NATO Courses, July 1978.

NATO Advanced Study Institute on Signal Processing of Multimedia
Il Ciocco Resort Hotel, Tuscany, Italy.

Joint Source and Channel Coding

Pierre Duhamel (1)
Olivier Rioul, and Seyed Barham ZahirAzami (2)
École Nationale Supérieure des Télécommunications (ENST), Paris, France.
(1): Signal and Image processing Dpt. (2): Communications and Electronics Dpt.

Abstract. This paper first presents joint source and channel coding in an information theoretic manner, and precises what can be expected from such an approach compared to a classical procedure. Then, we present the main methods that have been already proposed for performing joint source and channel coding and relate them to the general framework.

1 Introduction

Joint source and channel coding has attracted a lot of attention recently, after a few years where only a small group of people were working hard on this topic. For people first jumping in the field, the first impression can be somewhat awkward: what are these people doing, if Shannon has shown the separation theorem? By the way, many people remember Shannon's work only vaguely. This is the first question we shall address: we carefully explain the origin of the separation theorem, provide the source bound, as well as the channel bound, and provide the optimum theoretically performance attainable (OPTA) that can be obtained when sending a given source on a given channel. When doing so, the expectations one can have when working jointly on source and channel coding can be made explicit.

Then, we provide a generalized framework in which many recent approaches can be stated. Finally, we give a comprehensive overview of the litterature on joint source and channel coding, insisting on the common characteristics of the proposed approaches.

In summary, the purpose of this paper is to merge theoretical comprehension of the problem together with the presentation of practical algorithms.

2 Information theoretic preliminaries

This section gives the basic information theoretic tools that are needed to fully understand the Shannon bounds in their various versions. In fact, it is quite common in the channel coding field to check the performance of some proposed system against the best attainable performance. In source coding, even if one knows the existence of $R(D)$ curves, actual performances are seldom checked against bounds. Furthermore, it is less known that the same kind of bound exists for the joint source/channel coding situation. This point is clarified in this section.

2.1 Situation of interest

Deriving bounds clearly requires a simple model of the situation of interest. We use here the classical transmission model, as depicted on figure 1. Let us consider carefully the situation, as well as the constraints that are required for the results to have practical usefulness.

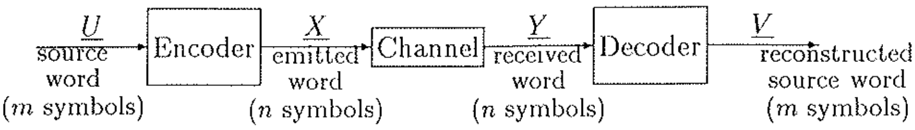

Figure 1: Shannon's paradigm

We consider only block processing: inputs as well as all variables in the scheme are vectors, as defined below:

- The initial and reconstructed source words have m components (the source symbols), $\underline{U} = (U_1, U_2, \dots, U_m)$ and $\underline{V} = (V_1, V_2, \dots, V_m)$.
- The channel input and output words have n components (the channel symbols) $\underline{X} = (X_1, X_2, \dots, X_n)$, and $\underline{Y} = (Y_1, Y_2, \dots, Y_n)$

All the above variables are modelized as random variables that can be discrete or continuous. For this reason, we have adopted the following notation: $\sum\!\!\!\!\!\!\int$ denotes summing, where a discrete sum is considered for discrete variables, and an integral (continuous)is considered for continuous variables (the measure symbol $d(.)$ will be omitted, for simplicity). For example, the expectation reads

$$\mathbf{E}\, X = \sum\!\!\!\!\!\!\!\int_{\underline{X},} x p(\underline{x})$$

. This common notation allows to address at the same time a large variety of sources and channels.

The source is characterized by its probability distribution $p(\underline{u}) = p(u_1, \cdots, u_m)$. For example, a Gaussian source is modelled by $p(\underline{u}) = \frac{1}{\sigma_u\sqrt{2\pi}} \exp(-\frac{\|u\|^2}{2\sigma_u^2})$, where σ_u^2 is the source variance. A binary symmetric source (BSS) is modelled by a uniform distribution $p(\underline{u}) = \frac{1}{2^m}$.

The channel is described by its impact on the input: given an input $\underline{x}$, it provides an output $\underline{y}$, and a probabilistic model description is given by the transition probabilities $p(\underline{y}|\underline{x})$, i.e. the probability distribution of the output $\underline{y}$, for a given input $\underline{x}$. For example, an additive white Gaussian noise (AWGN) Channel is modelled by

$$p(\underline{y}|\underline{x}) = \frac{1}{\sigma_b\sqrt{2\pi}} \exp(-\frac{||y-x||^2}{2\sigma_b^2})$$

, where σ_b^2 is the noise variance. A binary symmetric channel (BSC) is modelled by its raw error probability p, such that $p(\underline{y}|\underline{x}) = p^d(1-p)^{n-d}$ where $d = d_H(\underline{x}, \underline{y})$, the Hamming distance between the channel input and output.

Any type of transmission can be considered: Gaussian signal onto a Gaussian channel, or onto a BSC, or a binary signal on a Gaussian or BS channel, or even multivalued signals. The equations we provide hold in all these cases, unless otherwise stated.

Of course, there is a need for adapting the source to the channel: this is the role of the encoder. The encoder is given by $\underline{x} = \mathcal{C}(\underline{u})$. Specifying $\mathcal{C}$ amounts to designing channel codewords characterized by their probability distribution $p(\underline{x})$.

Conversely, the decoder has the task to convert the channel output to a reconstructed word in the source domain. The decoder is given by $\underline{v} = \mathcal{D}(\underline{y})$, and specifying $\mathcal{D}$ amounts to designing source codewords according to the distribution $p(\underline{v}|\underline{u})$.

At this point, our model is not complete. When tuning transmission systems, one has more degrees of freedom: For example, one can choose the power of the source, and increasing the source power clearly also improves the performance of the system. Hence, it is realistic to consider some constraint on it, in order to take the cost of the system into account. In the same way, one has to choose a distortion measure on the reconstructed source $\underline{V}$: will it be a probability of error, a mean square error (MSE), or some other criteria? This depends on the precise situation.

Hence, one has to consider two constraints:

1. **Average cost per channel input:**

$$\frac{1}{n}\mathbf{E}\ N(\underline{X}) \leqslant P$$

   where $N(\underline{x})$ is a cost function, chosen in such a way that

$$N(\underline{x}) = \sum_{i=1}^{n} N(x_i) \geqslant 0.$$

   For example, the channel input can be power limited $\frac{1}{n}\mathbf{E}\ \|\underline{X}\|^2 \leqslant P$. This is a constraint on $p(\underline{x})$.

2. **Reliability criterion**;

$$\frac{1}{m}\mathbf{E}\ d(\underline{U}, \underline{V}) \leqslant D$$

   where $d(\underline{u}, \underline{v})$ is some distance, chosen in such a way that

$$d(\underline{u}, \underline{v}) = \sum_{i=1}^{m} d(u_i, v_i) \geqslant 0.$$

This is a constraint on $p(\underline{v}|\underline{u})$ because $\mathbf{E}\ d(\underline{U}, \underline{V}) = \sum\!\!\!\!\!\!\int_{\underline{u},\underline{v}} p(\underline{u})p(\underline{v}|\underline{u})d(\underline{u}, \underline{v})$.

A few common situations are:

- Lossless coding, in which $D = 0$;
- *Mean Square Error* (lossy case): $\frac{1}{m}\mathbf{E}\ \|\underline{U} - \underline{V}\|^2 \leqslant D$
- Bit Error Rate (BER): $\mathcal{P}_e = \frac{1}{m}\mathbf{E}\ d_H(\underline{U}, \underline{V}) \leqslant D$, where $d_H(.,.)$ denotes Hamming distance.

Note that the distortion measure is considered here as a constraint, rather as a criterion to be minimized. The reason is that information theory concentrates on the *global rate*, defined as

$$\rho = \frac{n}{m} \quad \text{channel symbol/source symbol}$$

and tries to minimize this global rate ρ under constrainsts (1) and (2). On the other hand, practically, one usually minimizes D under given ρ and constraint (1).

It is rather easy to derive a source/coding performance bound, without showing that it is actually attainable. This is obtained through the definition of the mutual information between $\underline{X}$, $\underline{Y}$, two random variables or vectors (again, discrete or continuous).

