Chapter 6

Adaptive Quantization Without Side Information[1]

Contents

6.1 Introduction

The most successful methods for lossless compression of data, such as arithmetic coding [55, 116, 80], Lempel-Ziv coding [120] or dynamic Huffman coding [35, 54, 111], are all adaptive (see [6] for an extensive review of lossless compression). While the initial work on entropy coding (e.g. Huffman coding) relied on knowing, or measuring, the source distribution, adaptive schemes make no prior assumptions on the source statistics, which the coders try to learn. Here, we are concerned with designing

[1] For publications related to this chapter see [75]

adaptive quantization algorithms which exhibit characteristics similar to those of the abovementioned adaptive lossless compression schemes. In the most general context we can define our problem as that of adapting some or all of the parameters of a quantizer/entropy coder system (including bin sizes, reconstruction levels, codeword lengths and dynamic range) to the changing statistics of an input source. We make few assumptions on the source and, in particular, we allow it to be have long term dependencies and show varying "local" behavior. We will assess the performance of the adaptive schemes by comparing their rate-distortion characteristics with those achievable by means of non-adaptive schemes,

We are concerned with systems where the adaptation occurs based *only* on the causal past so that both encoder and decoder can adapt in the same manner, and no extra information needs to be sent. We propose to split the adaptation algorithm in two parts (see Fig. 6-1(a)):

1. Model estimation: Based on the previous N samples we estimate the distribution function of the source.

2. Quantizer design: for the given estimated distribution the new quantizer parameters are computed.

The advantage of splitting the algorithm in this manner is that well known quantizer design techniques can then be used (such as the Lloyd-Max algorithm in the case of constant rate quantizers). If we correctly estimate the distribution then we are guaranteed optimality.

A further question arises as to how much memory should be used in estimating the distribution. Clearly, if the source input were independent identically distributed (i.i.d), it would be reasonable to accumulate statistics over a long time window. Conversely, if the source input distribution were changing over time, shorter windows would have to be used. If the window size is kept constant, choosing an appropriate

(b)

Figure 6-1: Adaptive quantization algorithm. (a) The adaptation algorithm can be decomposed in two parts (i) the estimation of the input distribution based on past samples and (ii) the design of the new quantizer *given* the estimated distribution. (b) In the simplest case the adaptive algorithm uses a fixed finite window to estimate the distribution. In a more general case it would be necessary to change the speed of adaptation as well, so that the window size would also change over time.

6.1.1 Related work

Once we have established our objective we first examine related work in several areas. Note that our general formulation raises a number of different issues some of which have been tackled in other contexts.

6.1.1.1 Adaptive quantization

We first summarize some of the approaches proposed for adaptive quantization. As stated above we consider only schemes that do not require overhead information, i.e. we look at backward adaptation rather than forward adaptation. Bit allocation is a form of forward adaptive quantization which requires the input sequence, or a segment of it, to be known by the encoder, which then transmits the quantizer choice as overhead [39]. From that perspective, the schemes described in Chapters 2 and 4 are examples of forward adaptive quantization, which operate with fixed codebooks. Recent work [20, 28] looks at forward adaptive quantization and describes a procedure to optimally obtain the various codebooks from a training set of data.

We mention two different approaches to backward adaptive quantization. In [49, 25, 12] the objective is to adjust the support region of a scalar quantizer, so that this quantizer can be used in conjunction with a predictor in a DPCM system. The original idea of Jayant's adaptive quantizer [49, 41, 50] was to change the support region based on the previous quantized sample, while in [25] more than one sample of memory is used. Finally in [12] both the support region and the bin sizes can be adjusted, although the bin sizes are restricted to a finite set of values.

A somewhat different problem is tackled in [15] where an initial tree structured vector quantizer (TSVQ) is first designed, with a rate higher than the rate available for transmission. Then the adaptive algorithm chooses which subtree of the previously designed tree has to be used at every instant. Both encoder and decoder keep counts of the number of samples that corresponded to each of the nodes of the tree, and they select the subtree which minimizes the expected distortion (under the assumption that future samples will have the same distribution as past ones).

Note that all these systems use (implicitly or explicitly) simple models of the source to determine the changes in quantization. For instance [49] assumes that the sources are relatively smooth but have varying dynamic range so that the role of the adaptation is to estimate the changes in the variance of the source (so that the dynamic range of the quantizer is adapted) while a uniform quantizer is used. Similarly, the assumption in [15] is that the initially designed tree-structured codebook is sufficiently representative of the expected input signals, so that the adaptive algorithm can find a "good" sub-tree at any given time. In our work, the aim is to explicitly determine a model for the source from the data known to encoder and decoder, and then adapt the quantization scheme to get the best performance for the given model.

6.1.1.2 Adaptive lossless compression

The topic of adaptation has been extensively dealt with in the area of lossless data compression. The two main approaches to adaptive lossless compression are model–based (e.g. Arithmetic Coding (AC) or adaptive Huffman coding) and dictionary-based (e.g. Lempel Ziv (LZ) coding), where the adaptivity comes from dynamically updating, respectively, the model and the dictionary. We refer to [6] for an extensive survey of lossless compression techniques.

We will concentrate on the AC algorithm [55, 80, 116] as it is closer to some of the main ideas in our work. In the simpler case of a binary source, the encoder has to update the probabilities of the 0's and 1's. If the source is stationary and the model is correct then AC can provide a performance very close to the first order entropy. However, in real life environments, where sources need not be stationary, the performance of the algorithm is determined by how well it adapts to the changing statistics of the source. In that sense, the model tracking part of the AC algorithm plays an essential part in the system performance.

The problem of tracking the changing source statistics in the Q-coder, IBM's implementation of AC, is dealt with in [79]. The main problem is to find, for every newly arrived symbol, whether the occurrence is “normal”, i.e. consistent with the current model, or “not-normal”, i.e. unexpected within the current model. The proposed solution is to have different rates of change in the model so that the estimated probability of the most likely symbol will change slowly, while the estimated probability of the least likely symbol will change faster. The basic idea is that unlikely events (such as the occurrence of the least likely symbol) may signal a change in the distribution. These ideas also highlight the main trade-off in defining an adaptive coding algorithm. Because of the need to adapt to changing statistics, the scheme of [79] will perform worse than a static algorithm for an i.i.d. source. A similar trade-off can also be seen in the context of adaptive filtering, where in least mean square (LMS) type algorithms fast convergence comes at the price of noisy behavior if the source is stationary [83].

As another example of adaptation in the context of lossless coding, it has been shown that the Huffman coding tree can be modified “on the fly” so that the code would adapt to changing statistics, or learn them starting with no prior knowledge [35, 54, 111]. A first approach to generate these statistics would be to choose the number of samples N over which symbol occurrences are counted. However, a fully adaptive scheme would also require a procedure to change N, if necessary, during the coding process in order to improve the performance (we would thus have a parameter $N(n)$ as in Fig. 6-1(b)). Recent work [46] presents a solution to this last question, at the cost of some complexity, by proposing that the window size N be updated by choosing, among several possible sizes, the one producing a code with better compression.

Finally, it is worth noting that the question of what constitutes a good model for random data is a topic of interest not only for compression but also as a problem per se. Indeed, the minimum description length (MDL) technique introduced by Rissanen

[93, 94] provides the link between these two problems by establishing the asymptotic optimality of describing a distribution with a set of parameters that requires the least total number of bits to be encoded when counting *both* the bits needed to describe the model and the bits needed to encode the occurrences of the different symbols within the model. This criterion has been shown to also provide an asymptotically optimal universal code for data generated by a stationary source.

6.2 Adaptation algorithm

In this section we describe the various components of the adaptive quantization scheme as depicted in Fig. 6-1(b). For each of the elements we will formulate the objective and examine some solutions. The algorithm consists of three main blocks, a first one estimates the source probability density function (pdf), a second one updates the quantizer to maximize the performance for the given estimated distribution, and finally the third one decides what subset of the decoded sequence should be used to compute the model.

6.2.1 Estimation of input distribution

In this section we will study the problem of estimating the input source pdf, $f(x)$. Note that we refer to $f(x)$ as the pdf by abuse of language. We are computing an approximation to $f(x)$ based on a subset of the previously transmitted samples and assuming that those samples were generated according to some underlying model. In that sense our aim is to find a model that best "explains" the data under the sole assumption of being smooth, and obviously the model we obtain will depend on which set of data we are trying to model. Our goal is to:

Objective 6.1 *Given the N most recent quantized sample occurrences $\hat{x}(n-N)$, $\hat{x}(n-N+1)$, …, $\hat{x}(n-1)$, where N might be a constant or can be changed by* *the speed adaptation algorithm, find an estimate $\hat{f}(x)$ of the probability distribution function of the source, $f(x)$.*

We will use the following notation. The quantizer has L reconstruction levels r_i with $L-1$ decision levels denoted $b_1, \ldots, b_{L-1}$. Additionally, we have counted over the last N samples how many fell in each bin. These counts are denoted $n_0, \ldots, n_{L-1}$, where n_0 and n_{L-1} are the number of samples that fell in the "outer" bins. Our goal is to, given the knowledge of $n_0, \ldots, n_{L-1}$ and $b_1, \ldots, b_{L-1}$, find a good approximation $\hat{f}(x)$. From the observed data we can deduce that:

$$P_i \;=\; \int_{b_i}^{b_{i+1}} f(x)\,dx \;=\; \frac{n_i}{N}, \text{ for } i = 0, \ldots, L-1, \text{ and } b_0 = -\infty, b_L = +\infty. \quad (6.1)$$

Although strictly speaking the equality holds only in the limit as N goes to infinity, it is a sufficiently good approximation.

