Adaptive Scalar Quantization Without Side Information

Antonio Ortega, *Member, IEEE,* and Martin Vetterli, *Fellow, IEEE*

Abstract— **In this paper, we introduce a novel technique for adaptive scalar quantization. Adaptivity is useful in applications, including image compression, where the statistics of the source are either not known *a priori* or will change over time. Our algorithm uses previously quantized samples to estimate the distribution of the source, and does not require that side information be sent in order to adapt to changing source statistics. Our quantization scheme is thus backward adaptive. We propose that an adaptive quantizer can be separated into two building blocks, namely, model estimation and quantizer design. The model estimation produces an estimate of the changing source probability density function, which is then used to redesign the quantizer using standard techniques. We introduce nonparametric estimation techniques that only assume smoothness of the input distribution. We discuss the various sources of error in our estimation and argue that, for a wide class of sources with a smooth probability density function (pdf), we provide a good approximation to a "universal" quantizer, with the approximation becoming better as the rate increases. We study the performance of our scheme and show how the loss due to adaptivity is minimal in typical scenarios. In particular, we provide examples and show how our technique can achieve signal-to-noise ratios (SNR's) within 0.05 dB of the optimal Lloyd–Max quantizer (LMQ) for a memoryless source, while achieving over 1.5 dB gain over a fixed quantizer for a bimodal source.**

I. Introduction and Related Work

ADAPTIVITY is a key feature in the most popular methods for lossless data compression, such as arithmetic coding (AC) [2]–[4], Lempel–Ziv coding [5], or dynamic Huffman coding [6]–[8] (see [9] for an extensive review of lossless compression).

The classical works on entropy coding (e.g., Huffman coding) and optimal quantizer design (e.g., Lloyd–Max quantization—LMQ), propose methods to achieve optimal performance for sources that can be fitted to a certain model. In adaptive AC implementations, adaptation is achieved by estimating this model "on the fly," with no prior assumptions. In this work, we propose that estimating the source model from past inputs should be an essential building block in adaptive scalar quantization as well. While the same could be argued for vector quantization, we concentrate here in the scalar case since going to higher dimensions complicates the modeling process. Note also that an adaptive quantizer can be used as part of a differential pulse code modulation (DPCM) loop and we are thus not restricting ourselves to memoryless sources. We can define our problem as that of adapting some or all of the parameters of a scalar 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 statistical characteristics of the source and, in particular, we allow it to have long-term dependencies and to 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 nonadaptive schemes. We study 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.

While adaptive source models have been used to varying extents in many adaptive quantization schemes, here we propose to explicitly separate the adaptation algorithm into two parts [see Fig. 1(a)], as follows:

1) *Source modeling*: Based on the previous N quantized samples, we model the source, for instance, by estimating its probability density function (pdf).
2) *Quantizer design*: For the given estimated statistics, new quantizer parameters are computed.

Separating the algorithm into these two building blocks, an approach similar to that used in adaptive AC, allows us to use well-known quantizer design techniques (such as the Lloyd–Max algorithm in the case of constant rate quantizers) and concentrate on the task of estimating the model. If we correctly estimate the distribution, then we are guaranteed optimality. The main contributions of this work are:

1) to make an explicit separation of modeling and quantizer design;
2) to study methods to model a source based on quantized samples;
3) to consider nonparametric, piecewise linear models for the pdf's.

Nonparametric models are selected (see Section II-A) so as to minimize the number of assumptions needed. We also study the amount of memory that should be used in estimating the

Manuscript received August 7, 1995; revised September 5, 1996. The work of A. Ortega was supported in part by the Fulbright Commission and the Ministry of Education of Spain and by the National Science Foundation under Grant MIP-9502227 (CAREER). Part of this work was performed while at the Department of Electrical Engineering and Center for Telecommunications Research, Columbia University. The work of M. Vetterli was supported in part by the National Science Foundation under Grant MIP-93-21302. The associate editor coordinating the review of this work and approving it for publication was Prof. Michael W. Marcellin.

A. Ortega is with the Department of Electrical Engineering–Systems, Integrated Media Systems Center, University of Southern California, Los Angeles, CA 90089 USA (e-mail: ortega@sipi.usc.edu).

M. Vetterli is with the Departement d'Electricité, Ecole Polytechnique Fédérale de Lausanne, EPFL, CH-1015 Lausanne, Switzerland, and the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA.

Publisher Item Identifier S 1057-7149(97)03080-7.

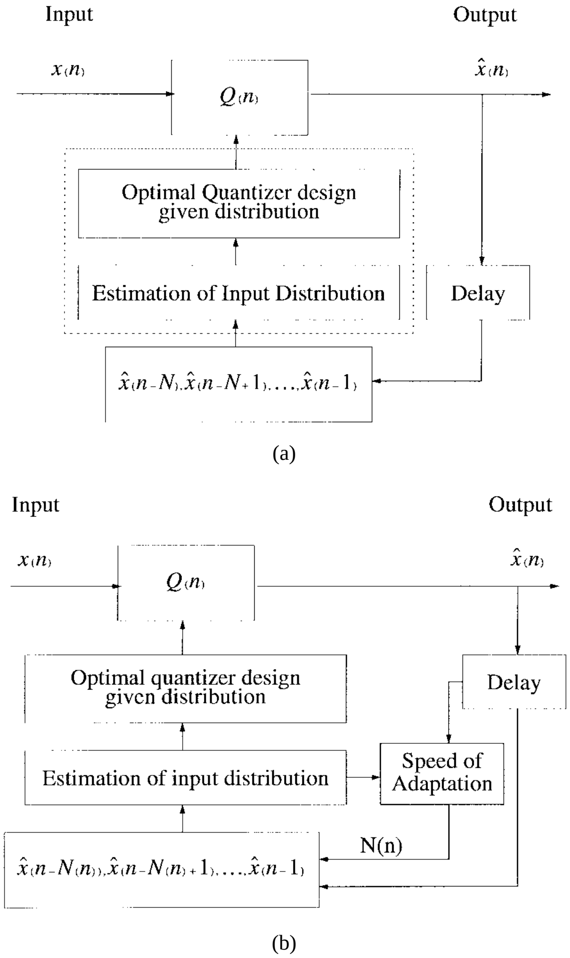

Input
Output
x(n)
Q(n)
x̂(n)
Optimal Quantizer design given distribution
Estimation of Input Distribution
Delay
x̂(n-N),x̂(n-N+1),...,x̂(n-1)
(a)
Input
Output
x(n)
Q(n)
x̂(n)
Optimal quantizer design given distribution
Delay
Estimation of input distribution
Speed of Adaptation
N(n)
x̂(n-N(n)),x̂(n-N(n)+1),...,x̂(n-1)
(b)

Fig. 1. Adaptive quantization algorithm. (a) The adaptation algorithm can be decomposed into 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.

distribution (i.e., N). Clearly, if the input source 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 size for a given type of source will result in suboptimal performance for other sources. We are thus interested in systems, such as that depicted in Fig. 1(b), where the window size or, equivalently, the speed of adaptation, can be changed over time.