The *mutual information* is a *dependancy measure between* $\underline{X}$ *and* $\underline{Y}$ and is defined as:

$$I(\underline{X},\underline{Y}) = \sum\!\!\!\!\!\!\int_{\underline{x},\underline{y}} p(\underline{x},\underline{y}) \log_2 \frac{p(\underline{x},\underline{y})}{p(\underline{x})p(\underline{y})}$$

It is measured in bits (binary units). One can easily understand that it is a dependancy measure: it corresponds to the average amount of "information" that the knowledge of one realization of X brings to knowledge of Y. When applied to vectors (e.g. $I(\underline{X},\underline{Y})$ and $I(\underline{U},\underline{V})$) it denotes bits/word, while in terms of the individual source or channel symbols $\frac{1}{m}I(\underline{U},\underline{V})$ and $\frac{1}{n}I(\underline{X},\underline{Y})$ are measured in bits/symbol.

The plain definition given above can also be rewritten for quantities concerning the channel as:

$$I(\underline{X},\underline{Y}) = \sum\!\!\!\!\!\!\int_{\underline{x},\underline{y}} p(\underline{y}|\underline{x})p(\underline{x}) \log_2 \frac{p(\underline{y}|\underline{x})}{\sum_{\underline{x}'} p(\underline{y}|\underline{x}')p(\underline{x}')}$$

When written as such, $I(X,Y)$ depends on (1) the probability density of the input: $p(\underline{x})$ and, (2), the transition probabilities $p(\underline{y}|\underline{x})$.

Concerning the source-related quantities, one has:

$$I(\underline{U},\underline{V}) = \sum\!\!\!\!\!\!\int_{\underline{u},\underline{v}} p(\underline{v}|\underline{u})p(\underline{u}) \log_2 \frac{p(\underline{v}|\underline{u})}{\sum_{\underline{u}'} p(\underline{v}|\underline{u}')p(\underline{u}')}$$

which depend on the probability density of the source: $p(\underline{u})$ and, (2), the reconstruction error probabilities $p(\underline{v}|\underline{u})$.

From the flow graph of figure 1 , and the above comment, it seems intuitively obvious (and can be proved rigorously) that

$$I(\underline{U},\underline{V}) \leqslant I(\underline{X},\underline{Y})$$

i.e. any processing cannot increase the mutual information. This is widely known in information theory as the *data processing theorem*. We may rewrite this theorem as

$$\frac{1}{m} I(\underline{U},\underline{V}) \leqslant \rho \frac{1}{n} I(\underline{X},\underline{Y}) \tag{1}$$

Recall that, as usual in information theory, we intend to minimize the global rate ρ. In the above inequality, the lower bound on ρ will be minimum if $I(\underline{U},\underline{V})$ is minimized and $I(\underline{X},\underline{Y})$ is maximized. These quantities (min of $I(\underline{U},\underline{V})$ and max of $I(\underline{X},\underline{Y})$) are precisely the quantities that are important in information theory:

Capacity: The capacity of a channel is obtained by maximizing $I(\underline{X},\underline{Y})$ on all possible $p(\underline{x})$ for a given channel, characterized by $p(\underline{y}|\underline{x})$. Of course, this has to be undertaken under constraint (1), and taking the supremum on n:

$$C(P) = \sup_n \max_{p(\underline{x})} \{ \frac{1}{n} I(\underline{X},\underline{Y}) \mid \frac{1}{n} \mathbf{E}\ N(\underline{X}) \leqslant P \}.$$

Rate-distortion: The Rate-Distortion curve $R(D)$ is obtained by minimizing $I(\underline{U},\underline{V})$ on all possible $p(\underline{v}|\underline{u})$ and m for a given $p(\underline{u})$ under constraint (2):

$$R(D) = \inf_m \min_{p(\underline{v}|\underline{u})} \{\frac{1}{m} I(\underline{U},\underline{V}) \mid \frac{1}{m} \mathbf{E}\, d(\underline{U},\underline{V}) \leqslant D\}.$$

Note that in the lossless case ($D = 0$), the $R(D)$ limit is exactly the entropy of the source ($\mathcal{H}(\underline{U}) = I(\underline{U},\underline{U})$). In practical situations, this quantity is bounded (for $\underline{U}$ taking finitely many values)

OPTA (*Optimum Performance Theoretically Attainable*): By taking both Shannon's limits, (1) becomes

$$\rho \geqslant \frac{R(D)}{C(P)}.$$

It turns out that this limit is actually attainable, and is the ultimate performance of our communication system. The proof that this limit is actually attainable can be found in [18].

Typical curve shapes are given below for a memoryless Gaussian source sent over a memoruless Gaussian channel. In this case the capacity reads

$$C(P) = \frac{1}{2}\log_2(1 + \frac{P}{\sigma_b^2})$$

shown on figure 2

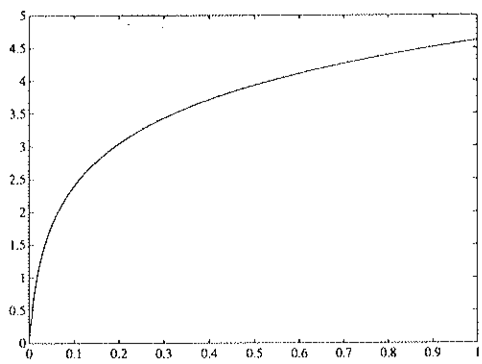

Figure 2: Channel capacity of a Gaussian channel (shape)

and the source rate-distortion function reads

$$R(D) = \frac{1}{2}\log_2(\frac{\sigma_u^2}{D})$$

shown on fig 3

The global rate thus reads :

$$\rho = \frac{R(D)}{C(P)} = \frac{\log_2(\frac{\sigma_u^2}{D})}{\log_2(1 + \frac{P}{\sigma_b^2})}$$

which depends on the channel SNR $\frac{P}{\sigma_b^2}$ and on the source SNR $\frac{\sigma_u^2}{D}$. For a given channel SNR, the global rate is only a dilation of the $R(D)$ curve.

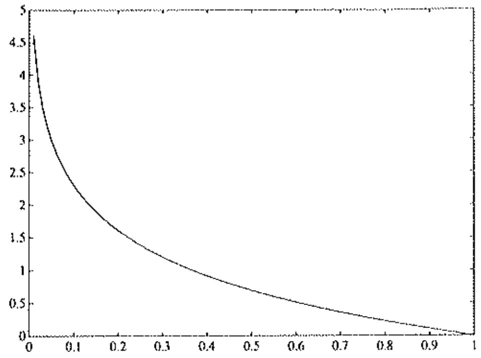

Figure 3: Rate-Distortion function for a Gaussian source (shape)

2.3 Three Shannon theorems

The previous considerations are very general, and often apply to any type of signal (continuous, discrete-valued, correlated, ...). (Note that Shannon's theorems were initially proved in the memoryless case, but can also be proved under more general assumptions)

However, the techniques that are often used for signal protection when sending it on some channel usually assume more specific properties, in order to obtain simpler algorithms, of wider application. The actual system is thus decomposed in more elementary tasks than depicted in figure 1. For example, the encoder can be separated in a transform, followed by vector quantization, followed by an index assignment, and so on... As a consequence, actual systems involve many processing blocks, and one of their tasks is to enforce the inputs or outputs of some blocks to meet some desired property. For example, the input of the channel coding device has usually to be binary and memoryless.

2.3.1 Channel Coding theorem

In channel coding, the source is assumed to be discrete: the "information word" can take M different values, with equal probability. The information word is usually represented in binary form, and thus correspond to $\log_2 M$ information bits.

Of course, in actual systems, a preprocessing is necessary so that this assumption reasonably holds. The encoder output is a code word $\underline{X}$, on n symbols. The channel coding rate is defined as $R_c = \frac{\log_2 M}{n}$ and its unit is in bits of information per coded symbol sent on the channel.

In this case, the OPTA equation reads

$$R_c = \frac{\log_2 M}{n} \leqslant C(P).$$

and *Shannon's channel coding theorem* states that for any arbitrary small $\epsilon > 0$, and for any $R_c < C(P)$, there exists a code of rate $\geqslant R_c$ such that the probability of error satisfies $P_e < \epsilon$.

2.3.2 Source Coding theorem

Here, the channel is assumed to be perfect, that is, $\underline{y} = \underline{x}$ (no errors). The source vector has size m, and the output of the source encoder is an information word which can take M different values. Hence, the source coding rate is $R_s = \frac{\log_2 M}{m}$ encoded bits per source symbol.

In this case, the OPTA inequality gives :

$$R_s = \frac{\log_2 M}{m} \geqslant R(D)$$

Shannon's source coding theorem states that for any arbitrary small $\epsilon > 0$, and for any $R_s > R(D)$, there exists a code of rate $\leqslant R_s$ such that the distortion introduced by this code is $\leqslant D + \epsilon$.