The task of determining $\hat{f}(x)$ is complicated by the fact that we are limiting ourselves to accessing only the quantized data. The problem can be separated into two parts: (i) estimating $\hat{f}(x)$ in the two outer bins, where we can only rely on knowing one of the boundaries (this is equivalent to estimating the dynamic range of the source), and (ii) estimating $\hat{f}(x)$ within the inner bins, where we know the boundaries.

In the more general case, since we have a finite amount of available data we can choose a set of P, $P \geq L$, points, $x_0, \ldots, x_{P-1}$ and our objective will be to find $\hat{f}(x_0), \ldots \hat{f}(x_{P-1})$, while $\hat{f}(x)$ can be linearly interpolated at other points x. The x_i can be chosen arbitrarily within the estimated dynamic range of the source, say $[b_0, b_L]$. The task of approximating the dynamic range will be dealt with in more detail in section 6.2.1.3.

Assume, thus, $[b_0, b_L]$ given and choose $P \geq L$ points which, for simplicity we assume equally spaced. Further assume that the pdf $f(x)$ that we are trying to approximate is smooth in some sense. Then we can aim at finding $\hat{f}$ such that

$$\int_{b_i}^{b_{i+1}} \hat{f}(x)\,dx \ = P_i, \quad \text{for} \quad i = 0, \ldots, L-1. \tag{6.2}$$

Since $\hat{f}$ is a piecewise linear approximation we can write the equations (6.2) as a function of the P unknowns $\hat{f}(x_0), \ldots \hat{f}(x_{P-1})$. This can be seen as a typical *inverse problem* which in the case of $P > L$ is overdetermined. Several techniques can be used to regularize such problems [81]. We outline a linear regularization method that has the advantage of resorting to the pdf smoothness assumption. An excellent description of these techniques can be found in [81].

Clearly, if $P > L$ there are many possible solutions which meet the constraints of (6.2). For a large enough P linearizing is a good approximation and we can thus write the constraints as

$$\sum_{k=0}^{P-1} \hat{f}(x_k) r_i(x_k)(x_{k+1} - x_{k-1})/2 = P_i, \quad \forall i = 0 \ldots L-1, \tag{6.3}$$

where $r_i(x_k) = 1$ if $x_k \in [b_i, b_{i+1}]$ and $r_i(x_k) = 0$ otherwise. Assuming for simplicity, equally spaced x_k and normalizing everything so that $x_{k+1} - x_k = 1$, we can write in matrix form

$$\mathbf{R} \cdot \mathbf{f} = \mathbf{p} \tag{6.4}$$

where $\mathbf{f}$ is the vector of the P unknown “knots” in the piecewise linear approximation, $\mathbf{p}$ is the vector of the L observed samples densities and $\mathbf{R} = \{r_{ik}\} = \{r_i(x_k)\}$ is the $P \times L$ matrix which determines what $\hat{f}(x_k)$ should be considered in each of the L constraints.

The basic idea of the linear regularization methods is to first relax the constraint of choosing $\hat{f}(x_k)$ to exactly match the observed frequencies of occurrence as in (6.4). We thus introduce a cost $\mathcal{M}$ which measures how much a solution deviates from a perfect fit

$$\mathcal{M} = |\mathbf{R} \cdot \mathbf{f} - \mathbf{p}|^2. \tag{6.5}$$

Additionally, we introduce a second cost $\mathcal{S}$ that will measure the "smoothness" of the resulting $\hat{f}(x)$. For instance, if we expect a priori $\hat{f}(x)$ not too deviate much from a linear function we can introduce a cost based on an approximation to the second derivative $\hat{f}''(x)$ so that

$$\mathcal{S} = \int (\hat{f}''(x))^2 \, dx \simeq \sum_{k=0}^{P-1} (-\hat{f}(x_k) + 2\hat{f}(x_{k+1}) - \hat{f}(x_{k+2}))^2. \tag{6.6}$$

Which can be then be also expressed in matrix form as

$$\mathcal{S} = \mathbf{f} \cdot \mathbf{B^T} \cdot \mathbf{B} \cdot \mathbf{f} \tag{6.7}$$

where $\mathbf{B} = \{b_{ik}\}$, with $b_{ik} = -1$ for $k = i$, $b_{ik} = 2$ for $k = i+1$, $b_{ik} = -1$ for $k = i+2$, and zero elsewhere, for $i = 1, \ldots, P-2$ and $k = 1, \ldots, P$.

Now, combining the two costs $\mathcal{M}$ and $\mathcal{S}$ and choosing a real positive number λ we can find $\hat{f}(x_k)$ to minimize:

$$\min(\mathcal{M} + \lambda\mathcal{S}) = \min(|\mathbf{R} \cdot \mathbf{f} - \mathbf{p}|^2 + \lambda(\mathbf{f} \cdot \mathbf{B^T} \cdot \mathbf{B} \cdot \mathbf{f})) \tag{6.8}$$

a least squares problem which can be solved using standard techniques. We refer to [81] for the details.

Thus we can find for each value of λ a set of points $\hat{f}(x_k)$ that yield an approximation to the pdf. There are two main advantages to using this technique:

1. It does not require an accurate estimation of the "outer boundaries". A good guess for the outer boundaries suffices, as the matching and smoothness criterion will guarantee smoothly decaying tails for the approximated distribution.

2. It provides an elegant way of including in the estimation any available prior knowledge about the smoothness of the pdf to be estimated.

However, there are also drawbacks in this approach, as for instance the relatively high number of points that are required, P, and, most importantly, the relevance of a good choice for the parameter λ which determines the relative weights to the smoothness and matching criteria. Potentially, an iterative procedure, where several λ's are tried until an appropriate solution is found, may be required. For these reasons we now propose a simpler approach that requires only $P = L$ points and involves no iterations.

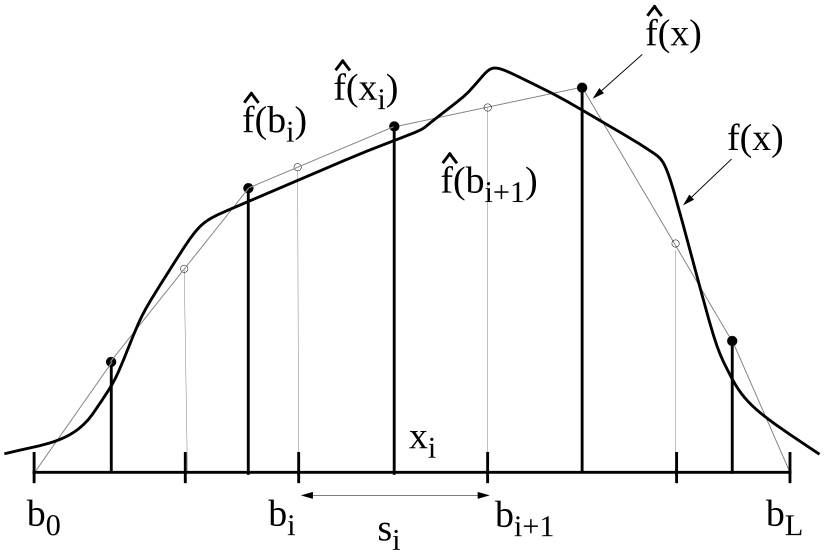


Figure 6-2: Notation used in the model estimation algorithm. The b_i's denote the decision levels, with b_0 and b_L denoting the outer boundaries of the finite support approximation. The x_i are the knots of the piecewise linear approximation. In this figure there are as many knots as bins but in general one can use more knots than bins. Note that we depicted an $f(x)$ which is non zero outside the range determined by b_0, b_L to emphasize the fact that these two boundaries have to be estimated and that the operation introduces some error.

6.2.1.1 A simple non-iterative approach

Assume again that we have chosen the boundaries b_0 and b_L, such that $\hat{f}(b_0) = \hat{f}(b_L) = 0$, as our estimate of the dynamic range (Refer to Fig. 6-2). Furthermore, assume that we estimate that our choice of b_0, b_L is expected to "leave out" a fraction of the tail of the distribution such that $\int_{-\infty}^{b_0} f(x)dx = \int_{b_L}^{+\infty} f(x)dx = P_{out}$ (the details will be explained in section 6.2.1.3). Then, denoting $P_0' = P_0 - P_{out}$ and $P_{L-1}' = P_{L-1} - P_{out}$ with $P_i' = P_i$ for $i = 1, \ldots, L-2$, we can choose $P = L$ points x_i at which we need to calculate the function values $\hat{f}(x_i) = f_i$ such that $\hat{f}$ will meet the constraint of (6.2). To restrict the number of degrees of freedom we arbitrarily choose the x_i to be center of each of the inner bins [2].

Now we can write the integrals over each bin of the piecewise linear approximation as

$$\hat{P}_i = \int_{b_i}^{b_{i+1}} \hat{f}(x)\,dx = \frac{1}{2}(\hat{f}(x_i) + \hat{f}(b_i))(x_i - b_i) + \frac{1}{2}(\hat{f}(x_i) + \hat{f}(b_{i+1}))(b_{i+1} - x_i) \quad (6.9)$$

where $\hat{f}(b_i)$ can be found by linear interpolation

$$\hat{f}(b_i) = (b_i - x_{i-1})\frac{\hat{f}(x_i) - \hat{f}(x_{i-1})}{(x_i - x_{i-1})} + \hat{f}(x_{i-1}) \quad (6.10)$$

and we have $\hat{f}(b_0) = 0$ and $\hat{f}(b_L) = 0$. Note that, since we have only one "knot" per bin, each of the equations (6.9) involves at most three unknowns $\hat{f}(x_{i-1}), \hat{f}(x_i), \hat{f}(x_{i+1})$ so that the system we have to solve is

$$\mathbf{T} \cdot \mathbf{f} = \mathbf{p}' \quad (6.11)$$

[2] Note that we arbitrarily choose the $x_i = b_i + (b_{i+1} - b_i)/2$ as the "fixed" points in our piecewise linear approximation. We do this for simplicity but one could also define the estimation problem as one where both f_i and x_i have to be chosen to satisfy the constraint. By allowing more degrees of freedom this method could provide smoother estimated distributions

where $\mathbf{T}$ is a $L \times L$ tridiagonal matrix and $\mathbf{p}'$ denotes the vector of observed probabilities (with the corrected tails). Efficient gaussian substitution methods can be used to solve this system [81].

