, provided there exists an estimator $\hat{\theta}_n : \mathcal{X}^n \to \Theta$, whose error is upperbounded by $O(1/n)$. In other words, provided that the class of distribution $\mathcal{P}$ is learnable, a lower bound of the minimax redundancy scales as $\frac{\#\text{ of parameter}}{2} \log n$. This result can be instantiated for many subclasses, such as those in Table 2.4: i.i.d., Markov and compound process [47, Sec. 3.1.9.]. Given a family of memoryless probability distributions $\mathcal{P} = \{P(.|s), s \in \mathcal{S}\}$ defined on the alphabet $\mathcal{X}$, a *compound* source is such that a probability distribution is chosen and used from the beginning to the end.

An upper bound for minimax regret. The upper bound follows an achievability argument. More precisely, the code based on the NML distribution is used, and by definition of the minimum, the minimax regret is necessarily smaller than the Maximum regret achieved by NML. Interestingly, the resulting upper bound for minimax regret achieves the same growth rate as (2.16). Therefore, the minimax regret and redundancy are asymptotically equivalent and satisfy [27, Th. 7.5.]

$$r(n) - K_1 \leq \bar{R}_n(\mathcal{P}) \leq R_n^*(\mathcal{P}) \leq r(n) + K_2 \tag{2.17}$$

where the function $r(n)$ and the constants K_1, K_2 depend on the class of distributions considered $\mathcal{P}$, see Table 2.4.

Class $\mathcal{P}$	**Asymptotic equivalent function $r(n)$**
i.i.d. process [27, Th. 7.5]	$\frac{\lvert\mathcal{X}\rvert - 1}{2} \log n$
mth order Markov process [27, Th. 7.5]	$\lvert\mathcal{X}\rvert^m \frac{\lvert\mathcal{X}\rvert - 1}{2} \log n$
compound process with $\lvert\mathcal{S}\rvert$ states from [27, Th. 7.4]	$\lvert\mathcal{S}\rvert \frac{\lvert\mathcal{X}\rvert - 1}{2} \log n$

Table 2.4: Asymptotic equivalent function to the Minimax regret and Minimax redundancy for some class of processes.

An important consequence of this result is that, for all the classes of processes in Table 2.4, the excess rate (in bit per source symbol) vanishes asymptotically:

$$\lim_{n\to\infty} \frac{1}{n} \bar{R}_n(\mathcal{P}) = \lim_{n\to\infty} \frac{1}{n} R_n^*(\mathcal{P}) = 0, \tag{2.18}$$

and the convergence rate of the excess rate is

$$\frac{\#\text{ of parameters}}{2} \cdot \frac{\log n}{n}. \tag{2.19}$$

So, VL universal coding does not incur any asymptotic loss with respect to distribution aware coding.

The two-part algorithm (estimate the distribution, encode it, and encode the data with the estimated distribution) achieves the same growth rate as the minimax regret for a large variety of probability class $\mathcal{P}$. Unfortunately, the direct implementation of the one-part code using NML distributions, which achieves the optimal redundancy, is impractical. Indeed, $\{Q_n^{\mathrm{NML}}\}_n$ does not define a process distribution, since Q_n^{NML} is not the marginal of Q_{n+1}^{NML}. So, a practical scheme based on NML must process the sequence as a whole. More precisely, for a given input sequence length n, one needs to (i) compute the probability distribution Q_n^{NML}, (ii) determine the length of each input sequence (i.e. $\forall x^n$, compute $-\log Q_n^{\mathrm{NML}}(x^n)$), and finally (iii) find the codewords as in [26, Th. 5.5.1.] (Kraft inequality).

Instead, the *two-part algorithm* (also called two stage in [27]), is the very simple and natural way to implement universal coding [65]. It consists of:

- estimating the distribution, and encoding it.
- encoding the data sequence x^n according to the estimated distribution.

The difficulty of the two-part algorithm lies in the encoding of the distribution [45, p.16, chap. 5]. First attempts to solve this problem considered avoiding encoding the distribution, which led to the one-part universal coding scheme (2.14) and (2.11). Fortunately, the distribution encoding problem has been solved in [4]. This allowed to show that the two-part algorithm achieves the same growth rate as in Table 2.4 for a large variety of distribution class [45, chap. 15]. For instance, for the class of i.i.d. processes, the optimal distribution estimate is the type (that counts the occurrence of each symbol), and the growth rate is $\frac{|\mathcal{X}|-1}{2} \log n$ [27, Example 6.1].

Historically, the two-part algorithm was proposed in [65], before the one-part algorithm [66]. Indeed, coding in two parts seems a very natural and practical way to tackle universality. However, it leads to the difficult problem of finding a good code for the distribution. This was the motivation for finding a one part code, that avoids to code the distribution.

Another practical and minimax optimal two-part algorithm: encode the ith symbol based on the distribution estimated from the first $i-1$ symbols. Another practical scheme consists in a sequential learning and encoding approach. At iteration i,

- estimate the distribution based on the first $i-1$ symbols of the sequence,
- encode the i-th symbol of the sequence based on the estimated distribution.

Without loss of optimality, the encoding of the sequence can be performed with a prefix code which allows sequential encoding such as arithmetic coding (see the Paragraph on prefix codes p. 12), such that the decoder, upon decoding of the i-th symbol has access to the first $i-1$ symbols and can reconstruct the distribution estimate. Therefore, there

is no need to encode the distribution estimate. This represents a gain with respect to the two-part algorithm described above. However, the encoding is suboptimal since the distribution estimate is less accurate. On the whole, both effects compensate, and it is shown that this sequential encoding achieve the same growth rate as the NML scheme, [27, Th. 6.3] for i.i.d. processes, and [27, Th. 6.5, Th 6.6] for Markov processes.

2.3 Universal source coding in a wide set of distributions: the model selection problem and the minimum description length principle

Universal coding over large distribution sets (for which minimax criteria don't exist) leads to a model selection problem. Universal coding refers to the problem of compressing an input sequence with minimum coding length, when the distribution of the input sequence is unknown. Section 2.2 showed that we can construct universal codes, i.e. codes whose length has the same growth rate as a distribution-aware code, according to a minimax criterion, i.e. with uniform convergence over the set of possible distributions. Unfortunately optimizing with the minimax criterion can only be solved within a rather small set of distribution. Indeed, for large classes such as the set of ergodic stationary processes [89] or the set of Markov processes of an unknown finite order [88], quantitative characterizations of minimax criteria do not exist. Instead, the search spaces must be small set of distributions, (for instance the set of i.i.d processes, or the set of Markov processes of order m, for some fixed m).

One way to deal with larger classes, is to remove the uniform convergence condition. The goal becomes: seek a code that achieves the same asymptotic length as a distribution-aware code, but non-uniformly i.e. with a convergence rate that depends on the distribution. Such a code is said to be *weakly universal*, and by contrast a *strongly universal* code refers to a universal code with *uniform* convergence over the distribution set. For instance, the Lempel-Ziv code [108, 109] is weakly universal over the set of stationary ergodic processes defined over a finite alphabet [43, p. 31].

There is another way to construct a universal code ranging over a wide set of distributions without completely sacrificing strong universality. It consists in selecting non-uniformly the best distribution class, while selecting uniformly the distribution within the class. This method is referred to as the Minimum Description Length (MDL) principle [65, 45, 5], and is one way to solve the more general *model selection problem*.

To formalize MDL, we review some terminology. Following [45, p. 15] [86, p. 14], a *model* refers to a set of probability distributions of the same form, e.g. the "i.i.d. model", or the "first-order Markov model". A *model class* instead refers to a family of models, e.g. "the Markov class" (the model class of all Markov chains of each order). An *hypothesis*, (also called point hypothesis in [45, p. 15]) is an instantiation of a model. It refers to a single distribution, e.g. a Markov chain of a fixed order, with all parameter values specified.

The idea behind MDL is that "the statistical model best fitting to the data is the one that leads to the shortest description, taking into account that the model itself must also be described" [27, Sec. 8.2]. This principle can be instantiated with a two part universal

code, and the optimal model is then:

$$q_{MDL} = \arg\min_{q\in\mathcal{Q}} \underbrace{\underbrace{\underbrace{L(q)}_{\text{one model}} + \underbrace{L(Q_q)}_{\text{one distribution within the model}}}_{\text{one hypothesis}} + \underbrace{L(x^n|Q_q)}_{\text{data}}}_{} \tag{2.20}$$

where $L(q)$ bits allow to identify the model $\mathcal{M}_q$ within the model class,
$L(Q_q)$ bits allow to identify one distribution Q_q within the model $\mathcal{M}_q$,
$L(x^n|Q_q)$ bits allow to encode the sequence x^n using the coding distribution Q_q.
The notation L refers to both coding and counting the length of the coded sequence. To illustrate these definitions with an example, we can consider the Markov class $\{\mathcal{M}_q\}_{q\in\mathcal{Q}}$, where q stands for the order of the Markov distribution. Here, model selection is therefore equivalent to identifying the chain order that best fits the data. To identify an hypothesis, i.e. a distribution, first the order has to be coded with $L(q)$ bits, and then the parameter of the distribution with $L(Q_q)$ bits. If the chosen order is 0 (i.i.d. process), the parameters of the distribution are the type (the histogram). Finally, the data sequence $x^n = (x_1, ..., x_n)$ is compressed, with an arithmetic code [26, Sec. 13.3].

The MDL principle can also be implemented with a one part universal code [66] and the model selection becomes:

$$q_{MDL} = \arg\min_{q\in\mathcal{Q}} \underbrace{L(q)}_{\text{one model}} + \underbrace{L(x^n|\mathcal{M}_q)}_{\text{data}} \tag{2.21}$$

where $L(x^n|\mathcal{M}_q)$ is the codelength of the data obtained with a minimax optimal code. However, in the following, we will rather focus on two-part universal code, as there exists practical implementations of these two-part codes, that remain minimax optimal.

The model selection problem results from the fact that there is no redundancy rate for large classes of processes. The question of model selection arises in many problems where a task has to be performed, which depends on the distribution of the data, and when this distribution is unknown. Compression, as discussed here, is an example, but any learning task such as prediction or classification are other examples. A very natural approach is then to first estimate the distribution or the function (for prediction, classification), and then process the data. The estimation then poses a new question: how many parameters in the function, or in the distribution should be chosen? The discussion about universal compression and the construction of the one part universal code shows that there is no need to estimate the distribution, and therefore no need to estimate the order of the model if the set of the possible distributions is limited. In other words, it is only because redundancy rates don't exist for large classes of processes that indeed model selection is a problem.

2.4 Universal compression of real data: image and video

Compressing images and videos is a universal compression problem, as the joint distribution of the pixels is not known. However, image and video compression have three more specificities that have not been tackled in Sec. 2.2:

1. The model of the data is unknown and definitely far more complex than stationary ergodic. For instance, the pixels are highly dependent (within an image but also between successive images), and this dependence is not stationary.
2. The encoding and mostly decoding complexities must be low.
3. The compression is lossy.

Efficient lossy compression can be realized with uniform scalar quantization followed by lossless compression. This incurs a loss of only 1.53 dB or 0.255 bit/sample with respect to the optimal rate-distortion (RD) function irrespective of the distribution. Lossy coding refers to the case where the decoded vector is only an approximation of the original source. Rate Distortion (RD) theory derives the information theoretical bounds for lossy compression. In particular, the RD function $R(D)$ gives, for a given maximum average distortion D, the smallest compression rate among all possible codes. It also shows that vector quantizer achieves the optimal RD performance, provided the vector size tends to infinity. However, complexity of vector quantization prevents its use in practical system such as video compression algorithms.

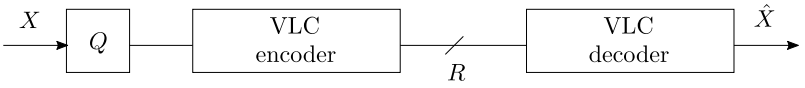

Figure 2.2: Efficient lossy compression: Uniform scalar quantizer and VL lossless compression.

