

Reduced-Complexity Deterministic Annealing for Vector Quantizer Design

Kemal Demirciler

*Institute of Advanced Technology Research and Development, Eastern Mediterranean University,
6 Bozova Sokak, Yenisehir, Lefkosa, Mersin 10, Turkey
Email: kemaldemirciler@yahoo.com*

Antonio Ortega

*Integrated Media Systems Center, Department of Electrical Engineering, University of Southern California,
3740 McClintock Avenue, Los Angeles, CA 90089-2564, USA
Email: ortega@sipi.usc.edu*

Received 16 September 2004; Recommended for Publication by John Sorensen

This paper presents a reduced-complexity deterministic annealing (DA) approach for vector quantizer (VQ) design by using soft information processing with simplified assignment measures. Low-complexity distributions are designed to mimic the Gibbs distribution, where the latter is the optimal distribution used in the standard DA method. These low-complexity distributions are simple enough to facilitate fast computation, but at the same time they can closely approximate the Gibbs distribution to result in near-optimal performance. We have also derived the theoretical performance loss at a given system entropy due to using the simple soft measures instead of the optimal Gibbs measure. We use the derived result to obtain optimal annealing schedules for the simple soft measures that approximate the annealing schedule for the optimal Gibbs distribution. The proposed reduced-complexity DA algorithms have significantly improved the quality of the final codebooks compared to the generalized Lloyd algorithm and standard stochastic relaxation techniques, both with and without the pairwise nearest neighbor (PNN) codebook initialization. The proposed algorithms are able to evade the local minima and the results show that they are not sensitive to the choice of the initial codebook. Compared to the standard DA approach, the reduced-complexity DA algorithms can operate over 100 times faster with negligible performance difference. For example, for the design of a 16-dimensional vector quantizer having a rate of 0.4375 bit/sample for Gaussian source, the standard DA algorithm achieved 3.60 dB performance in 16 483 CPU seconds, whereas the reduced-complexity DA algorithm achieved the same performance in 136 CPU seconds. Other than VQ design, the DA techniques are applicable to problems such as classification, clustering, and resource allocation.

Keywords and phrases: deterministic annealing, complexity reduction, vector quantization, stochastic relaxation, Gibbs distribution, codebook initialization.

1. INTRODUCTION

Vector quantization is a source coding technique that approximates blocks (or vectors) of input data by one of a finite number of prestored vectors in a codebook. The challenge is to find the set of vectors (or quantization levels) such that a given criterion for the total distortion between the actual source and the quantized source is as small as possible under a constraint on the overall rate [1]. Since distortion depends on the codebook design, vector quantizer design is a key optimization problem to determine the performance of a VQ-based system [2, 3, 4, 5].

The traditionally used VQ design approach is the generalized Lloyd algorithm (GLA), also referred to as the LBG algorithm [6]. The GLA is an iterative descent algorithm, which converges to a final codebook relatively quickly, but such that

the resulting codebook is only locally optimal. This is because the algorithm gets trapped in a local minimum of the distortion (energy) surface to which the initial codebook is closest. Consequently, the performance of GLA can be poor compared to that of a globally optimal quantizer.

A powerful approach to reduce the sensitivity of the algorithm to the initial codebook is the introduction of *randomness*. Several randomized optimization techniques have been investigated in the past. In [7] such “random search” techniques are discussed, where the idea is to randomly perturb the system at each iteration and determine the resulting change in performance. In some of its variations, a perturbation is only accepted if the performance increases, otherwise it is rejected; and in other variations, perturbations that decrease performance are also accepted under certain conditions. In general, if a random search technique allows

temporary decreases in an objective function with nonzero probability, then the algorithm is in the class of *stochastic relaxation* (SR) [8, 9], or stochastic local search techniques.

An important SR technique is simulated annealing (SA) [9], where in each iteration a new codebook is generated in the neighborhood of the old one, and the new codebook is accepted or rejected according to the Metropolis algorithm [10]. If sufficient computational resources are devoted, the SA algorithm is guaranteed to yield globally optimal solutions [11]. A *reduced-complexity* quantizer design based on SR is proposed in [9], achieving similar or slightly better results than SA in much less time under similar iteration schedules. Basically, the reduced-complexity SR algorithm is the generalized Lloyd algorithm together with a stochastic perturbation step that could either be on the encoder (SR-C) or the decoder (SR-D). Another method that uses a similar randomized search technique is suggested in [12] and has an average performance comparable to SR-D but with higher complexity. In the above approaches, random search moves were allowed on the energy surface in order to give the system the ability to avoid local minima.

Unlike these SR techniques, a deterministic annealing (DA) approach for optimal vector quantizer design puts the problem in a probabilistic framework, and deterministically optimizes the probabilistic objective function in each iteration [13]. In DA there are no random moves on the energy (cost) surface. At high temperatures, the energy surface is smoothed, so that the algorithm starts at the global minimum on the smoothed energy surface. Through a careful annealing schedule, the algorithm traces the global minimum as the energy surface assumes its nonconvex “rugged” form with decreasing temperature. The Gibbs distribution is used to associate sample vectors in the training set with codevectors since it maximizes the entropy under the constraint of a given average distortion. Note that the sample vector-codevector associations are not one-to-one, but rather one-to-many. In other words, each sample vector in the training set is assigned to *all* codevectors with a given probability: a sample vector gets assigned to each sample vector with a probability that increases as the distance to the sample vector decreases.

The DA method can construct high-performance vector quantizers by avoiding local minima. The main obstacle to its widespread utilization is the very significant computation complexity involved. Note that while codebook design is a one-time cost in many applications, there are others where codebook adaptation is required. Our focus then is on reducing codebook design complexity for these situations. The computation complexity of the DA approach is due to three factors: first, computation of the Gibbs distribution to obtain each association probability involves the computation of an exponential term; second, this function has to be computed for a very large number of pairings (as many pairings as codevectors, for each of the training samples); third, this process has to be repeated for each iteration of the annealing process. Our proposed technique will address all these factors that contribute to the complexity by (i) defining simple functions to compute the association probability, (ii) reducing the

number of association probabilities to compute per training vector, and (iii) speeding up the iterations in the annealing process.

Although the literature is rich in fast search vector quantization techniques, where the aim is to have low encoding complexity [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26], there is relatively less work toward low-complexity VQ design techniques [9, 27, 28].

Our proposed reduced-complexity deterministic annealing approach is based on using soft information processing with simplified assignment measures. In digital communications engineering, soft information is “a reliability measure over the sample space of the investigated random variable.” More specifically when considering multiple decoding choices for a received noisy signal, soft information provides a measurement of how much confidence there is in making each possible decoded signal choice [29, 30, 31]. By analogy, in our VQ design, we will call soft information the relative “strength” of the association of each codevector to each sample vector. We will refer to this formulation as a *soft vector quantizer* (SVQ) design, where the encoding is “soft” in that each input vector in the training set gets assigned to multiple codewords. We develop reduced-complexity DA techniques through the design of simple soft measures that can mimic the effect of the Gibbs distribution used in the standard DA. Hence, while the designed soft measures are simple enough to facilitate fast computation, they also keep the performance penalty to a minimum by mimicking the Gibbs distribution’s functionality. We also derive a theoretical analysis of the performance loss when using a simplified measure instead of the optimal one, and further use the result to derive optimal annealing schedules for the proposed simple soft measures. In contrast to the standard DA which starts with essentially a single codevector and increases the size of the codebook through iterations, in SVQ the design starts with the required number of codevectors and optimizes their locations through iterations. It is also observed, and empirically shown, that the importance of a codevector for a given sample vector (in terms of the amount of probability mass associated with it) decreases exponentially fast with the distance from the sample vector, even at relatively high temperatures. Hence, major computational gains can be obtained with negligible performance degradation by considering only the *nearest few* codevectors from each sample vector. We present experimental evidence indicating that through these techniques significant performance gains are achieved by the SVQ algorithms over the traditionally used GLA and over SR-D, where the latter is widely thought to provide near-optimal performance. Compared to the standard DA, the results show drastic reductions in computational complexity with very small sacrifice in performance. It is also shown that using the SR technique [9] in combination with the SVQ algorithms results in further improvement in performance, although the added benefits of the SR techniques are less when the performance of the SVQ approaches the optimal.

This paper is organized as follows. Section 2 briefly introduces the standard DA method for VQ design and

points out its computational complexities. In Section 3, the reduced-complexity Gibbs distribution and low-complexity soft information measures for VQ design are proposed and analyzed. Optimal annealing schedules for the low-complexity soft information measures are also derived. Section 4 presents experimental results comparing the performances of the proposed algorithms with that of GLA, SR, and the standard DA on various Gauss-Markov, speech, and image sources. The effect of codebook initialization is also investigated. Finally, Section 5 concludes the paper.