6.2.1.2 The zero frequency problem

So far we have seen how to estimate the distribution from the available sample counts but a further question remains as to whether the counts can be used as is or some additional assumptions have to be made. In particular the zero frequency problem [6, 115] arises: if for a certain i (not one of the outer bins) we have $n_i = 0$ should we assume that the source has probability 0 of producing samples in that range? or, conversely, should we assume that the set of samples is not of significant enough size? We adopt the solution advocated, for instance, in probability estimation for arithmetic coders, and we add one count to all counters to avoid the problem [55, 80]. Using some such technique is particularly important when N is small and we have a rapidly varying source.

6.2.1.3 Estimation of the dynamic range

The remaining task is to determine the points b_0, b_L at which we estimate the pdf to be zero. Note that this problem is especially relevant in the simple method of Section 6.2.1.1. Indeed, while in the more general case, a sufficiently large number of interpolation points $P > L$ enables us to produce a model with smoothly decaying tails, in the scheme of Section 6.2.1.1 we are restricted to a $\hat{f}(x)$ with a linear decay tail. More precisely, in the general case as $P \gg L$, the tail in the outer bins can have several linear segments, thus achieving a smoothly decaying tail, while in Section 6.2.1.1 we are restricted to just one such segment. If we chose points b_0, b_L which overestimate the true dynamic range of the input source we may have cases where the result of solving (6.11) would yield negative values for $\hat{f}(x_0), \hat{f}(x_{L-1})$. We are thus interested in having good estimates of the dynamic range of the source. More formally, our objective is to:

Objective 6.2 *Find b_0 and b_L, defined as the points such that we estimate the source pdf to be "almost zero". For these points we will have by definition $\hat{f}(b_0) = 0$ and $\hat{f}(b_L) = 0$.*

The difficulty here stems from the fact that we have limited information: we know that n_0, resp. n_{L-1}, samples fell below b_1, resp. above b_{L-1}, but we need to use some of our assumptions to estimate b_0 and b_L. Obviously the main assumption is that the outer bins should contain the tails of the distribution. Based on the available information, i.e. the counts n_i, the current decision levels b_i, $i = 1, L-1$, and b_0^{old} and b_L^{old} the dynamic range estimates obtained in the previous iteration, we will consider three cases as follows (we outline the algorithm for adjusting b_0, but the same ideas apply for b_L):

1. if $n_0 = 0$, i.e. the outer bin is empty, we readjust the boundaries so that $b_0 = b_1$ (unless the adjacent bin is also empty), and we then "split" one of the inner bins (e.g. the one where we observed more samples), say i, and we assign $n_i/2$ samples to each of the newly formed bins. Thus we choose the new b_0 such that, at least based in our latest observation, we have $\hat{f}(b_0) = 0$.

2. if $n_0/(b_1 - b_0^{old}) > n_1/(b_2 - b_1)$ then clearly our current estimate is incorrect since we assume smoothly decaying tails for the distribution and we are observing more "sample density" in the outer bin. We have to expand the quantizer range and thus choose the new boundaries so that the two adjacent bins have the same sample density, thus we pick $b_0 = b_1 - (n_0/n_1)(b_2 - b_1)$.

3. the two previous cases occur when there is a large enough disparity between our current estimate and the "true" short term source distribution. When our estimate is sufficiently good that neither (1) nor (2) apply, we assume that the tail of the distribution is gaussian. The mean of the distribution is estimated by finding the average (weighted by the n_i) of the current reconstruction values. Under the assumption of a gaussian tail distribution and knowing the number of samples that fell on the two outer bins we can estimate the variance of the distribution. Finally we choose the outer boundary so that the tail beyond b_0 has a probability of less than some threshold P_{out} and thus the requirement that $\hat{f}(b_0) \simeq 0$ is met. Usually a threshold is needed since under the gaussian the input source distribution does not have finite support. Furthermore, this threshold will have to depend on the number of bits of the quantizer and should not be very small since, as mentioned earlier, overestimating the dynamic range of the source may result in non-admissible (i.e. negative) solutions for the $\hat{f}(x)$.

Note that cases (1) and (2) have to be dealt with separately since they represent cases where our previous estimates are incorrect and therefore would result in incorrect mean and variance estimates. Furthermore, it is clear that, since we carry a running memory, cases (1) and (2) would not occur if we updated the quantizer sufficiently often. For instance, we note how in [49, 41], where the dynamic range is estimated after each quantized sample is received, no such situations arise. In that sense (1) and (2) are safeguards to enable a less complex operation of the algorithm. It should also be pointed out that the estimation of the dynamic range is not that important when the more general approach is used. Finally, in some cases, as for instance image processing applications, the boundary estimation is not as critical since the source has inherently finite range.

6.2.2 Quantizer design for the estimated distribution

The ideas of the previous section have provided a way of computing an estimate of the source distribution. The objective of the next building block, see Fig 6-1(b), is to

Objective 6.3 *Redesign the quantizer for the given distribution $\hat{f}$. This can be done by using an optimal quantizer design algorithm which assumes $\hat{f}$ as the input distribution.*

As an example, we can design a constant rate quantizer simply using the Lloyd-Max algorithm for the given piecewise linear approximation. The task is to choose a new set of bin boundaries b_i', as well as the corresponding reconstruction levels r_i', such that the expected distortion for the distribution $\hat{f}(x)$ is minimized. Note that, as is the case with Huffman coding for example, one can guarantee optimality provided the model matches the source distribution. More formally, the algorithm fixes the outer boundaries b_0' and b_L' and then iterates the following two steps:

Algorithm 6.1 *Lloyd-Max algorithm:*

At iteration k:

Step 1: *Centroid condition (CC): choose the reconstruction levels to be the centroids of the current bins. For $i = 0, \ldots, L-1$*

$$r_i^k = \frac{1}{S_i^k} \int_{b_i^{k-1}}^{b_{i+1}^{k-1}} x\hat{f}(x)\,dx \quad \text{where} \quad S_i^k = \int_{b_i^{k-1}}^{b_{i+1}^{k-1}} \hat{f}(x)\,dx,$$

Step 2: *Nearest neighbor condition (NNC): use the nearest neighbor rule to find the new bin boundaries. For $i = 0, \ldots, L-2$:*

$$b_{i+1}^k = \frac{r_{i+1}^k - r_i^k}{2}$$

Step 3: *If some convergence criterion is met stop. Otherwise, $k \rightarrow k+1$, go to* **Step 1**.

Note that obviously b_0 and b_L are not changed in the optimization since these are not decision levels and are only used for the purpose of defining the model $\hat{f}(x)$. Because linear approximations to the distribution are used, determining the centroids can be done in closed form, at a low cost in complexity.

The same framework can be used with a variable rate entropy constrained design [22] where we keep step (1) as above but now have, for a given choice of a Lagrange multiplier λ, a positive real number:

Algorithm 6.2 *Entropy constrained quantization[22];*

At iteration k:

Step 1: *As in Lloyd-Max algorithm.*

Step 2: *compute the estimates of the entropy of each of the current bins, for $i = 0, \ldots, L-1$,*

$$H_i^k = -\log_2 \frac{1}{N} \int_{b_i^{k-1}}^{b_{i+1}^{k-1}} \hat{f}(x)\, dx.$$

Step 3: *select the boundaries b_{i+1}^k so that each value of x gets assigned to the bin that is optimal in the sense of minimizing the lagrangian cost. Thus given the reconstruction levels r_i^k, we have that $r(x)$, the reconstruction level assigned to point x, is*

$$r(x) = \arg \min_{r_i^k, i=0,\ldots,L-1} ((x - r_i^k)^2 + \lambda \cdot H_i^k).$$

Step 4: *If some convergence criterion is met stop. Otherwise, $k \rightarrow k+1$, go to* **Step 1**.

It is important to note that once we have estimated a model (i.e. chosen the $\hat{f}(x_i)$) the model is not modified by the algorithm that redesigns the quantizer. Furthermore, since our system keeps a running memory of the counts for each bin (the counters are not reset to zero after the quantizer has been redesigned) we also change the counters to adjust for the new bin sizes. Therefore, after the quantizer design stage, and calling b'_i and n'_i, respectively, the new bin boundaries and the updated estimated bin counts, we have that:

$$n'_i = N \cdot \int_{b'_i}^{b'_{i+1}} \hat{f}(x)\, dx. \tag{6.12}$$

6.2.3 Determining the speed of adaptation

The remaining block to be defined in the encoder of Fig. 6-1(b) is that in charge of determining the speed of adaptation. Our aim here is to:

Objective 6.4 *Dynamically determine at every iteration the number of past samples N that should be used in estimating the pdf.*

The classes of error produced by the choice of memory can be separated into two classes:

(a) Non-significant data: if not enough memory is used we may be dealing with a non-significant (in a statistical sense) set of data and our estimation will necessarily be erroneous.