A remarkable fact [44, page 2334] is that, at high rates, uniform scalar quantization with VL coding (see the scheme in Fig. 2.2) attains performance within 1.53 dB (or 0.25 bit/sample for low distortion) of the best possible RD function. Moreover the result holds quite generally: for any source distribution, for sources with memory, for any distortion function (nondecreasing in the error magnitude). Last but not least, the result holds even approximately when the rate is not too large (see Fig. 2.3), and holds exactly for exponential densities (provided the quantization levels are placed at the centroids). Due to these results, uniform scalar quantization with VL coding (also called entropy-constrained scalar quantization) is extensively used in modern video coding.

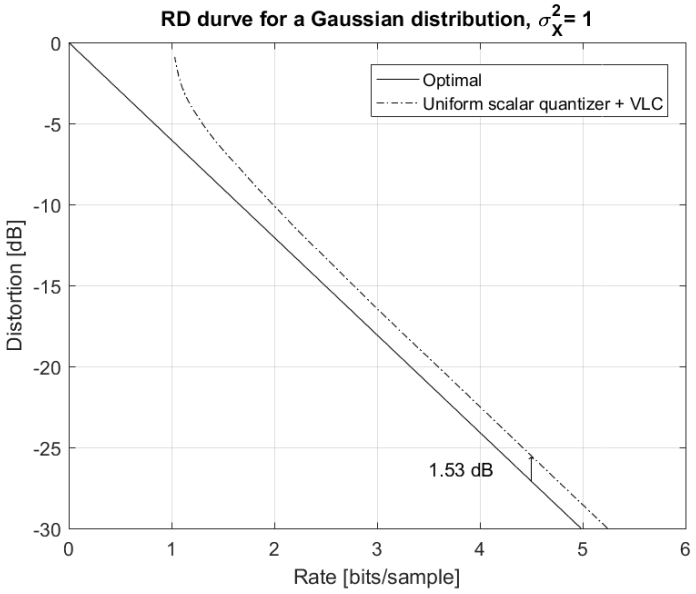

Figure 2.3: RD functions for a Gaussian source. At high rates, the RD curve of the uniform scalar quantizer with VL coding (dashed line) is 1.53 dB above the optimal RD curve (solid line). Moreover, the highrate regime occurs for relative small values of R ($R > 2$).

The complexity issue and the pixel dependence issue are tackled with a divide and conquer approach: Transform coding, where the transform removes the dependencies between the pixels, and the entropy coder the redundancy due to the non-uniformity of the pixel intensity distribution. Motivated by its quasi-optimality (within 1.53 dB), we stick to uniform scalar quantization with VL code. A consequence is that the VL code becomes the element that needs to exploit the dependency of the quantized samples. Indeed, not exploiting the memory incurs a loss. More precisely, the independence bound on entropy [26, Th. 2.6.6] states, that for a random discrete vector X^n, representing the quantized samples, the joint entropy satisfies

$$H(X^n) \leq H(X_1) + \ldots + H(X_n) \tag{2.22}$$

with equality iff the random variables $X_1, \ldots, X_n$ are mutually independent. So, lower compression rate can be achieved when the data are jointly compressed (joint entropy (LHS)), rather than separately (sum of the individual entropies (RHS)). However, this is a negative result with respect to the complexity. Indeed, to achieve the joint entropy as a compression rate, one needs to store the joint probability distribution, and this storage grows exponentially with the number of symbols considered. The complexity here is mostly a storage issue, as an efficient implementation based on conditional probability distributions and arithmetic code exists.

A positive interpretation of (2.22) exists, by exploiting the condition of equality. Indeed, if the vector x^n is transformed with an invertible mapping $T : x^n \mapsto y^n = Tx^n$ such that the transformed coefficients are nearly independent, then

$$H(X^n) = H(Y^n) \lesssim H(Y_1) + \ldots + H(Y_n) \tag{2.23}$$

where the first equality is the case of equality in the data processing equality [26, Ex. 2.2 and 2.4], and the second almost equality follows from the independence bound in the joint entropy (2.22). So, applying a transform, allows to decorrelate the symbols such that separate encoding of the symbols becomes nearly optimal.

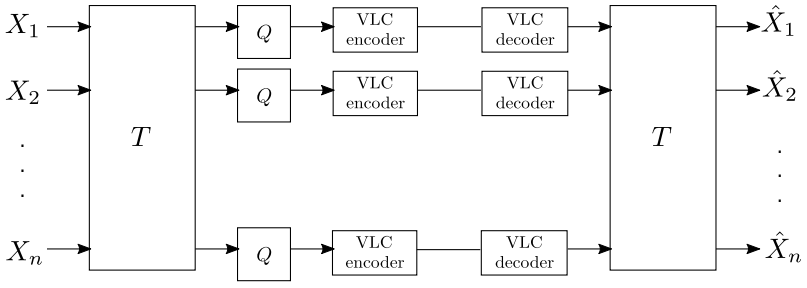

Figure 2.4: Transform coding.

Optimal transform for compression of a Gaussian source with perfect distribution knowledge is the Karhunen-Loève transform (KLT). The most popular result related to the optimality of KLT is in approximation theory [86, Lemma 23.1 Th 23.2]. There it is shown that, among all linear transforms, KLT provides a low rank approximation with minimal expected distortion with the original vector. This version [86, Lemma 23.1 Th 23.2] of the optimality of KLT is interesting as it shows that KLT is not only optimal among all orthogonal transforms [52, prop 7.1], but also more generally among all linear transforms.

Less known is the optimality of KLT in RD theory. More precisely, among all linear transforms and at a given rate, the Karhunen-Loève transform (KLT) minimizes the expected distortion of a Gaussian vector [52, prop 7.2]. Then, each component of the transformed vector can be encoded separately using uniform scalar quantization and VL code [52, prop 7.2][26, Th. 10.3.3]. The rate allocation between the components is determined by the eigenvalues of the covariance matrix, and more rate is allocated to the components corresponding to the largest eigenvalues. Optimality of KLT is shown for Gaussian vectors because closed form expression of the RD function of Gaussian vector exists. However, it may hold for more general processes. Indeed, in approximation theory, the optimality of KLT holds irrespective of the signal distribution. Therefore, this technique, called transform coding, is widely used in image and video coding.

Universal transform. When the distribution is not known, the covariance matrix can be estimated from the data, and KLT is used with this estimated covariance matrix. This is called PCA (principal component analysis). However, the estimated covariance matrix needs to be sent, which might be very costly in the case of non stationary sources. Instead, a preferred technique in video coding is to use an approximation of the KLT, the discrete cosine transform (DCT), which is signal independent (no need to send any information about the transform), and is optimal for Markov chains of order 1, with high correlation [52, p. 389].

Predictive coding. Another divide and conquer approach is predictive coding. Exploiting the memory is not based on a linear transform, but on a linear filter. There, a linear prediction based on past reconstructed values is removed from the samples and the resulting prediction residual is quantized. The primary goal is to decorrelate the samples, as in Transform coding. Another interpretation, is that the predictor reduces the variance of the variable to be scalar quantized [44, page 2331]. Finally, both transform and predictive coding are equivalent asymptotically i.e. when the number of coefficients of the transform and of the prediction filter both go to infinity [52, p. 403].

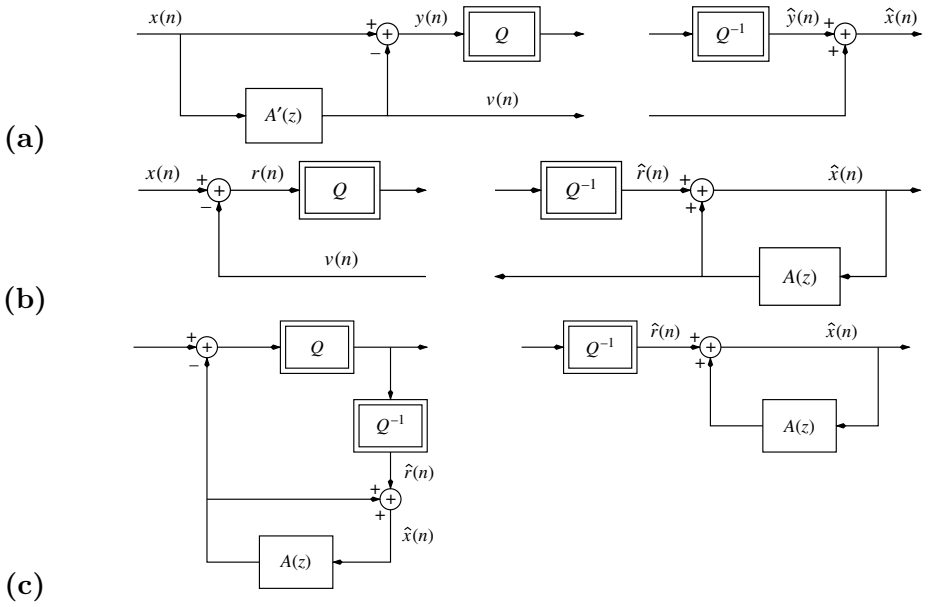

Figure 2.5: Predictive coding explained in Figures, from [57, Fig. 1.3 and 1.4]. The principle of predictive coding is best explained with (a). A prediction $v(n)$ of the current symbol $x(n)$ is computed from the input sequence $\{x(n)\}_n$, with the filter $A'(z)$, and is subtracted from the current symbol $x(n)$ to yield the residue $y(n)$. This residue is then quantized, and this is the compressed version of $x(n)$. This scheme is however impractical as it requires to send the prediction $v(n)$. To avoid sending additional data, the prediction is computed on the quantized coefficients at the decoder (b), but this scheme is still not satisfactory as it requires a feedback loop. Finally, the feedback loop is avoided in (c), by duplicating the decoder at the encoder. This scheme also avoids sending any additional data (neither feedforward nor feedback).

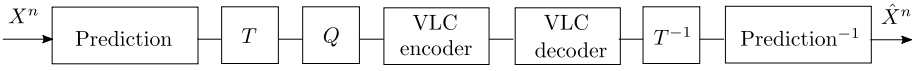

Figure 2.6: Overall video compression scheme.

2.5 My contributions to universal compression: introduction

Compression for images and videos requires to construct, for each part of the image where the distribution is assumed to be stationary and ergodic, a universal code that consists of: the identification of the model ($q \in \mathcal{Q}$), then the identification of the distribution within the model ($Q_q \in \mathcal{M}_q$) and finally the encoding of the sequence with this distribution as in (2.20). Here, the word *model* is used in the sense in which it is usually employed in statistics, i.e. a family of distributions. By contrast in information theory, a *model* refers to a single distribution and a *model class* to a family of distributions [45, p. 175]. In the following, I will use the information theoretical terminology unless otherwise specified, because it allows to use the classical denomination "model based coding".

With this new terminology, the length of the description (2.20) becomes

$$\underbrace{\underbrace{L(q)}_{\text{one class}} + \underbrace{L(Q_q)}_{\text{one distribution}}}_{\text{one model}} + \underbrace{L(x^n|Q_q)}_{\text{data}} = \underbrace{\ell(C_{\mathcal{Q}}(q)) + \ell(C_q(Q_q))}_{\text{one model}} + \underbrace{\ell(C(x^n|Q_q))}_{\text{data}} \tag{2.24}$$

where $C_{\mathcal{Q}}$ stands for the code of the class, C_q stands for the code of the distribution within the class, and C for the code of the data sequence. To make it more concrete, $(C_{\mathcal{Q}}, C_q)$ can, for instance, include the identification of the predictor within a set of predefined predictors (intra mode), the motion vectors, a flag bit to determine whether a block is inter coded or rather intra coded. C contains all the encoding steps of the data sequence: prediction, transform, entropy coder.