2.3.3 Separation theorem: OPTA

Thanks to the first two Shannon's theorems, it is more or less obvious that (at least theoretically), the source/channel bound can be approached as close as desired using the following strategy:

- The source coding procedure gives a source rate as close as possible to its bound $R(D)$, for a given distortion $\leqslant D$;
- The channel coding procedure has the task of transmitting the resulting information bits with "no" errors (P_e arbitrarily small), while the channel rate is as close as desired from its bound $R_c \approx C(P)$.

By doing so, one indeed obtains a global rate close to Shannon's limit $\rho \approx R(D)/C(P)$ (but higher), with a total distortion $\approx D$ (but smaller).

This is the content of *Shannon's source and channel coding theorem.* Note that, under these conditions, the channel has negligible contribution to the overall distortion: only by the source compression distorts the source.

2.4 Trivial examples

In Shannon's strategy as explained above, each task (source compression as well as channel coding) is incredibly difficult, and practically requires the use of huge data blocks. This motivated the search for making the source and channer coders work together, with the hope that this joint effort will be less costly than for a separated system, while providing performances close to the optimum.

At first glance, this approach does not seem to be compatible with Shannon's results. However we show below two trivial examples for which such an approach can even be optimal.

Example 1 Consider a memoryless binary symmetric source (BSS) to be transmitted via a memoryless Binary Symmetric Channel (BSC) with error probability p, at a global rate $R = 1$. We choose the Bit-Error Rate (BER) as a distortion measure D. In this case, it can be shown that $R_s^* = R(D) = 1 - H_2(D)$, where $H_2(x) = x \log_2 \frac{1}{x} + (1 - x) \log_2 \frac{1}{1-x}$ is the binary entropy function. The channel capacity is $R_c^* = 1 - H_2(p)$, and Shannon's bound becomes

$$\frac{R(D)}{C} = \frac{1 - H_2(D)}{1 - H_2(p)} \leqslant R = 1$$

which simplifies to

$$D \geqslant p.$$

Following Shannon, an optimal procedure is to build a source coder characterized by $R_s \approx 1 - H_2(D)$ for $D = p$, and a channel coder with rate $R_c \approx 1 - H_2(p)$, realizing almost no error.

However, it is easily realized that, by sending straightforwardly the source on the channel, we obviously have $D = p$ and $\rho = 1$. As a result, another optimal solution is: "don't do anything", which is the simplest system. Note that at this other optimal solution, all errors are introduced by the channel, while there is no source compression. This is the opposite situation compared to Shannon's approach.

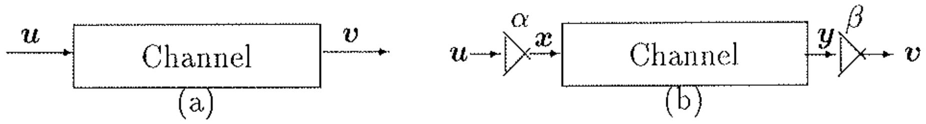

Figure 4: Two optimal systems: (a) binary symmetric and (b) Gaussian source and channel.

Example 2 The situation is quite similar when considering a memoryless Gaussian source with variance σ_u^2 to be transmitted through a Gaussian channel (adding a variance σ_b^2), with global rate $\rho = 1$. The natural distortion measure here is the quadratic norm.

In this case, the optimal rate-distortion function is $R(D) = \frac{1}{2}\log_2 \gamma_s$, where $\gamma_s = \sigma_x^2/D$ is the source to noise ratio,and the capacity is $C(P) = \frac{1}{2}\log_2(1+\gamma_c)$, where $\gamma_c = P/\sigma_b^2$ is the channel to noise ratio. In this case, the Shannon bound is given by

$$\frac{R(D)}{C(P)} = \frac{\log_2 \gamma_s}{\log_2(1+\gamma_c)} \leqslant \rho = 1,$$

which simplifies to

$$\gamma_s \leqslant 1 + \gamma_c.$$

A trivial optimal solution is that illustrated on figure 4 (b), where the gains α and β are such that $\frac{1}{\beta} = \alpha + \frac{1}{\alpha}\frac{\sigma_b^2}{\sigma_u^2}$. It is easily checked that, here again, the OPTA is also attained.

2.5 Possible benefits

These two examples, although quite trivial once the actual problem is understood, are compatible with Shannon theorems. They only show that a "separated system" is not the only solution at the optimum: a solution involving a joint optimization can heavily reduce the complexity of the global system, while maintaining the performance close to the optimum.

Another situation can also appear: due to some external contraint (i.e. complexity, processing delay, ...) the source encoder or the channel encoder can be largely suboptimal. In this case, an optimal solution taking the constraints into account is *not* the cascade of the suboptimal blocks. This is a well known fact. Consider an example fairly obvious in the Signal Processing area: the filter banks. Although ideal filters can lead to perfect reconstruction (PR), the best PR filterbank is *not* only the best approximation of the ideal brickwall filter...

This tends to show that actual gains in terms of performances can be obtained by joint source/channel coding, compared to a classical separated system. However, this gain is expected to be negligible if one uses close to the optimum separated source and channel coders.

Conversely, if the design of the system is heavily constrained (complexity, delay, ...), one can expect noticeable gains when using jointly optimized source and channel coders (or even a fully integrated source/channel coder).

Why it is not always as simple The previous trivial examples are not really usable, since they work only under the following restrictions:

- When $\rho = 1$, rate for which the system performances are limited by the noise level in the channel;
- When the channel and the source are ideally "adapted" to each other.

A really useful result would require more flexibility and robustness towards (i) the global rate (ii) the source, (iii) the channel.

The problem, once stated as such, however, is quite intractable in its full generality. People who have addressed the problem directly in this way know that it is in fact very difficult to obtained performances that are better than those of the trivial system for $\rho = 1$, even when trying to work with larger global rates.

This is the reason why, practically, most results were obtained by starting from a separated communication system, and trying to tune some blocks of the system according to a joint source/channel criterion. Many variants were obtained, depending on the channel models, the blocks that are merged, and so on This is the way we present actual systems in section 5, while making the connection with the theory we just outlined.

We now devote a few paragraphs to describing the basic tools that are used in such a process (section 2.6). These tools are somewhat generalized in section 4.

2.6 Other situations of interest

Note also that the above study was completed in the (classical) situation of a one way, one-to-one communication link. However, other situations are also to be considered, since the problem can be stated in a quite different manner. Two such examples are:

- *Broadcast channels:* This is a one to many transmission situation. Here, the tuning cannot be performed for each user, and the problem is rather to allow each user to recover as much as he or she can from the received signal. This is not really the case in the classical situation considered above, which is more on a "all or nothing" basis: either you recover the full quality signal, or you completely loose it. The solution in this case clearly involves embedded coders and progressive protection.
- *Channels with feedback:* This situation has already been studied in the channel coding situation. Shannon has shown that feedback does not increase capacity. However, it is much easier to get very close to the capacity of the channel. Thus, it should be easier to obtain performances close to the OPTA situation.

3 Some basic tools used in source and channel coding

As explained above, both source coding and channel coding operations are usually performed separately. By doing so, one is working on a subset of fig. 1, and one makes another quantity (the *index*) apparent.

3.1 Vector Quantization

A common tool for data compression is *vector quantization* (VQ). It is a redundancy removal process that makes effective use of four interrelated properties of vector parameters: linear dependencies (correlations), nonlinear dependencies, shape of the *probability density function* and vector dimensionality itself. In fact, it is exactly the general situation of source compression, with full flexibility (yet usually large complexity)

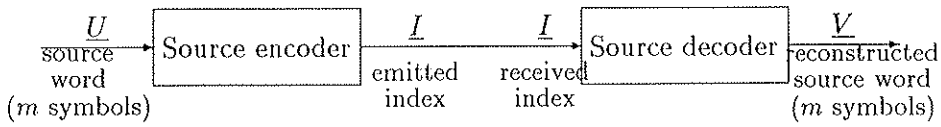

Figure 5: Source coder

Let $\underline{u} = [u_1 u_2 ... u_m]^T$ be an m-dimensional vector whose components $\{u_k, 1 \leqslant k \leqslant m\}$ are real-valued continuous amplitude random variables, (also usually assumed to be of zero-mean, stationary and ergodic), $\underline{v}$ the output of the VQ (another real-valued, discrete-amplitude, m-dimensional vector). We write $\underline{v} = q(\underline{u})$, where q is the quantization operator.