(b) Sources with memory: if the source statistics (as determined by time averages over finite windows) change over time then an excess of memory will not permit sufficient adaptivity and will result in loss of performance.

Note that if we were quantizing a i.i.d. source with unknown statistics we could use a training mode operation [38] where the quantizer learns the statistics of the source during a certain period of time and afterwards the adaptivity is switched off. Similarly, one could operate the quantizer alternatively in training and stationary modes according to whether the current measured statistics agree with previously measured ones.

In our experiments we choose to keep two set of counters, one accumulating the long term statistics, the other accumulating the latest pattern of sample arrivals. We choose to use the short term data to estimate the model only if the difference between short and long term data exceeds a threshold. In this way, we try to detect the changes in statistics while avoiding always using a short term estimate, and thus risking having to deal with non-significant data.

6.3 Convergence of the adaptive quantizer

In this section we study in more detail the behavior of the adaptive algorithm. While in Section 6.4 we will deal with the quantization performance on real sequences of samples here we deal with the performance bounds in ideal situations as well as with the convergence and the asymptotic properties of the algorithm. The building blocks of the adaptive algorithm, as determined by their objectives (see Objectives 6.1, 6.3, and 6.4) each introduce errors by estimating the parameters of the source based in the quantized data, rather than on the actual samples.

We will concentrate on both the asymptotic and stationary performance of the algorithm. While the dynamic behavior is also of interest it is much more difficult to characterize since one would have to start by establishing some measure of variability for a source. We refer to the discussion on choosing the speed of adaptation (see Section 6.2.3).

Also we will not compare the errors introduced by the different approaches that were presented for approximating $f(x)$ with a piecewise linear function. Although we have set up a general framework for determining the approximation, given some smoothness criterion, we find the simpler approach of Section 6.2.1.1 to be sufficiently good for our purposes. All the results presented in this section and Section 6.4 were obtained using the approach of Section 6.2.1.1.

6.3.1 Asymptotic performance under fine quantization assumption

We briefly note, without a rigorous proof, that our algorithm is asymptotically optimal for an i.i.d. source, under the fine quantization assumption. In the asymptotic case we are interested on the behavior of the algorithm as (a) we gather statistics over an arbitrarily long time, i.e. N large, and (b) the number of quantization levels L becomes large.

The statistics gathered from the decoded data become arbitrarily close to the true statistics of the input pdf as the number of observed samples increases. By the law of large numbers, for a stationary input pdf, if N is the total number of observed samples and $n_i(N)$ is the number of samples that fell in bin i, we have that,

$$\lim_{N\to\infty} \frac{n_i(N)}{N} = \int_{b_{i-1}}^{b_i} f(x)dx,$$

where $f(x)$ is the source pdf and the b's are the bin boundaries. Therefore, if we wait a long enough time the variance of the estimated statistics can be made arbitrarily small.

Also when L increases the error we make in approximating a smooth input pdf with a piecewise linear function decreases. This is analogous to the arguments used in high resolution quantization analyses [39]. Therefore, for a sufficiently large number of quantizer levels the performance can be made arbitrarily close to that of the optimal quantizer for the given input pdf, and thus, quite obviously, the error due to generating the model based on the quantized levels can be minimized.

Therefore, as N and L increase our approximation gets arbitrarily close to the true pdf and thus the resulting quantizer is arbitrarily close to optimal.

6.3.2 Stationary performance

We now look at the behavior when the source input is again i.i.d. but we make no assumptions on L. This case is of interest as it provides for practical cases (i.e. with small number of bins) a measure of how close the adaptive quantizer is to the optimal performance. Since we noted that under stationarity the measured bin counts approach those dictated by the true distribution, we assume here that we know $f(x)$ and we thus measure solely the error due to using a piecewise linear approximation $\hat{f}(x)$, instead of the true pdf $f(x)$. Moreover, we also assume that the input pdf has known finite support so as to ignore the errors derived from the estimation of the dynamic range for a non-finite support distribution.

Note that the Lloyd iteration converges to a solution that meets the two optimality conditions, namely, the centroid condition (CC) and the nearest neighbor condition (NNC) [39], however it is not guaranteed to converge to a global optimal.

We will consider two main issues in the stationary behavior, (1) whether for the piecewise linear functions that we consider the Lloyd-Max iteration yields a global optimum and (2) the performance degradation due to using past quantized samples to obtain $\hat{f}(x)$. Section 6.4 will present examples of this behavior in actual sources. By resorting to the two following examples we seek to isolate the effect of the quantized data from the error in determining the dynamic range and the error due to estimating the P_i from a finite set of past data.

6.3.2.1 Example 1: Closed form solution

Consider first a simple example where we use a 3-bin quantizer with $f(x) = 3/4(1-x^2)$ for $-1 \leq x \leq 1$, $f(x) = 0$, elsewhere. We start by choosing an initial quantizer and for each of the initial conditions we compute the piecewise linear approximation. Since we assume a 3-bin quantizer and a finite-support, symmetric pdf, only one parameter, namely the boundary between inner and outer bin, is needed to fully specify the quantizer. Here solutions that meet both CC and NNC can be obtained in closed form. In Fig. 6-3 we plot the quantizer that meets these two conditions (i.e. the result of a Lloyd-Max iteration) for each of the possible initial conditions. A first comment to be made is that the initial conditions, i.e. the initial quantizer from which $\hat{f}(x)$ will be estimated, do matter. However, note that the dependency on the initial condition is small. Indeed all the solutions (vertical axis in Fig. 6-3) are fairly close to the optimal one for the true pdf $f(x)$. A second conclusion is that in this particular example there is a unique solution that meets the optimality conditions (Nearest neighbor and centroid) on $\hat{f}(x)$, and therefore the Lloyd-Max algorithm on the approximated pdf would also find the global optimum.

If we now use our adaptive algorithm successively on the known distribution, we can find out to what solution the algorithm would converge. Here we apply the adaptive algorithm for given initial conditions and we find a new quantizer using the results of Fig. 6-3, then we take the new quantizer as the initial condition and so on, until we converge. Indeed we observe (see Fig. 6-4) that after just one or two iterations the algorithm converges to a unique solution. We also note that the same final result is reached no matter what the initial conditions were.

6.3.2.2 Example 2: solution using Lloyd-Max with known pdf

Our second example examines the iterative solution that can be obtained using the adaptive algorithm on a known pdf. We no longer seek closed form solutions and thus we can consider quantizers with larger number of bins. Here we use a quadratic pdf (convolution of 3 uniform distributions) and we also note that the initial conditions determine the result. Refer Fig. 6-5. In the experiment we use initially a quantizer that is a linear combination of the Lloyd-Max quantizer for the true pdf and a uniform

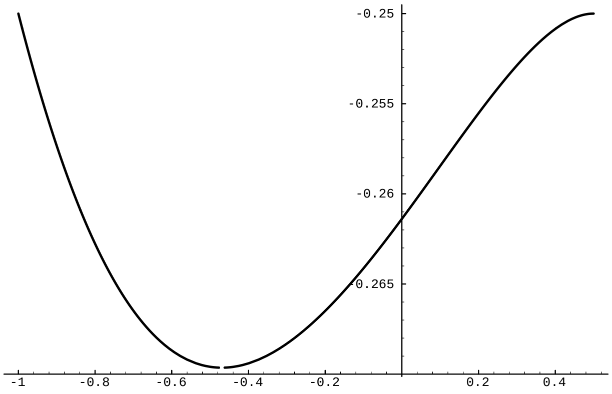


Figure 6-3: Example 1. 3-bin case and pdf $f(x) = 3/4(1-x^2)$ for $-1 \leq x \leq 1$, $f(x) = 0$, elsewhere. We represent the unique solution for the boundary (vertical axis) that meets the centroid and nearest neighbor conditions, for each of the initial conditions on the boundary (horizontal axis). Note that regardless of the initial conditions the range of solutions achieved is very restricted and close to the optimal solution obtained on the true pdf. As we have only three bins and the pdf is symmetric around zero, the quantizer is completely determined by a single parameter, the boundary of the middle bin.

quantizer. Note how the initial choice of quantizer is not too critical and, as should be expected, even less so in the case where a larger number of bins is used. Fig. 6-5 represents the result after using a single Lloyd-Max iteration on the approximated function. If we then use the new quantizer as the initial condition we will observe again that we have convergence.

Fig. 6-6 represents an example of the successive application of the algorithm (with a 4 level quantizer). Note that the first iteration (when the algorithm is started with a uniform quantizer as the initial condition) is already very close to the convergence value.

We can also measure the performance when using the iterative solution for different number of quantizer levels. Our results are shown in Fig 6-7. We observe that the iterative application of our algorithm converges to a unique solution. Furthermore

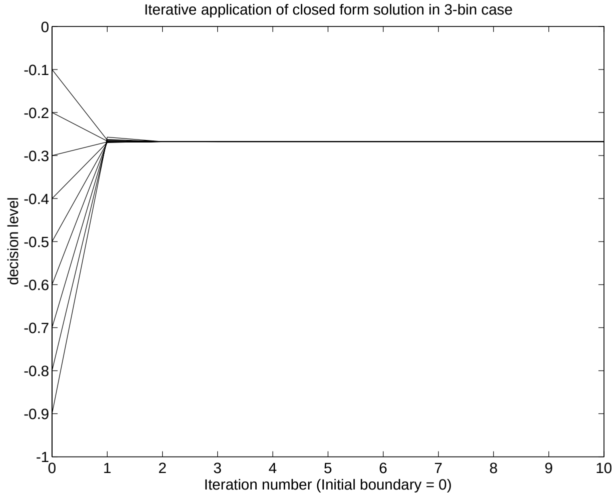


Figure 6-4: Example 1. Result of applying successively the algorithm to the three bin case with $f(x) = 3/4(1 - x^2)$. Note that convergence is very fast. Just two iterations are sufficient. The vertical axis represents the choice for the initial quantizer. The horizontal axis represents the number of iterations.

we see that the loss due to adaptivity is minimal, and diminishes as the number of levels increases. This figure shows the lower bound of the error due to adaptivity.