Statistics suggest to use a worst case criterion with respect to the true distribution of the data (see Sec. 2.2). In information theory instead, a classical criterion is rather an average criterion. For instance, in video standardization, the overall compressed length is averaged over a set of typical videos which is determined at the beginning of the standardization process. The optimization can then be written as:

$$\min_{C_{\mathcal{Q}}, C_q, C} \sum_{P \in \mathcal{P}} \mathbb{E}_{X^n} \left[\min_{q \in \mathcal{Q}} \underbrace{\ell(C_{\mathcal{Q}}(q)) + \ell(C_q(Q_q))}_{\text{one model}} + \underbrace{\ell(C(X^n|Q_q))}_{\text{data}} \right] \tag{2.25}$$

There are many difficulties in solving this optimization problem. The first open issue is the optimal code C for the data sequence under complexity constraint. In fact, when there is no complexity constraint, the optimal code C for the data sequence is well known: it is an entropy coder, which exploits both redundancies due to the distribution and the memory. If the complexity must be low, then a divide and conquer approach is applied (see Sec. 2.4). There, the redundancy due to the memory is exploited by a transform, and/or a predictor. However, the memory is not optimally exploited as the images are not Gaussian distributed and the conditions for the optimality of the KLT are not satisfied. In other words, after the transform, the data are decorrelated but not independent. So to compensate for the suboptimality of the divide and conquer approach, the entropy coder takes into account remaining dependencies, but in a limited way as only local dependencies are exploited in the context based arithmetic coder Cabac [55].

The second difficulty is to find a good code $(C_{\mathcal{Q}}, C_q)$ for the model [45, p 17]. This difficulty was indeed a motivation for the work on the one part code [66], which avoids the encoding of the model. Another difficulty related to the code $(C_{\mathcal{Q}}, C_q)$ for the model, is that it requires to determine the set of possible models $\mathcal{Q}$. Hence the challenges in compressing visual data are to efficiently:

(i) determine a **model class** $\mathcal{Q}$ relevant for visual data,
(ii) **code the model** $(C_{\mathcal{Q}}, C_q)$, and
(iii) **code the data** according to the model C.

My contributions were to propose processing tools adapted to the images by trading between the description lengths of the model and the data. Extreme cases were developed, where either all the description was devoted to the model only, or to the data only. Moreover, I developed user-centric video compression schemes where the characteristics of the user are taken into accounts. These characteristics can be additional limitations, such as low complexity at the decoder side (distributed video coding), or, rather new functionalities offered to the user such as free navigation within a multiview scene (free viewpoint television).

2.6 Coding without sending the model: a fixed transform learned on the data

Publications related to this topic:

[30] T. Dumas, A. Roumy, and C. Guillemot, Shallow sparse Autoencoders versus sparse coding algorithms for image compression, in IEEE International Conference on Multimedia and Expo (ICME), 2016.

[31] T. Dumas, A. Roumy, and C. Guillemot, Image compression with stochastic Winner-Take-All Autoencoder, in ICASSP (IEEE International Conference on Acoustics, Speech, and Signal Processing), 2017.

[32] T. Dumas, A. Roumy, and C. Guillemot, Autoencoder based Image compression: can the learning be quantization independent? in ICASSP (IEEE International Conference on Acoustics, Speech, and Signal Processing), 2018.

The model of an image requires a long description, because the model is complex, but also because it changes rapidly, such that sending the model is very costly. Therefore, fixed transforms are used in order to avoid sending the model. For instance, Jpeg image compression uses DCT, and Jpeg 2000 wavelets. The drawback of these transforms is that they are optimal for very restricted distribution families: DCT is optimal in the RD sense for Gaussian Markov processes of order 1 with high correlation [52, p. 389 et Prop 7.2], and wavelets for sparse Markov processes of order 1 [58]. However, pixels do not satisfy these optimality conditions. So, additional transforms have been proposed to further decorrelate the transform coefficients, by exploiting the distribution of the transform coefficients. For instance, the coding schemes [72, 71] are nonlinear wavelet transforms based on the Generalized lifting scheme [91]. To avoid sending the model, the parameters of the filter are learned for a class [73] (in that case, the coding scheme is specific to a type of images such as remote sensing images), or derived from the context [70]. In all these schemes, the filters are designed such as to reduce the encoding rate (more precisely the energy of the coefficients), but not in a RD sense.

We therefore proposed to learn a transform with a **deep neural network** (Deep NN) architecture on a large image dataset, where the transform is learned in order to optimize a RD criterion. Moreover, the characteristics of the uniform scalar quantizer are explicitly used in the learning. The learning is performed offline, and the optimized transform is stored at both the encoder and the decoder. One drawback of learned NNs is that they are very much dedicated to a task. For instance, previous constructions of NN based transform were tuned for one RD tradeoff. Therefore, we also constructed a **unique transform** for the whole range of distortions.

A first remarkable fact regarding the RD performance of our NN based transform coding scheme (see Fig. 2.7), is that the learned transform outperforms Jpeg2000. Indeed, Fig. 2.7 shows the RD performance of a compression scheme based on different learned transforms (yellow, green, and red curves), that all outperform Jpeg2000 (black curve). The learned transform coding still cannot bridge the gap with predictive coding (intra coding of H265), except at very low bitrate, but in Intra coding the compression not only relies on a transform but also on prediction, which is known to outperform the transform coding scheme. A second remarkable fact is that both Jpeg2000 and H265 use context-based entropy coders, where the context allows to take into account some memory in the data at the VL encoder input. In our approach instead, a very **simple symbolwise entropy coder** is used.

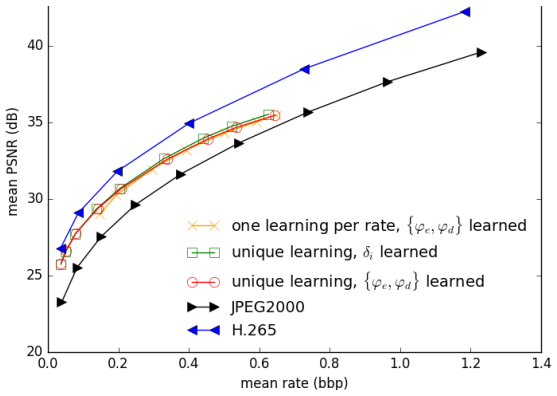

Figure 2.7: RD performance of Transform coding, with a deep transform learned with a RD criterion on a dataset of images [32, Fig. 5]. Performance averaged over 24 images.

In conclusion, motivated by the fact that models of images are complex and change very rapidly, we developed a strategy based on a fixed transform to avoid sending the model, which may require a long description due to both its complexity and its dynamic. Unlike previous standards, the transform is not longer optimized for a class of theoretical distributions for which optimality results can be derived. Instead, the transform is learned, to get closer to the true model of images. Also a deep architecture is used for its approximation power. Finally, the **fixed deep transform**, which we designed, and which does not require to send any information about the model, outperforms classical transform such as DCT and Wavelets, even if a very simple symbolwise entropy coder is used (whereas both Jpeg and Jpeg2000 use context based entropy coders).

2.7 Coding by sending the model only: super-resolution based video coding

Publications related to this topic:

[8] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel, Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding," in BMVC (British Machine Vision Conference), 2012.

[9] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel, Neighbor embedding based single-image super-resolution using Semi-Nonnegative Matrix Factorization," in ICASSP (IEEE International Conference on Acoustics, Speech, and Signal Processing), 2012.

[10] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel, Compact and coherent dictionary construction for example-based superresolution," in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 2013.

[11] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel, K-WEB: Nonnegative dictionary learning for sparse image representations," in IEEE International Conference on Image Processing (ICIP), Sep. 2013.

[12] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel, Super-resolution using Neighbor Embedding of Back-projection residuals," in 18th International Conference on Digital Signal Processing (DSP), Jul. 2013.

[14] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel, Single-image super-resolution via linear mapping of interpolated selfexamples," IEEE Transactions on Image

Processing, vol. 23, pp. 5334–5347, Dec. 2014.

Major publication:

[13] **M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel, "Video super-resolution via sparse combinations of key-frame patches in a compression context," in 30th Picture Coding Symposium (PCS), Dec. 2013.**

The two Previous Sections proposed schemes where the model is not send. Another extreme consists in sending the model only. Consider a video scheme, in which the receiver mostly receives Low Resolution (LR) images and sometimes High Resolution (HR) images, called KeyFrames (KF) (Fig. 2.8). At the receiver, super-resolution (SR) is used to increase the resolution, such that HR is the final rendered resolution.

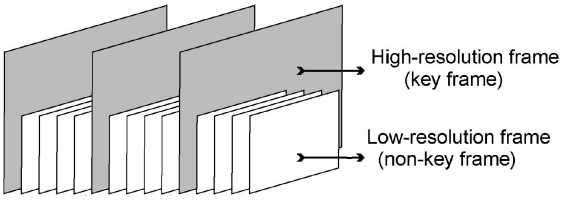

Figure 2.8: Sent and received Image Sequence in the SR based video coding scheme. The receiver extrapolates the LR frames and renders a full HR sequence.

This video compression scheme based on SR can occur in *different scenarios*. For instance, it may be a *choice* to send compressed videos as in Fig. 2.8. Or, it results from the *network constraints*. Indeed, to adapt to the variations in the network bandwidth in video streaming, the video compression rate is lowered by reducing the spatial resolution. To compensate for this variation in resolution, SR is used to increase the resolution when the received resolution is low, and therefore propose a constant resolution whatever the network state is.

SR based video compression is an "all in the model" compression scheme. The SR algorithm first builds a dictionary of pairs of LR and HR patches from the two KF which surround the GOP of LR frames. Then, LR patches are decomposed onto the LR dictionary (column b, steps 1 and 2 in Fig. 2.9), and the SR algorithm uses both this LR decomposition and the dictionary to enhance the resolution (column b, step 3 in Fig. 2.9).

More precisely, for each patch of the current LR frame, a nearest neighbor search is performed (column b, step 1 in Fig. 2.9) in the LR dictionary built from the KF. Note that the patches in the dictionary have different characteristics. Indeed, they can represent either a smooth area, an edge or a texture. Each type of patch has a different complexity since the number of DCT coefficients to sufficiently well approximate a patch varies and depends on the type of the patch, see Fig. 2.10. In a way, the type of the patch (smooth/edge/texture) is similar to a class of distributions, and the NN search is similar to selecting a distribution within the class distribution. The second step consists in computing weights to better approximate a LR patch, which can be seen as specifying the distribution within the class. (Note that another originality of the proposed method in [13] is to not only consider the current frame but also corresponding patches in the KF (column c. in Fig. 2.9). This insures some temporal consistency.) Finally, the weights are used with the HR dictionary to generate the HR patch. So here, the KFs act as the description of the model class, and the LR image as the identification of one distribution

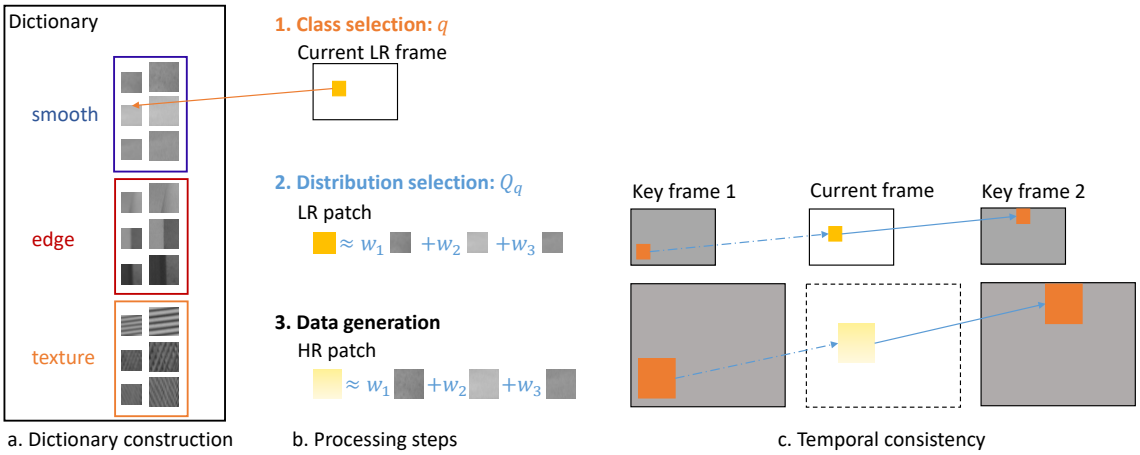

Figure 2.9: Principle of the SR algorithm.

within the class. KF and LR are the only transmitted data. Indeed, no residue is sent between the true and super-resolved image. Therefore, this scheme is an *all in the model* compression scheme.