2. VECTOR QUANTIZER DESIGN BY DETERMINISTIC ANNEALING

In the deterministic annealing algorithm proposed by Rose et al. [13], the main principle is the application of a probabilistic hierarchical clustering process, where each sample vector in the training set is associated to a cluster with a certain degree of membership. Each cluster R_j is represented by a codevector c_j . Thus, the distortion (energy) function to be minimized is an expected distortion function,

$$E\{D\} = \sum_x \sum_j P(x \in R_j) d(x, c_j), \quad (1)$$

where $d(x, c_j)$ is the distortion measure incurred in representing sample vector x by codevector c_j , and $P(x \in R_j)$ is the probability that x belongs to the cluster represented by c_j . The probability distribution used to define the associations is the Gibbs distribution, which maximizes the entropy under the constraint (1) [13]:

$$P(x \in R_j) = \frac{e^{-\beta d(x, c_j)}}{\sum_{i=0}^{|\mathfrak{R}|-1} e^{-\beta d(x, c_i)}}, \quad (2)$$

where $|\mathfrak{R}|$ is the cardinality of the cluster set, $\{R_0, R_1, \dots, R_{|\mathfrak{R}|-1}\}$. Notice that the distribution in (2) is a form of soft information. In other words, it gives a reliability value for assigning the sample vector x to cluster R_j over the sample space of the cluster set. The parameter β is a term that is inversely proportional to the temperature in the annealing process. Hence, at infinite temperature, which corresponds to $\beta = 0$, the probability associations are uniform: $P(x \in R_j) = 1/|\mathfrak{R}|$, for all x, j . This means that each sample vector x is equally assigned to all the clusters. As β gets large, that is, the temperature is lowered, the probability assignments for a sample vector x start to favor clusters closer to x ; the closer a cluster representative c_j to x , the higher its probability assignment. In the limit as $\beta \rightarrow \infty$, each sample vector gets assigned exactly to one cluster, namely, the cluster corresponding to the nearest codevector. We refer to this as the *hard* assignment, as opposed to the *soft* assignment where a sample vector gets assigned to more than one representative.

The codevector locations are defined as the weighted average of the sample vectors, where the weights are the probability associations of the sample vectors to the specific codevector being considered:

$$c_j = \frac{\sum_x x \cdot P(x \in R_j)}{\sum_x P(x \in R_j)}. \quad (3)$$

Thus, at $\beta = 0$ all cluster representatives are at the center of mass of the training set,

$$c_j = \frac{(1/|\mathfrak{R}|) \sum_x x}{(1/|\mathfrak{R}|) \sum_x 1} = \frac{1}{K} \sum_x x, \quad \forall j, \quad (4)$$

where K is the number of sample vectors in the training set. Essentially, at $\beta = 0$, there is, only one cluster (or Voronoi region), which is the whole set, and a single representative codevector at its center of mass. The hierarchical design algorithm in [13] starts the annealing process with the whole training set as one cluster at $\beta = 0$, gradually increases β , and reoptimizes by solving (3) at each β . As β is increased, the probability associations start to get “harder”, that is, more biased towards the closest codevector, and the system goes through a sequence of splitting of the clusters at *phase transitions* until the required number of clusters (or codevectors) are reached. The main focus in [13] is the derivation of the critical values of β , denoted by β_c , at which these phase transitions occur. The authors show that in order to be able to attain the global minimum, the splitting of the clusters should be at these critical moments, which correspond to optimal cluster splitting temperatures. Note that β does not control the size of the codebook; the system goes through a sequence of phase transitions until the required number of representatives (codevectors) is reached. During the annealing process whenever β reaches β_c for an existing cluster, that cluster splits into smaller clusters. In the limit $\beta \rightarrow \infty$, the associations become hard and each sample vector is associated with one representative as in the GLA algorithm.

The work by Rose et al. [13] provides the theoretical framework explaining how the DA approach avoids local minima, and shows that through the proposed annealing process a global minimum can be achieved. However, for practical applications this algorithm has some drawbacks: in particular, the annealing of the temperature has to be very slow especially in the vicinity of β_c , and the association probabilities for each sample vector have to be calculated for *all* codevectors. Such complexity may be excessive for many applications. In the next section, we present and analyze reduced-complexity techniques for VQ design that result in very significant computational gains with negligible performance degradation. In the sequel, we will refer to the method explained in this section as the *standard* DA.

3. REDUCED-COMPLEXITY DETERMINISTIC ANNEALING

3.1. Introduction

In our proposed *soft vector quantizer* (SVQ) algorithms, we formulate the vector quantizer design problem in a probabilistic framework as in the standard DA. However, unlike standard DA where each training vector is associated in probability with *all* of the codevectors, in SVQ algorithms each training vector is allowed to be associated with only a *subset* of the codevectors. These probability associations provide the relative reliability of each of the codevectors that the training vector can be mapped to and are a function of the relative distances of the codevectors to the training vector. The cost of the computation of the Gibbs soft assignment in (2), which involves exponentials, is high; if we count each of the basic operations (addition, subtraction, and multiplication) to take one floating point operation, *flop*, then an exponential computation takes 8 flops, and since soft assignments for *all* codevectors have to be updated for all sample vectors in every iteration in the standard DA, this results in a system of very high computational complexity. In order to reduce the computational complexity of the system, we would like to define and use a simpler distribution, preferably one that does not involve exponential terms. We define a general “simple” distribution as

$$p_0(c_i | x_k) = \frac{\mu(x_k, c_i)}{\sum_j \mu(x_k, c_j)}, \quad (5)$$

where $\mu(x_k, c_i)$ is a computationally easy-to-compute measure of the *goodness of match of codevector c_i to sample vector x_k* . The denominator is the sum of the goodness of matches with respect to a subset of N codevectors that are most relevant to x_k , where $N \leq |C|$ and $|C|$ is the codebook size. Note that, when $N = |C|$, all of the codevectors are regarded as relevant. Therefore, in (5) the softness of the assignment can be controlled by adjusting N (as N is reduced the assignment becomes hard). Recall that in (2) the term β acts as a *softness control* factor (i.e., as β increases assignments get harder), but for any given β assignment to all the codevectors was required. Thus, using a simple function, $\mu(x_k, c_i)$, coupled with $N \ll |C|$ can result in major computational gains at the expense of some reduction in performance, as only a fraction of codevector assignments is required.

For a given set of soft assignments, $p_0(c_i | x_k)$, for all i , for all k , the codevector locations can be computed as the weighted average of the sample vectors as in (3),

$$\begin{aligned} c_i^* &= \sum_k x_k p_0(x_k | c_i) = \frac{\sum_k x_k p(x_k) p_0(c_i | x_k)}{\sum_k p(x_k) p_0(c_i | x_k)} \\ &= \frac{\sum_k x_k p_0(c_i | x_k)}{\sum_k p_0(c_i | x_k)}, \end{aligned} \quad (6)$$

where sample vector probabilities are assumed to be uniform, $p(x_k) = 1/K$, and where K is the size of the training set. The general iterative framework for updating the soft assign-

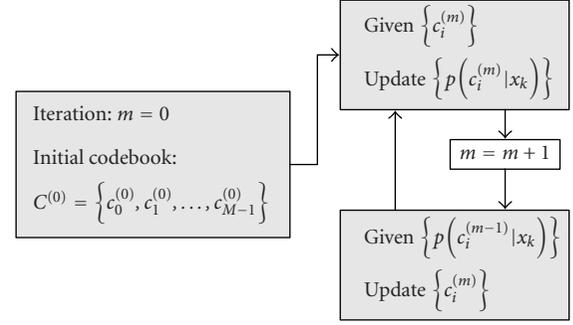


FIGURE 1: The iterative procedure showing the updating of the soft assignments and the codevectors.

ments and codevector locations is shown in Figure 1. Note that this framework is independent of the type of soft assignment used and can be applied for any choice of $\mu(x_k, c_i)$.

Ideally, in any annealing algorithm, the annealing temperature should start at a very high level (theoretically infinite) and gradually cool down to zero. However, as we have seen in the standard DA, this results in a very slow convergence. In our proposed SVQ algorithm, the temperature is not infinite at the start; we demonstrate that starting with a low temperature and with fixed (required) number of codevectors, it is possible to achieve near-optimal performance, even though starting with a low temperature means starting the algorithm with a nonconvex energy surface. We show that introduction of controlled randomness into the iterations, as in standard SR, helps in reducing the impact of this nonconvexity on the final design.