6.3.2.3 Discussion

So far we have seen in Example 1 that using the Lloyd-Max algorithm on a piecewise linear approximation can produce a global minimum. More generally, in order for the global minimum to be reached, a sufficient condition [33, 103] is to have a log-concave $\hat{f}(x)$. For instance a concave piecewise linear approximation can be shown to be log-concave and thus will yield a global minimum. Generally speaking, we should expect that if a solution exists for the underlying pdf $f(x)$ then a good approximation $\hat{f}(x)$ will also be well behaved.

In both examples considered we observed convergence of the Lloyd iteration. This observation was confirmed by the experiments performed on real sources. In the above examples the only source of error was due to using $\hat{f}(x)$ in the quantizer design, since

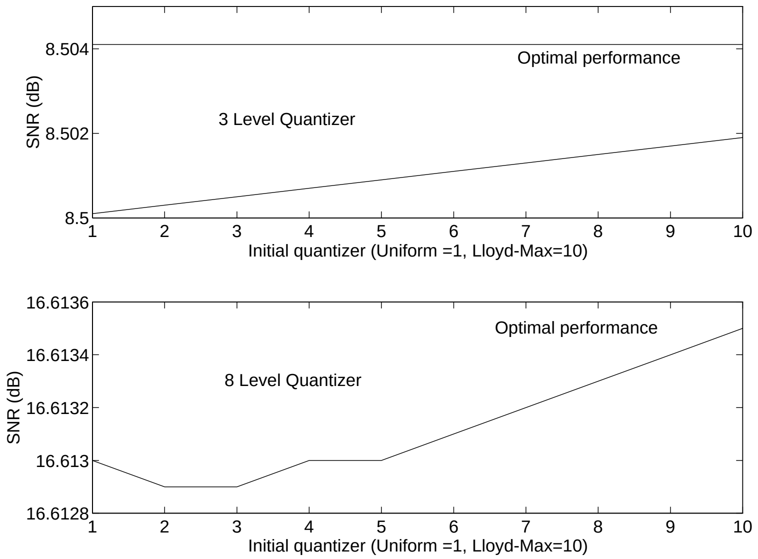


Figure 6-5: SNR obtained after running the Lloyd-Max iteration on the $\hat{f}(x)$ obtained with different starting conditions. The horizontal axis represents the different choices for the initial quantizer with 1 being the uniform quantizer and 10 the Lloyd-Max quantizer obtained on $f(x)$. The top line indicates the performance of the Lloyd-Max quantizer design for the true pdf. The top graph represents the 3-bin case, the bottom one the 8-bin case. Note how the degradation due to using the approximation is smaller, as expected, in the 8-bin case.

the P_i were computed directly from the known pdf $f(x)$. However, in the case of actual sources, the errors in estimating P_i and the boundaries would both prevent a perfect convergence. In other words, even for an i.i.d., the quantizer obtained in the process of successively estimating $\hat{f}(x)$ and running the Lloyd-Max algorithm will probably change slightly at each iteration, because the samples observed over the latest interval may give slightly different estimates for P_i.

To summarize, we have shown examples of how our algorithm performs when the effects of boundary and P_i estimation are not considered. We have observed that the procedure of successively matching the observed counts and re-designing the quantizer converged for pdf's similar to those that we would be expecting in real cases.

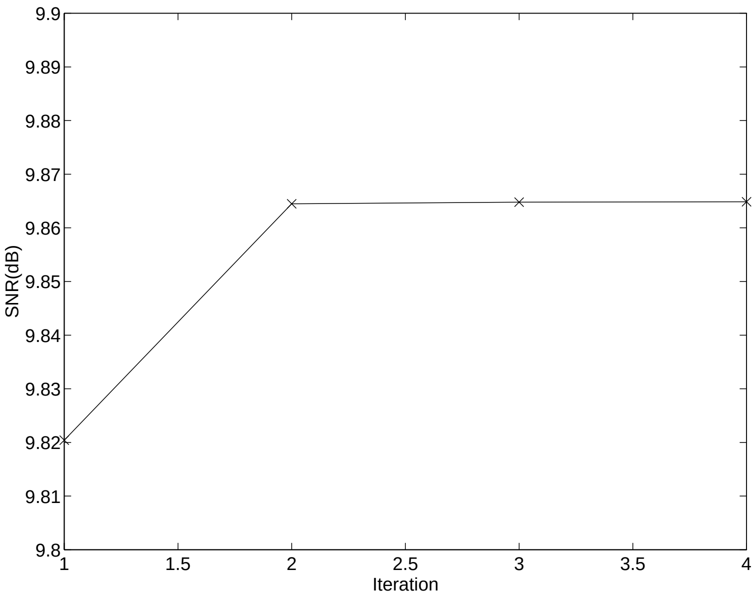


Figure 6-6: SNR obtained after running successive Lloyd-Max iterations where at each stage the true pdf is used to generate the counts that will produce $\hat{f}(x)$. A 2 bit quantizer is used. The resulting quantizer is used as the starting condition of the following iteration. Note that convergence is fast and that even after only one iteration (with a uniform quantizer as the initial condition) the SNR is very close to that attained at convergence.

6.4 Experimental results

In this section we present several examples to illustrate the performance of our adaptive quantization. We are concerned with the advantages of adaptivity in situations where the input pdf, as measured by the gathered statistics, changes over time. We will also show examples for i.i.d. sources where we should be experiencing some performance loss due to the adaptivity of the algorithm. Most examples are provided for fixed rate quantizers at a rate of two bits per sample. The examples with variable rate quantization indicate the achieved SNR vs. entropy trade-off. Note that we use the normalized SNR, $\log(\sigma_x^2/\sigma_r^2)$ where σ_x^2 and σ_r^2 are respectively the variance of the signal and that of the error. When we are dealing with non time-varying signals, we use averages over windows as our estimates for the variance.

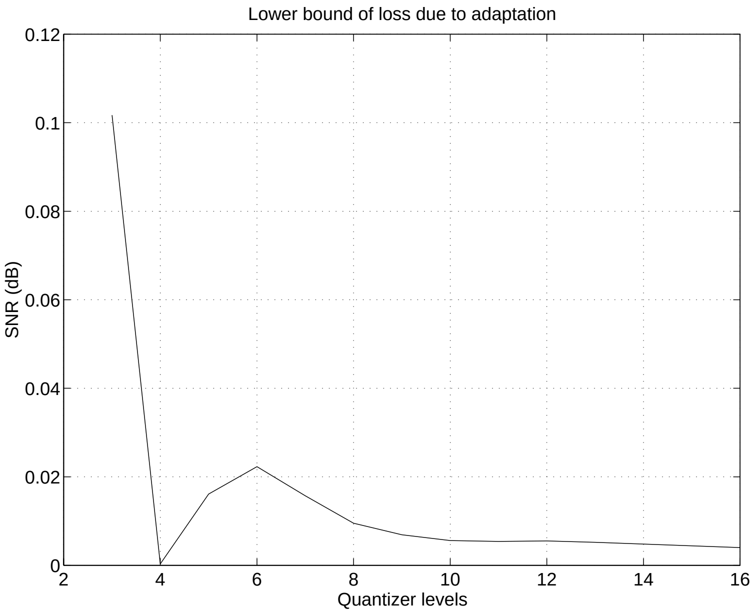
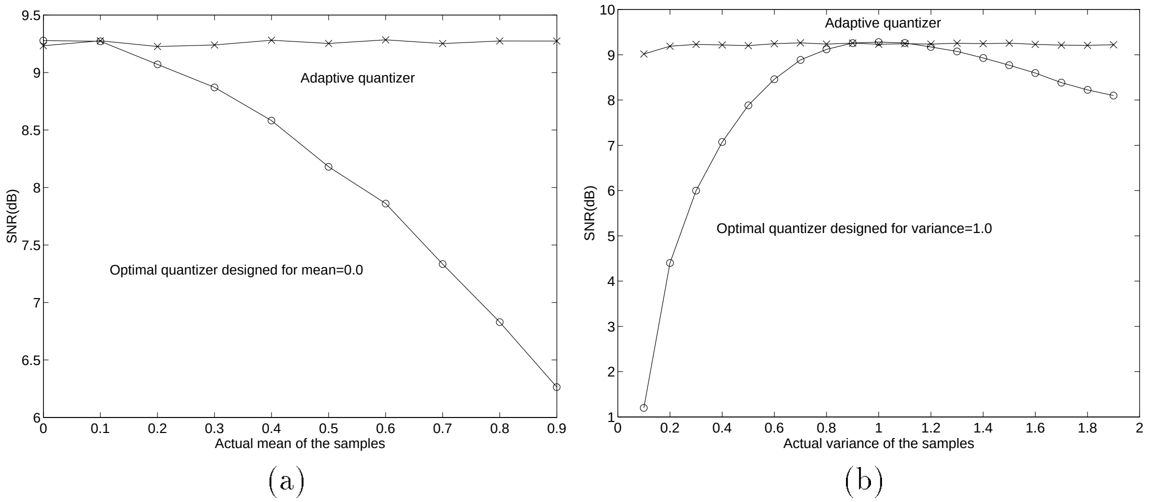


Figure 6-7: Loss in performance due to the adaptivity for several numbers of quantizer levels. The curve depicts the difference in SNR between the optimal Lloyd-Max quantizer and the quantizer obtained after iterating our algorithm. Note that the loss diminishes as the number of level increases. Also note that the decrease is not strictly monotonic. The error incurred between $f(x)$ and $\hat{f}(x)$ in the approximation is clearly strictly monotonic in the number of quantizers, but this may not be the case as far as the loss in performance is concerned.