Note that in our choices we are seeking to avoid making assumptions on the nature of the source to be quantized. However, there are two underlying assumptions that we expect to be met in order for our method to work. First, we assume that the source pdf will be smooth, e.g., that the pdf is continuous (higher regularity would be even better). Similarly, when it comes to the time variance of the sources, we assume either i.i.d. or independent samples with slow variation. We believe these two assumptions to be valid for most practical cases of interest.

We now briefly review some of the relevant prior work.

A. Adaptive Quantization

While some recent work has demonstrated adaptive vector quantization schemes that do require side information to specify modifications on existing codebooks [10]–[12], or to choose among a predetermined set of codebooks [13], here we focus on backward adaptive schemes, which do not require overhead information. Examples of backward adaptive quantization can be found in [14]–[16], where the objective is to adjust the support region of a uniform scalar quantizer so that this quantizer can be used in conjunction with a predictor in a DPCM system. Adaptation is based on one [14], [17], [18] or more [15] of the previously quantized samples. In [16], 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 [19] where an initial tree-structured vector quantizer (TSVQ) is designed off line with a rate higher than the rate available for transmission. At any given time a subtree of the original tree is used, namely, the one that minimizes the expected distortion (under the assumption that future samples will have the same distribution as, recent, past ones). Encoder and decoder keep counts of the number of samples that corresponded to each of the nodes and use this information to generate the distortion estimates.

Note that these systems use (implicitly or explicitly) simple models of the source to determine the changes in quantization. For instance, [14] assumes that the source distributions are relatively smooth (and a uniform quantizer is thus suitable) but have varying dynamic range so that the role of the adaptation is to estimate the changes in the variance of the source. Similarly, the assumption in [19] 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” subtree 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.

B. Adaptive Lossless Compression

The two main approaches to adaptive lossless compression [9] 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 will concentrate on the AC algorithm [2]–[4] as it is closer in spirit to our approach. In an AC scheme, the encoder keeps track of the probabilities of the input symbols. If the source is i.i.d. and the model is correct, then AC can achieve a rate very close to the source entropy. However, in real-life environments, the performance of the algorithm is determined by how well it adapts to the changing statistics of the source. In that sense, model tracking plays an essential part in the AC algorithm performance. The general idea [20] is to determine 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. Because of the need to adapt to changing statistics, this scheme will perform worse than a static algorithm for an i.i.d. source [20].

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 adapts to changing statistics, or learn them starting with no prior knowledge [6]–[8]. 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. 1(b)]. Recent work [21] presents a solution to this last question and proposes that the window size N be updated by choosing, among several possible sizes, the one producing a code with better compression for the last received symbols.

Finally, it is worth noting the connection between data modeling and data compression. Indeed, the minimum description length (MDL) principle introduced by Rissanen [22], [23] 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.

C. Outline of the Paper

The paper is organized as follows. In Section II, we describe the various components of the adaptive quantization scheme as depicted in Fig. 1(b). For each of the building blocks we define the objective and examine some solutions. In Section III, we analyze the convergence properties of the algorithm. We consider the asymptotic behavior as well as the various sources of error in the algorithm. Finally, in Section IV, we provide some experimental results to demonstrate the advantages and disadvantages of adaptive quantization over static approaches.

II. Adaptive Scalar Quantization Algorithm

In this section, we will describe the three building blocks of the adaptive quantizer, namely, source pdf estimation, quantizer adaptation, and adjustment of adaptation speed [see Fig. 1(b)].

A. Estimation of Input Distribution

1) General Case: Given a discrete time input $x(1), \cdots, x(n)$ we are seeking to find a model that best "explains" the data. To do so, we assume that the last N samples have been generated by an i.i.d. source, and we estimate the pdf of that source, $f(x)$, based on information available at both encoder and decoder, i.e., the previously quantized samples. We only assume that the underlying source has a smooth (e.g., continuous) pdf and that the source changes slowly, so that modeling the past N samples as if they had been generated by an i.i.d. source is reasonable. Our objective here is as follows.

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

We consider a quantizer with L reconstruction levels $r_0, \cdots, r_{L-1}$ and $L-1$ decision levels denoted $b_1, \cdots, b_{L-1}$. Additionally, denote n_i, for $i = 0, \cdots, L-1$, the number of samples, out of the N most recently transmitted, which were quantized to the ith bin. n_0 and n_{L-1} are the number of samples that fell in the "outer" bins and, obviously, $\sum_i n_i = N$. Our goal is, given the knowledge of $n_0, \cdots, n_{L-1}$ and $b_1, \cdots, b_{L-1}$, to find a good approximation $\hat{f}_n(x)$.[1] From the observed data we can deduce that

$$\begin{aligned} P_i &= \int_{b_i}^{b_{i+1}} f(x)\,dx \\ &\simeq \frac{n_i}{N}, \qquad \text{for } i = 0, \cdots, L-1 \\ &\qquad\qquad \text{and } b_0 = -\infty,\, b_L = +\infty. \end{aligned} \tag{1}$$

Although strictly speaking, the equality holds only in the limit as N goes to infinity, we use n_i/N to approximate P_i.

The problem we have formulated is in fact very general and is encountered, for instance, in statistics when fitting a model to some data. There are, however, two major differences between our work and standard model estimation problems: i) we estimate a model *based on quantized data*, and ii) we insist on *using a nonparametric approach*.

By restricting ourselves to the quantized data, we will operate with reduced information and thus our model estimation technique will perform worse than those described for standard model estimation problems. In particular, we may have to separate our model estimation problem into two parts: i) estimating $\hat{f}(x)$ in the two outer bins, where we can rely on knowing only 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 both bin boundaries.

A parametric approach would restrict the set of candidate models to a family of pdf's determined by a vector parameter θ. For example, one could try to find an estimate $\hat{f}(x|\theta)$ with $\theta = (m, \sigma)$ such that $\hat{f}$ is a Gaussian pdf with parameters m and σ. In our work we choose to use estimates $\hat{f}$ that are piecewise linear approximations to the underlying pdf. We are obviously estimating a finite set of parameters that uniquely describe $\hat{f}$ but still deem this approach nonparametric in that *we do not expect the input pdf to also be piecewise linear*. Rather, we choose a model (piecewise linear) that is expected to provide a reasonably good fit for a wide class of input pdf's (but which will not give a "perfect" approximation) instead of choosing a specific parametric source model (e.g., Gaussian).