3.2. Reduced-complexity Gibbs distribution for VQ design

We know that as a result of soft association every sample vector x_k has a certain degree of belonging to *all* of the codevectors in the codebook. However, when we take all the soft associations into account, the effect of very small soft associations on (6) and on the converged codebook is negligible. Thus, in order to reduce the complexity, we compute the soft associations only for the N closest codevectors and set the other $|C| - N$ associations to zero. Note that the size of the codebook is not changed. This approach requires computing the distance from one input to all codevectors (which is required for any VQ design) and identifying the N nearest codevectors. Note that after the first few iterations the N nearest codevectors for training vectors need not be computed in every iteration, because the displacement of the codevectors from one iteration to the next tend to be small. Periodic updates of the list of N closest codevectors can be used to prevent changes in this list leading to inaccurate assignment probabilities. Denoting $N(x_k, N)$ to be the nearest N codevectors from a given x_k , the soft information is computed by

$$p(c_i | x_k) = \frac{e^{-\beta d(x_k, c_i)}}{\sum_{c_j \in N(x_k, N)} e^{-\beta d(x_k, c_j)}}, \quad (7)$$

TABLE 1: Average performance and running time comparison for $N = |C| = 128$ and $N = 4$. The source is uncorrelated Gaussian, the vector dimensions are 16, and the soft information measure is reduced-complexity Gibbs distribution. The results are averaged over 20 experiments (details on experimental setup are in Section 4).

N	Ave. SNR	Ave. CPU time
4	3.595 dB	136 s
128	3.598 dB	16483 s

where $d(x_k, c_i) = \|x_k - c_i\|^2$. Note that with this kind of exponential function the assignment probabilities decay very rapidly, even when β is low.

3.2.1. Fixed number of associations

One can first consider an approach with a fixed N . Experimentally, we have found $N = 4$ to be a good tradeoff value between performance and complexity. In other words, results obtained by setting $N = 4$ and with $N = |C|$ (i.e., using all the codevectors in the codebook) resulted in negligible performance difference, however, the computational savings are significant, especially for large codebooks (e.g., $|C| > 128$). A comparison of $N = 4$ and $N = |C| = 128$ using the same annealing schedule is given in Table 1. The loss in performance incurred by considering only the nearest 4 codevectors for each sample vector instead of the whole codebook is negligible for all practical purposes, while a very significant complexity reduction (about a factor of 120 speedup) is achieved.

The proposed algorithm is shown in Figure 2, where the iterations start with $N = 4$, $\beta = 1/(4\sigma_X^2)$, where σ_X^2 is the source variance, and an initial codebook C_0 . The initial value of β , $\beta = 1/(4\sigma_X^2)$, is found empirically to be a good starting value. At each iteration, we gradually increase β ($\kappa > 1.0$), update the soft information according to (7), and reoptimize the codevector locations using (6). We can then apply the codevector perturbation method of [9]. As β increases, the *softness* of the codevector associations decreases. In the limit, when all the probability mass is assigned to the nearest codevector for all sample vectors, we reach the nearest neighbor condition.

3.2.2. Variable number of associations

When we need to design quantizers for very large codebook sizes (e.g., $|C| = 512, 1024, \dots$), it is useful to use N larger than 4 (e.g., 10, 12, 15, \dots). However, we know that while the N th furthest away codevector from a given sample vector plays an important role (has large probability mass) in the early iterations, its importance decreases in each iteration. As the temperature decreases, the probability mass is gradually transferred from the distant to the closer codevectors. Hence, after a while, the N th codevector will contain negligible mass and it can be discarded without any significant effect on the final performance. Thus to simplify the computation without affecting the performance, we can modify our algorithm as follows: whenever the average probability mass of the nearest $N - 1$ codevectors, $PM(N - 1)$ exceeds a certain mass π

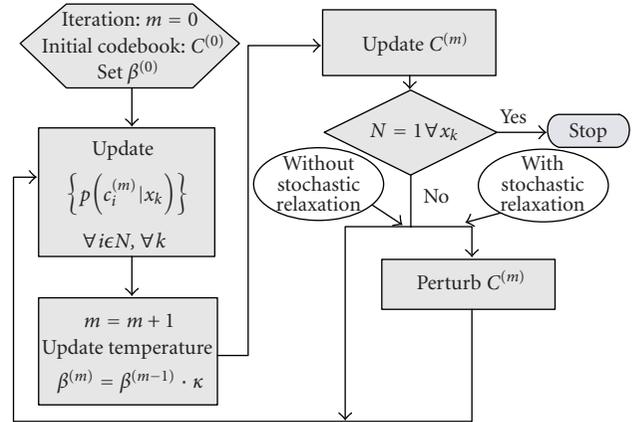


FIGURE 2: Flowchart for the reduced-complexity Gibbs soft assignment measure algorithm.

(e.g., $\pi = 0.99$), N is reduced by one, $N = N - 1$, that is,

$$\text{if } PM(N - 1) = \frac{1}{K} \sum_k \sum_{c_i \in N(x_k, N-1)} p(c_i | x_k) \pi, \quad (8)$$

then $N = N - 1$,

where K is the size of the training set. Gradually decreasing N , as the temperature changes, results in considerable complexity reduction, except when N becomes small. This is because reducing N requires computing $PM(N - 1)$ at each iteration, which adds to the overall complexity. Therefore, in our algorithm, when N reaches a small value, for example, $N = 4$, the process of gradual reduction of N stops.

3.3. Low-complexity soft information measures for VQ design

As previously stated, in order to reduce further the computational complexity of the system, we can use in (5) a less complex distribution than the optimal Gibbs distribution. One of the simplest distributions that readily comes to mind is the “inverse Euclidean distance” distribution, in which, for a given sample vector x_k , the “importance” of the codevectors decreases with increasing distance from x_k . “Inverse Euclidean distance” in a soft information measure can be defined as

$$p(c_i | x_k) = \frac{1/d_i}{\sum_{j=0}^{N-1} 1/d_j}. \quad (9)$$

The distances in (9) are the Euclidean norms between x_k and the codevectors (n is the vector dimension), $d_i = d(x_k, c_i) = \sqrt{(x_{k,0} - c_{i,0})^2 + (x_{k,1} - c_{i,1})^2 + \dots + (x_{k,n-1} - c_{i,n-1})^2}$. The number of codevectors to be taken into consideration for each x_k can be determined by a circle centered on x_k with a radius R . R is chosen so that it includes the N nearest codevectors to x_k . The radius R decreases from one iteration to the next, $R^{(m)} = R^{(m-1)}\rho$, where $0 < \rho < 1.0$. Note that as

since Gibbs is the optimal soft measure. In this section, we will show how this performance loss can be reduced if, in the low-complexity approach, we can find temperature reduction schedules that approximate those from the Gibbs β schedule. To achieve this, we will want to minimize the L_1 distance between the two distributions [32],

$$\|p_0(c|x) - p_G(c|x)\|_1 = \sum_i |p_0(c_i|x) - p_G(c_i|x)|. \quad (14)$$

More specifically, we would like to find the spread reduction schedule R_x for a given Gibbs β schedule such that (14) is minimized. Note that minimizing (14) is related to minimizing the relative entropy between $p_0(c|x)$ and $p_G(c|x)$, $D(p_0(c|x)||p_G(c|x))$, since we know from [32] that

$$D(p_0(c|x)||p_G(c|x)) \geq \frac{1}{2 \ln 2} \|p_0(c|x) - p_G(c|x)\|_1^2 \quad (15)$$

with equality when $p_0 = p_G$. Although it is intuitive that in order to minimize the performance difference between a simplified soft measure and the optimal soft measure, the relative entropy between the two should be minimized, we refer to the appendix for a more formal justification. The error analysis in the appendix shows that at a given system entropy (softness) the performance loss in terms of distortion between two distributions (soft measures) is a function of the relative entropy between the distributions, so that we can show that minimizing the relative entropy results in minimizing the distortion penalty paid for using a simplified soft measure.