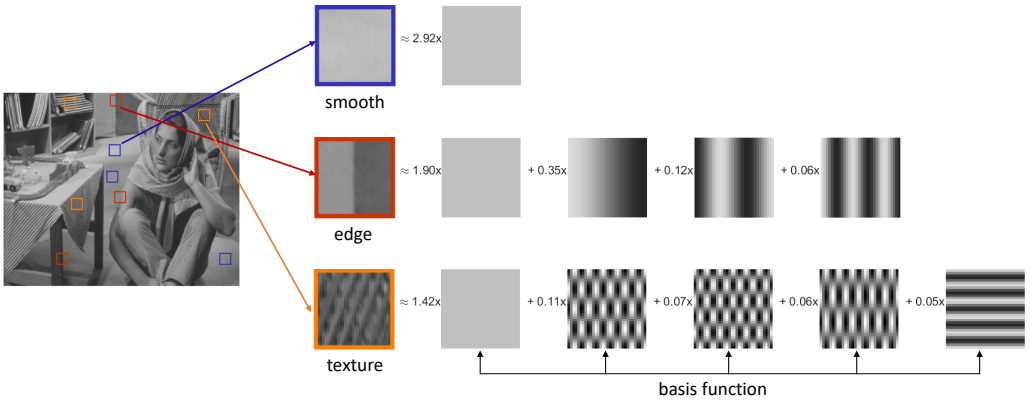

Figure 2.10: Patches in an image have different types (smooth area, edge, and texture) and therefore have different complexities measured as the number of DCT coefficients required to efficiently approximate a patch.

The second remarkable fact about this compression approach is that the code of the model is the one used for the data. Indeed, the distribution is sent as an empirical distribution i.e. samples. Accordingly, the encoding of this empirical distribution is done using a video compression scheme (HEVC in this setup).

Interestingly, this proposed compression algorithm based on SR achieves **similar compression efficiency as HEVC** while sending possibly two spatial resolutions [13]. This is noticeable since **scalable** video encoders suffer some penalty [107] (12.8% for the all intra configuration and 13.7% for the RandomAccess configuration [93]). The penalty of scalable video encoders is sometimes explained by the redundant sampling of predictive video coders: indeed to encode a high resolution frame, the number of residue samples equals the resolution of the enhanced but also base layer. This is the reason why wavelet based scalable video compression schemes were proposed since they are non-redundant

[15, 60, 94]. These coding schemes indeed only experienced little penalty with respect to the non scalable H.264 AVC scheme [80], but had the drawback to not be compatible with H264 AVC. Interestingly, the SR based compression scheme is compatible with the standardized H.26x coding schemes, and shows similar compression results to the ones of H.265 HEVC the latest of the standardized coding schemes. So, it shows that we need to go to a significant subsampling of the residue to cancel the scalability penalty of standardized video coding schemes.

In conclusion, the proposed video compression scheme, based on SR, allowed us to analyze an *all-in-the-model* compression scheme. The *all-in-the-model* compression scheme allows to draw several conclusions. First, coding the model is considered to be a difficult problem. The SR based video compression scheme shows that this can be circumvented by *coding the model as data*. Second, it is widely believed that scalable video compression schemes are suboptimal [93], especially when they are based on the standardized hybrid coding schemes (H.26x). This SR based algorithm shows that, on the contrary, a video can be scalably encoded without any additional penalty.

Other contributions: Super-resolution efficient both in terms of distortion and complexity. We proposed novel techniques capable of producing a high-resolution (HR) image from a single low-resolution (LR) image. A first algorithm with complexity constraint (external dictionary) was proposed that improves the PSNR by at least 1 dB with respect to state of the arts methods. The innovations were efficient image patch representation [8], patch estimation [9] and dictionary learning [10, 12]. Then, relaxing the complexity constraint, we proposed a method based on an internal dictionary, that improved state-of-the art methods by about 0.8dB [14].

2.8 Distributed source coding: model selection and impact of mismatch model

Publications related to this topic:

[101] V. Toto-Zarasoa, A. Roumy, and C. Guillemot, "Maximum Likelihood BSC parameter estimation for the Slepian-Wolf problem," IEEE Communications Letters, vol. 15, no. 2, pp. 232–234, Feb. 2011.

[40] E. Dupraz, A. Roumy, and M. Kieffer, "Source Coding with Side Information at the Decoder and Uncertain Knowledge of the Correlation," IEEE Transactions on Communications, vol. 62, no. 1, pp. 269 – 279, Jan. 2014.

Major publication:

[102] V. Toto-Zarasoa, A. Roumy, and C. Guillemot, "Source modeling for Distributed Video Coding," IEEE Trans. on Circuits and Systems for Video Technology, vol. 22, no. 2, Feb. 2012.

Also [75, 96, 100, 95, 48, 99, 97, 98, 35, 37, 36, 40, 39, 38].

Distributed source coding (DSC) refers to the compression of many correlated sources without communication between the sources. An instance of this problem is source coding with side information (SI) at the decoder. Here, two correlated source sequences x^n and y^n are given and the source sequence x^n needs to be compressed. Upon compression, the side information y^n is not known. By contrast, the decoder knows y^n and receives the compressed version of x^n (SW coding scheme in Fig. 2.11).

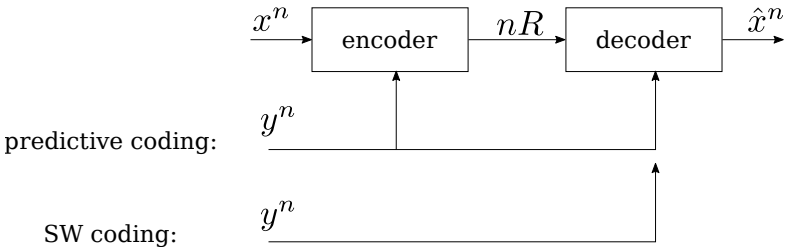

Figure 2.11: Source coding with SI at the encoder and the decoder (predictive coding). Source coding with SI at the decoder only (Slepian Wolf (SW) coding).

Surprisingly, Slepian and Wolf (SW) [90] showed that the schemes with (predictive, Fig. 2.11) and without (SW, Fig. 2.11) SI at the encoder, achieve the same compression rate. However, the two encoding schemes are completely different, as the predictive encoder can use explicitly the realization of the other source y^n, whereas the SW encoder relies only on the joint distribution between X^n and Y^n, referred to as the model. Therefore, SW coding is also called *model-based coding*. More precisely, the model is used at both encoder and decoder but in a different way:

- to determine the code and the compression rate, at the encoder,
- to estimate x^n from y^n, at the decoder.

Knowing the model is a key assumption, for not needing the SI at the encoder in the SW scheme. Indeed, without this knowledge, the compression rate increases from $H(X|Y)$ to $H(X)$ (for memoryless sources), i.e. as for a scheme without SI. However, in practical systems, the model may not be known. Therefore, when the encoding function is linear [92], we first showed that this model can be efficiently inferred at the decoder. The trick consists in deriving the maximum likelihood estimate of the model parameters knowing degraded versions of the information available at the decoder (i.e. the compressed version of the SI y^n and not the SI itself), and refining it with an Expectation Maximization (EM) algorithm. Indeed, the originality of the methods relies in the initialization. Indeed, a direct implementation of the model parameter estimation problem with respect to all information available at the decoder relies on a hidden variable, the source sequence. And this problem can either be solved jointly (sequence and parameter) with a prohibitive compelxity or with an EM algorithm, but where the initialization is tricky. Instead, the model parameter estimation with respect to the compressed version of the SI, and not the SI itself, can be solved according to a ML criterion. This approach holds quite generally for binary memoryless sources with an additive channel [101], non-binary sources [40], sources with memory and back or reverse correlation channel [102]. Moreover, we characterized the effect of an imperfect knowledge of the model at the encoder on the compression rate and proposed practical schemes to achieve these rates [38]. Indeed, the lack of model knowledge can significantly degrade the performance since the optimal compression rate is reduced to a worst case [40]. The degradation was evaluated for different source models Both results (possibility to learn the model the decoder, and significant degradation at the encoder without model) motivates the use of a feedback channel to tune the compression rate in practical systems.

As an application, we considered Distributed Video Coding (DVC), a video compression system that builds upon the idea of distributed source coding in order to achieve efficient video compression while maintaining a low complexity at the encoder. Despite recent advances, distributed video compression rate-distortion performance is not yet at the level

of predictive coding. The key issues we considered to bring DVC to a level of maturity closer to predictive coding were:

1. finding a new model well suited for the sources in DVC. The new model includes
   (i) two classes: new (inter) correlation model (i.e. between the sources),
   (ii) the non uniformity of each source [99],
   (iii) the memory of each source (Hidden Markov models are used) [97].
2. finding the information theoretical compression rates for these models
3. estimating the model parameters [101] and the model class (correlation type).

Note that the novelty in the approach results from the integration of memory and non-uniformity in the entropy coder. Indeed, integration of these properties are straightforward in source codes such as arithmetic code [26, Sec. 13.3]. However, due to the duality between channel coding and DSC [90, 21], the entropy coder is here a channel code, which requires special handling. The Distributed Video Coding system [102] that integrates the enhancement that we propose here demonstrates a quality-versus-rate improvement by up to 10.14%, with respect to its elder version. A simplified model with non-uniform sources is also proposed that achieves an improvement by up to 5.7%.

In DSC, the lack of model knowledge at the encoder can significantly degrade the compression performance. However, even achieving these degraded performances is challenging, since the decoder also needs the model knowledge. It was shown that the lack of model knowledge at the decoder can be circumvented. The trick consists in estimating the source parameters optimally according to the ML criterion but with respect to a subset of the information available at the decoder. This estimate allows to efficiently initialize an EM algorithm, known to be very sensitive to the initialization. As for the application to DVC, we proposed new models and their model estimation methods, that achieved up to 10.14% bitrate saving with respect to state-of-the-arts DVC implementations. The novelty here resided in the integration of non-uniformity and memory in the entropy coder for DSC, which turns out to be a channel code.

2.9 Uncertainty on the side information available at the decoder: Free-viewpoint Television (FTV)

Publications related to this topic:

[74] A. Roumy, "An Information theoretical problem in interactive Multi-View Video services," IEEE Communications Society Multimedia Communications Technical Committee (ComSoc MMTC) E-Letter, vol. 11, pp. 11–16, March 2016.

[33] E. Dupraz, T. Maugey, A. Roumy, and M. Kieffer, "Transmission and Storage Rates for Sequential Massive Random Access," arXiv:1612.07163, 2017.

Major publication:

[76] A. Roumy and T. Maugey, "Universal lossless coding with random user access: the cost of interactivity," in Proceedings IEEE International Conference on Image Processing, Sep. 2015.

The lack of model knowledge at the encoder can significantly degrade the compression performance (see Sec.2.8). For instance, when many SIs are available at the encoder, and only one SI out of the set is available at the decoder, without the encoder knowing which one from the set (see Fig. 2.12), then the optimal compression is given by the worst case i.e. with respect to the least correlated SI [28].

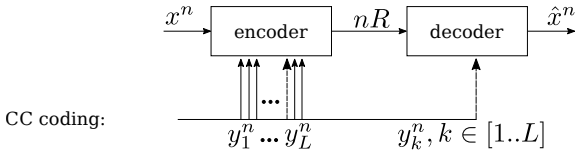

Figure 2.12: Source coding with SI at the decoder and set of SIs at the encoder (Compound code CC).

This setup occurs for instance in Free-viewpoint television (FTV). Indeed, one user chooses a viewpoint, i.e. an image, and can change it at anytime. Compression is performed offline, the compressed video is stored at a server, and the users make requests to the server. When an image is compressed, the encoder knows that the decoder can have in memory a previously requested image, but does not know which one, since this depends on the navigation of the user. In this case, the optimal compression is determined by the least correlated image that could be in the users memory.