2) A Nonparametric Piecewise Linear Approximation: In general, we can choose a set of P points, $x_0, \cdots, x_{P-1}$, $P \geq L$, and make it our objective to find $\hat{f}(x_0), \cdots \hat{f}(x_{P-1})$. $\hat{f}(x)$ can then 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 II-A5.

[1] To simplify the notation, in the remainder of the paper we do not use the subscript n in $\hat{f}_n(x)$, but it should be clear that different $\hat{f}(x)$'s are obtained every time the estimated input pdf is updated.

Assume, thus, $[b_0, b_L]$ given and choose $P \geq L$ points. Our goal is then, under the pdf smoothness assumption, to find $\hat{f}$ such that

$$\int_{b_i}^{b_{i+1}} \hat{f}(x)\,dx = P_i, \qquad \text{for } i = 0, \cdots, L-1 \tag{2}$$

where $P_i = n_i/N$. Since $\hat{f}$ is a piecewise linear approximation, we can write the equations in (2) as a function of the P unknowns $\hat{f}(x_0), \cdots, \hat{f}(x_{P-1})$. This can be seen as a typical *inverse problem*, which in the case of $P > L$ is underdetermined. Several techniques can be used to regularize such problems (see [24] for an excellent description of these techniques). We outline a linear regularization method that has the advantage of resorting to the pdf smoothness assumption.

Clearly, if $P > L$, there are many possible solutions that meet the constraints of (2). For a large enough P linearizing is a good approximation, and we can thus write the constraints as

$$\sum_{k=0}^{P-1} \left(\tfrac{1}{2}\right)[\hat{f}(x_{k+1}) + \hat{f}(x_k)] r_i(x_k)(x_{k+1} - x_k) = P_i \qquad \forall i = 0, \cdots, L-1 \tag{3}$$

where $r_i(x_k) = 1$ if $(x_{k+1} + x_k)/2 \in [b_i, b_{i+1}]$ and $r_i(x_k) = 0$, otherwise. Assuming, for simplicity, equally spaced x_k and normalizing so that $x_{k+1} - x_k = 1$, we can write in matrix form

$$\mathbf{R} \cdot \mathbf{f} = \mathbf{p} \tag{4}$$

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 probability masses, P_i, and $\mathbf{R} = \{r_{ik}\} = \{r_i(x_k)\}$ is the $P \times L$ matrix, which determines which $\hat{f}(x_k)$ 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 (4). We thus introduce a cost $\mathcal{M}$, which measures how much a solution deviates from a perfect fit

$$\mathcal{M} = |\mathbf{R} \cdot \mathbf{f} - \mathbf{p}|^2. \tag{5}$$

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

$$\begin{aligned}\mathcal{S} &= \int [\hat{f}''(x)]^2\,dx \\ &\simeq \sum_{k=0}^{P-1} [-\hat{f}(x_k) + 2\hat{f}(x_{k+1}) - \hat{f}(x_{k+2})]^2.\end{aligned} \tag{6}$$

This can be also expressed in matrix form as

$$\mathcal{S} = \mathbf{f}^{\mathbf{T}} \cdot \mathbf{B}^{\mathbf{T}} \cdot \mathbf{B} \cdot \mathbf{f} \tag{7}$$

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 $i = 1, \cdots, P - 2$ and $k = 1, \cdots, P$.

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

$$\min(\mathcal{M} + \lambda\mathcal{S}) = \min[|\mathbf{R} \cdot \mathbf{f} - \mathbf{p}|^2 + \lambda(\mathbf{f}^{\mathbf{T}} \cdot \mathbf{B}^{\mathbf{T}} \cdot \mathbf{B} \cdot \mathbf{f})]. \tag{8}$$

This is a least squares problem that can be solved using standard techniques (see [24] 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, as follows:

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 easy 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, such as the relatively large 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. Note that, for the appropriate choice of the smoothness parameter and sufficiently large P, this approach would in fact converge to the true distribution, while the technique we present next would still generate an approximation error. However this may not be a decisive consideration since, in general, the "right" smoothness parameter will not be known.

3) A Simple Noniterative Approach: Assume that the boundaries b_0 and b_L are chosen so that $\hat{f}(b_0) = \hat{f}(b_L) = 0$, as our estimate of the dynamic range (refer to Fig. 2). Furthermore, assume that we estimate that our choice of b_0, b_L is expected to "leave out" fractions of the tails 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 II-A5). Then, denoting $P'_0 = P_0 - P_{out}$ and $P'_{L-1} = P_{L-1} - P_{out}$ with $P'_i = P_i$ for $i = 1, \cdots, L - 2$, 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 (2) for P'_i. To restrict the number of degrees of freedom, we arbitrarily choose the x_i to be the center of each of the inner bins.[2]

[2] Note that we choose for simplicity the $x_i = (b_i + b_{i+1})/2$ as the "fixed" points in our piecewise linear approximation. Alternatively, we could have defined the estimation problem as one where both f_i and x_i have to be chosen to match the observed input and possibly some smoothness constraints, in a manner similar to the method in Section II-A2.

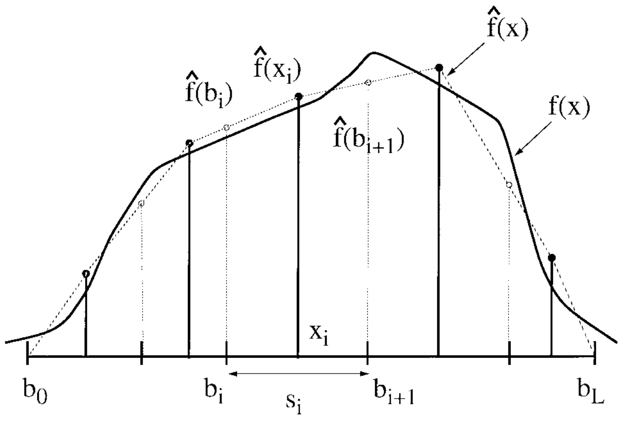


Fig. 2. Notation used in the model estimation algorithm. The b_i's denote the decision levels, with b_0 and b_L denoting the outer boundaries of the finite support approximation. The x_i's are the knots of the piecewise linear approximation. In this figure, there are as many knots as bins but in general one can use more knots than bins. Note that we depicted an $f(x)$, which is nonzero 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.