We can show that the relative entropy is approximately minimized when the variances of the two distributions, $p_0(c|x)$ and $p_G(c|x)$, are equal. The variances of $p_G(c|x)$ and $p_0(c|x)$, respectively, are (the lower limits of the integrals start from zero because we use absolute distances between sample vector and each of the codevectors):

$$\begin{aligned} \text{var}_G &= \mathbb{E}_G\{z^2\} - (\mathbb{E}_G\{z\})^2 \\ &= \frac{\int_{z=0}^{\infty} z^2 e^{-\beta z^2} dz}{\int_{z=0}^{\infty} e^{-\beta z^2} dz} - \left(\frac{\int_{z=0}^{\infty} z e^{-\beta z^2} dz}{\int_{z=0}^{\infty} e^{-\beta z^2} dz} \right)^2 \end{aligned} \quad (16)$$

$$= \frac{1}{2\beta} - \left(\sqrt{\frac{1}{\pi \cdot \beta}} \right)^2 = \frac{\pi - 2}{2\pi \cdot \beta},$$

$$\begin{aligned} \text{var}_0 &= \mathbb{E}_0\{z^2\} - (\mathbb{E}_0\{z\})^2 \\ &= \frac{\int_{z=0}^{R_x} z^2 (R_x - z) dz}{\int_{z=0}^{R_x} (R_x - z) dz} - \left(\frac{\int_{z=0}^{R_x} z (R_x - z) dz}{\int_{z=0}^{R_x} (R_x - z) dz} \right)^2 \\ &= \frac{R_x^2}{6} - \left(\frac{R_x}{3} \right)^2 = \frac{R_x^2}{18}. \end{aligned} \quad (17)$$

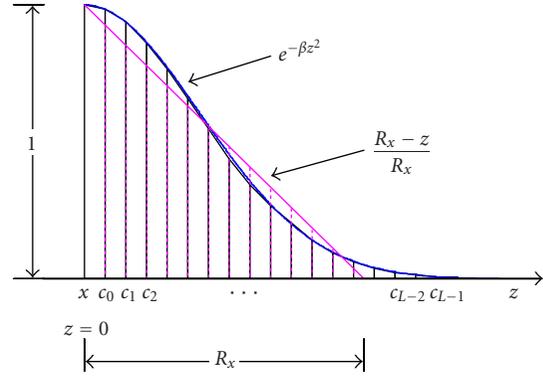


FIGURE 4: An instance of the Gibbs function with parameter β and an instance of the triangle function with parameter R_x are shown. There are L codevectors at increasing distance from sample vector x .

Equating (16) and (17), and solving for R_x , we get

$$R_x = \sqrt{\frac{9(\pi - 2)}{\pi \cdot \beta}}. \quad (18)$$

Hence, using (18) we can obtain a schedule for R_x given a schedule for β .

We have used the setup in Figure 4 to show that for a given β for the Gibbs distribution, the spread R_x obtained by (18) for the triangle distribution minimizes the relative entropy. In the figure, there is a set of L codevectors at increasing distances from a sample vector x . For each β in a set $\{\beta_1, \beta_2, \dots, \beta_m\}$, the soft Gibbs assignments of the codevectors is computed using (7) with $N = L$. Then, through an exhaustive search, the spread R_x that gives the soft assignments using (13) which minimizes the relative entropy $D(p_0(c|x)||p_G(c|x))$ is obtained.

The resulting minimum relative entropy curve is shown in Figure 5 by the solid line. This is compared with the result obtained using (18) by the dashed curve. We can see that the derived relation in (18) can approximate well the minimum relative entropy curve, and hence the best R_x schedule for a given β schedule. The error is due to the fact that we approximate the relative entropy using the variance of the two functions.

The reduced-complexity Gibbs algorithm and the low-complexity soft measure algorithm for the triangular membership function using two different spread reduction schedules are used to design codebooks of sizes 128 and 256 (for details on the experiments, see Section 4). The results are shown in Table 2. Of the two schedules for triangular soft information measure, the first one is the geometric spread reduction given in (11), and the second one is obtained using (18) and the Gibbs schedule (referred to as Gibbs guided spread reduction in Table 2). Observe from the results in Table 2 that the performance of the triangular soft information measure using the Gibbs guided spread reduction schedule outperformed the geometric spread reduction when both are operated with the same number

TABLE 2: Comparing the geometric and the Gibbs guided spread (temperature) reduction for the triangular membership function for the design of 128 and 256 sized codebooks for uncorrelated Gaussian source with vector dimension 16.

Codebook size	Reduced-complexity Gibbs soft information measure ($\kappa = 1.005$)		Low-complexity soft information measure (triangular) Geometric spread reduction ($\rho = 0.995$)		Low-complexity soft information measure (triangular) Gibbs guided spread reduction	
	Ave. SNR	Ave. CPU time	Ave. SNR	Ave. CPU time	Ave. SNR	Ave. CPU time
128	3.595 dB	136 s	3.392 dB	91 s	3.411 dB	91 s
256	5.210 dB	329 s	4.919 dB	232 s	4.952 dB	232 s

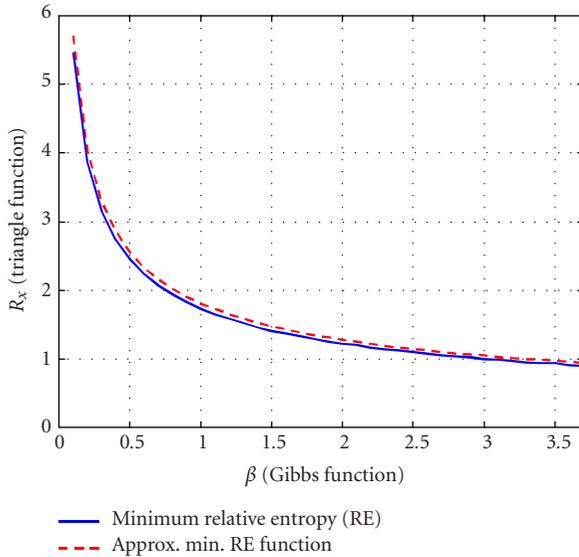


FIGURE 5: Plot showing the minimum relative entropy between the triangle soft measure and the Gibbs soft measure at various spread R_x and β pairs. The solid curve is obtained by an exhaustive search for R_x that gives the minimum relative entropy for a given value of β . The dashed curve is obtained using the derived relationship between R_x and β to give the minimum relative entropy.

of iterations. Therefore, for a given β -schedule the relation in (18) provides a better R_x -schedule than the geometric reduction. Note that β -schedule is itself geometric, $\beta^{(m)} = \beta^{(m-1)} \cdot \kappa$. But since the Gibbs soft information measure is the optimal measure, following the β -schedule for the simple soft information measure results in an increase in performance as demonstrated in Table 2. Note also that to obtain the β -schedule the Gibbs algorithm need not be run, it can be obtained using $\beta^{(m)} = \beta^{(m-1)} \cdot \kappa$, $\kappa > 1.0$.

4. EXPERIMENTAL RESULTS

We now present the results obtained when our algorithms were used to design codebooks of various sizes and sources. The results are compared with other algorithms of interest, namely, GLA, SR-D, and standard DA. Our quoted execution times (CPU times) are based on those obtained with an *Intel PIII-550 MHz* machine.

The first set of training sources we considered was two cases of first-order Gauss-Markov sources, one with correlation coefficient $\alpha_0 = 0.0$ (uncorrelated source) and the other with $\alpha_0 = 0.9$ (correlated source). We divided 16 384 samples into 1024 16-dimensional training vectors, and designed codebooks of sizes 32, 64, 128, and 256 for both training sets, where the initial codebooks were obtained randomly from the training sets. Since both GLA and SR-D are sensitive to the choice of the initial codebooks, in order to investigate the effect of initialization, we have also designed codebooks of sizes 32, 64, 128, and 256, where the pairwise nearest neighbor (PNN) algorithm [33] is used to obtain the initial codebooks. For this we have used the uncorrelated Gaussian source with 4096 16-dimensional training vectors.

The second source examined was a segment of human speech sampled at 8 kHz and partitioned into 2048 16-dimensional vectors, and we have designed five codebooks of sizes 16, 32, 64, 128, and 256. The final source considered was obtained by extracting 8192 16-dimensional vectors (corresponding to 4×4 blocks) from two 512×512 monochrome training images from the USC image database, with each pixel amplitude quantized to 8 bits. Four codebooks of sizes 32, 64, 128, and 256 were designed using this training set, and the performance of these codebooks is tested in coding the image “Lena” which was outside of the training set. The effect of the PNN initialization on the speech and the image sources is also demonstrated.

We designed codebooks for the following algorithms where in the plots an “a” appended to an algorithm name means that stochastic perturbation was used (e.g., SVQ-Ga would mean the same as SVQ-G but without perturbation).

- (1) SVQ-G: soft vector quantizer design using the reduced-complexity Gibbs distribution as the soft measure, and with stochastic perturbation.
- (2) SVQ-E: soft vector quantizer design using the inverse Euclidean distance distribution as the soft measure, and with stochastic perturbation.
- (3) SVQ-T: soft vector quantizer design using the height-defined distribution with triangular membership function as the soft measure, and with stochastic perturbation.
- (4) VQ-DA: vector quantizer design using the standard deterministic annealing [13].
- (5) SR-D: vector quantizer design using the reduced-complexity decoder perturbation algorithm [9].