To lower the compression rate, the server could decode the whole video, and reencode the requested images. However, this would lead to an extreme complexity at the server. It is however possible to achieve the same compression rate, while maintaining a low complexity at the server [76]. The idea consists in allowing the server to extract bits from the compressed bitstream see Fig. 4.1.

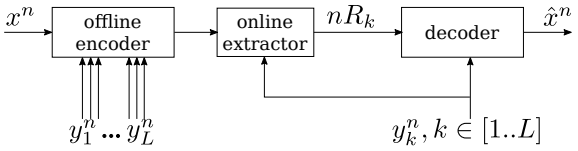

Figure 2.13: Source coding with bitextraction at the server.

Uncertainty on which SI is available at the decoder occurs in the context of FTV, where users can freely choose and change their viewpoint at anytime. This uncertainty can significantly degrade the compression performance. However, we have shown that this can be circumvented if the server is allowed to extract samples from the bitstream, corresponding to the compressed bitstream.

Chapter 3

Transmission and universality

Universal transmission, or universal channel coding, is the problem of sending data over a noisy channel without knowledge of the channel transition probability. If universality does not incur any loss in data compression (at least asymptotically), this is not the case for the transmission over noisy channels. Indeed the lack of knowledge has two possible impacts. More precisely, for the case of a memoryless channel with a single path:

- the transmitter can not predict the rate to encode the data, because the channel statistics are not known. Therefore, the best performance is determined by the worst channel [46, Th. 3.2.1]. To compensate for this negative result, a feedback channel, if possible, is used to inform the encoder of the channel statistics. If not, then an outage will occur, i.e. cases where no transmission is possible. This problem has been studied in [61].
- the lack of knowledge has also an impact on the decoder. Indeed, if the decoder uses a wrong decoding metric, then the performance might be degraded. Unfortunately, the optimal performance under mismatched decoding [56] is still an open problem. However, there exists cases where the decoding metric has no impact on the performance. This occurs for an erasure channel. This also occurs if sequencewise ML decoding is performed, and if the channel is either a binary symmetric channel (BSC) or a finite alphabet input Additive White Gaussian Noise Channel (AWGN). Indeed, in both cases, the ML decoding is equivalent to minimizing a distance (Hamming for BSC and Euclidean for AWGN).

So the impact of the lack of channel knowledge is well studied when the channel has a single path. However, many questions remain regarding the case of multipath channels. Therefore, we studied the question of imperfect channel knowledge in the context of multipath channels (Sec. 3.1).

Moreover, the solution of a feedback channel to circumvent the lack of knowledge at the transmitter may not always be usable, for instance if there is a delay constraint. For these cases, the source has to be adapted to the channel. This is studied in Sec. 3.2.

3.1 Transmission without perfect channel knowledge: Equalization and channel parameter estimation

Publications related to this topic:

[83] N. Sellami, A. Roumy, and I. Fijalkow. A proof of convergence of the MAP turbo-detector to the AWGN case. IEEE Trans. on Signal Processing, 56(4):2716 – 2724, April 2008.

[50] I. H. Kacem, N. Sellami, A. Roumy, and I. Fijalkow, Training sequence length optimization for a turbo-detector using decision-directed channel estimation," Research Letters in Communications, 2008.

[85] N. Sellami, M. Siala, A. Roumy, and I. Kammoun, MAP sequence equalization for imperfect frequency selective channel knowledge," European Transactions on Telecommunications, vol. 21, no. 2, pp. 121–130, 2010.

Major Publication:

[82] N. Sellami, A. Roumy, and I. Fijalkow, The impact of both a priori information and channel estimation errors on the MAP equalizer performance," IEEE Trans. on Signal Processing, pp. 2716–2724, July 2006.

Also [19, 20, 84, 81]

Multipath channels occur in wireless scenarios where the omnidirectional waveform sent is reflected (against building for instance) and the receiver gets multiple and delayed copies of the emitted signal. This occurs much more in urban areas. After sampling, the receivers gets a discrete convolution of the input signals and needs to deconvolve the input. This deconvolution, usually referred to as equalizer, can be performed either optimally in the sense of the maximum a posteriori (MAP) criterion with a complexity that grows exponentially with the number of multipath or with a linear filter with or without feedback loop. In the latter case, the complexity grows only linearly with the number of multipaths [63]. More formally, let $\mathbf{X}$ stand for the random vector that models the input of the channel, $\mathbf{H}$ be the vector of the channel coefficients (i.e. the coefficients of the filter), $\mathbf{Y}$ be the output of the channel, and $\mathbf{x}, \mathbf{y}, \mathbf{h}$ their realizations. Let us further assume that the channel is fixed during the transmission of the block $\mathbf{X}$. The classical MAP equalizer solves

$$\hat{\mathbf{x}} = \arg\max_{\tilde{\mathbf{x}}} \mathbb{P}(\mathbf{X} = \tilde{\mathbf{x}} | \mathbf{Y} = \mathbf{y}, \mathbf{H} = \mathbf{h}) \tag{3.1}$$

These equalizers all assume that the coefficients of the channel are perfectly known. If this is not the case, then a learning sequence known by both the encoder and the decoder is sent, which allows the receiver to first estimate the coefficients of the channel. Then, the algorithm becomes

$$\hat{\mathbf{x}} = \arg\max_{\tilde{\mathbf{x}}} \mathbb{P}(\mathbf{X} = \tilde{\mathbf{x}} | \mathbf{Y} = \mathbf{y}, \mathbf{H} = \hat{\mathbf{h}}) \tag{3.2}$$

where $\hat{\mathbf{h}}$ is the estimate of the channel.

For infinite length learning sequence, the channel can be perfectly estimated. However, for the sake of efficiency, the whole bandwidth can not be used for learning the channel, (the primary goal is to send data). So, the learning sequence is finite and the estimation is thus not perfect (due to the additive thermal noise).

Another solution is to consider the problem on the whole

$$\hat{\mathbf{x}} = \arg\max_{\tilde{\mathbf{x}}} \mathbb{P}(\mathbf{X} = \tilde{\mathbf{x}} | \mathbf{Y} = \mathbf{y}) \tag{3.3}$$

to introduce the hidden variable $\mathbf{h}$ and to iterate between the computation of the symbol densities and the estimation of the channel coefficients.

$$\hat{\mathbf{x}}_{i+1} = \arg\max_{\tilde{\mathbf{x}}} \mathbb{P}(\mathbf{X} = \tilde{\mathbf{x}} | \mathbf{Y} = \mathbf{y}, \mathbf{H} = \hat{\mathbf{h}}_i) \tag{3.4}$$

$$\hat{\mathbf{h}}_{i+1} = \mathbb{E}_{\mathbb{P}(\mathbf{X}=\hat{\mathbf{x}}_{i+1}|\mathbf{Y}=\mathbf{y},\mathbf{H}=\hat{\mathbf{h}}_i)}[LMS(\hat{\mathbf{x}}_{i+1})] \tag{3.5}$$

This problem can be solved with an iterative solution such as the expectation maximization. The algorithm introduces the hidden variable $\mathbf{h}$ and iterates between the computation of the symbol densities and the estimation of the channel coefficients. This algorithm is however very sensitive to the initialization as, the problem (3.3) is not convex. Therefore the need for a first good estimate of the channel. But if an initialization is provided, through for instance a learning sequence, then the statistical characteristics of this channel estimate should be taken into account. This leads to the following optimization problem:

$$\hat{\mathbf{x}} = \arg\max_{\tilde{\mathbf{x}}} \mathbb{P}(\mathbf{X} = \tilde{\mathbf{x}} | \mathbf{Y} = \mathbf{y}, \hat{\mathbf{H}} = \hat{\mathbf{h}}) \tag{3.6}$$

We proposed an iterative solution to this problem in [85].

A second contribution is the analysis [82] of a MAP equalizer, which receives an estimate of the channel coefficients. The difficulty in this analysis relies on the fact that the receiver does not perform linear transformations, but rather performs an exhaustive search among all possible input sequences. This search can be made in an efficient way, such that the computation grows linearly with the length of the sequence (the exponentially growing cost is in the length of the channel only). Still to perform an analysis, all possible errors need to be considered. Our analysis follow the same line and a closed form expression of the probability of error at the equalizer output as a function of the accuracy of the channel estimate is derived. The analysis was then extended to the case of a turbo-detector (iterative receiver MAP equalizer and MAP channel decoder) [81] and to the case of MIMO receivers [19].

This analysis was the starting point to design the system and perform resource allocation. In particular, we optimized the training sequence length and showed that the shortest is the better. Indeed, using more symbols to learn the channel will lower the error rate of the equalizer but also lower the data throughput. [50] shows that on the whole the shortest the training sequence the better. This analysis was performed under realistic assumption of an efficient receiver. First, a turbo detector (concatenation of a MAP equalizer and convolutional decoder, shown to get rid of the channel) was used. Second, the channel estimate is also improved as the symbols are estimated (Least mean square estimate).

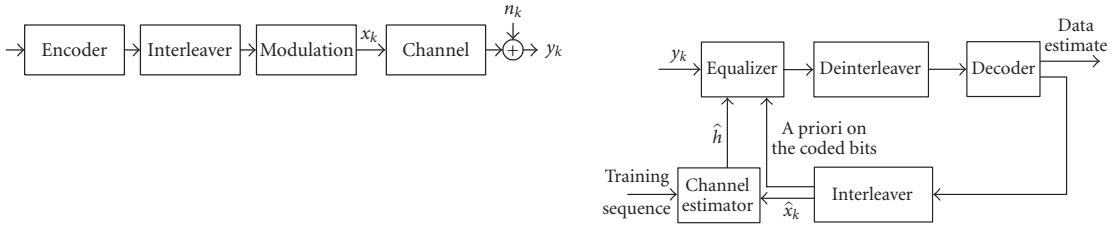

Figure 3.1: Transmitter and Turbo receiver.

Finally, the tools developed to analyze the negative effect of parameter uncertainties can also quantify the positive effect of some side information (for instance a priori information in a turbo scheme). We could then analyze the convergence of a turbo detector

and show that the effect of a multipath channel can be removed [83]. This is a well know result when there is no noise. And it is interesting to see that the result still holds in the presence of noise, provided a channel code is used, and that a turbo scheme is used at the receiver.

Uncertainty of the channel was studied under the assumption that only the channel coefficients are unknown but the noise level is known at both encoder and decoder. This setup was motivated by the fact that there exist indeed ways to avoid the impact of an uncertainty on the noise level. It was shown that the fact that a channel has multipath is not harmful. In fact, when the coefficients of the mutipath channel are not known, the effect on the channel can completely be compensated even in the presence of noise. Moreover, when the channel coefficients need to be estimated, then it was shown that the length of the training sequence can be as short as possible.

3.2 Unified method for transmission with adaptation to the channel: Unequal Erasure Protection, adaptation to the source: File Bundle Protection

Publications related to this topic:

[67] V. Roca, A. Roumy, B. Sayadi, "The Generalized Object Encoding (GOE) Approach for the Forward Erasure Correction (FEC) Protection of Objects and its Application to Reed-Solomon Codes over GF(2^8)", IETF RMT Working Group, July 2012.

[68] V. Roca, A. Roumy, B. Sayadi, "The Generalized Object Encoding (GOE) LDPC-Staircase FEC Scheme", IETF RMT Working Group, Oct 2012

[69] V. Roca, A. Roumy, B. Sayadi, "The Need for Extended Forward Erasure Correction (FEC) Schemes: Problem Position", IETF RMT Working Group, July 2012.

[78] A. Roumy, V. Roca and B. Sayadi. "Memory Consumption Analysis for the GOE and PET Unequal Erasure Protection Schemes", ICC 2012.

Major Publication:

[79] A. Roumy, V. Roca, B. Sayadi, R. Imad,, "Unequal Erasure Protection and Object Bundle Protection with the Generalized Object Encoding Approach", INRIA Research Report RR-7699, July 2011.