The values of $\underline{V}$ are to be taken from a finite set of L elements: $\underline{v} = \{v_i, 1 \leqslant i \leqslant L\}$, which is called a *codebook*. The design of a codebook consists of partitioning the m-dimensional space of the random vector $\underline{U}$ into L non overlapping regions or cells $\{C_i, 1 \leqslant i \leqslant L\}$ and associating with each cell C_i a vector $\underline{I}_i$. The VQ is designed so as to minimize a given error criterion. The most usual criterion is the average Euclidean distance D, that is minimized over a large number of samples.

$$D = \mathbf{E}\ d(\underline{v}_n, \underline{u}_n) \tag{2}$$

$$d[\underline{u}, \underline{v}] = \sum_{k=1}^{N} (v_k - u_k)^2 \tag{3}$$

which simplifies, assuming ergodicity and stationarity, to

$$D = \sum_{i=1}^{L} \int_{u \in C_i} d(\underline{u}, \underline{v}_i) p(\underline{u}) d\underline{u}.$$

A well known algorithm for VQ design is the *Lindé-Buzo-Gray algorithm* (LBG) [16]. This algorithm is also known as *generalized Lloyd algorithm* (GLA) or *K-means algorithm* and is based on an iterative use of two concepts:

- *Nearest neighbor condition*: Each input vector shall be encoded into its closest codevector. This is the result of optimizing the encoder for a given decoder.
- *Centroid condition*: The optimum codevector assignment for each cell is the centroid of all input vectors being encoded to that cell. This is the result of optimizing the decoder for a given encoder.

3.2 Binary Channel Coding

Channel coding (Also known as *error protection* and *error correcting coding* (ECC).) consists of various methods that add some protection to the message given at the output of the source coding process. This is done by adding some redundancy to the message which is used later in the channel decoder to detect and to correct the errors due to the channel noise.

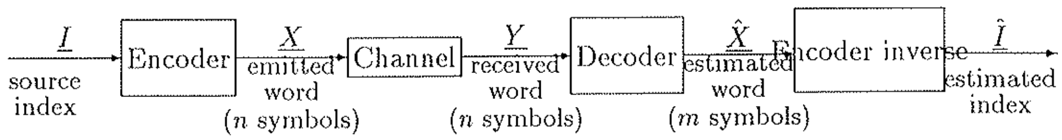

Figure 6: Channel coder

As depicted on figure 6, the information index is translated to some channel codeword by the channel encoder. What is received at the input of the decoder in generally not a codeword. Usually, the decoding process begins first by an estimation of the codeword that has most likely been sent on the channel, followed by an inverse mapping to recover the estimated information index.

One usually tries to minimize the overall error probability P_e. Assuming equiprobable indexes, the optimal solution is given by maximizing over all possible $\underline{I}$ the conditional probability $p(\underline{y}|\underline{x})$. This is known as the Maximum Likelihood detection.

When working on a BSC, this amounts to computing

$$\hat{\underline{x}} = \text{argmin}_{\underline{x}\in\{\text{code}\}} d_H(\underline{x}, \underline{y}).$$

When working on a Gaussian channel, one has to work with the Euclidean distance

$$\hat{\underline{x}} = \text{argmin}_{\underline{x}\in\{\text{code}\}} ||\underline{x} - \underline{y}||^2.$$

3.3 Hierarchical Protection

One way to maintain the performance in the noisy environment transmission is to better protect the more "sensitive" information bits which are suspected to contribute to greater errors. This method is known as *unequal error protection* (UEP) in the literature. Another use of the hierarchy of information will be discussed in section 7.2, page 21.

As an example one can mention an LPC vocoder. The human auditory system is more sensitive to pitch and voicing errors than the errors in the other LPC parameters. In the LPC-10 algorithm, pitch and voicing are encoded so as to prevent single-bit transmission errors from causing gross pitch and voicing perturbations, while no channel coding is provided for the other parameters.

As another example, in one realization of the CELP vocoder, the most significant bits of the binary representations of the codevectors are more sensitive to channel errors than the least significant bits. This property has been used to protect only the most significant bits [20].

Of course, one can imagine a progressive use of channel coders: use the very simple channel coders (even none at all) for the least sensitive bits and the stronger channel coders for more sensitive bits. This approach can be employed in networking problems where many types of data with different sensitivities to noise are to be transmitted. In [10] an example of such a system is explained: for each bit, a factor of sensitivity to channel error is defined. Using this factor, the optimal error rate allowed for each bit that minimizes the effects of channel noise, is estimated. Finally, a UEP coder is used to achieve different levels of protection.

In our opinion, UEP is a preliminary step compared to full joint optimization: In a UEP-based system, the study of the sensitivity of errors on certain bit is used to determine the level of protection that has to be given to it. However, it is not clear whether UEP leads to poorer performances than full source/channel coding or not.

As an example, consider the transmission of a memoryless uniform source on a BSC. Figure 7 compares the Shannon source/channel bound (OPTA) for this situation to the one that is obtained by the following procedure: First, the (real-valued) Gaussian source is represented in terms of individual bit streams, weighted by 2^{-i}. Then, the system send separately the various bit streams, with the appropriate rate corresponding to its importance in the representation of teh Gaussian source. More precisely, we are comparing the unconstrained bound (denoted as Shannon bound) to that which would be obtained by a separate system in which each bitstream is considered as such, and optimally tuned according to its importance in the Gaussian source. This is the best situation an UEP-based system would achieve. Clearly, both bounds are close to each other, and when considering practical situations, with coders of similar complexity, it is not clear which procedure will win.

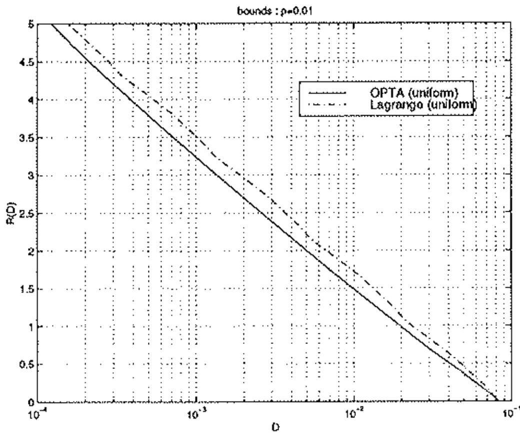

Figure 7: Unconstrained OPTA vs "bitstream" OPTA (Lagrangian bound)

4 Generalized Lloyd Algorithm

As explained above, information theory aims at minimizing the global rate. In actual situations, however, the overall system is generally constrained, and when the general architecture is decided, one tries to carefully tune its parameters in order to *minimize the distortion* D, for a given global rate ρ. (In this section, the underlines are dropped for convenience, although all quantities are random vectors)

This work is classically performed separately on the source and channel: First minimize the distortion, assuming that the channel does not introduce any error, and then tune the error correcting codes in such a way that the transmission errors are not "disturbing".

However, this can be performed jointly by a procedure very similar to a Lloyd algorithm, as used in classical vector quantization.

The aim here is to minimize *jointly* the distortion introduced by the joint source coder. The underlying criterion is easily written as:

$$D = \frac{1}{m}\mathbf{E}\,||U - V||^2 = \frac{1}{m}\sum\!\!\!\!\!\!\int_u p(u)\sum\!\!\!\!\!\!\int_y p(y|\mathcal{C}(u))||u - \mathcal{D}(y)||^2$$

In a Lloyd algorithm, the minimization of D is performed by an iterative procedure, in two steps:

1st step: Generalized centroid condition The encoder $\mathcal{C}(.)$ is fixed, and one optimizes the decoder $\mathcal{D}(.)$. This minimization is easily performed by first rewriting the criterion as:

$$D = \frac{1}{m}\sum\!\!\!\!\!\!\int_y \left[\sum\!\!\!\!\!\!\int_u p(u)p(y|\mathcal{C}(u))||u - \mathcal{D}(y)||^2\right]$$

The term between brackets is a distortion term, depending only on y, say D_y, and the minimum D is the sum of the minimal contributions for each y , as given by: $D_{\min} = \sum\!\!\!\!\!\!\int_y D_{y\min}$. Hence, the optimal decoder $\boldsymbol{v} = \mathcal{D}^*(y)$ providing the minimum distortion is obtained by setting the derivative of D_y with respect to $v = \mathcal{D}(y)$ to zero:

$$0 = \sum\!\!\!\!\!\!\int_u p(u)p(y|\mathcal{C}(u))(u - \mathcal{D}^*(y)),$$

and we obtain:

$$\mathcal{D}^*(y) = \frac{\sum\!\!\!\!\!\!\int_u p(u)p(y|\mathcal{C}(u))u}{\sum\!\!\!\!\!\!\int_u p(u)p(y|\mathcal{C}(u))} \quad (4)$$

$$= \frac{\sum_x p(y|x)\sum\!\!\!\!\!\!\int_{u|\mathcal{C}(u)=x} p(u)u}{\sum_x p(y|x)\sum\!\!\!\!\!\!\int_{u|\mathcal{C}(u)=x} p(u)} \quad (5)$$

$$= \frac{\sum_x p(y|x)\sum\!\!\!\!\!\!\int_{u|\mathcal{C}(u)=x} p(u)u}{\sum_x p(y|x)p(x)} \quad (6)$$

Here (4) is true whatever the type of signals and channels, and (5) only holds when the channel symbols take discrete values. This formula shows a strong similarity with the classical centroid update, but for the appearance of the weighting by $\sum_x p(y|x)$ which is only due to the channel model.