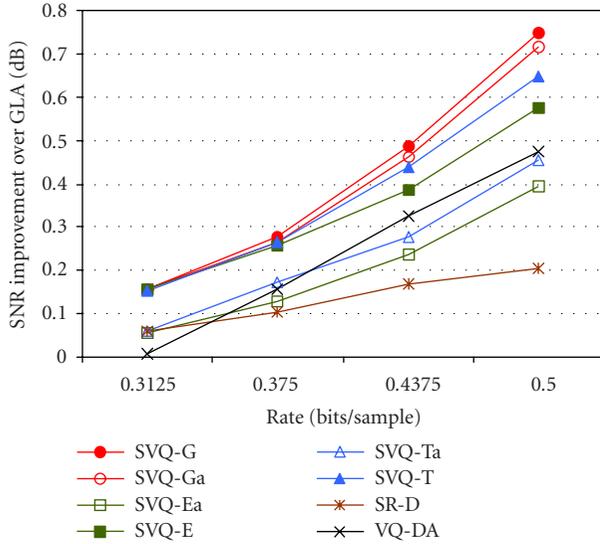


FIGURE 6: Improvements over GLA for Gaussian source; vector dimension = 16 samples/vector.

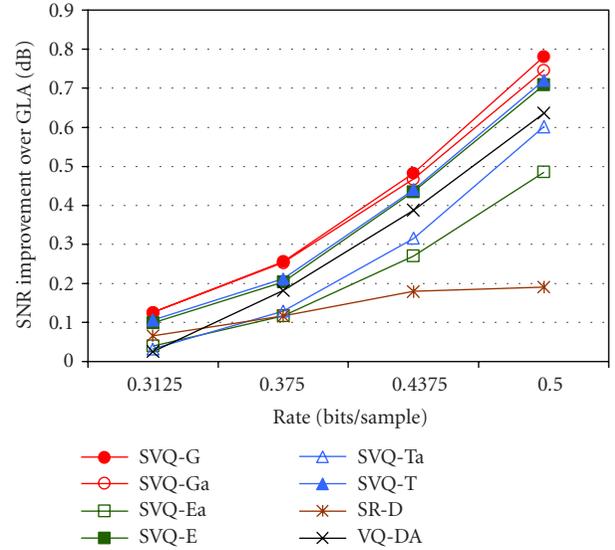


FIGURE 7: Improvements over GLA for Gauss-Markov source; vector dimension = 16 samples/vector.

(6) GLA: vector quantizer design using the generalized Lloyd algorithm [6].

In the cases where PNN initialization is not used, for each algorithm, except VQ-DA, the average performances for 20 different initial codebooks are computed. To allow us to compare the average performances of the different algorithms, the same set of initial codebooks is used. Recall that VQ-DA uses the center of mass of the training set as the initial codebook, so its performance with this initial condition is recorded. In the cases where PNN initialization is used, a unique initial codebook is obtained from the training set. The performance measure used for the image source is peak signal-to-noise ratio (PSNR) and for the others is signal-to-noise ratio (SNR), defined as $PSNR = 10 \cdot \log_{10}(255^2/D)$ and $SNR = 10 \cdot \log_{10}(P_s/D)$, where P_s is the signal power and D is the distortion per sample. The SR-D algorithm was run for 200 iterations as given in [9], and the GLA was run until convergence.

4.1. Gauss-Markov sources

The performances of the first 5 algorithms (listed above, both with and without perturbation) with initial codebooks obtained randomly from the training set are compared with the GLA performances in Figures 6 and 7. In all cases, the reduced-complexity DA algorithms (SVQ) achieved significant improvements over the traditionally used GLA and over SR-D, which is said to give near-optimal results [9]. From the figures we observe that the SVQ-G algorithm (reduced-complexity Gibbs distribution) performed better than the other SVQ algorithms; however, the performance of SVQ-T is competitive. Note the progression of performances of the low-complexity soft information measures: the performance improves from the inverse Euclidean distance soft measure (SVQ-E and SVQ-Ea) to the triangular soft measure (SVQ-

T and SVQ-Ta). This was an expected result since the triangular soft measure was designed to better approximate the optimal Gibbs distribution. Note also the gain achieved by the stochastic relaxation (SR) in the SVQ algorithms compared to nonstochastic cases. The gain ranges from a high 0.2 dB for SVQ-E to a low 0.02 dB for SVQ-G algorithms. It should be noted that the better an algorithm performs *without* the SR, the lesser the additional gain achieved by the SR in the SVQ algorithms. In other words, as an algorithm comes closer to the global optimum using the principles of soft information processing, it requires less help from the SR to attain an improved performance. In the limit, granting enough computational resources for the full power of the soft information processing to be utilized, the global optimum can be reached without requiring any help from SR. But as the results demonstrate, for reduced-complexity DA approaches, SR has a positive effect in the improvement of the performances with negligible computational complexity.

The results for VQ-DA (standard DA) were obtained starting with all the sample vectors being equally associated with all the codevectors, which dictates an initial codebook where all the codevectors are at the center of mass of the training set. The simulations were conducted with a conservative annealing schedule, where it took over 120000 CPU seconds (about 24 hours) for the codebook of size $|C| = 256$ to converge. Recall that in VQ-DA the probability associations are computed to all codevectors for each sample vector, thus the algorithm executes very slowly especially for large codebooks. The figures show that the performance of VQ-DA compared to reduced-complexity DA algorithms is inferior in all cases. Moreover, the SVQ algorithms run much faster than VQ-DA, requiring 350 CPU seconds for $|C| = 256$ and 16-dimensional vectors. While, if enough computational resources are allocated, VQ-DA is expected to be very close to

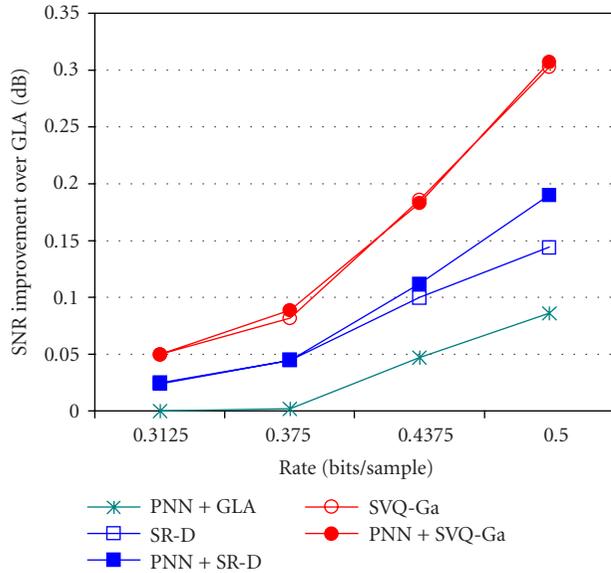


FIGURE 8: The effect of PNN initialization as improvement over GLA. Source is Gaussian, vector dimension = 16 samples/vector.

optimal as shown in [13], the performance of the reduced-complexity DA algorithms proved that for most practical applications the expected performance of VQ-DA does not justify its computational burden.

Both GLA and SR-D algorithms are sensitive to the initial codebooks. Hence, in order to investigate the effect of initialization on these algorithms and on our proposed algorithms, we have used the PNN initialization for the codebooks [33]. In Figure 8, we show the performances of the 4 codebooks on (uncorrelated) Gaussian source as improvement over the PNN initialized GLA. For clarity of presentation, we have only included the SVQ-Ga performance from our proposed algorithms; the other SVQ algorithms behave comparatively the same with SVQ-Ga as in Figure 6. Note from the figure that the PNN initialization improves the GLA and SR-D algorithms, however, the SVQ-Ga algorithm is not affected. This is a positive result for the SVQ algorithms for it shows that they can evade the local minimum dictated by the initial codebook, and hence are insensitive to the choice of the initial codebook. The PNN and its fast but sub-optimal version, fast-PNN, require $O(K^3)$ and $O(K \log K)$ time, respectively, where K is the size of the training set [33]. The results presented in Figure 8 are obtained using the full search PNN algorithm (with complexity $O(K^3)$) in order to get the best possible results with the GLA and the SR-D algorithms. The fast-PNN initialization would result in reduced performance; it is shown in [33] that the fast-PNN algorithm increased the coding error by 0.4–0.6 dB for image sources compared to full search PNN. The SVQ algorithms outperform both GLA and SR-D algorithms without the complexity of the initialization process, which gets computationally more impractical as the size of the training set increases. The running time for the generation of the PNN codebooks from a training set of 4096 16-dimensional vectors was 2374 CPU seconds,

and the design of the size 256 codebooks for GLA, SR-D, and SVQ-Ga algorithms on average was 44, 366, and 1552 CPU seconds on the same machine. Therefore, with the PNN initialization, the total running times for the GLA and the SR-D algorithms are higher than the SVQ-Ga algorithm. Since SVQ-Ga performs the same with and without the initialization, the SVQ-Ga algorithm outperforms GLA and SR-D in less running time.