Networks experience packet losses at the Internet Protocol (IP) level. Indeed, the physical layer introduces impairments in the form of symbol errors. Then, check sum are applied at the data link and/or physical layer that rejects the damaged frames. If the physical layer is wireless, the packet loss ratio (called Block Error Rate in [2]) can be rather high. For instance, 3GPP [2, Annex E] adopted different end-to-end profiles leading to a packet loss rate between 0.2375 % (for wideband calls) and 2.6375 % (for super-wideband calls) [54].

Retransmission and Forward Error Codes (FEC) can cope with erasures but introduce delays. Several solutions exist in the transport layer and above, to ensure that the client can recover the data in the presence of packet losses; for instance, retransmission (in unicast with the TCP protocol [62]; in multicast, with the Reliable Mutlicast Transport Protocol [59], where lost packets are also recovered with local retransmission thanks to a designated receiver to avoid acknowledgment implosion) or protection with Forward Error Codes (FEC) [53, 106]. Both strategies implies delays which may not always be compatible with real time streaming:

- delay for retransmission is at least the Round Trip Time (RTT);
- delay with FEC is the time needed to send all the encoded packets, which can be up to the number of encoded packets defined by the smallest code rate: $R_{\min} = \frac{k}{n_{\max}}$, if there is no feedback to stop transmission, when a sufficient number of packets is received.

Unequal erasure protection helps meeting a tighter delay constraint. Indeed, there is another degree of freedom: the amount of source data. Indeed, if the source can be split into importance subfiles (for instance scalable video encoding [49], or data of interframe versus intraframe in a video), it is possible to further decrease the delay with UEP. UEP can be implemented with retransmission [18] or FEC [16, 105, 3, 79]. Here we consider solutions based on FEC codes because:

- delays incurred by retransmission are always bigger than delays by FEC (due to RTT)
- FEC based protocols can easier scale up and deal with a large number of receivers than retransmission based protocols

More precisely, in UEP, the original k packets of the source are split into a base layer (k_1 packets) and some enhancement layers ($k_2, ...$). Then, each layer is protected with a different erasure channel code, the base layer being more protected than the other layers: $R_{\min} = \frac{k}{n_{\max}} = R_1 = \frac{k_1}{n_1} \leq R_2 = \frac{k_2}{n_2}$. Therefore, assuming that $R_{\min} = R_1$ and since by construction $k_1 \leq k$, we have $n_1 = \frac{k_1}{k} n \leq n$, introducing a smaller delay for the base layer.

Existing solutions to achieve unequal erasure protection are equivalent in terms of erasure resilience. They differ in terms of computational complexity and memory consumption.

- joint encoding of all layers: [16], [105] (with expanding window),
- separate encoding of the layers and joint packetization of the encoded data: as in Priority Encoding Transmission (PET) [3]
- separate encoding of the layers and separate packetization: as in Generalized Object Encoding (GOE) [79]

Separate encoding of the layers is asymptotically optimal over an erasure channel [17], so all methods listed above perform the same in terms of erasure resilience. However, the methods differ in terms of computational complexity and memory consumption. First, joint encoding is more complex than separate encoding as the data of all layers have to be jointly processed. Second, joint packetization [3] is more complex than separate encoding [79] as all encoded data need to be stored and concatenated, by taking into account the amount of data in each layer (k_i) and its protection level (R_i). Moreover, PET [3] suffer more from rounding than GOE and requires to recompute the data partition each time the size of the source data changes.

Interestingly the GOE approach can also handle small source packet sizes. Transport protocol have fixed packet size. In the case of small source packet sizes, the packets might be rather empty, as classical protocols consider to put the data of a single source in each packet. Thanks to a new abstraction level, the GOE signalization can also handle concatenate small packets, saving badnwidth. This approach can be seen as a variable to fixed length code.

Unequal erasure protection (UEP) is a tool to adapt a priori to the channel losses, while meeting a delay constraint. Since the adaption is done prior to transmission, the system must be dimensioned to the worst case. But different levels of protection are provided such that the delay can be lowered. GOE is a framework to implement UEP with separate encoding and packetization. The novelty lies in the introduction of a new level of abstraction that allows to deal with different parts of an object to be sent over the network. This is done through the GOE signalization compatible with any transport protocol with or without feedback (TCP, ALC) and is therefore backward compatible. It is as efficient as state of the art methods in terms of erasure efficiency but less complex in terms of computation and memory. Notably, GOE can also handle small packets and concatenate them with an appropriate signalization.

Chapter 4

Conclusion and perspectives

This manuscript presented the main research results obtained since completing my PhD. A general conclusion can now be drawn: the model is central to the design of efficient compression and transmission schemes. The term "model" is used here in its information theoretical meaning, see Sec. 2.3. An interesting question concern the accuracy of the model, i.e. how well the model should capture the data, in the case of compression, or the channe,l in the case of transmission. Indeed, the most accurate model is not necessarily the one that leads to the best performance. This is seen for instance in the case of universal source coding performed with a single distribution, or with a unique transform, whatever the image to be compressed is, Sec. 2.6. Similarly, in the case of transmission, it was shown that the optimal description of the channel with a turbo detector, is the shortest one, Sec. 3.1. Nevertheless, in some cases, a better model can improve. For instance, in the case of DVC, our results showed that a model that better fits the signal characteristics, improves significantly the compression performance, see Sec. 2.8.

This leads to the question raised in the title of the manuscript: "is it necessary to send the model?" And it can now be concluded that there is no unique answer. Indeed, in the case of compression, it was shown that two extreme compression schemes with or without sending the model (see Sec. 2.6 and Sec. 2.7) achieve comparable performance. The fact that there is no unique answer to this question is to be welcomed as it allows to adapt to the application constraints (complexity, delay, ...). However, it is important to stress that not sending the model does not mean not modeling. Indeed, in the deep transform, the model is learned from a database. But, the model has properties which depend on the choice to send or not the model. More precisely, when the model is not sent, the model must avoid overfitting to be able to generalize to any king of image. By contrast, when the model is sent, overfitting is a welcomed property (especially, when the data is not sent as in Sec. 2.7).

To summarize, there are two main issues concerning the model. The first one is related to the design of algorithms dedicated to a large variety of processing tasks. For instance, it includes the estimation of the model, and then the processing of the data according to this model. In the case of compression, these tasks are the construction of the entropy coder, the transform, the predictor, and the distribution estimator. For the transmission, it means estimating the channel and decoding the data according to the estimated channel. The second issue is to evaluate the impact of the model. In the case of compression, this means to develop a code for the data according to the model, and measure the codelength of this code. Measuring the impact of the model in the case of transmission requires to decode the data according to the channel, and to evaluate the performance of the decoder. These tasks require skills in coding/information theory, but also signal processing, and

vision, areas in which I contributed.

Perspectives. My research program addresses the challenges faced by the emergence of new types of data, new imaging modalities, and also new practices in terms of user interaction [7]. This creates the need for efficient and possibly new algorithms and their analysis, which I will tackle, building on the achieved results, and on the acquired skills.

Compression and diffusion for interactive user experience: Massive Random Access to large databases. Video traffic represents 80% of the data exchanged over the Internet in 2018[1]. Moreover, the amount of these videos exchanged, might triple from 2016 to 2021[1]. Internet of Things (IoT) brings even more data exchange increase, since these data are not only created and requested by human beings but also by objects and machines. Indeed, it is anticipated that, in 2018, the total generated data will be 400 ZB (1 ZB=10^{21} B)[2].

One characteristic of these data is their high redundancy. For the case of videos, multi-view acquisition is now widely used, with a tremendous number of possible views: at least 100 views are considered for FTV [1], which leads to a raw datarate of about 124 Gbit/s (for HD videos at 50Hz). Note that the number of 100 is chosen for testbed experiments but practical implementation will consider significantly more views. Moreover, the resolution of visual sensors keeps increasing, and new modalities have emerged (plenoptic, 360^o cameras) [7], leading to high correlation within the pixels but also among the views. Regarding IoT for SmartCities, it is usually deployed with overinstrumentation, which leads to highly spatially and temporally correlated data. It is therefore of great importance to exploit these redundancies to reduce the traffic and the storage.

My research project aims at compressing these large databases. The high redundancy of the data suggests to compress the data together, which leads to a single and non-cuttable compressed bistream. On the other hand, the database is so big that users may not be interested in the whole database. This occurs exactly in the context of FTV. Users visualize one view at a time. This also applies for databases collecting data from a large-scale sensor network (such as Smart Cities). In fact these two problems are instances of a more general problem that can be called massive random access (MRA) to large databases. Indeed, MRA concerns databases that are so large that, to be stored on a single server, the data have to be compressed efficiently, meaning that the redundancy/correlation between the data have to be exploited. The dataset is then stored on a server and made available to users that may want to access only a subset of the data. Such a request for a subset of the data is indeed random, since the choice of the subset is user-dependent. Finally, massive requests are made, meaning that, upon request, the server can only perform low complexity operations (such as bit extraction but no decompression/compression).

Compression rates have been derived under simple model assumptions [77], see Sec. 2.9. It has been shown that the data can be sent at the same compression rate as if there were no random access to the database. The cost of the interactivity is in the storage only: the data has to be stored according to the least correlated SI. This cost is rather limited, as it stays in the same order of magnitude as the compression rate without interactivity. In particular, it does not scale with the number of possible SIs i.e. the number of navigation paths allowed. However, to construct practical systems, many issues remain open. I

[1]Cisco Visual Networking Index: Forecast and Methodology, 20162021 https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html

[2]Cisco Global Cloud Index: Forecast and Methodology, 20162021 White Paper, Fig. 24 https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html#_Toc503317525

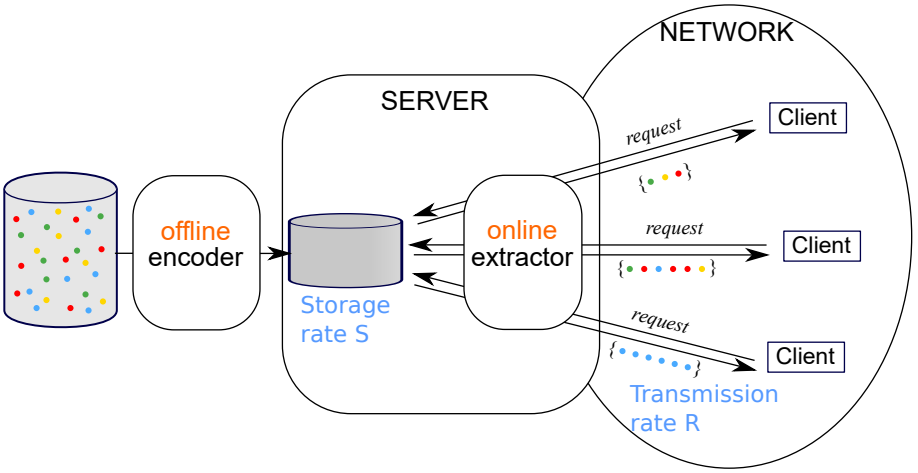

Figure 4.1: MRA (massive random access). A database, containing highly correlated data, is compressed into a single file in order to exploit the redundancy. Users can access any subpart of the database. The requested subpart is delivered without transmitting the whole database, and also without encoding/decoding the data at the server.

am currently leading a project called InterCom, for Interactive Communication[3] on this research topic. The first goal is to derive optimal compression rates in terms of storage rate but also transmission rate in more realistic scenarios (see Fig. 4.1). The second goal is to construct practical schemes for both applications: FTV and IoT.

Open problems are:

- Building an entropy coder for MRA. The entropy coder is the core of the compression algorithm. The proof in [77] relies on the construction of embedded codes, and this suggests the use, in practice, of variable rate channel codes. Here, and by contrast with existing codes, the characteristics of the data, i.e. the correlation between the sources, have to be taken into account.
- Zero error vs vanishing error rate. The proof relies [77] on a vanishing error argument. However, in practice, the scheme must provide zero error for any length. This is of great importance for lossless applications (building health monitoring, for instance). This is also needed to avoid error propagation in the context of lossy video compression implemented with a divide and conquer approach (Sec. 2.4).
- Universal coding. We have shown that, at infinite length, the lack of model knowledge does not impact the compression rates [34]. However, to answer the question of universal MRA coding, and select the optimal model class, there is a need to characterize the impact of a wrong model on the compression performance at finite length.
- Real data have to be preprocessed before being entropy coded. Indeed, in a practical system, the entropy coder can only tackle limited and remaining dependencies. Therefore a need for preprocessing to handle most of the redundancy. Moreover, this preprocessing tasks will depend on the type of data (image, video vs meteorological data[4]), and on their representation (point cloud, meshes, pixel values).