Note that, even if it is not easily demonstrated, this formula corresponds to the classical result that this optimum decoder computes the conditional expectation of the source vector based on the channel output, i.e.: $\mathcal{D}(y) = \mathbf{E}\,(U|Y = y)$.

2nd step: Generalized nearest neighbor condition Here we optimize the encoder $\mathcal{C}(.)$ for a given decoder $\mathcal{D}(.)$. First rewrite the distortion as:

$$D = \frac{1}{m}\sum\!\!\!\!\!\!\int_u p(u)\left[\sum\!\!\!\!\!\!\int_y p(y|\mathcal{C}(u))||u - \mathcal{D}(y)||^2\right]$$

and the optimal encoder is obtained as:

$$C^*(u) = \text{argmin}_u \sum\!\!\!\!\!\!\int_y p(y|\mathcal{C}(u))||u - \mathcal{D}(y)||^2$$

which is the minimum of the "generalized distance" (taking into account the effect of the channel) between u and its reconstructed value $\mathcal{D}(y)$. When there is no channel error (i.e. $p(y|x) = \delta(x,y)$), this formula reduces to the classical one. Note that, when the channel is not very noisy, this is a reasonable approximation. Now, if v takes discrete values (depending on the channel model), this formula is easily rewritten:

$$\begin{aligned} C^*(U) &= \text{argmin}_u \sum_v ||u - v||^2 \sum\!\!\!\!\!\!\int_{y|\mathcal{D}(y)=v} p(y|\mathcal{C}(u)) && (7) \\ &= \text{argmin}_u \sum_v ||u - v||^2 p(v|x) && (8) \end{aligned}$$

The algorithm As classically done, the algorithm repeatedly performs both steps until the distortion (or the encoder/decoder) remains stable. Note that equations (6) and (8) can be practically evaluated in many situations.

5 Practical approaches

The methods that have been proposed for performing joint source and channel coding were initially performed on simple models. Since then, more and more efficient algorithms were proposed, and we propose here an organised review of these methods, pointing to the previous considerations relying on information theory and the generalized Lloyd algorithm when relevant.

These methods are presented with reference to a scheme in which all tasks that have to be completed in sequence are explicitly shown. Depending on the assumptions on which these methods rely, some blocks are merged, and have to perform a more complex task which is then to be optimized for minimum distortion under noisy channel conditions.

5.1 Communication Model

A general model of the transmission block diagram is depicted on figure (8):

- the message emitted from the source is first passed from a transformation block (for taking correlations into account)
- source compression (quantization) is performed in order to eliminate some redundancy;
- *index assignment* (IA), also known as *Labeling* then gives a good bit pattern to each codevector;
- the resulting bits are then protected by a channel encoder;
- the modulation shapes the signal before transmitting it to the physical channel;
- this channel introduces some perturbations, according to some model
- a series of "inverse" operations: demodulator, channel decoder, inverse index assignment, codebook search and inverse transformation are applied to recover the original message.

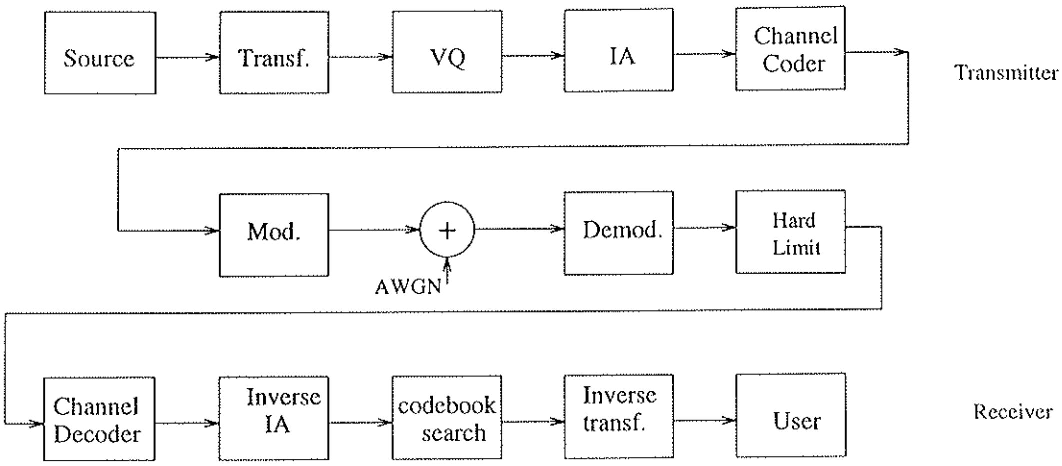

Figure 8: Block diagram of the transmission system.

This general model can be simplified in different ways. In fact, each method described in this paper makes its own assumption on the model and combines some of the blocks in figure (8) into a single block and/or easily omits some of the blocks. For example, a BSC, simply models the modulator, channel noise, demodulator and hard limiter set as in one block. Some methods make a single block from two or three other blocks and apply some optimization routines to it. As an extreme example, in Modulation Organized Vector Quantization, all the blocks: vector quantization, index assignment, channel coding and modulation are merged together and locally optimized.

5.2 Index Assignment

The Indexing step makes the translation of the discrtete real-valued centroids to some code (the index) rthat is transmitted to the channel.

In a perfect channel situation, and if the source coder has used VQ, any index assignment will have the same performance. The reason for this is that after LBG, all centroids have equal probability, and index assignment does not change the bit rate. Furthermore, it does not affect the average distortion, either.

However, when a non-trivial channel is present, this assignment plays an important role in determining the overall VQ performance. Basically, LBG by itself does not provide any protection against channel noise because any change of bit can redirect one codevector to any other one in the codebook. So, even a low *bit error rate* (BER) can heavily distort the signal if no index assignment strategy is used.

Once the origin of the problem is recognized, the task to be assigned to the IA is clear: channel errors should change some index to other ones that are likely to be "close" to the initial one. Hence, the problem is too find an IA for which the distance (to be chosen according to the channel model) between indexes is similar to the distance between centroids (Euclidian distance). This is called *pseudo Gray coding* in the literature[31].

It must be noticed that the IA is an *non polynomial* (NP)-complete task since there are $\frac{(2^b)!}{2^b \times b!} = \frac{(2^b-1)!}{b!}$ possible distinct combinations to assign $L = 2^b$ codevectors to L codewords. The 2^b and the $b!$ factors in the denominator eliminate respectively the symmetric cases and the bit permutation cases. This results 8.3×10^{499} distinct possible combinations for $b = 8$ bits.

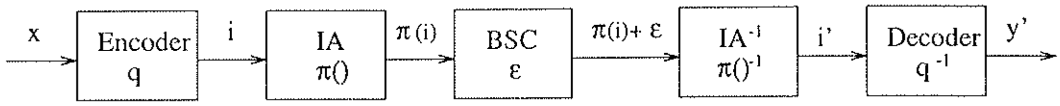

Figure 9: Block diagram of the VQ based coding system used over a noisy channel.

Of course, some strategies provide better initial estimates of the codebook than other ones. It has been soon recognized that the VQ codebook design stragegy known as *splitting* training was efficient with that respect. In the splitting technique, the training begins with a few (possibly just one) codevectors. Each codevector is then divided into two sister codevectors each time with the small perturbations, and these new centroids are updated. Farvardin has observed [7] that when the splitting technique [16] is used for VQ training, the resulting codebook has a *natural* ordering that can somehow protect the signal in the presence of channel errors. This is due to the splitting mechanism which makes *sister* codevectors behave similarly. However, this is not entirely efficient because if an error occurs on the first splitted bits, the resulting distortion can be much greater.