4.2. Speech source

The performance on the speech source using the three algorithms, GLA, SR-D, and SVQ-Ga, with and without the codebook initialization, is shown in Figures 9 and 10. In Figure 9, the performance improvement over GLA and in Figure 10, improvement over PNN initialized GLA are shown. Note that while the performance improvement of SVQ-Ga over GLA is large (0.95 dB at 0.5 bit/sample), compared with the PNN initialized GLA the improvement is rather modest. But note again that the effect of the initialization is very small on the SVQ-Ga performance, whereas improvements of 0.85 dB and 0.2 dB are obtained at 0.5 bit/sample for GLA and SR-D, respectively, after initialization. Therefore, as in the Gaussian source, the SVQ-Ga renders the initialization unnecessary.

4.3. Image source

The last source considered was the image source, where the results are shown in Figure 11 for the coding of the image source “Lena.” As in the previous two source cases, the SVQ-Ga performance is practically not sensitive to the initial codebook initialization and it outperformed the GLA and the SR-D algorithms by 0.3–0.4 dB and 0.2–0.3 dB, respectively, both being initialized with PNN. Therefore, as in the Gaussian and the speech sources, the SVQ-Ga outperformed the PNN + GLA and PNN + SR-D without the need of initialization.

5. CONCLUSION

In this paper, we have designed reduced/low-complexity methods for deterministic annealing (DA) for the vector quantizer design problem, which we called *soft vector quantizer* (SVQ) design algorithms. The proposed low-complexity soft measures are used as the soft association probabilities in the probabilistic framework of the DA to reduce the computational cost compared to the optimal Gibbs soft measure used in the standard DA. Although the simple soft measures significantly reduce the computational complexity of the system, this improvement comes at a price since these soft measures are not the optimal distributions. Hence, we have also derived the theoretical performance loss for using a simplified measure instead of the optimal measure, and used the result to derive optimal annealing schedules for the proposed simple soft measures. We have demonstrated that using the derived optimal schedule for the low-complexity soft measures increases the quality of the final codebook compared to using a geometric reduction schedule which is usually suggested in the annealing algorithms. We have also shown that the low-complexity DA methods benefit from the stochastic

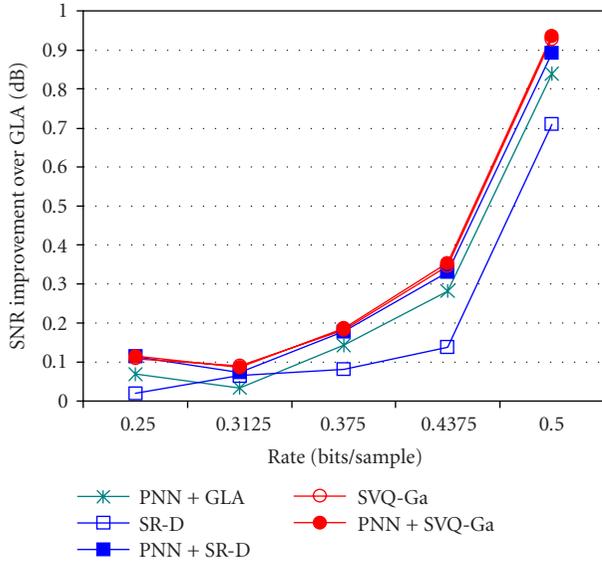


FIGURE 9: Improvements over GLA for human speech source sampled at 8 kHz; vector dimension = 16 samples/vector.

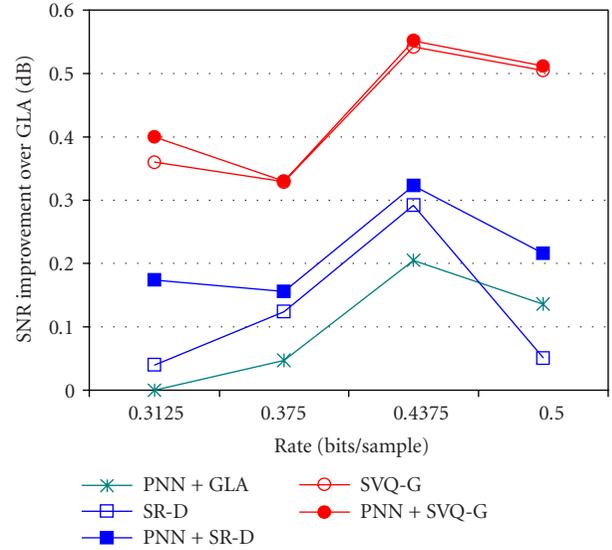


FIGURE 11: Improvements over GLA for the coding of image source "Lena." Vector dimension = 16 samples/vector (corresponding to 4×4 blocks).

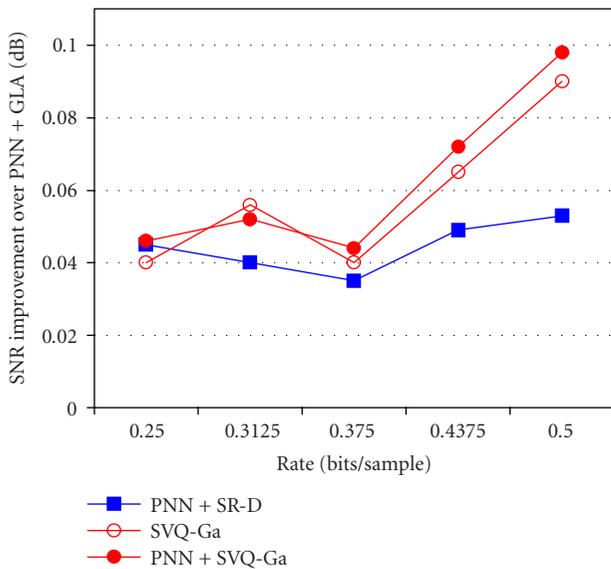


FIGURE 10: Improvements over PNN-initialized GLA for human speech source sampled at 8 kHz; vector dimension = 16 samples/vector.

relaxation techniques with decreasing benefits as the performance approaches the optimal.

We have demonstrated the effectiveness of our low/reduced-complexity DA (SVQ) algorithms by designing codebooks for a variety of sources, namely, Gauss-Markov, speech, and image, at different rates. In each case, the proposed SVQ algorithms significantly improved the quality of the final codebooks compared to the traditionally used GLA and compared to the SR-D algorithm, where the latter is accepted as a benchmark reference by some researchers to

be a VQ design technique that performs near-optimally. We have also investigated the effect of codebook initialization on GLA, SR-D, and SVQ algorithms and showed that, while GLA and SR-D receive major benefit from this initialization at the expense of increased computational complexity, the SVQ algorithms are able to attain the same performance without the need of initialization. Hence, the SVQ algorithms are not sensitive to the choice of the initial codebook and outperform codebook initialized GLA and SR-D algorithms. Compared to the standard DA, the computational complexity of the SVQ algorithms is shown to be drastically reduced. Using the same annealing temperature, the SVQ algorithms run more than a factor of 100 faster than the standard DA algorithm with negligible performance difference. We believe that the proposed algorithms, with their significantly higher performance over the widely used GLA and SR-D, and with their low-computational complexity with negligible performance difference compared to the standard DA, have proved themselves to be important VQ design techniques.

APPENDIX

Our proposed triangular soft information measure significantly reduces the computational cost of the soft assignments compared to the optimal Gibbs soft measure, at the cost of some loss in performance. In this appendix, we derive the penalty paid in distortion for using the simplified soft measure instead of the optimal one at a given system entropy (softness), and show that minimizing the relative entropy between the two measures minimizes the distortion penalty.