[3] https://intercom.cominlabs.u-bretagneloire.fr/

[4] Note that the compression of meteorological data is far from anecdotal. In 2017, the weather channel websites received every minute a deluge of 18 million forecast requests - an increase of 22 % from the previous year, https://www.domo.com/blog/data-never-sleeps-5/.

Image Video compression based on learned models. Despite significant improvements obtained with the video compression standard HEVC, there is still a need to further compress the data. One of the goals is to have a single decoding algorithm that could handle any new modalities such as High Dynamic Range, 360^o videos. Indeed, a project, called Versatile Video Coding[5] (VVC), has been launched in April 2018 to develop a new video coding standard. The very good performance obtained with the learning of the transform suggests that it may be fruitful to introduce learning in other processing tasks, such as the prediction, the entropy coder, or the bit allocation problem.

Acquisition adapted to the sensed signal and optimization according to information theoretical criteria. Compressed sensing (CS) is an efficient acquisition scheme, where the acquisition is made through random linear measurements of the data while performing dimensionality reduction. The reconstruction is performed by solving underdetermined linear systems under a sparsity a priori constraint, i.e. the data can be transformed in a domain such that only very few coefficients are non zero. CS is of particular interest when the classical acquisition scheme already multiplexes the signal, and when the signal is sparse in some basis. In that case, it is possible to subsample the acquired measurements and still reconstruct the signal. This is exactly the case in Magnetic resonance imaging (MRI). Another possible application is when the sampling is performed in raster mode, and that there is a physical way to perform multiplexing. This is the case with the single pixel camera.

CS stems from the signal processing community and first analyses mostly concern the ability to recover the data. More precisely, these analyses are worst case as they consider a deterministic signal and look for properties on the measurement matrix that guarantee necessary and/or sufficient conditions for perfect reconstruction of any possible sparse vector [41]. Another question of interest is the study of CS as a communication tool. To do this, average analysis over the signal is required, where conditions are computed such that perfect recovery occurs with high probability. We already obtained preliminary results regarding the analysis of CS as a communication tool [23, 25] and distributed CS [24] under the hypothesis that the support is perfectly known. (For distributed CS, we also proposed practical schemes [22].) A remaining question is under which conditions can the support of the sparse signal be recovered with high probability. Such average analyses have been performed for reconstruction algorithms, where the reconstructed signal at convergence can be characterized. For instance, Basis Pursuit [103], Matched Filter and linear MMSE [64], thresholded linear MMSE [104], Maximum Likelihood and Approximate message passing in [64], Symbol-by-Symbol MAP and thresholded Lasso in [104]. When the reconstruction is based on a greedy algorithm (matching pursuits, etc...), the analysis is very tricky due to the iterative nature of the algorithm, which introduces dependence among iterations. An interesting question is therefore under which conditions (size of the measurement matrix, distribution of the coefficients) can the support of the sparse signal be recovered by a greedy and iterative algorithm such as orthogonal matching pursuit.

[5] https://news.itu.int/versatile-video-coding-project-starts-strongly/

Bibliography

[1] *Call for Evidence on Free-Viewpoint Television: Super-Multiview and Free Navigation.* ISO/IEC JTC1/SC29/WG11 MPEG2015/N15348, June 2015, http://mpeg.chiariglione.org/sites/default/files/files/standards/parts/docs/w15348_rev.docx.

[2] 3GPP, *TS 26.132. Speech and video telephony terminal acoustic test specification*, 3rd Generation Partnership Project (3GPP) Std., June 2017.

[3] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1737–1744, Nov 1996.

[4] A. Barron and T. Cover, "Minimum complexity density estimation," *IEEE Transactions on Information Theory*, vol. 37, no. 4, p. 10341054, 1991.

[5] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2743–2760, Oct 1998.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *IEEE International Conference on Communications, (ICC)*, 1993, pp. 1064–1070.

[7] N. Bertin and A. Roumy, "New generation of audiovisual sensors and the challenge of dimensionality (nouvelle génération de capteurs audiovisuels : les défis de dimensionalités)," Data Science Symposium, Invited talk, Nov. 2015.

[8] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel, "Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding," in *BMVC (British Machine Vision Conference)*, 2012. [Online]. Available: http://hal.inria.fr/hal-00747054

[9] ——, "Neighbor embedding based single-image super-resolution using Semi-Nonnegative Matrix Factorization," in *ICASSP (IEEE International Conference on Acoustics, Speech, and Signal Processing)*, 2012. [Online]. Available: http://hal.inria.fr/hal-00747042

[10] ——, "Compact and coherent dictionary construction for example-based super-resolution," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013. [Online]. Available: http://hal.inria.fr/hal-00875964

[11] ——, "K-WEB: Nonnegative dictionary learning for sparse image representations," in *IEEE International Conference on Image Processing (ICIP)*, Sep. 2013. [Online]. Available: http://hal.inria.fr/hal-00876018

[12] ——, "Super-resolution using Neighbor Embedding of Back-projection residuals," in *18th International Conference on Digital Signal Processing (DSP)*, Jul. 2013. [Online]. Available: http://hal.inria.fr/hal-00876020

[13] ——, "Video super-resolution via sparse combinations of key-frame patches in a compression context," in *30th Picture Coding Symposium (PCS)*, Dec. 2013. [Online]. Available: http://hal.inria.fr/hal-00876026

[14] ——, "Single-image super-resolution via linear mapping of interpolated self-examples," *IEEE Transactions on Image Processing*, vol. 23, pp. 5334–5347, Dec. 2014.

[15] V. Bottreau, M. Benetiere, B. Felts, and B. Pesquet-Popescu, "A fully scalable 3d subband video codec," in *Proceedings 2001 International Conference on Image Processing*, 2001.

[16] A. Bouabdallah and J. Lacan, "Dependency-aware unequal erasure protection codes," *Journal of Zhejiang University - Science A*, vol. 7, 2006.

[17] S. Boucheron and M. R. Salamatian, "About priority encoding transmission," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 699–705, Mar 2000.

[18] K. Bouchireb and P. Duhamel, "Transmission Schemes for Scalable Video Streaming in a Multicast Environment," in *2008 International Wireless Communications and Mobile Computing Conference*, Aug 2008, pp. 419–424.

[19] S. Chaabouni, N. Sellami, and A. Roumy, "Lower bounds on the performance of the MAP equalizer with a priori over MIMO systems," in *Proc.of ISSPA, International Symposium on Signal Processing and its Applications*, Sharjah, UAE, Feb. 2007.

[20] ——, "The Impact of a priori information on the MAP equalizer performance with M-PSK modulation," in *Proc.of 15th European Signal Processing conference, EUSIPCO*, Poznan, Poland, Sept. 2007.

[21] J. Chen, D. He, and A. Jagmohan, "On the duality between Slepian–Wolf coding and channel coding under mismatched decoding," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 4006–4018, 2009.

[22] G. Coluccia, E. Magli, A. Roumy, and V. Toto-Zarasoa, "Lossy compression of distributed sparse sources: a practical scheme," in *European Signal Processing Conference (EUSIPCO)*, Sep. 2011. [Online]. Available: http://hal.inria.fr/inria-00629001/en/

[23] G. Coluccia, A. Roumy, and E. Magli, "Exact Performance Analysis of the Oracle Receiver for Compressed Sensing Reconstruction," in *ICASSP (IEEE International Conference on Acoustics, Speech, and Signal Processing)*, 2014.

[24] ——, "Operational Rate-Distortion Performance of Single-source and Distributed Compressed Sensing," *IEEE Transactions on Communications*, 2014.

[25] ——, "Mismatched sparse denoiser requires overestimating the support length," in *ICASSP (IEEE International Conference on Acoustics, Speech, and Signal Processing)*, 2017.

[26] T. Cover and J. Thomas, *Elements of information theory, second Edition.* Wiley, 2006.

[27] I. Csiszar and P. Shields, "Information theory and statistics: A tutorial," *Foundations and Trends in Communications and Information Theory*, vol. 1, no. 4, pp. 417–528, 2004. [Online]. Available: http://dx.doi.org/10.1561/0100000004

[28] S. C. Draper and E. Martinian, "Compound conditional source coding, Slepian-Wolf list decoding, and applications to media coding," in *IEEE International Symposium on Information Theory*, 2007.

[29] P. Duhamel and M. Kieffer, *Joint Source-Channel Decoding: A Cross-Layer Perspective with Applications in Video Broadcasting over Mobile and Wireless Networks.* Academic Press, 2010.

[30] T. Dumas, A. Roumy, and C. Guillemot, "Shallow sparse Autoencoders versus sparse coding algorithms for image compression," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2016.

[31] ——, "Image compression with stochastic Winner-Take-All Autoencoder," in *ICASSP (IEEE International Conference on Acoustics, Speech, and Signal Processing)*, 2017.

[32] ——, "Autoencoder based Image compression: can the learning be quantization independent?" in *ICASSP (IEEE International Conference on Acoustics, Speech, and Signal Processing)*, 2018.

[33] E. Dupraz, T. Maugey, A. Roumy, and M. Kieffer, "Transmission and Storage Rates for Sequential Massive Random Access," arXiv:1612.07163, 2017. [Online]. Available: https://arxiv.org/abs/1612.07163

[34] ——, "Transmission and Storage Rates for Sequential Massive Random Access." [Online]. Available: https://arxiv.org/abs/1612.07163

[35] E. Dupraz, A. Roumy, and M. Kieffer, "Source Coding with Side Information at the Decoder: Models with Uncertainty, Performance Bounds, and Practical Coding Schemes," in *International Symposium on Information Theory and its Applications 2012*, Oct. 2012. [Online]. Available: http://hal-supelec.archives-ouvertes.fr/hal-00727780

[36] ——, "Codage de Sources avec Information Adjacente et Connaissance Imparfaite de la Corr élation : le problème des cadrans," in *Actes de la 24ème édition du colloque Gretsi*, Brest, France, Sep. 2013, pp. 1–4, iD286 ID286. [Online]. Available: http://hal-supelec.archives-ouvertes.fr/hal-00935785

[37] ——, "Codage Distribué dans des Réseaux de Capteurs avec Connaissance Incertaine des Corrélations," in *Actes de la 24ème édition du colloque GRETSI*, Brest, France, Sep. 2013, pp. 1–4, iD390 ID390. [Online]. Available: http://hal-supelec.archives-ouvertes.fr/hal-00935788

[38] ——, "Practical Coding Scheme for Universal Source Coding with Side Information at the Decoder," in *Proceedings of the Data Compression Conference*, Snowbird, États-Unis, Mar. 2013, pp. 1–11. [Online]. Available: http://hal-supelec.archives-ouvertes.fr/hal-00819491

[39] ——, "Universal Wyner-Ziv Coding for Gaussian Sources," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 2013, pp. 1–4. [Online]. Available: http://hal-supelec.archives-ouvertes.fr/hal-00819493

[40] ——, "Source Coding with Side Information at the Decoder and Uncertain Knowledge of the Correlation," *IEEE Transactions on Communications*, vol. 62, no. 1, pp. 269 – 279, Jan. 2014. [Online]. Available: http://hal-supelec.archives-ouvertes.fr/hal-00935847

[41] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel, 2013.

[42] A. E. Gamal and Y.-H. Kim, *Network Information Theory*. New York, NY, USA: Cambridge University Press, 2012.

[43] E. Gassiat, *Codage universel et identification d'ordre par sélection de modèles*, Société Mathmatique de France, Ed., 2014.

[44] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, Oct 1998.