A general solution to the IA problem is to perform the VQ design first and then permute the indices in such a way that the resulting codebook becomes more robust against channel noise. It is shown in [5] that a non negligible reduction in distortion can be obtained through a well designed IA rather than a random one.

Note that this strategy is *not* a simultaneous optimization of source and channel coding: only a "translation step" is optimized. A more global strategy will be discussed in the next section. Note also that, when trying to address directly a general problem, it is quite often found that solutions depend on some implicit indexing, that greatly influence the system performances.

The problem can be formulated simply as explained in figure (9):

Simulated Annealing Since IA is an NP-complete problem, Farvardin used *simulated annealing* (SA) to solve it [7]. SA is a Monte Carlo algorithm which has been widely used to solve combinatorial problems [12], and is recalled below. An appropriate *temperature* variable, T, is to be defined. This variable is initialized to a high value T_m, in the beginning of the process and is decreased progressively until a sufficiently small value T_f, is reached. A high value of T signifies a high degree of randomness while a low value of it means that nothing is left at random. A high value of T at the begining of the process, permits to avoid many local optima.

The SA algorithm can theoretically give the global optimum solution, unconditionally on the initial state, provided that the initial value, T_m, and the schedule of decreasing T, are chosen appropriately. Unfortunately, this is difficult to achieve and therefore good optima from SA might be difficult to obtain in most practical cases.

As an example, Farvardin reported a *signal-to-noise ratio* (SNR) of about 8.95 dB for SA, compared to 8.87 dB for a naturally organized LBG with splitting. The test parameters were: $\epsilon = 10^{-2}$, $N = b = 8$ bits for a first order Gauss-Markov source with correlation coefficient $\rho = 0.9$.

Binary Switching Algorithm Another algorithm for an optimum IA was proposed by Zeger and Gersho [31]: *binary switching algorithm* (BSA). In BSA, to each codevector $\underline{v}$ is assigned a cost function $C_\pi(\underline{v})$. This cost function is a measure of the contribution to the total distortion due to the possible channel errors when $\underline{v}$ is decoded, assuming a certain permutation, π. Then the codevectors are sorted in decreasing order of their cost values. The vector that has the largest cost, say $\underline{v}$, is selected as a candidate to be switched first.

A trial is conducted: $\underline{v}$ is *temporarily* switched with each of the other codevectors to determine the potential decrease in the total distortion $D_\pi = \sum_{k=0}^{L-1} C_\pi(\underline{v}_k)$, following each switch. The codevector which yields the greatest decrease in D_π when switched with $\underline{v}$ is then switched *permanently* with it. The algorithm is then repeated for the next highest cost and so on.

Although a global optimal IA is not necessarily obtained by BSA, good locally optimal solutions have been reported [31]. Simulation tests have been made with a first order Gauss-Markov source as well as an *independent identical distribution* (iid) and speech waveform. As an example, for $\epsilon = 10^{-2}$, $N = 4$, $b = 8$ bits, 1.5 dB gain has been achieved compared to the initial state.

Link with the generalized Lloyd procedure Clearly, such IA strategies are small subsets of a full source/channel optimisation procedure. this will be quite explicit when making the connection with the general strategy explained in section 4.

In fact, this IA problem cannot really be cast into the generalized Lloyd framework defined previously, since changing an index assignment amounts to changing *both* the encoder and the decoder. This explains the complexity of the task : an IA strategy tries to optimize (partially) both sides of the system (emission and reception.

However, it has many advantages: the rest of the system remains unchanged, each coder is tuned separately, and it can greatly improve the performances. Its drawback are easily understood: the complexity of the tuning (but it has to be done once for all), the link with plain VQ (for obtaining centroids with equal probability), and the combinatorial nature of the optimization (it is not quite sure than constraining the search to an indexing simplifies the procedure...)

6 Simultaneous Optimization of Quantizer and Channel Coder

When trying to make the connection with the generalized Lloyd procedure, one recognizes that the centroid values should also be adapted to the presence of a specific channel model. In this case, the full VQ is matched to the minimization of the global distortion. Figure 10 illustrates the block diagram for this situation.

We outline below two methods: "channel optimized vector quantization" which is a generalization of LBG for the noisy channel transmission and "self organizing hyper cube" which is a generalization of Kohonen map into higher dimensions.

6.1 Channel Optimized Vector Quantization

Farvardin proposed a joint optimization for the source and the channel coders [8, 9]. It is in fact a straightforward application of the generalized Lloyd procedure in section 4, applied on a BSC, with a quadratic distortion as a criterion to be minimized. first rewrite the general distortion measure involving both the quantization error and the

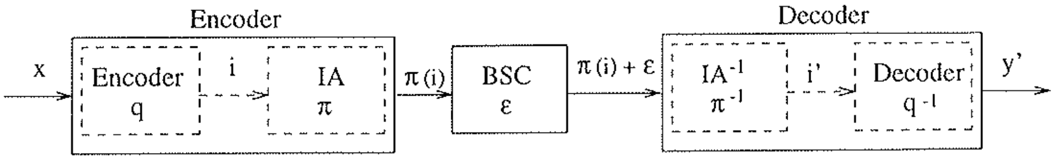

Figure 10: Block diagram of the VQ based coding system used over a noisy channel. Here, the IA is included in the encoding process.

error due to channel perturbation [2] in the case of a BSC. Here $\mathcal{C}(u)$ denotes the index associated with emitted centroid $\underline{x}$ and $\mathcal{D}(\underline{y})$ denotes the centroid associated to the received index $\underline{y}$. $p(\underline{y}|\mathcal{C}(\underline{u}))$ represents the channel effects: probability that some emitted index $\mathcal{C}(\underline{u})$ is changed to index $\underline{y}$.

The resulting algorithm is very similar to the LBG algorithm and is named *channel optimized vector quantization* (COVQ). The cells, $\mathcal{C}(\underline{u})$, are updated according to equation (6) [9], and the centroids $\mathcal{D}(\underline{y})$ according to (8).

In a few words, each input vector $\underline{u}$ is classified into the cell with the least *expectation of distortion*, while $\mathcal{D}(\underline{y})$ represents the centroid of all input vectors that are decoded into $\mathcal{D}(\underline{y})$, even if the received index, $\underline{y}$, is different from the emitted one $\mathcal{C}(\underline{x})$. Of course, both equations can be simplified into the LBG learning equations by simply assuming that:

$$p(\underline{y}|\mathcal{C}(\underline{x})) = \begin{cases} 1 & : \quad \underline{y} = \mathcal{C}(\underline{x}) \\ 0 & : \quad \underline{y} \neq \mathcal{C}(\underline{x}) \end{cases} \tag{9}$$

This way, LBG can be regarded as a special case of COVQ when the parameter of the BSC is zero..

It can be shown that the obtained optimum encoding cells are convex polyhedrons and that some cells might vanish thus creating *empty cells* [8]. This means that the system trades quantization accuracy for less sensitivity to channel noise. Figure (11) shows an example of COVQ for a two-dimensional (N=2), three-level (L=3) VQ and a *discrete memoryless channel* (DMC) with the parameters as in the following Table.

Probability transition matrix $P(i|j)$ in the DMC example.

$i\|j$	1	2	3
1	$1-2\epsilon$	ϵ	ϵ
2	2ϵ	$1-4\epsilon$	2ϵ
3	ϵ	ϵ	$1-2\epsilon$

This figure illustrates that when the channel noise is large, there is a risk that some cells vanish. Assuming that there are L' nonempty encoding cells, $L' \leqslant L$, only L' codewords need to be transmitted. Of course, any of L binary codewords may be received and therefore the codebook must remain of size L. It is interesting to observe the analogy that exists between the presence of empty cells (codevectors with no corresponding input vector) and the added redundancy in channel coding.