6.4.1 Advantages of adaptivity

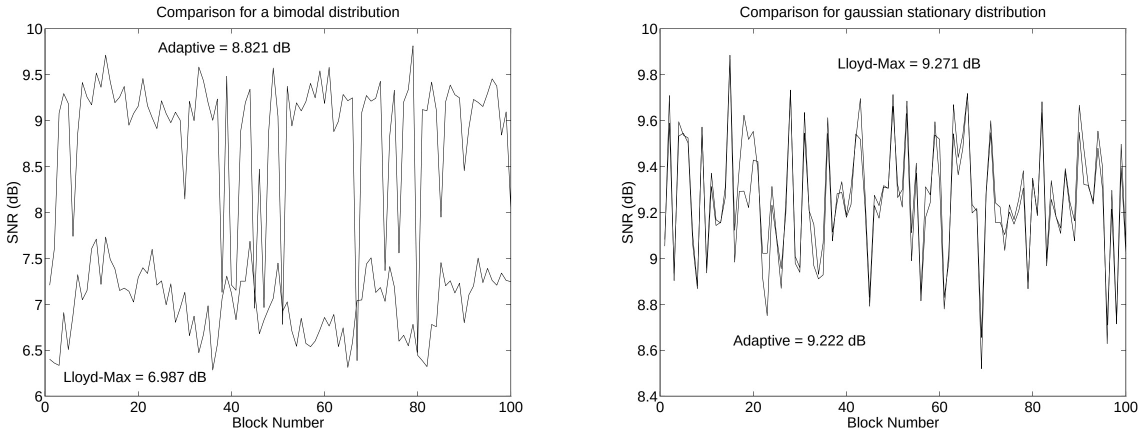
An adaptive algorithm can be useful even in the case of i.i.d. sources. In particular, adaptive schemes can learn the distribution "on the fly" (for instance, they could operate in "training mode" part of the time, typically at the beginning of the transmission). Furthermore, because they are not designed for a specific distribution they do not suffer the shortcoming of loss of performance in the face of mismatch between the actual source distribution and the one that was assumed in the design. Two examples of this can be seen in Figs. 6-8 (a) and (b), where the behavior of the adaptive algorithm and a Lloyd-Max quantizer are compared when the mean and variance of the source, respectively, do not match those assumed in the design.



(a) (b)

Figure 6-8: Comparison between adaptive and Lloyd-Max algorithms. (a) Mean mismatch. The Lloyd-Max quantizer is designed for a zero mean gaussian source. The adaptive algorithm maintains its performance constant. The variance of the source was 1. (b) Variance mismatch. The Lloyd-Max quantizer is designed for a variance 1 gaussian source. As the mismatch becomes significant the adaptive algorithm clearly out performs the Lloyd-Max quantizer.

A second advantage of using an adaptive algorithm is that it can outperform systems that are designed considering only long term statistics, by attempting to find short term trends in the data. As an example, Fig 6-9(a) shows the performances of the Lloyd-Max algorithm (trained on the sequence) and the adaptive algorithm for a bimodal source which randomly switches between two states each producing different mean and variance. When an i.i.d. source is considered though, the adaptive approach will be less effective although, as shown in Fig. 6-9(b) for a gaussian distribution, only marginally so. Note that the results we present were obtained using the adaptive algorithm with the *same* parameters for both types of sources (i.e. both the time between quantizer updates, and all the thresholds were fixed at the same level in both cases). Fig. 6-10 shows that the advantage of adaptivity can also be obtained within an entropy constrained variable rate quantization framework [22].



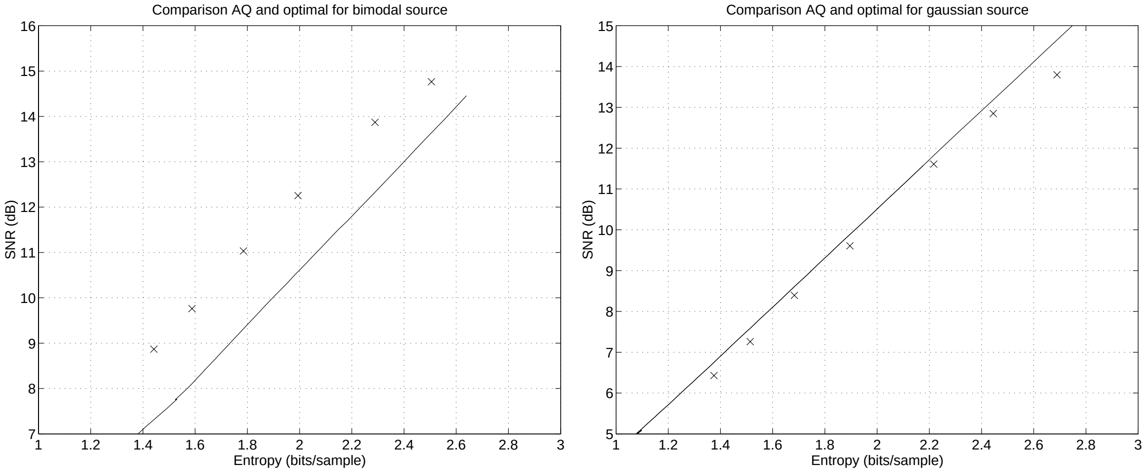
(a) (b)

Figure 6-9: Comparison of performance of Lloyd-Max and the adaptive algorithm. The SNR is the average measured over blocks of 2000 samples. (a) When a bimodal source is considered, the performance is much better than a Lloyd-Max design based on the complete sequence. The source switches between two states each producing different mean but same variance ($\sigma^2 = 1$). (b) When a stationary gaussian source ($\sigma^2 = 1$) is considered, the loss due to the adaptation is minimal.

6.4.2 Loss due to adaptivity

In this section we briefly discuss the performance of our adaptive algorithm for i.i.d. sources and show how the loss due to operating with estimates of the distribution, rather than the samples themselves as is the case in the Lloyd-Max design, is minimal. In our experiment, we use the adaptive algorithm but initialize it with the optimal Lloyd-Max quantizer trained on the source, rather than a uniform quantizer as is usually the case. In this way, since our first "guess" was optimal, the loss in performance is due exclusively to the adaptivity. Table 6.1 summarizes our results.

In Table 6.1 the recurrence time is the period between consecutive quantizer updates. The memory (measured in units of the recurrence times) represents the number of samples that are considered to generate the new quantizer. For instance a memory of 1.25 implies that the previous 50 samples are used when the recurrence time is 40, a memory of $+\infty$ means that all previous samples are considered at



(a) (b)

Figure 6-10: Comparison of performance of Lloyd-Max and the adaptive algorithm in the entropy constrained case. The average entropy of the quantizer is used. (a) When a bimodal source is considered, the performance is much better than a Lloyd-Max design based on the complete sequence. (b) When a stationary gaussian source ($\sigma^2 = 1$) is considered, the loss due to the adaptation is minimal.

every update. We note that, as the number of samples becomes small the main factor becomes the "non-significance" error, i.e. not enough information is used in updating the quantizers. This error can be overcome by appropriate choice of the speed of adaptation. Conversely, for long update intervals the main factor becomes the error introduced by the algorithm itself due to its manipulating quantized data, rather than the original samples as in the Lloyd-Max algorithm. This error can be seen to be very small.

6.5 Conclusions and future work

We have described an adaptive quantization algorithm which learns the source distribution from the quantized data and adapts the quantizer parameters using well known design methods. As an example we have demonstrated adaptive fixed rate and entropy constrained scalar quantizers which use a piecewise linear approximation

	Recurrence time T (samples)			
Memory (times T)	40	200	400	2000
1.25	8.824	9.157	9.220	9.259
1.67	8.903	9.210	9.241	9.264
2.5	9.109	9.240	9.257	9.266
5	9.154	9.260	9.265	9.267
$+\infty$	9.241	9.264	9.266	9.267

Table 6.1: Performance at different speeds of adaptation for a stationary source. Note that the adaptive algorithm was initialized with the optimal quantizer as designed by the Lloyd-Max algorithm on the actual data. The Lloyd-Max performance is 9.271 dB.

of the source distribution and rely on Algorithms 6.1 and 6.2 to update the quantizer parameters.

Future research will concentrate on extending these ideas to more general environments (e.g. VQ), and exploring its suitability for DPCM applications. Applications to image compression, e.g. in quantization of subbands, will also be considered. Further work is needed on the problem of estimating the boundaries and determining the speed of adaptation.

References

[1] JPEG technical specification: Revision (DRAFT), joint photographic experts group, ISO/IEC JTC1/SC2/WG8, CCITT SGVIII, August 1990.

[2] Special issue on packet speech and video. IEEE J. on Sel. Areas in Comm., Apr. 1991.

[3] Special issue on packet video. IEEE Trans. on Circ. and Sys. for Video Tech., Jun. 1993.

[4] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Trans. on Image Proc.*, 1(2):205–220, Apr. 1992.

[5] ATM Forum. *ATM User-Network Interface Specification, Version 3.0.* Prentice-Hall, 1993.

[6] T. C. Bell, J. G. Cleary, and I. H. Witten. *Text Compression.* Prentice Hall, Englewood Cliffs, NJ, 1990.

[7] R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

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

[9] D. P. Bertsekas. *Dynamic Programming.* Prentice-Hall, 1987.