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

$$\begin{aligned}\hat{P}_i &= \int_{b_i}^{b_{i+1}} \hat{f}(x)\,dx \\ &= \tfrac{1}{2}\,[\hat{f}(x_i) + \hat{f}(b_i)](x_i - b_i) \\ &\quad + \tfrac{1}{2}\,[\hat{f}(x_i) + \hat{f}(b_{i+1})](b_{i+1} - x_i), \\ &\qquad i = 0, \cdots, L-1, \end{aligned} \tag{9}$$

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

$$\hat{f}(b_i) = (b_i - x_{i-1})\,\frac{\hat{f}(x_i) - \hat{f}(x_{i-1})}{x_i - x_{i-1}} + \hat{f}(x_{i-1}) \tag{10}$$

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 (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

$$\mathbf{T} \cdot \mathbf{f} = \mathbf{p}' \tag{11}$$

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 [24].

4) 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 [9], [25] 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 [2], [4]. Using some such technique is particularly important when N is small and we have a rapidly varying source.

5) 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 II-A3. Indeed, while in the 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 II-A3 we are restricted to a $\hat{f}(x)$ with linearly decaying tails. 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 II-A3 we are restricted to just one such segment. If we chose points b_0, b_L that overestimate the true dynamic range of the input source, we may have cases where the result of solving (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 here is as follows.

Objective 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 resort to 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 the most samples, $i = \arg\max_i n_i$), 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 greater "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 the current estimate and the "true" short-term source distribution. When the estimate is sufficiently good that neither 1) nor 2) apply, we assume that the tail of the density function is Gaussian and we determine b_0 so that $\int_{-\infty}^{b_0} f(x)\,dx \simeq P_{out}$, refer to [26] for the details.

Note that Cases 1) and 2) have to be dealt with separately, since they represent situations where our previous estimates are incorrect. However, 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 [14], [17], where the dynamic range is estimated after each quantized sample is received, no such situations arise. However, it may not be practical to recompute the model as frequently and thus 1) and 2) are needed to enable a less complex operation of the algorithm. It should also be pointed out that the estimation of the dynamic range is not as important when the more general approach is used. Finally, in some cases, for instance, in image processing applications, the boundary estimation is not as critical, since the source has inherently finite and known range.

B. 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. 1(b)], is as follows.

Objective 3: Redesign the quantizer for the given density function $\hat{f}$. This can be done by using an optimal quantizer design algorithm that assumes $\hat{f}$ as the input pdf.

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. The algorithm fixes the outer boundaries b'_0 and b'_L and then uses the standard Lloyd–Max design procedure [27] iteratively to find the new boundaries and reconstruction levels. The only difference is that $\hat{f}(x)$ is used to compute the optimal reconstruction levels, i.e., the centroids of the source distribution. Thus,

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

where

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

is the centroid of the ith bin, where k is the iteration number.

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 [28], where again the only modification to the standard algorithm is that we operate with our estimated pdf $\hat{f}(x)$. For example, the entropy estimates required by this algorithm can be found as

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

where H_i^k is the entropy estimate for the ith bin at the kth iteration.

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 quantizer design algorithm. 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

$$n'_i = \text{ROUND}\left[N \cdot \int_{b'_i}^{b'_{i+1}} \hat{f}(x)\,dx\right] \tag{12}$$

where n'_i is rounded to the nearest integer, and we ensure that $\sum n'_i = N$. In this manner, the bin counts are reset every time the model is recomputed. This allows us to maintain a running memory longer than the model update period, without taking into account the fact that the samples were quantized with various quantizers (typically, a different quantizer during each update period).

C. Determining the Speed of Adaptation

The remaining block to be defined in the encoder of Fig. 1(b) is that in charge of determining the speed of adaptation. Our aim here is to choose the "memory" of the algorithm, as in the following objective.

Objective 4: Dynamically determine every time the quantizer is updated the number of past samples N that should be used in estimating the pdf.

The errors produced by the choice of memory can be separated into the following two classes.

a) *Nonsignificant data*: If not enough memory is used (N small), we may be dealing with a nonsignificant (in a statistical sense) set of data and our estimation could 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 (N large) will not permit sufficient adaptivity and will result in loss of performance.

Note that if we were quantizing an i.i.d. source with unknown statistics, we could use a training mode operation [29] 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 keep two sets of counters, one accumulating the long-term statistics, the other accumulating the latest pattern of sample distribution. 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 nonsignificant data. Other ways of weighting the past are possible (and can be combined with the approach just described). For example, in our experiments we use exponential weights for the past counts in long term statistics by multiplying the current bin counts by a “forget factor” $0 < \alpha < 1$ before updating the counts.

III. Convergence of the Adaptive Quantizer

While in Section II we proposed using “optimal” quantizer design algorithms, it should be clear that these are guaranteed to be optimal only for the set of samples on which they were trained. Given that our approach entails training the quantizer on the fly based on past *quantized* samples, there is no reason to expect optimal behavior. In this section we study the sources of error intrinsic to our scheme and analyze its asymptotic behavior. As will be demonstrated by the examples of Section IV with real sequences of samples, the suboptimality incurred is very limited, although pdfs that would make the algorithm fail are also possible. We will concentrate on the case of a stationary memoryless input source. While the dynamic behavior (characterizing how the algorithm performs when the local statistics of the source change) is also of interest, it is more difficult to analyze. Such an analysis is left for future work.

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, we find the simpler approach of Section II-A3 to be sufficiently good for our purposes. All the results presented in this section and Section IV were obtained using the approach of Section II-A3.

A. Sources of Error

Our adaptive scalar quantization scheme introduces two sorts of errors which, in general, preclude that the optimal quantizer for the given source be achieved. These are:

1) *Modelization* or *approximation error*, which results from designing the new quantizer based on a certain model; thus optimal performance would not be achieved in general unless the input pdf exactly matches the model;
2) *Estimation error*, which results from building the model on the fly based on a finite memory of previous samples, rather than based on the complete set of samples as required to achieve optimality.

Ideally, if the approximation error is zero and the estimation error goes to zero as the sample size grows to infinity, we would have a *universal quantizer*. Otherwise, the quantizer will have a certain *redundancy*.

In general, the approximation error is not going to be zero. First, we might not have enough information to get an exact fit to the probability density function $f(x)$. Second, for computational reasons, we might choose an approximation to the true probability density function (e.g., a piecewise linear fit as in our algorithm). This inherent mismatch to the true probability density function will lead to a suboptimal quantizer design during the Lloyd–Max procedure.

The estimation error of the pdf’s integral over the bins will go down to zero, since the variance of the estimated probability of each bin goes to zero inversely proportionally to the number of sample points (see Section III-B).

Let us consider the approximation problem in more detail. First, assume that the probability density function is parametric with k parameters. Assuming that the bins cover the region of support of the random variable, then we need $k+ 1$ bins to specify the distribution, and usually this is sufficient, as pointed out in [30]. For example, a Gaussian random variable is specified by the probability measure on three intervals covering the real line. For more complex parametric distributions, the intervals might have to be chosen with some care, and finding the parameters from the probability measures could be involved. In these parametric cases, we are thus able to identify the pdf exactly from the quantized data, and can obtain a universal quantizer for the parametric family.