For a given soft assignment measure (conditional probability), $p(c|x)$, we have the expected distortion and the

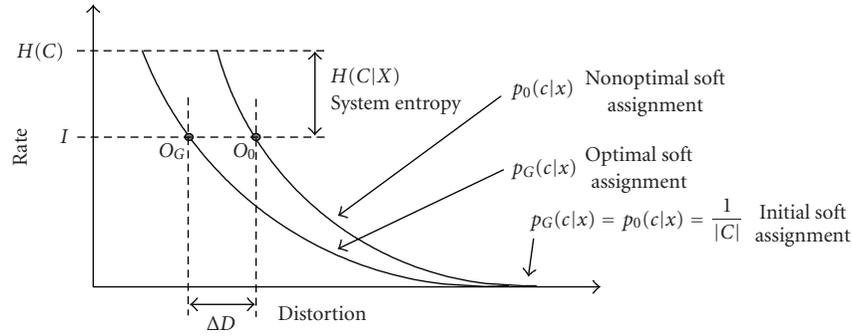


FIGURE 12: Convergence of the optimal and nonoptimal soft assignment measures. Starting with equal, uniform soft assignments, the optimal soft assignment measure achieves a lower distortion than the nonoptimal soft measure. At a given system entropy level $H(C|X)$, the difference in distortion is shown as ΔD . O_G and O_0 show the points of minimum distortion attainable without exceeding rate I , for the optimal and nonoptimal assignment measures, respectively. $|C|$ is the codebook size, and $H(C)$ is the codebook rate.

average mutual information in (A.1) and (A.2), respectively.

$$D(p(c|x)) = \sum_x p(x) \sum_c p(c|x) d(x, c), \quad (\text{A.1})$$

$$I(X; C) = \sum_x p(x) \sum_c p(c|x) \log \frac{p(c|x)}{p(c)}. \quad (\text{A.2})$$

Let P_I be the set of all I -admissible soft assignment measures,

$$P_I = \{p(c|x) : I(X; C) \leq I\}, \quad (\text{A.3})$$

and hence, for fixed I ,

$$D(R) = \min_{p(c|x) \in P_I} D(p(c|x)), \quad (\text{A.4})$$

where $R(D) = \min_{p(c|x) \in P_I} I(X; C)$. Now, let $p_G(c|x)$ and $p_0(c|x)$ be two different soft assignment measures, and assume that $p_G(c|x)$ is the optimal I -admissible soft assignment measure for some rate I , $p_G(c|x) \in P_I$, and the expected distortion corresponding to $p_G(c|x)$ is $D(p_G(c|x))$. Let the other soft assignment measure, $p_0(c|x)$, be defined as

$$p_0(c|x) = p_G(c|x) + \Delta p(c|x) \quad \forall c, x. \quad (\text{A.5})$$

We require that (A.5) satisfy two conditions:

$$\sum_c \Delta p(c|x) = 0 \quad \forall x, \quad (\text{A.6})$$

$$\Delta I = I_0(X; C) - I_G(X; C) = 0. \quad (\text{A.7})$$

The condition in (A.6) is required so that $p_0(c|x)$ in (A.5) is a valid pmf, $\sum_c p_0(c|x) = 1$, and the condition in (A.7) requires that the difference in mutual information be zero so that $p_0(c|x) \in P_I$ and the system entropy remains unchanged. We would like to obtain the difference in the expected distortion, $\Delta D = D(p_0(c|x)) - D(p_G(c|x))$, subject to the conditions in (A.6) and (A.7). The situation is depicted in Figure 12, where the results of the iterations are plotted for rate versus distortion for two assignment measures.

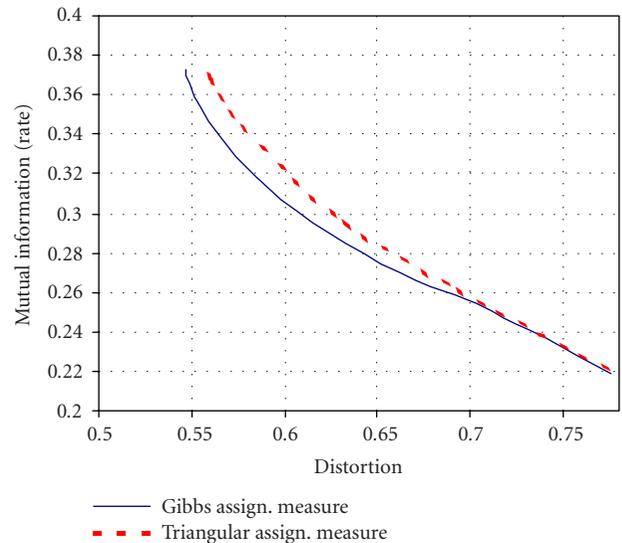


FIGURE 13: Plot showing the convergence of two experiments, one with Gibbs soft assignment measure and the other with triangular soft assignment measure, for the design of a codebook of size 64, vector dimension 16, and the source type zero-mean, unit variance Gaussian.

A real simulation result is shown in Figure 13. The curves are obtained by starting with uniform sample vector-codevector assignments (maximum softness) and plotting rate against the distortion obtained using the intermediate codebooks in each iteration (nearest sample vector-codevector assignments). Note that the conditional entropy $H(C|X)$ is the system entropy and it measures the softness of the assignments, where $I(X; C) = H(C) - H(C|X)$ and $H(C)$ is the codebook rate. For example, in Figure 13, where the codebook size is $|C| = 64$ and the vector dimension is $n = 16$, the codebook rate is $\log_2 |C|/n = 0.375$ bit/sample. Through iterations the softness of the assignments is reduced until finally we reach the hard assignments, where $H(C|X) = 0$ and each sample vector x is assigned to a single codevector with probability one.

We will start by expanding $\Delta I = I_0(X; C) - I_G(X; C)$,

$$\begin{aligned} \Delta I &= \sum_x p(x) \sum_c p_0(c|x) \log \frac{p_0(c|x)}{p_0(c)} \\ &\quad - \sum_x p(x) \sum_c p_G(c|x) \log \frac{p_G(c|x)}{p_G(c)}. \end{aligned} \quad (\text{A.8})$$

Substitute $p_G(c|x) = p_0(c|x) - \Delta p(c|x)$ from (A.5) into (A.8):

$$\begin{aligned} \Delta I &= \sum_x p(x) \sum_c p_0(c|x) \log \frac{p_0(c|x)}{p_0(c)} \\ &\quad - \sum_x p(x) \sum_c (p_0(c|x) - \Delta p(c|x)) \log \frac{p_G(c|x)}{p_G(c)} \\ &= \sum_x p(x) \sum_c \Delta p(c|x) \log \frac{p_G(c|x)}{p_G(c)} \\ &\quad + \sum_x p(x) \sum_c \left[p_0(c|x) \log \frac{p_0(c|x)}{p_G(c|x)} \right. \\ &\quad \quad \left. - p_0(c|x) \log \frac{p_0(c)}{p_G(c)} \right]. \end{aligned} \quad (\text{A.9})$$

Simplifying the above expression, we get

$$\begin{aligned} \Delta I &= \sum_x p(x) \sum_c \Delta p(c|x) \log \frac{p_G(c|x)}{p_G(c)} \\ &\quad + D(p_0(c|x) \| p_G(c|x)) - D(p_0(c) \| p_G(c)). \end{aligned} \quad (\text{A.10})$$

The optimal distribution is the Gibbs distribution, $p_G(c|x) = e^{-\beta d(x,c)} / \sum_{c'} e^{-\beta d(x,c')}$, where $p_G(c)$ is uniform. Let $p_G(c) = \zeta$, a constant for all c , and substitute the Gibbs distribution in the first term in (A.10):

$$\begin{aligned} &\sum_x p(x) \sum_c \Delta p(c|x) \log \frac{p_G(c|x)}{p_G(c)} \\ &= \sum_x p(x) \sum_c \Delta p(c|x) \log \left(\frac{e^{-\beta d(x,c)}}{\zeta \cdot \sum_{c'} e^{-\beta d(x,c')}} \right) \\ &= \sum_x p(x) \sum_c \Delta p(c|x) \left(-\frac{\beta d(x,c)}{\ln 2} - \log \sum_{c'} \zeta \cdot e^{-\beta d(x,c')} \right). \end{aligned} \quad (\text{A.11})$$

Therefore,

$$\begin{aligned} &\sum_x p(x) \sum_c \Delta p(c|x) \log \frac{p_G(c|x)}{p_G(c)} \\ &= \frac{-\beta}{\ln 2} \underbrace{\sum_x p(x) \sum_c \Delta p(c|x) d(x,c)}_{=\Delta D} \\ &\quad - \sum_x p(x) \left(\log \sum_{c'} \zeta \cdot e^{-\beta d(x,c')} \right) \underbrace{\sum_c \Delta p(c|x)}_{=0 \quad \forall x} \\ &= \frac{-\beta}{\ln 2} \Delta D. \end{aligned} \quad (\text{A.12})$$

Substituting (A.12) into (A.10), and using the condition in (A.7) that $\Delta I = 0$, we get

$$\Delta I = \frac{-\beta}{\ln 2} \Delta D + D(p_0(c|x) \| p_G(c|x)) - D(p_0(c) \| p_G(c)) = 0. \quad (\text{A.13})$$

Finally, the difference in the expected distortion is

$$\Delta D = \frac{\ln 2}{\beta} [D(p_0(c|x) \| p_G(c|x)) - D(p_0(c) \| p_G(c))]. \quad (\text{A.14})$$

Note that for large vector dimensions [32], $-D(p_0(c) \| p_G(c)) \cong D(p_G(c) \| p_0(c))$. Hence, for large dimensions, the penalty paid in terms of distortion at a given system entropy for using the nonoptimal soft assignment measure, $p_0(c|x)$, instead of the optimal one, $p_G(c|x)$, is

$$\Delta D = \frac{\ln 2}{\beta} [D(p_G(c) \| p_0(c)) + D(p_0(c|x) \| p_G(c|x))]. \quad (\text{A.15})$$

Note also that $p_G(c)$ is uniform and $p_0(c)$ depends on $p_0(c|x)$. Hence, minimizing the conditional relative entropy, $D(p_0(c|x) \| p_G(c|x))$, in (A.15) minimizes ΔD .