[10] J.-C. Bolot and T. Turletti. A rate control mechanism for packet video in the internet. In *Proc. of Infocom'94*, pages 1216–1223, Toronto, Jun. 1994.

[11] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. on Commun.*, 31(4):532–540, April 1983.

[12] A. R. Calderbank, S. W. McLaughlin, and D. F. Lyons. A low complexity two-stage adaptive vector quantizer. In *Proc. of the 25th CISS*, pages 582–587, Baltimore, Md., Mar. 1991.

[13] K.M. Chandy, U. Herzog, and L. Woo. Parametric analysis of queueing networks. *IBM J. Res. Develop.*, pages 36–42, January 1975.

[14] R.F. Chang, W.T. Chen, and J.S. Wang. Image sequence coding using adaptive nonuniform tree-structured vector quantization. *Journal of Visual Comm. and Image Representation*, 2(2):166–176, Jun. 1991.

[15] R.F. Chang, W.T. Chen, and J.S. Wang. Image sequence coding using adaptive tree-structured vector quantisation with multipath searching. *IEE Proc. I*, 139(1):9–14, Feb. 1992.

[16] C.-T. Chen and A. Wong. A self-governing rate buffer control strategy for pseudoconstant bit rate video coding. *IEEE Trans. on Image Proc.*, 2(1):50–59, Jan. 1993.

[17] W.-T. Chen, R.-F. Chang, and J.-S. Wang. Image sequence coding using adaptive finite-state vector quantization. *IEEE Trans. on Circ. and Sys. for Video Tech.*, 2(1):15–24, Mar. 1992.

[18] T.-C. Chiueh and R. H. Katz. Multi-resolution video representation for parellel disk arrays. In *Proc. of ACM Multimedia Conf. '93*, Anaheim, CA, June 1993.

[19] J. Choi and D. Park. A stable feedback control of the buffer state using the controlled lagrange multiplier method. *IEEE Trans. on Image Proc.*, 1994. To appear.

[20] P. A. Chou, M. Effros, and R. M. Gray. A vector quantization approach to universal noiseless coding and quantization. *IEEE Trans. on Info. Th.*, 1994. Submitted.

[21] P. A. Chou, T. Lookabaugh, and R. M. Gray. Optimal pruning with applications to tree-structured source coding and modeling. *IEEE Trans. on Info. Th.*, IT-35:299–315, March 1986.

[22] P. A. Chou, T. Lookabaugh, and R. M. Gray. Entropy-constrained vector quantization. *IEEE Trans. on Signal Proc.*, 37(1):31–42, Jan. 1989.

[23] C. Constantinescu and J. A. Storer. On-line adaptive vector quantization with variable size codebook entries. In *Proc. of the Data Compression Conference*, pages 33–41, Mar. 1993.

[24] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

[25] S. Crisafulli and R. B. Bitmead. Adaptive quantization: solution via nonadaptive linear control. *IEEE Trans. on Comm.*, 41(5):741–748, May 1993.

[26] N. Dal Degan, R. Lancini, P. Migliorati, and S. Pozzi. Still images retrieval from a remote database: the system imagine. *Signal Processing: Image Communications*, 5(3):219–234, May 1993.

[27] P. Douglas and M. Vetterli. Statistical analysis of the output rate of two variable bit-rate video coders. In *Proc. of the 3rd Packet Video Workshop*, Morristown, NJ, March 1990.

[28] M. Effros, P. A. Chou, and R. M. Gray. Variable dimension weighted universal vector quantization and noiseless coding. In *Proc. of the Data Compression Conference, DCC'94*, pages 2–11, Snowbird, UT, Mar. 1994.

[29] A. Eleftheriadis, S. Pejhan, and D. Anastassiou. Algorithms and performance evaluation of the Xphone multimedia communication system. In *Proc. of the ACM Multimedia 93 Conf.*, pages 311–320, Anaheim, CA, Aug. 1993.

[30] H. Everett. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11:399–417, 1963.

[31] N. Farvardin and J. W. Modestino. On the overflow and underflow problems in buffer-instrumented variable-length coding of fixed-rate memoryless sources. *IEEE Trans. on Info. Th.*, IT-32:516–532, November 1986.

[32] M. L. Fisher. The Lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, January 1981.

[33] P. E. Fleischer. Sufficient conditions for achieving minimum distortion in a quantizer. *IEEE Int. Conv. Rec.*, pages 104–111, 1964.

[34] G. D. Forney. The Viterbi algorithm. *Proc. of the IEEE*, 61:268–278, March 1973.

[35] R. S. Gallager. Variations on a theme by Huffman. *IEEE Trans. on Info. Th.*, IT-24(6):668–674, Nov. 1978.

[36] M. W. Garrett. *Contributions Toward Real-Time Services on Packet Switched Networks.* PhD thesis, Dept. of Electrical Eng., Columbia Univ., 1993.

[37] M. W. Garrett and M. Vetterli. Joint source/channel coding of statistically multiplexed real time services on packet networks. *IEEE/ACM Trans. on Networking*, 1(1):71–80, February 1993.

[38] A. Gersho and D. J. Goodman. A training mode adaptive quantizer. *IEEE Trans. on Info. Th.*, IT-20(6):746–749, Nov. 1974.

[39] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, 1992.

[40] M. Goldberg and H. Sun. Frame adaptive vector quantization for image sequence coding. *IEEE Trans. on Comm.*, COM-33:629–635, May 1988.

[41] D. J. Goodman and A. Gersho. Theory of an adaptive quantizer. *IEEE Trans. on Comm.*, COM-22(8):1037–1045, Aug. 1974.

[42] D. Gross and C. Harris. *Fundamentals of Queueing Theory.* John Wiley, New York, 2nd edition, 1985.

[43] T. Hamada, M. Saito, and S. Matsumoto. Optimal quantization by viterbi algorithm for HDTV coding. In *Proc. 4th Intl. Workshop on HDTV*, Turin, September 1991. (Abstract).

[44] B. G. Haskell and R. L. Schmidt. A low bit-rate interframe coder for the videotelephone. *Bell Systems Tech. J.*, 54:1475–1495, October 1975.

[45] H. Heeke. A traffic-control algorithm for ATM networks. *IEEE Trans. on Circ. and Sys. for Video Tech.*, 3(3):182–189, Jun. 1993.

[46] H.-C. Huang and J.-L. Wu. Windowed huffman coding algorthim with size adaptation. *IEE Proc. I*, 140(2):109–113, Apr. 1993.

[47] J.-Y. Huang and P. M. Schultheiss. Block quantization of correlated gaussian random variables. *IEEE Trans. on Commun.*, 11:289–296, September 1963.

[48] N. Jayant, J. Johnston, and R. Safranek. Signal compression based on models of human perception. *Proc. of the IEEE*, Oct. 1993.

[49] N. S. Jayant. Adaptive quantization with a one-word memory. *Bell Sys. Tech. J.*, 52(7):1119–1144, Sept. 1973.

[50] N. S. Jayant and P. Noll. *Digital Coding of Waveforms: Principles and Applications to Speech and Video.* Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[51] G. Keesman, I. Shah, and R. Klein-Gunnewiek. Bit-rate control for MPEG encoders. *Signal Processing: Image Communication*, 1993. Submitted.

[52] Y. H. Kim and J. Modestino. Adaptive entropy-coded subband coding of images. *IEEE Trans. on Image Proc.*, 1(1):31–48, Jan. 1992.

[53] K. Knowlton. Progressive transmission of grey-scale and binary pictures by simple efficient and lossless encoding schemes. *Proceedings of the IEEE*, 68(7):885–896, July 1980.

[54] D. E. Knuth. Dynamic Huffman coding. *J. Algorithms*, (6):163–180, 1985.

[55] G. G. Langdon and J. Rissanen. Compression of black-white images with arithmetic coding. *IEEE Trans. on Comm.*, COM-29(6):858–867, Jun. 1981.

[56] J.-P. Leduc. Control of the image quality for digital TV and HDTV codecs. In *Proc. of EUSIPCO '92*, pages 1497–1500, Brussels, 1992.

[57] J.-P. Leduc and S. D'Agostino. Universal VBR videocodecs for ATM networks in the Belgian broadband experiment. *Image Communication*, 3:157–165, June 1991.

[58] D. LeGall. MPEG: a video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, April 1991.

[59] A. Lewis and G. Knowles. Image compression using the 2-D wavelet transform. *IEEE Trans. on Image Proc.*, 1(2):244–250, Apr. 1992.

[60] D. W. Lin, M.-H. Wang, and J.-J. Chen. Optimal delayed-coding of video sequences subject to a buffer-size constraint. In *Proc. of SPIE Visual Communications and Image Processing '93*, Cambridge, MA, Nov. 1993.

[61] M. Liou. Overview of the px64 kbit/s video coding standard. *Comm. of the ACM*, 34(4):59–63, Apr. 1991.

[62] M. Macedonia and D. Brutzman. MBone provides audio and video across the Internet. *Computer*, 27(4):30–36, Apr. 1994.

[63] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J.D. Robbins. Performance models of statistical multiplexing in packet video communications. *IEEE Trans. on Comm.*, 36:834–843, July 1988.

[64] M. Malak and J. Baker. An image database for low bandwidth communication links. In *Proc. of the Data Compression Conference*, Snowbird, Utah, March 1991.

[65] M. W. Marcellin and T. R. Fischer. Trellis coded quantization of memoryless and gauss-markov sources. *IEEE Trans. on Comm.*, 38(1):82–93, January 1990.

[66] M. Minoux. *Mathematical Programming: Theory and Algorithms.* Wiley, 1986.