However, there are several reasons a nonparametric approximation such as that proposed here could be preferable. First, the intervals over which the probability is measured in our application correspond to the quantization bins. Therefore, we may have situations where the parametric model estimation problem is ill conditioned for the given quantization levels. Second, the goodness of our approximation to a certain arbitrary input source will hinge on how well such a source can be described by our particular parametric model. Also, complexity considerations may come into play, and we might settle for an approximation which can perform reasonably well (without achieving a perfect fit) for a large class of sources. The quality of this approximation will depend on the number of bins and the interpolation function used to approximate the probability density function.

Instead of a parametric family, we may only assume smoothness properties on the probability density function, which can then be used in fitting the data.[3]Note that to measure the “goodness of fit” we do not consider the quality of the approximation to the pdf, but rather the obtention a good LMQ. Indeed, work reported in [30] and [33] has shown that in many instances a good approximation to the LMQ can be found using our method, even when a relatively high error is made in matching the true pdf with a piecewise linear approximation. Quantifying the additional mean square error (MSE) due to an LMQ design based on an approximated pdf is beyond the scope of this work, but is an interesting question in its own right. Numerical evidence shown in Section IV indicates that a reasonable fit gives negligible errors in the cases investigated.

B. Asymptotic Performance Under Fine Quantization Assumption

Let us consider now the high rate case, that is, when the number of bins grows to infinity. Again assume that the pdf is smooth (e.g., continuous and differentiable). Then, it is clear

[3] Actually, if the probability density function belongs to a certain smoothness class, e.g., Lipschitz-1, there exist estimation procedures that allow an exact fit. However, these use a number of bins that increases with the number of samples, e.g., as $n^{1/3}$, and they would have to be modified so as to satisfy our constant rate requirement [31], [32].

that the approximation error will go to zero, and again, we will obtain a universal scheme (but only in the limit of infinite rate).

As for the estimation error, we note that our algorithm is asymptotically optimal for an i.i.d. source, under the fine quantization assumption. In the asymptotic case, we are interested in the behavior of the algorithm as i) we gather statistics over an arbitrarily long time, i.e., N large, and ii) 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 source as the number of observed samples increases. By the law of large numbers, for a stationary input source, 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_i's are the bin boundaries. Therefore, the variance of the estimated statistics can be made arbitrarily small for large enough N.

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 [27]. Thus, 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 made arbitrarily small.

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.

C. Approximation Error for Memoryless Sources

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 eliminate the effect of the estimation error and concentrate on the approximation 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 nonfinite 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) [27]; however, it is not guaranteed to converge to a globally optimal solution. Section IV will present examples of this behavior in actual sources. The following toy example seeks 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.

1) Example—Solution Using Lloyd–Max with Known pdf: Let us consider the iterative solution that can be obtained using the adaptive algorithm on a known pdf (i.e., no estimation error). In our example (Figs. 3–5) we use a quadratic pdf (convolution of three uniform distributions). We first note that the initial conditions affect the result of the iteration (refer to Fig. 3). In the experiment we initially use a quantizer that is a linear combination of the LMQ for the true pdf and a uniform 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. 3 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 again observe convergence.

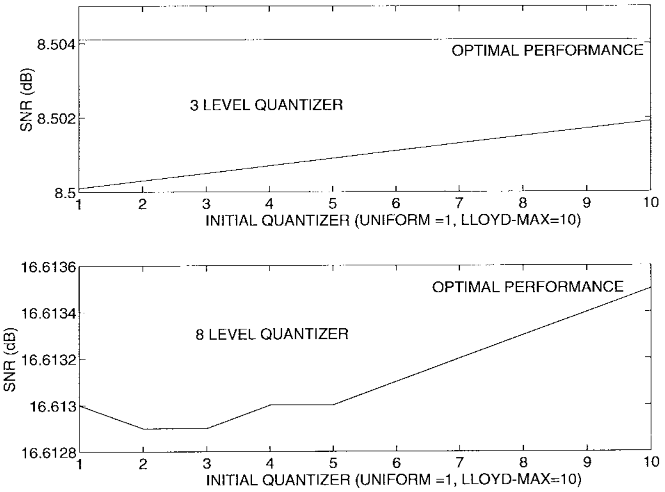


Fig. 3. SNR obtained after running the Lloyd–Max iteration on the $\hat{f}(x)$ obtained with different starting conditions for the quadratic pdf. The horizontal axis represents the different choices for the initial quantizer with 1 being the uniform quantizer and 10 the LMQ obtained on $f(x)$. The top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

Fig. 4 represents an example of the successive application of the algorithm (with a four-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. 5. We observe that the iterative application of our algorithm converges to a unique solution. Furthermore we see that the loss due to the approximation error is minimal, and diminishes as the number of levels increases. This figure shows the lower bound of the error due to adaptivity.

2) Discussion: Our previous example shows that the Lloyd–Max algorithm converges for the considered piecewise linear approximation. More generally, in order for the iteration to produce a global minimum, a sufficient condition [34], [35] 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

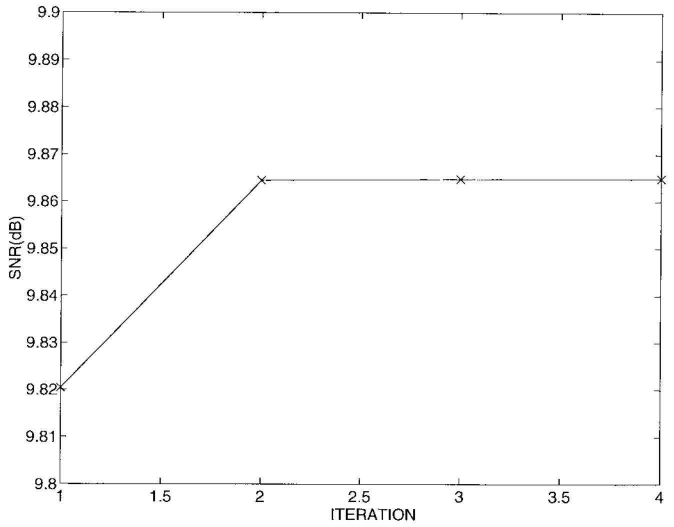


Fig. 4. SNR obtained after running successive Lloyd–Max iterations where, at each stage, the true pdf is used to generate the counts that will produce $\hat{f}(x)$. A 2-b quantizer is used. The resulting quantizer is used as the starting condition of the following iteration. Note that convergence is 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.