[45] P. D. Grunwald, *The Minimum Description Length Principle*. MIT Press, 2007.

[46] T. Han, *Information-spectrum methods in information theory*. Springer, 2003.

[47] T. Han and K. Kobayashi, *Mathematics of Information and Coding*, ser. Translations of Mathematical Monographs. American Mathematical Society, 2002, vol. 203.

[48] C. Herzet, V. Toto-Zarasoa, and A. Roumy, "Error-resilient non-asymmetric Slepian-Wolf coding," in *IEEE Intl. Conf. on Communications, ICC*, 2009.

[49] ITU-T and I. JTC1, "Joint Scalable Video Model JSVM-8.6," Tech. Rep., 2007.

[50] I. H. Kacem, N. Sellami, A. Roumy, and I. Fijalkow, "Training sequence length optimization for a turbo-detector using decision-directed channel estimation," *Research Letters in Communications*, 2008.

[51] J. C. Kieffer, "A unified approach to weak universal source coding," *IEEE Transactions on Information Theory*, vol. 24, pp. 674–682, 1978.

[52] J. Kovacevic and M. Vetterli, *Wavelets and Subband Coding*, ser. Prentice-Hall signal processing series. Prentice Hall PTR, 1995.

[53] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft, *The Use of Forward Error Correction (FEC) in Reliable Multicast*, IETF Std. RFC 3453, Dec. 2002.

[54] N. Majed, S. Ragot, X. Lagrange, and A. Blanc, "Delay and quality metrics in Voice over LTE (VoLTE) networks: An end-terminal perspective," in *ICNC 2017 : International Conference on Computing, Networking and Communications : Communications QoS and System Modeling*, 2017, pp. 643–648.

[55] D. Marpe, H. Schwarz, and T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, p. 620636, July 2003.

[56] N. Merhav, G. Kaplan, A. Lapidoth, and S. S. Shitz, "On information rates for mismatched decoders," *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 1953–1967, Nov 1994.

[57] N. Moreau, *Tools for Signal Compression*. Wiley, 2011.

[58] P. Pad and M. Unser, "Optimality of operator-like wavelets for representing sparse ar(1) processes," *IEEE Transactions on Signal Processing*, vol. 63, no. 18, pp. 4827–4837, Sept 2015.

[59] S. Paul, K. K. Sabnani, J. C. H. Lin, and S. Bhattacharyya, "Reliable multicast transport protocol (RMTP)," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 407–421, Apr 1997.

[60] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," in *Proceedings 2001 International Conference on Signal Processing*, 2001.

[61] P. Piantanida, G. Matz, and P. Duhamel, "Outage behavior of discrete memoryless channels under channel estimation errors," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 4221–4239, Sep 2009.

[62] J. Postel, *Transmission Control Protocol*, IETF Std. RFC 0793, Sept. 1981.

[63] J. Proakis and M. Salehi, *Digital Communications*. McGraw-Hill, 2008.

[64] G. Reeves and M. Gastpar, "Approximate Sparsity Pattern Recovery: Information-Theoretic Lower Bounds," *IEEE Trans. on Information Theory*, vol. 59, no. 6, pp. 3451–3465, June 2013.

[65] J. Rissanen, "Modeling By Shortest Data Description," *Automatica*, vol. 14, pp. 465–471, 1978.

[66] ——, "Universal coding, information, prediction and estimation," *IEEE Transactions on Information Theory*, vol. 30, pp. 629–636, 1984.

[67] V. Roca, A. Roumy, and B. Sayadi, "The Generalized Object Encoding (GOE) Approach for the Forward Erasure Correction (FEC) Protection of Objects and its Application to Reed-Solomon Codes over GF(2^8)," IETF RMT Working Group, Work in Progress: <draft-roca-rmt-goe-fec-02.txt>, Jul. 2012. [Online]. Available: http://tools.ietf.org/html/draft-roca-rmt-goe-fec-02

[68] ——, "The Generalized Object Encoding (GOE) LDPC-Staircase FEC Scheme," IETF RMT Working Group, Work in Progress: <draft-roca-rmt-goe-ldpc-01.txt>, July 2012. [Online]. Available: http://tools.ietf.org/id/draft-roca-rmt-goe-ldpc-01.txt

[69] ——, "The Need for Extended Forward Erasure Correction (FEC) Schemes: Problem Position," IETF RMT Working Group, Work in Progress: <draft-roca-rmt-extended-fec-problem-00.txt>, Jul. 2012. [Online]. Available: http://tools.ietf.org/html/draft-roca-rmt-extended-fec-problem-00

[70] J. C. Rolon, E. Mendonca, and P. Salembier, "Generalized lifting with adaptive local pdf estimation for image coding," in *2009 Picture Coding Symposium*, 2009.

[71] J. C. Rolon, A. Ortega, and P. Salembier, "Modeling of contours in wavelet domain for generalized lifting image compression," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009.

[72] J. C. Rolon and P. Salembier, ""generalized lifting for sparse image representation and coding"," in *2007 Picture Coding Symposium*, 2007.

[73] J. C. Rolon, P. Salembier, and X. Alameda, ""image compression with generalized lifting and partial knowledge of the signal pdf"," in *2008 15th IEEE International Conference on Image Processing*, 2008.

[74] A. Roumy, "An Information theoretical problem in interactive Multi-View Video services," *IEEE Communications Society Multimedia Communications Technical Committee (ComSoc MMTC) E-Letter*, vol. 11, pp. 11–16, March 2016.

[75] A. Roumy, K. Lajnef, and C. Guillemot, "Rate-adaptive turbo-syndrome scheme for Slepian-Wolf Coding," in *41st Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2007.

[76] A. Roumy and T. Maugey, "Universal lossless coding with random user access: the cost of interactivity," in *Proceedings IEEE International Conference on Image Processing*, Quebec, Canada, Sep. 2015.

[77] ——, "Universal lossless coding with random user access: the cost of interactivity," in *IEEE International Conference on Image Processing (ICIP)*, 2015, *Among 10% best papers.*

[78] A. Roumy, V. Roca, and B. Sayadi, "Memory Consumption Analysis for the GOE and PET Unequal Erasure Protection Schemes," in *IEEE International Conference on Communications*, Feb. 2012. [Online]. Available: http://hal.inria.fr/hal-00668826

[79] A. Roumy, V. Roca, B. Sayadi, and R. Imad, "Unequal erasure protection and object bundle protection with the generalized object encoding approach," INRIA Research Report RR-7699, Jul. 2011. [Online]. Available: http://hal.inria.fr/inria-00612583/en/

[80] P. Schelkens, Y. Andreopoulos, J. Barbarien, T. Clerckx, and F. Verdicchio, "A comparative study of scalable video coding schemes utilizing wavelet technology," in *Proc. SPIE Photonics East Wavelet Applications in Industrial Processing*, 2003.

[81] N. Sellami, A. Roumy, and I. Fijalkow, "Performance analysis of the MAP equalizer within an iterative receiver including a channel estimator," in *IEEE Vehicular Technology Conference, VTC'F04*, Sept. 2004.

[82] ——, "The impact of both a priori information and channel estimation errors on the MAP equalizer performance," *IEEE Trans. on Signal Processing*, pp. 2716 – 2724, July 2006.

[83] ——, "A proof of convergence of the MAP turbo-detector to the AWGN case," *IEEE Trans. on Signal Processing*, vol. 56, no. 4, pp. 2716 – 2724, April 2008.

[84] N. Sellami, M. Siala, A. Roumy, and I. Kammoun, "Generalized MAP: sequence detection for non ideal frequency selective channel knowledge," in *Proc.of ICASSP, IEEE Int. Conference on Acoustics, Speech and Signal Processing*, Honolulu, USA, April 2007.

[85] ——, "MAP sequence equalization for imperfect frequency selective channel knowledge," *European Transactions on Telecommunications*, vol. 21, no. 2, pp. 121–130, 2010.

[86] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. New York, NY, USA: Cambridge University Press, 2014.

[87] C. E. Shannon, "A mathematical theory of communication, Part I, Part II," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.

[88] P. Shields and B. Weiss, "Universal redundancy rates for the class of b-processes do not exist," *IEEE Transactions on Information Theory*, vol. 41, no. 2, pp. 508–512, Mar 1995.

[89] P. C. Shields, "Universal redundancy rates do not exist," *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 520–524, Mar 1993.

[90] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[91] J. Sole and P. Salembier, "Generalized lifting prediction optimization applied to lossless image compression," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 695–698, Oct 2007.

[92] V. Stankovic, A.D.Liveris, Z. Xiong, and C. Georghiades, "On code design for the Slepian-Wolf problem and lossless multiterminal networks," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1495 –1507, april 2006.

[93] K. Suehring, B. Li, K. Sharman, V. Seregin, and A. Tourapis, "JCT-VC AHG report: HEVC HM, SCM, SHM and HDRTools software development and software technical evaluation (AHG3). Doc. JCTVC-AD0003," MPEG-H, MPEG-H Group, , January 2018. [Online]. Available: http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=10822

[94] C. Tillier and B. Pesquet-Popescu, "3d, 3-band, 3-tap temporal lifting for scalable video coding," in *Proceedings 2003 International Conference on Image Processing*, 2003.

[95] V. Toto-Zarasoa, E. Magli, A. Roumy, and G. Olmo, "On Distributed Arithmetic Codes and Syndrome Based Turbo Codes for Slepian-Wolf Coding of Non Uniform Sources," in *European Conf. on Signal Processing, EUSIPCO*, 2009.

[96] V. Toto-Zarasoa, A. Roumy, and C. Guillemot, "Rate-adaptive codes for the entire Slepian-Wolf region and arbitrarily correlated sources," in *IEEE Intl. Conference on Acoustic, Speech and Signal Processing*, Las Vegas, USA, Apr. 2008.

[97] ——, "Hidden Markov model for Distributed Video Coding," in *International Conference on Image Processing (ICIP)*, Sept. 2010.

[98] ——, "Non-asymmetric Slepian-Wolf coding of non-uniform Bernoulli sources," in *International Symposium on Turbo Codes (ISTC)*, Sept. 2010.

[99] ——, "Non-uniform source modeling for Distributed Video Coding," in *European Signal Processing Conference (EUSIPCO)*, Aug. 2010.

[100] V. Toto-Zarasoa, A. Roumy, C. Guillemot, and C. Herzet, "Robust and Fast Non Asymmetric Distributed Source Coding Using Turbo Codes on the Syndrome Trellis," in *IEEE Intl. Conf. on Acoustic, Speech and Signal Processing, ICASSP, Taipei*, 2009.

[101] V. Toto-Zarasoa, A. Roumy, and C. Guillemot, "Maximum Likelihood BSC parameter estimation for the Slepian-Wolf problem," *IEEE Communications Letters*, vol. 15, no. 2, pp. 232–234, Feb. 2011. [Online]. Available: http://hal.inria.fr/inria-00628996/en/

[102] ——, "Source modeling for Distributed Video Coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 22, no. 2, Feb. 2012. [Online]. Available: http://hal.inria.fr/inria-00632708/en/

[103] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via Orthogonal Matching Pursuit," *IEEE Trans. on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[104] A. Tulino, G. Caire, S. Verdú, and S. Shamai, "Support Recovery with sparsely sampled free random matrices," *IEEE Trans. on Information Theory*, vol. 59, no. 7, pp. 4243–4271, July 2013.

[105] D. Vukobratovic and V. Stankovic, "Unequal error protection random linear coding strategies for erasure channels," *IEEE Transactions on Communications*, vol. 60, no. 5, pp. 1243–1252, May 2012.

[106] M. Watson, A. Begen, and V. Roca, *Forward Error Correction (FEC) Framework*, IETF Std. RFC 6363, June 2011.

[107] M. Wien, H. Schwarz, and T. Oelbaum, "Performance Analysis of SVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1194–1203, Sept 2007.

[108] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, pp. 337–343, 1977.

[109] ——, "Compression of individual sequences by variable rate coding," *IEEE Transactions on Information Theory*, vol. 24, pp. 530–536, 1978.