[67] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization.* Wiley, 1988.

[68] A. N. Netravali and B. G. Haskell. *Digital Pictures: Representation and Compression.* Plenum Press, New York, 1988.

[69] M. T. Orchard and K. Ramchandran. An investigation of wavelet-based image coding using an entropy-constrained quantization framework. In J. A. Storer and M. Cohn, editors, *Proc. of the Data Compression Conf.*, pages 341–350, Mar. 1994.

[70] A. Ortega, M. W. Garrett, and M. Vetterli. Toward joint optimization of VBR video coding and packet network traffic control. In *Proc. of the 5th Packet Video Workshop*, Berlin, March 1993.

[71] A. Ortega, K. Ramchandran, and M. Vetterli. Optimal buffer-constrained source quantization and fast approximations. In *Proc. IEEE Intl. Symp. on Circ. and Sys. ISCAS '92*, pages 192–195, San Diego, CA., May 1992.

[72] A. Ortega, K. Ramchandran, and M. Vetterli. Optimal bit allocation for MPEG coders and applications to buffer control. In *Proc. Picture Coding Symposium '93*, Lausanne, Switzerland, Mar. 1993. Abstract.

[73] A. Ortega, K. Ramchandran, and M. Vetterli. Optimal trellis-based buffered compression and fast approximations. *IEEE Trans. on Image Proc.*, 3(1):26–40, Jan. 1994.

[74] A. Ortega, J. I. Ronda, N. García, and P. J. Zufiria. A model based approach to buffer occupancy control for TV and HDTV coders. In *Proc. 4th Intl. Workshop on HDTV*, Turin, September 1991.

[75] A. Ortega and M. Vetterli. Adaptive quantization without side information. In *Int'l Conf. on Image Proc., ICIP'94*, Austin, Texas, Nov. 1994. To appear.

[76] A. Ortega and M. Vetterli. Multiple leaky buckets for increased statistical multiplexing of ATM video. In *Proc. of the 6th Packet Video Workshop*, Portland, OR, Sep. 1994. To appear.

[77] A. Ortega, Z. Zhang, and M. Vetterli. A framework for optimization of a multiresolution remote image retrieval system. In *Infocom'94*, pages 672–679, Toronto, Canada, June 1994.

[78] A. Ortega, Z. Zhang, and M. Vetterli. Modeling and optimization of a multiresolution image remote retrieval system. In *IS&T/SPIE Symp. on Electronic Imaging Sci. & Tech.*, pages 185–196, San Jose, CA, February 1994.

[79] W. B. Pennebaker and J. L. Mitchell. Probability estimation for the Q-coder. *IBM J. Res. Develop.*, 32(6):737–752, Nov. 1988.

[80] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon, and R. B. Arps. An overview of the basic principles of the Q-coder adaptive binary arithmetic coder. *IBM J. Res. Develop.*, 32(6):717–726, Nov. 1988.

[81] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, 2nd edition, 1992.

[82] J. G. Proakis. *Digital Communications.* McGraw-Hill, 1989.

[83] J. G. Proakis, C. M. Rader, F. Ling, and C. L. Nikias. *Advanced digital signal processing.* Macmillan, New York, 1992.

[84] K. Ramchandran, A. Ortega, and M. Vetterli. Bit allocation for dependent quantization with applications to MPEG video coders. In *Proc. of ICASSP'93*, volume V, pages 381–384, Minneapolis, MN, Apr. 1993.

[85] K. Ramchandran, A. Ortega, and M. Vetterli. Efficient quantization for a multiresolution HDTV coder. In E. Dubois and L. Chiariglione, editors, *Signal Processing of HDTV, IV, Proc. of the Intl. Workshop on HDTV '92, Kawasaki, Japan, Nov. 1992*, pages 699–706. Elsevier, 1993.

[86] K. Ramchandran, A. Ortega, and M. Vetterli. Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders. *IEEE Trans. on Image Proc.*, 1994. To appear.

[87] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate–distortion sense. *IEEE Trans. on Image Proc.*, 2(2), April 1993.

[88] K. Ramchandran and M. Vetterli. Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility. *IEEE Trans. on Image Proc.*, 1994. To appear.

[89] K. Ramchandran and M. Vetterli. Syntax-constrained encoder optimization using adaptive quantization thresholding for JPEG/MPEG coders. In J. A. Storer and M. Cohn, editors, *Proc. of the Data Compression Conf.*, pages 146–155, Snowbird, Utah, March 1994.

[90] E. P. Rathgeb. Modeling and performance comparison of policing mechanisms for ATM networks. *IEEE J. on Sel. Areas in Comm.*, 9(3):325–334, April 1991.

[91] A. R. Reibman and B. G. Haskell. Constraints on variable bit-rate video for ATM networks. *IEEE Trans. on CAS for video tech.*, 2(4):361–372, December 1992.

[92] G. Rigolio, L. Verri, and L. Fratta. Source control and shaping in ATM networks. In *GLOBECOM'91, Phoenix*, 1991.

[93] J. Rissanen. Universal coding, information, prediction and estimation. *IEEE Trans. on Info. Th.*, IT-30(4):629–636, 1984 1984.

[94] J. Rissanen. *Stochastic complexity in statistical inquiry.* World scientific, Singapore, 1989.

[95] J. I. Ronda, A. Ortega, A. Fernández, and N. García. Predictive control of variable bit rate video coders. In *Proc. of the 6th European Signal Proc. Conf., EUSIPCO '92*, volume 3, pages 1369–1372, Brussels, August 1992.

[96] A. Sanz, C. Muñoz, and N. García. Approximation quality improvement techniques in progressive image transmission. *IEEE J. on Sel. Areas in Comm.*, SAC-2(2):359–373, March 1984.

[97] A. Segall. Bit allocation and encoding for vector sources. *IEEE Trans. on Inform. Theory*, IT-22:162–169, March 1976.

[98] P. Sen, B. Maglaris, N.E. Rikli, and D. Anastassiou. Models for packet switching of variable-bit-rate video sources. *IEEE J. on Sel. Areas in Comm.*, 7:865–869, June 1989.

[99] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. Journal*, 27:379–423, 1948.

[100] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. on Signal Proc.*, 41(12):3445–3462, Dec. 1993.

[101] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Trans. on Signal Proc.*, 36(9):1445–1453, September 1988.

[102] P. Skelly, M. Schwartz, and S. Dixit. A histogram-based model for video traffic behavior in an atm multiplexer. *IEEE/ACM Trans. on Networking*, 1(4):446–459, Aug. 1993.

[103] A. V. Trushkin. Sufficient conditions for uniqueness of a locally optimal quantizer for a class of convex error weighting functions. *IEEE Trans. on Info. Th.*, IT-82(2):187–198, 1982 1982.

[104] D. Tse, R. Gallager, and J. Tsitsiklis. Optimal buffer control for variable-rate lossy compression. In *Proc. of the 31st. Allerton Conf. on Comm. Control and Computing*, Oct. 1993.

[105] K.-H. Tzou. An intrafield DCT-based HDTV coding for ATM networks. *IEEE Trans. on Circ. and Sys. for Video Tech.*, 1(2):184–196, Jun. 1991.

[106] G. Ungerboeck. Channel coding with multilevel/phase signals. *IEEE Trans. on Info. Th.*, IT-28(1):55–67, January 1982.

[107] K. M. Uz, J. M. Shapiro, and M. Czigler. Optimal bit allocation in the presence of quantizer feedback. In *Proc. of ICASSP'93*, volume V, pages 385–388, Minneapolis, MN, Apr. 1993.

[108] K. M. Uz, M. Vetterli, and D. LeGall. Interpolative multiresolution coding of advanced television with compatible subchannels. *IEEE Trans. on CAS for Video Technology, Special Issue on Signal Processing for Advanced Television*, 1(1):86–99, March 1991.

[109] W. Verbiest, L. Pinnoo, and B. Voeten. The impact of the ATM concept on video coding. *IEEE J. on Sel. Areas in Comm.*, 6:1623–1632, December 1988.

[110] A. J. Viterbi and J. K. Omura. *Principles of Digital Communication and Coding*. McGraw-Hill, 1979.

[111] J. S. Vitter. Dynamic huffman coding. *ACM Trans. on Math. Software*, 15(2):158–167, 1989.

[112] G. K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, April 1991.

[113] P. H. Westerink, J. Biemond, and D. E. Boekee. An optimal bit allocation algorithm for sub-band coding. *Proc. of ICASSP*, pages 757–760, 1988.

[114] P. H. Westerink, D.E. Boekee, J. Biemond, and J.W. Woods. Subband coding of images using vector quantization. *IEEE Trans. on Comm.*, 36:713–719, 1989.

[115] I. H. Witten and T. C. Bell. The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. *IEEE Trans. on Info. Th.*, 37(4):1085–1094, Jul. 1991.

[116] I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, Jun. 1987.

[117] S.-W. Wu and A. Gersho. Rate-constrained optimal block-adaptive coding for digital tape recording of HDTV. *IEEE Trans. on Circuits and Sys. for Video Tech.*, 1(1):100–112, March 1991.

[118] Z. Xiong, N. P. Galatsanos, and M. T. Orchard. Marginal analysis prioritization for image compression based on hierarchical wavelet decomposition. In *Proc. of ICASSP '93*, pages 546–549, Minneapolis, MN, Apr. 1993.

[119] J. Zdepsky, D. Raychaudhuri, and K. Joseph. Statistically based buffer control policies for constant rate transmission of compressed digital video. *IEEE Trans. on Comm.*, 39(6):947–957, June 1991.

[120] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Trans. on Info. Th.*, IT-23(3):337–343, May 1977.