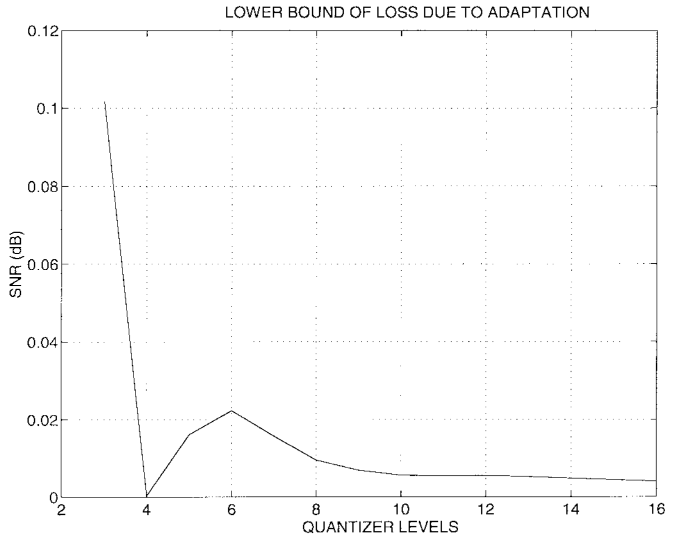


Fig. 5. Loss in performance due to the approximation error for several numbers of quantizer levels. The curve depicts the difference in SNR between the optimal LMQ and the quantizer obtained after iterating our algorithm. Note that the loss diminishes as the number of level increases. Also note that 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 quantizer levels, but this may not be the case as far as the loss in performance of the LMQ is concerned.

underlying pdf $f(x)$ then a good approximation $\hat{f}(x)$ will also be well behaved.

Convergence was also observed in the experiments performed on real sources. In the above example, we only suffered from the approximation error since the P_i were computed directly from the known pdf $f(x)$. However, in the case of actual sources, the errors in estimating both P_i and the bin boundaries would prevent a perfect convergence. In other words, even for an i.i.d. source, 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 due to the estimation error.

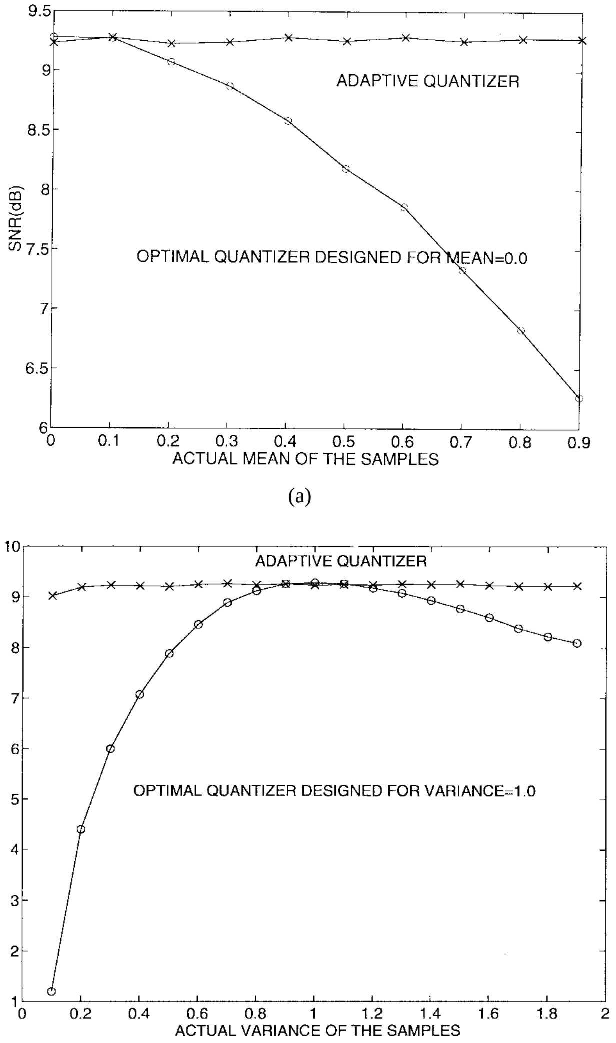
IV. Experimental Results

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 local 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 and to various sources of error described in the previous section. Most examples are provided for fixed rate quantizers at a rate of 2 b per sample. The examples with variable rate quantization indicate the achieved SNR versus 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 time-varying signals, we use averages over windows as our estimates for the variance. We use two types of sources in our experiments, memoryless Gaussian sources and "bimodal" sources, which are generated using a Markov chain (each of the two states of the Markov chain generates a memoryless Gaussian output).

A. 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. Examples can be seen in Fig. 6(a) and (b), where the behavior of the adaptive algorithm and an LMQ are compared when the mean and variance of the source, respectively, do not match those assumed in the design.

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. 7(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. 7(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. 8 shows that the advantage of adaptivity can also be obtained within an entropy-constrained variable-rate quantization framework [28].



(a)

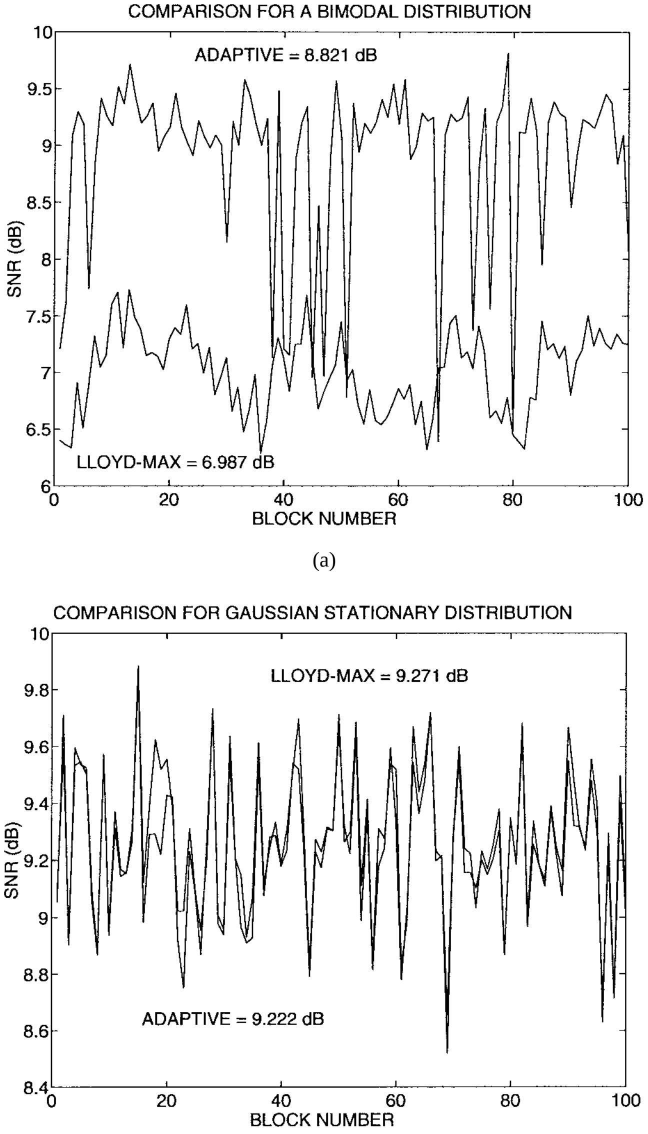
(b)

Fig. 6. Comparison between adaptive and Lloyd–Max algorithms. (a) Mean mismatch. The LMQ 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 LMQ is designed for a variance 1 Gaussian source. As the mismatch becomes significant, the adaptive algorithm clearly outperforms the LMQ.