Simulations have been reported [8, 9] for first order Gauss-Markov sources, as an example, for $\rho = 0.9$, $\epsilon = 10^{-2}$, $N = b = 8$. COVQ and naturally organized LBG with

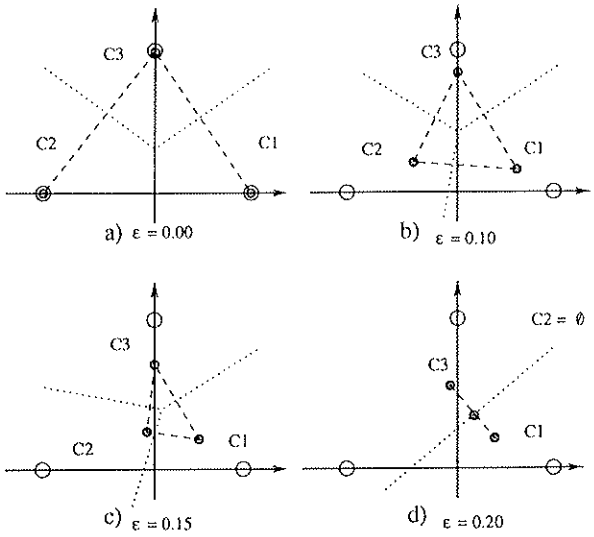

Figure 11: Figures (a), (b), (c) and (d) show the quantization cells for $\epsilon = 0.00, 0.10, 0.15$ and 0.20, respectively for a simple DMC. The codevectors get closer when ϵ increases and finally one of the cells, C_2, vanishes for $\epsilon = 0.20$. The c_i are the codevectors for a non noisy environment.

splitting have resulted in 9.70 dB and 8.87 dB, respectively. COVQ has had $L' = 26$ empty cells (out of $L = 256$), in this example.

6.2 Self Organizing Hyper Cube

When working with a BSC, it is clear that the channel introduces errors characterized by their Hamming norm, as explained above. However, the norm that characterizes the source errors is Euclidean. It is thus logical to propose a direct mapping from the input space to the Hamming space as a good way to build Vector Quantizers at the emitter side. This has been proposed in [28]; this mapping is roughly a b-dimensional generalization of the 2-dimensional Kohonen, also known as *self organizing map* and *competitive map*. [13, 14]. Hence it is named as *self organizing hyper cube* (SOHC). SOHC is trained with an algorithm similar to the Kohonen algorithm with some modifications: the codevectors are arranged in a b-dimensional cube (instead of a 2-dimensional map); the neighborhood function is defined in the hyper cube and Hamming distance (d_H) is used as the distance measure of the binary representations of the indices (instead of Euclidean distance in the Kohonen map).

As a result of such a definition of distance between the codevector indices, in SOHC, there is almost no difference between the quantized bits. In other words, the least and most significant bits have no sense in SOHC. Roughly speaking, the effect of noise on each bit is almost the same.

Examining figure (12), if we consider that the chosen codevector to be transmitted is 0000, a single bit of error can commute it to either of 1000, 0100, 0010 or 0001. Since all these codevectors are the first order neighbors of 0000 (with $d_H = 1$), this commutation does not contribute a gross error.

Adding the splitting technique to SOHC, improves further its performance [29]. In SOHC with splitting, each time that the codewords are splitted, the dimension of the codebook is increased, too. SOHC has been tested for quantizing and transmitting *log area ratio* (LAR) parameters of speech, over a BSC. Better objective results were

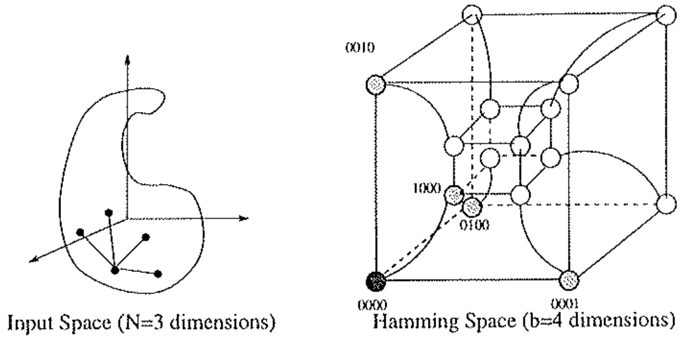

Figure 12: An example of SOHC. Left: input space. Right: SOHC. Codevector 0000 and its first order neighbors are highlighted in both spaces.

reported, compared to naturally organized VQ and Kohonen map, specially for high transition probabilities. For instance, with a transition probability $\epsilon = 10^{-2}$, $N = 10$, $b = 8$, the *spectral density distortion* (SD) [11] measure for SOHC, Kohonen map and naturally organized LBG with splitting were about 3.3 dB, 3.4 dB and 3.5 dB, respectively. With SOHC, a further protection is also possible, using some classic error control coding technique, since SOHC provides the bit patterns in which all the bits are (almost) equally likely to cause error.

7 Direct Modulation Organizing Scheme

Another possible source-channel configuration is the direct modulation organization. In this configuration, the encoder includes the modulator and benefits directly from the flexibility that is naturally present in a constellation. As shown in figure (13), the channel is considered with an *additive white Gaussian noise* (AWGN).

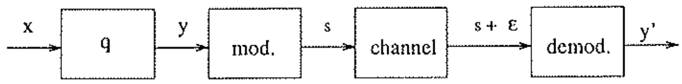

Figure 13: Block diagram of the Direct Modulation VQ based coding system, used over an AWGN channel. The source encoder, channel encoder and modulator are represented in one block. $s \in \{setofallconstellationpoints\}$.

Several works have been done in this field. To mention some, we can indicate a competitive learning algorithm which gives aft direct mapping from input space to the signal space is presented [25]; the hierarchical modulation, in which the constellation points are located to minimize the error expectation is explained [21, 4]. There exists some other works that we will not extend in this paper: joint optimization of three blocks (source coder, channel coder and modulator) [26]; Trellis coding and Lattice coding which are special kinds of covering the signal space by the constellation points [15].

7.1 Modulation Organized VQ

Withdrawing any binary representation, Skinnemoen proposed the *modulation organized vector quantization* (MORVQ) [25]. This method uses a quantizer which maps the codevectors directly into the constellation plane. It makes efficient use of the Kohonen learning algorithm to map the N-dimensional input space to the 2-dimensional signal space, in such a manner that the close codevectors in the modulation space, are assigned to the close points in the input space. This property is obtained by proper use of a neighborhood function [13, 14] and the resulting codebook has some organized structure. Having this structure, most little changes due to channel noise make the output codevector to be one of the neighbors of the source vector and so the distortion will not be very important.

Skinnemoen observed a great difference between explicit error protection and the structure of a codebook. He states that any transmission system (with or without error protection) has a BER working threshold. Above that limit, the system's performance breaks down. The role of MORVQ is to increase this threshold. This is the great advantage of MORVQ; however, in MORVQ, no more channel coding can be added since it does not produce any intermediate bit pattern which can be processed by classical channel coders.

Good numeric results have been reported in quantizing first order Gauss-Markov sources and *line spectrum pairs* (LSP) parameters of speech spectrum in an AWGN channel. As an example, for quantizing LSP parameters with $N = 10$ and $L = 256$, SD was 2.11 dB and 7.82 dB, respectively for MORVQ and LBG, for a highly noisy channel. Also it is observed that for MORVQ, the degradation curve by increasing channel noise is rather smooth while for LBG there is a threshold above which the system performance drops rapidly.

7.2 Hierarchical Modulation

Ramchandran et al. have proposed in [21] a Multi-Resolution broadcast system. One basic idea in their proposition consists in partitioning the information into two parts: the coarse information and the refinement or the detail information. This approach is intended to be used in conjunction with transformation based source coding methods, like subband and wavelet coding, since they have a natural multiresolution interpretation. The coarse information is to be received correctly even in a very noisy transmission environment, while the detail information is mostly destinated to the receivers whose channels have better qualities (graceful degradation) This classification can even be made more precise, making several classes of importance.

It has to be noted that this approach is naturally well suited to a broadcast situation rather than to a point to point link. Like the previous approach, the idea is to match the transmission constellation to the source coding scheme, without merging both steps. Ramchandran et al proposed the use of a multi-resolution constellation as depicted in figure (14). The coarse information is carried by the clouds, while inside each cloud, the mini-constellations or *satellites* provide the details. The loss of coarse information is associated with the receiver inability to decipher correctly which cloud was transmitted while the loss of detail information occurs when the receiver confuses one intra-cloud signal point for another. This property is already present in any QAM constellation, but is reinforced by the uneven localization of the points in the cloud.

Of course, many other configurations could be thought of, yielding similar properties. The same idea has been used in conjunction with *Trellis modulation coding* (TMC) as well as with embedded channel coding [21].