ACKNOWLEDGMENTS

We would like to thank Professor Z. Zhang and Professor K. M. Chugg for their feedback in the derivation of the performance bound for the low-complexity measures. This paper is based upon work supported by the National Science Foundation under Grant no. 9730556, by the Charles Lee Powell Foundation, and by Integrated Media Systems Center (a National Science Foundation engineering research center at the University of Southern California).

REFERENCES

- [1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, Norwell, Mass, USA, 1992.
- [2] R. L. Joshi, H. Jafarkhani, J. H. Kasner, et al., "Comparison of different methods of classification in subband coding of images," *IEEE Trans. Image Processing*, vol. 6, no. 11, pp. 1473–1486, 1997.
- [3] E. Simoncelli and J. Portilla, "Texture characterization via joint statistics of wavelet coefficient magnitudes," in *Proc. IEEE 5th International Conference on Image Processing (ICIP '98)*, vol. 1, pp. 62–66, Chicago, Ill, USA, October 1998.
- [4] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 713–718, 1992.
- [5] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 5, no. 3, pp. 257–265, 1997.
- [6] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, no. 1, pp. 84–95, 1980.
- [7] L. A. Rastrigin, *Random Search in Optimization Problems for Multiparameter Systems*, "Zinatne" Publishing House, Riga, Latvia, 1965.
- [8] S. Geman and D. Geman, "Stochastic relaxation, Gibbs

- distribution, and Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, no. 6, pp. 721–741, 1984.
- [9] K. Zeger, J. Vaisey, and A. Gersho, "Globally optimal vector quantizer design by stochastic relaxation," *IEEE Trans. Signal Processing*, vol. 40, no. 2, pp. 310–322, 1992.
- [10] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [11] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, D. Reidel Publishing, Boston, Mass, USA, 1987.
- [12] U. Moller, M. Galicki, E. Baresova, and H. Witte, "An efficient vector quantizer providing globally optimal solutions," *IEEE Trans. Signal Processing*, vol. 46, no. 9, pp. 2515–2529, 1998.
- [13] K. Rose, E. Gurewitz, and G. C. Fox, "Vector quantization by deterministic annealing," *IEEE Trans. Inform. Theory*, vol. 38, no. 4, pp. 1249–1257, 1992.
- [14] K. Motoishi and T. Misumi, "Fast vector quantization algorithm by using an adaptive searching technique," in *Abstracts of the IEEE International Symposium on Information Theory*, San Diego, Calif, USA, January 1990.
- [15] K. T. Lo and W. K. Cham, "Subcodebook searching algorithm for efficient VQ encoding of images," *IEE Proceedings—Vision, Image and Signal Processing*, vol. 140, no. 5, pp. 327–330, 1993.
- [16] C.-H. Lee and L.-H. Chen, "A fast search algorithm for vector quantization using mean pyramids of codewords," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 1697–1702, 1995.
- [17] X. Wu and L. Guam, "Acceleration of the LBG algorithm," *IEEE Trans. Commun.*, vol. 42, no. 2–4, pp. 1518–1523, 1994.
- [18] S. C. Tai, C. C. Lai, and Y. C. Lin, "Two fast nearest neighbor searching algorithms for image vector quantization," *IEEE Trans. Commun.*, vol. 44, no. 12, pp. 1623–1628, 1996.
- [19] L. Torres and J. Huhuet, "An improvement on codebook search for vector quantization," *IEEE Trans. Commun.*, vol. 42, no. 2–4, pp. 208–210, 1994.
- [20] N. Moayeri, D. L. Neuhoff, and W. E. Stark, "Fine-coarse vector quantization," *IEEE Trans. Signal Processing*, vol. 39, no. 7, pp. 1503–1515, 1991.
- [21] J. H. Conway and N. J. A. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. 29, no. 6, pp. 820–824, 1983.
- [22] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory*, vol. 32, no. 4, pp. 568–583, 1986.
- [23] R. M. Gray and Y. Linde, "Vector quantizers and predictive quantizers for Gauss-Markov sources," *IEEE Trans. Commun.*, vol. 30, no. 2, pp. 381–389, 1982.
- [24] A. B. Nobel and R. A. Olshen, "Termination and continuity of greedy growing for tree-structured vector quantizers," *IEEE Trans. Inform. Theory*, vol. 42, no. 1, pp. 191–205, 1996.
- [25] C. F. Barnes, S. A. Rizvi, and N. M. Nasrabadi, "Advances in residual vector quantization: a review," *IEEE Trans. Image Processing*, vol. 5, no. 2, pp. 226–262, 1996.
- [26] P. Sriram and M. Marcellin, "Image coding using wavelet transforms and entropy-constrained trellis-coded quantization," *IEEE Trans. Image Processing*, vol. 4, no. 6, pp. 725–733, 1995.
- [27] K. Zeger and A. Gersho, "A stochastic relaxation algorithm for improved vector quantizer design," *Electronics Letters*, vol. 25, no. 14, pp. 896–898, 1989.
- [28] C.-H. Hsieh, P.-C. Lu, and J.-C. Chang, "Fast codebook generation algorithm for vector quantization of images," *Pattern Recognition Letters*, vol. 12, no. 10, pp. 605–609, 1991.
- [29] K. M. Chugg, A. Anastasopoulos, and X. Chen, *Iterative Detection: Adaptivity, Complexity Reduction, and Applications*, Kluwer Academic, Reading, Mass, USA, 2001.
- [30] K. M. Chugg, K. Demirciler, and A. Ortega, "Soft information for source channel coding and decoding," in *Proc. 33rd International Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 326–331, Pacific Grove, Calif, USA, October 1999.
- [31] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-input soft-output modules for the construction and distributed iterative decoding of code networks," *European Transaction on Telecommunications*, vol. 9, no. 2, pp. 155–172, 1998.
- [32] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley, New York, NY, USA, 1991.
- [33] W. H. Equitz, "A new vector quantization clustering algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 10, pp. 1568–1575, 1989.

Kemal Demirciler received the B.S. degree (summa cum laude) as the University Valedictorian in 1996, and the M.S. and Ph.D. degrees in electrical engineering in 1998 and 2003, respectively, all from the University of Southern California (USC). His undergraduate education is supported by a CASP (Cyprus American Scholarship Program) scholarship. At USC, he received International Engineering Consortium William L. Everitt Student Award of Excellence in 1995, University Trustees' Award for Academic Excellence in 1996, Tau Beta Pi Fellowship from 1996 to 1997, Powell Foundation Doctoral Merit Fellowship from 1996 to 1999, and Katherine Harumi Contant Foundation Fellowship from 1999 to 2000. He was the Chair and the Cochair of the Integrated Media Systems Center (IMSC-NSF Engineering Research Center at USC) Student Conferences in 2000 and 2001, respectively, and the Cochair of the International Information Technology Workshops in Nicosia, Cyprus, in 2001 and 2002. He is a Member of Phi Kappa Phi, Tau Beta Pi, and Eta Kappa Nu Honor Societies, and a Member of IEEE. He is currently a researcher at the Institute of Advanced Technology Research and Development, the Eastern Mediterranean University (EMU) in Cyprus. His research interests include image and video compression and optimal resource allocation.



Antonio Ortega received the Telecommunications Engineering degree from the Universidad Politecnica de Madrid (UPM), Madrid, Spain, in 1989, and the Ph.D. degree in electrical engineering from Columbia University, New York, NY, in 1994, where he was supported by a Fulbright scholarship. In 1994, he joined the Electrical Engineering-Systems department, the University of Southern California, where he is currently an Associate Professor and Director of the Signal and Image Processing Institute. He is a Senior Member of the IEEE and a Member of ACM, and currently chairs the IEEE Signal Processing Society, Image and Multidimensional Signal Processing (IMDSP) Technical Committee. He was the Technical Program Cochair of ICME 2002, has been Associate Editor for the IEEE Transactions on Image Processing, IEEE Signal Processing Letters, and EURASIP JASP. He received the 1997 IEEE Communications Society Leonard G. Abraham Prize Paper Award and the 1999 IEEE Signal Processing Society Magazine Award. His research interests are in the areas of multimedia compression and communications. They include topics such as rate control and video transmission over packet wired or wireless networks, complexity optimized compression algorithms, and compression for recognition and classification applications.