B. 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 LMQ 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 I summarizes our results.

In Table I, 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, and a memory of $+\infty$ means that all previous samples are considered at every update. We note that, as the number of samples becomes small, the main factor is the "nonsignificance" 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 is 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.



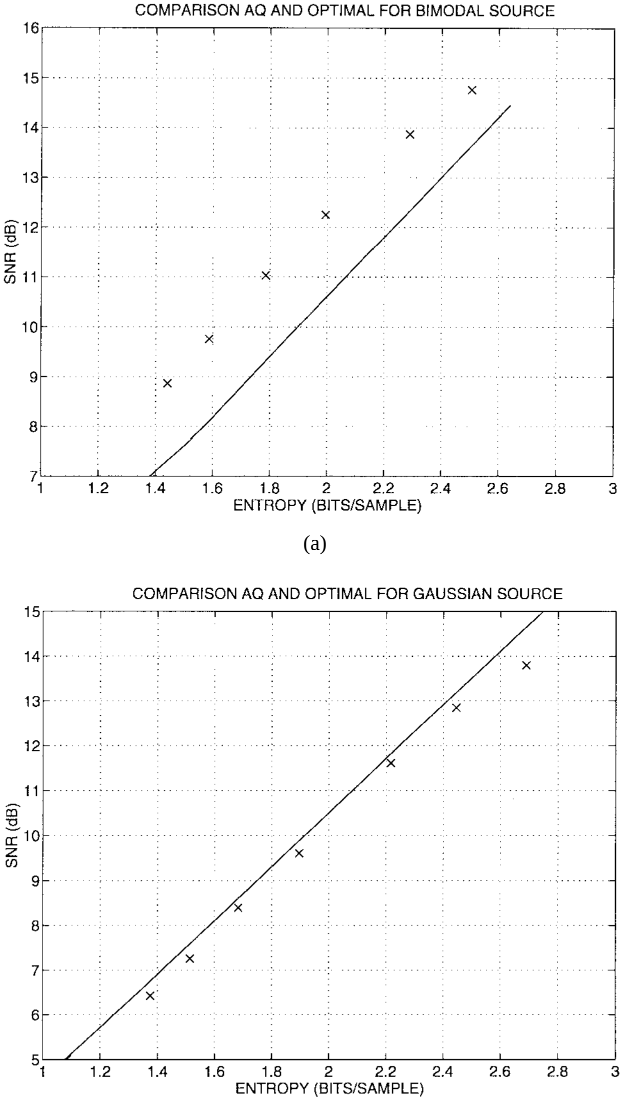
(a)

(b)

Fig. 7. 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.

V. Conclusions and Future Work

We have described an adaptive quantization algorithm, which learns the source distribution from the quantized data



(a)

(b)

Fig. 8. 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.

TABLE I
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. LMQ PERFORMANCE IS 9.271 dB

	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

and adapts the quantizer parameters using well-known design methods. As an example, we have demonstrated adaptive fixed-rate and entropy-constrained scalar quantizers that use a piecewise linear approximation of the estimated source distribution and rely on the well-known Lloyd–Max and entropy-constrained quantization design algorithms to update the quantizer parameters. Our backward adaptive algorithm can also be used for quantization schemes, such as trellis coded quantization (TCQ) [36] and scalar vector quantization (SVQ) [37], which rely on an underlying scalar quantizer [38]. Recent work has also extended the algorithm by allowing overhead information to be sent [33] and by considering piecewise polynomial approximations [39].

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, are being investigated [40]. Further work is needed on the problem of estimating the boundaries and determining the speed of adaptation.

ACKNOWLEDGMENT

The authors thank Dr. P. Chou of Xerox PARC for his detailed comments and suggestions on a first version of this manuscript, and Prof. B. Yu of the University of California at Berkeley for helpful discussions and for the preprints [30], [33].

REFERENCES

[1] A. Ortega and M. Vetterli, "Adaptive quantization without side information," in *Int. Conf. Image Processing, ICIP'94,* Austin, TX, vol. 3, pp. 856–860.

[2] G. G. Langdon and J. Rissanen, "Compression of black-white images with arithmetic coding," *IEEE Trans. Commun.,* vol. COMM-29, pp. 858–867, June 1981.

[3] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM,* vol. 30, pp. 520–540, June 1987.

[4] 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.,* vol. 32, pp. 717–726, Nov. 1988.

[5] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inform. Theory,* vol. IT-23, pp. 337–343, May 1977.

[6] R. S. Gallager, "Variations on a theme by Huffman," *IEEE Trans. Inform. Theory,* vol. IT-24, pp. 668–674, Nov. 1978.

[7] D. E. Knuth, "Dynamic Huffman coding," *J. Algorithms,* no. 6, pp. 163–180, 1985.

[8] J. S. Vitter, "Dynamic Huffman coding," *ACM Trans. Math. Softw.,* vol. 15, no. 2, pp. 158–167.

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

[10] Y. Steinberg and M. Gutman, "An algorithm for source coding subject to a fidelity criterion, based on string matching," *IEEE Trans. Inform. Theory,* vol. 39, pp. 877–886, May 1993.

[11] C. Chan and M. Vetterli, "Lossy compression of individual signals based on string matching and one pass codebook design," in *Proc. ICASSP'95,* Detroit, MI, pp. 2491–2494.

[12] Z. Zhang and V. K. Wei, "An on-line universal lossy data compression algorithm by continuous codebook refinement," in *Proc. IEEE Int. Symp. Information Theory, ISIT'94,* p. 262.

[13] M. Effros, P. A. Chou, and R. M. Gray, "Variable dimension weighted universal vector quantization and noiseless coding," in *Proc. Data Compression Conf., DCC'94,* Snowbird, UT, pp. 2–11.

[14] N. S. Jayant, "Adaptive quantization with a one-word memory," *Bell Syst. Tech. J.,* vol. 52, pp. 1119–1144, Sept. 1973.