Combelles et al. [4] have used the same idea of multi resolution coding, in conjunction with Turbo code, which was used to protect the coarse and detail information

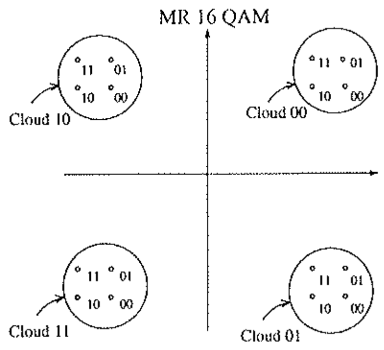

Figure 14: An example of Multi-Resolution Constellation. Each set of close points constitutes a cloud with four satellites points surrounding it. The detail information is presented in the satellites while, the course information is represented in the clouds. So there is 2 bits of coarse information and 2 bits for detail. Note also that the Gray code is used for numbering the satellites (and the clouds) in such a way that the codewords with Hamming distance equal to 2 are far from each other. This is like the application of Karnaugh map in digital design and can be used for larger constellations, too.

with $\frac{1}{2}$ and $\frac{3}{4}$ rates, respectively. They achieved 4 dB better performance for the coarse information while 2 dB degradation for the detail information, compared to a single resolution system using Turbo code, with the same overall spectral efficiency to obtain the same error rate (10^{-4}), while for a Rayleigh fading channel their simulation shows 5 dB of better performance for the coarse information and 3 dB degradation for the detail information.

In [21] an example is given where with a multi resolution system, the broadcast coverage radius (64 km) is much greater than for a single resolution system (45 km) while for the multi resolution system, the radius of full data availability is a little smaller (38 km).

8 Other approaches

Another frequent approach, which will not be detailed here is based on an explicit use on the residual correlations found in the information bit stream that is sent on the channel.

In fact, as outlined above, many processing blocks inside the source coding procedure aim at decorrelating the data. However, this decorrelation is not perfect, and can be used for protecting the errors that may arise when transmitting the data.

It was shown in [22] that this residual correlation could be used for reducing the errors introduced by the channel. This approach was elaborated in a series of other papers and explicitly models the correlated source [23], or tries to adapt the channel encoder to that situation [1].

9 Conclusion

This paper considers the joint source and channel problem from both a theoretical and practical point of view. As a result, we could present several methods in a unified framework.

This method also allowed to make the general mechanisms more precise. More specifically, the generalized Lloyd procedure, given in its most general formulation, and valid for any type of source and channel could lead to many applications in more specific cases.

Many questions still remain unanswered, for example:

It is clear that the joint optimization of a source and channel coder aims mainly at *simplicity*. What was it traded for? Namely, one has tuned the system according to some knowledge of the channel performance. If ever the sensitivity of the performances with respect to this tuning were too high, this could put the results into question.

Unequal error protection is a simple way of performing a precise adaptation between the source and the channel coder. This mechanism can be pushed much more than usually done, and in a practical situation, it is not clear yet which strategy will provide the most efficient system.

However, many other questions are yet unanswered, and many methods are still to be studied. The joint source and chanel coding is an area of increasing activity, which is likely to last for some time...

References

[1] F. Alajaji, N. Phamdo, and T. Fuja, " Channel codes that exploit the residual redundancy in CELP-encoded speech," *IEEE Trans. on Speech and Audio Processing* Sept. 1996, Vol. 4, pp. 325-336

[2] E. Ayanoglu, R.M. Gray, "The Design of Joint Source and Channel Trellis Waveform Coders", *IEEE Trans. on Information Theory*, vol. 33, no. 6, Nov. 1987, pp. 855-865.

[3] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*, Prentice-Hall, 1971.

[4] P. Combelles, D. Castelain, C. Berrou, "Système de Codage Canal Hiérarchique pour la Télévision Numérique", *Quatorzième colloque GRETSI*, Juan-les-Pins, France, Sep. 1993, pp. 615-618.

[5] J.R.B. De Marca, N.S. Jayant, "An Algorithm for Assigning Binary Indices to the Codevectors of a Multidimensional Quantizer", *Proc. of the IEEE International Conference on Communications*, Seattle, WA., June 1987, pp. 1128-1132.

[6] N. Farvardin, V. Vaishampayan, "Optimal Quantizer Design for Noisy Channels: An Approach to Combined Source-Channel Coding", *IEEE Trans. on Information Theory*, vol. 33, no. 6, Nov. 1987, pp. 827-838.

[7] N. Farvardin, "A Study of Vector Quantization for Noisy Channels", *IEEE Trans. on Information Theory*, vol. 36, no. 4, July 1990, pp. 799-809.

[8] N. Farvardin, V. Vaishampayan, "On the Performance and Complexity of Channel Optimized Vector Quantizers", *IEEE Trans. on Information Theory*, vol. 37, no. 1, Jan. 1991, pp. 155-160.

[9] N. Farvardin, "Speech Coding Over Noisy Channels", in *Speech Recognition and Coding: New Advances and Trends*, Springer 1995, pp. 387-404.

[10] K. Fazel, J.J. Lhuillier, "Application of Unequal Error Protection Codes on Combined Source-Channel Coding of Images", *Proc. of the International Conference on Communications*, 1990, pp. 320.5.1-320.5.6 (0898-0903).

[11] R.M. Gray, A. Buzo, A.H. Gray, Y. Matsuyama, "Distortion Measures for Speech Processing", *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 28, No. 4, pp. 367-376, Aug. 1980.

[12] S. Kirckpatrick, C.D. Gelatt, M.P. Vecchi, "Optimization by Simulated Annealing", *Science*, Vol. 220, pp. 671-680, May 1983.

[13] T. Kohonen, *Self Organization and Associative Memory*, Springer-Verlag, 1984.

[14] T. Kohonen, "The Self-Organizing Map", *Proc. of the IEEE*, Vol. 78, pp. 1468-1480, Sep. 1990.

[15] R. Laroia, N. Farvardin, S.A. Tretter, "On Optimal Shaping of Multidimensional Constellation", *IEEE Trans. on Information Theory*, vol. 40, No. 3, July 1994, pp. 1044-1056.

[16] Y. Lindé, A. Buzo, R.M. Gray, "An Algorithm for Vector Quantizer Design", *IEEE Trans. on Communications*, Vol. 28, No. 1, pp. 84-95, Jan. 1980.

[17] J.L. Massey, "Joint Source and Channel Coding", *NATO courses*, pp. 279-293.

[18] R.J. McEliece, *The Theory of Information and Coding*, Addison-Wesley, 1977.

[19] N. Moreau, *Techniques de Compression des Signaux*, Masson, 1995.

[20] K.K. Paliwal, B.S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame", *IEEE Trans. on Speech and Audio Processing*, Vol. 1, No. 1, pp. 3-14, Jan. 1993.

[21] K. Ramchandran, A. Ortega, K.M. Uz, M. Vetterli, "Multiresolution Broadcast for Digital HDTV Using joint Source/Channel Coding", *IEEE Trans. on Selected Areas in Communications*, Vol. 11, No. 1, pp. 3-14, Jan. 1993.

[22] K. Sayood and J.C. Borkenhagen, "Use of residual redundancy in the design of joint source-channel coders" *IEEE Trans. on Communications*, June 991, vol.39, pp.838-846

[23] K. Sayood, F. Liu, and J.D. Gibson, "A constrained joint source/channel coder design" *IEEE J. Selec. Areas on Com.* Dezc.1994, Vol.12, pp. 1584-1593

[24] C.E. Shannon, "A Mathematical Theory of Communication", *Bell System Technical Journal*, Vol. 27, no. 3-4, pp. 379-423 and 623-656, July and oct. 1948.

[25] P.H. Skinnemoen, *Robust Communication with Modulation Organized Vector Quantization*, Thesis, Norwegian Institute of Technology, 1994.

[26] V.A. Vaishampayan, N. Farvardin, "Joint Design of Block Source Codes and Modulation Signal Sets", *IEEE Trans. on Information Theory*, July 1992, pp. 1230-1248.

[27] S. Vembu, S. Verdú, Y. Steinberg, "The Source-Channel Theorem Revisited", *IEEE Trans. on Information Theory*, vol. 41, No. 1, pp. 44-54, Jan. 1995.

[28] S.B. Zahir Azami, B. Azimisadjadi, V. Tabatabaii, H. Borhani, "Self Organizing Hyper Cube and Some of Its Applications", *Proc. of the First Iranian Conference on Electrical Engineering*, Tehran, Iran, May 1993, pp. 464-472.

[29] S.B. ZahirAzami, G. Feng, "Robust Vector Quantizer Design Using Competitive Learning Neural Networks", Proceedings of European Workshop on Emerging Techniques for Communication Terminals (Toulouse, France), July 1997, pp. 72–75.

[30] S.B. Zahir Azami and P. Duhamel and O. Rioul, "Combined Source-Channel Coding for Binary Symmetric Channels and Uniform Memoryless Sources". Proc. of teh GRETSI conf. Sept. 97

[31] K. Zeger, A. Gersho, "Pseudo-Gray Coding", *IEEE Trans. on Communications*, Dec. 1990, pp. 2147-2156.