[15] S. Crisafulli and R. B. Bitmead, "Adaptive quantization: Solution via nonadaptive linear control," *IEEE Trans. Commun.,* vol. 41, pp. 741–748, May 1993.

[16] A. R. Calderbank, S. W. McLaughlin, and D. F. Lyons, "A low complexity two-stage adaptive vector quantizer," in *Proc. 25th CISS,* Baltimore, MD, Mar. 1991, pp. 582–587.

[17] D. J. Goodman and A. Gersho, "Theory of an adaptive quantizer," *IEEE Trans. Commun.,* vol. COM-22, pp. 1037–1045, Aug. 1974.

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

[19] R. Chang, W. Chen, and J. Wang, "Image sequence coding using adaptive tree-structured vector quantization with multipath searching," *IEE Proc. I,* vol. 139, pp. 9–14, Feb. 1992.

[20] W. B. Pennebaker and J. L. Mitchell, "Probability estimation for the Q-coder," *IBM J. Res. Develop.,* vol. 32, pp. 737–752, Nov. 1988.

[21] H.-C. Huang and J.-L. Wu, "Windowed Huffman coding algorithm with size adaptation," *IEE Proc. I,* vol. 140, pp. 109–113, Apr. 1993.

[22] J. Rissanen, "Universal coding, information, prediction and estimation," *IEEE Trans. Inform. Theory,* vol. IT-30, pp. 629–636, July 1984.

[23] ——, *Stochastic Complexity in Statistical Inquiry.* Singapore: World Scientific, 1989.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C,* 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[25] I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Trans. Inform Theory,* vol. 37, pp. 1085–1094, July 1991.

[26] A. Ortega, "Optimization techniques for adaptive quantization of image and video under delay constraints," Ph.D. dissertation, Dept. Electrical Eng., Columbia Univ., New York, 1994.

[27] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Boston: Kluwer, 1992.

[28] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Signal Processing,* vol. 37, pp. 31–42, Jan. 1989.

[29] A. Gersho and D. J. Goodman, "A training mode adaptive quantizer," *IEEE Trans. Inform. Theory,* vol. IT-20, pp. 746–749, Nov. 1974.

[30] B. Yu, "A statistical analysis of adaptive quantization based on causal past," in *Proc. IEEE Int. Symp. Information Theory, ISIT'95*

[31] L. Devroye, *A Course in Density Estimation.* Boston, MA: Birkhauser, 1987.

[32] B. Yu, personal communication, Mar. 1995.

[33] ——, "A statistical analysis of adaptive scalar quantization based on quantized past," *IEEE Trans. Inform. Theory,* 1996, submitted for publication.

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

[35] A. V. Trushkin, "Sufficient conditions for uniqueness of a locally optimal quantizer for a class of convex error weighting functions," *IEEE Trans. Inform. Theory,* vol. IT-82, pp. 187–198, 1982.

[36] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss–Markov sources," *IEEE Trans. Commun.,* vol. 38, pp. 82–93, Jan. 1990.

[37] R. Laroia and N. Farvardin, "Trellis-based scalar-vector quantizer for memoryless sources," *IEEE Trans. Inform. Theory,* vol. 40, pp. 860–870, May 1994.

[38] Y. Yoo and A. Ortega, "Adaptive quantization without side information using scalar-vector quantization and trellis coded quantization," in *Proc. 29th Asilomar Conf. Signals, Systems, and Computers,* Pacific Grove, CA, Oct./Nov. 1995.

[39] P.-W. Wong, "Progressively adaptive scalar quantization," in *Proc. Int. Conf. Image Processing, ICIP'96,* Lausanne, Switzerland, vol. 2, pp. 357–360.

[40] Y. Yoo, A. Ortega, and B. Yu, "Adaptive quantization of image subbands with efficient over-head rate selection," in *Proc. Int. Conf. Image Processing, ICIP'96,* Lausanne, Switzerland, vol. 2, pp. 361–364.

Antonio Ortega (S'92–M'95) was born in Madrid, Spain, in 1965. He received the Telecommunications Engineering degree from the Universidad Politecnica de Madrid (UPM) in 1989 and the Ph.D. degree in electrical engineering from Columbia University, New York, NY in 1994.

He was a research assistant at the Image Processing Group at UPM (1990). At Columbia University, he was a Graduate Research Assistant at the Center for Telecommunications Research from 1991 to 1994, and was supported by a scholarship from the Fulbright commission and the Ministry of Education of Spain. Since September 1994, he has been an assistant professor in the Electrical Engineering–Systems Department at the University of Southern California (USC). Los Angeles. At USC he is also a member of the Integrated Media Systems Center, the NSF Engineering Research Center, and the Signal and Image Processing Institute. His research interests are in the areas of image and video compression and communications. They include topics such as joint source-channel coding for robust video transmission, rate control, and video transmission over packet wired or wireless networks.

Dr. Ortega received the NSF Faculty Early Career Development (CAREER) award in 1995.

Martin Vetterli (M'86–SM'90–F'95) received the Dipl. El.-Ing. degree from ETH Zürich (ETHZ), Switzerland in 1981, the M.S. degree from Stanford University, Stanford, CA, in 1982, and the Doctorat ès Science degree from EPF Lausanne (EPFL), Switzerland, in 1986.

He was a Research Assistant at Stanford University and EPFL, and has worked for Siemens and AT&T Bell Laboratories. In 1986, he joined Columbia University, New York, where he was last an Associate Professor of electrical engineering and co-director of the Image and Advanced Television Laboratory. In 1993, he joined the University of California, Berkeley, where he is a Professor in the Department of Electrical Engineering and Computer Sciences. Since 1995, he is a professor of Communication Systems at EPF Lausanne, Switzerland, and since 1996, he chairs the Communications Systems Division. His research interests include wavelets, multirate signal processing, computational complexity, signal processing for telecommunications, digital video processing and compression, and wireless video communications.

Dr. Vetterli is the Area Editor for Speech, Image, Video, and Signal Processing of the IEEE TRANSACTIONS ON COMMUNICATIONS. He is also on the editorial boards of *Signal Processing, Image Communication, Annals of Telecommunications, Applied and Computational Harmonic Analysis,* and *The Journal of Fourier Analysis and Applications*. He received the Best Paper Award of EURASIP in 1984 for his paper on multidimensional subband coding, the Research Prize of the Brown Bovery Corporation (Switzerland) in 1986 for his thesis, the IEEE Signal Processing Society's 1991 Senior Award (with D. LeGall), and the Swiss Latsis Prize in 1996. He was a plenary speaker at the 1992 IEEE ICASSP in San Francisco, and is co-author (with J. Kovacevic) of *Wavelets and Subband Coding* (Englewood Cliffs, NJ: Prentice-Hall, 1995). He is a member of SIAM.