

Reduced Complexity Deterministic Annealing for Vector Quantizer Design¹

Kemal Demirciler

Integrated Media Systems Center
Dept. of Electrical Engineering
University of Southern California
3740 McClintock Ave., #400
Los Angeles, CA 90089-2564

Email: demircil@usc.edu

Antonio Ortega

Integrated Media Systems Center
Dept. of Electrical Engineering
University of Southern California
3740 McClintock Ave., #436
Los Angeles, CA 90089-2564

Email: ortega@sipi.usc.edu

Abstract -- 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 bits/sample for Gaussian source the standard DA algorithm achieved 3.60 dB performance in 16483 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.

¹ 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).

I. Introduction

Vector quantization is a source coding technique that approximates blocks (or vectors) of input data by one of a finite number of pre-stored 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 [11]. Since distortion depends on the codebook design, vector quantizer design is a key optimization problem to determine the performance of a VQ-based system [12] – [15].

The traditionally used VQ design approach is the generalized Lloyd algorithm (GLA) also referred to as the LBG algorithm [16]. 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 [17] 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) [3], [4], or stochastic local search techniques.

An important SR technique is simulated annealing (SA) [4], 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 [2]. If sufficient computational resources are devoted, the SA algorithm is guaranteed to yield globally optimal solutions [18]. A *reduced complexity* quantizer design based on SR is proposed in [4], 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 appended with a stochastic perturbation step either be on the encoder (SR-C) or the decoder (SR-D). Another method that uses a similar randomized search technique is suggested in [19] which has an average performance comparable to SR-D but has a 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 [6]. 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. And through a careful annealing schedule it traces the global minimum as the energy surface assumes its non-convex “rugged” form with the 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 they are one-to-many. In other words, each sample vector in the training set is assigned to *all* codevectors in probability: the closer a codevector to a sample vector, the higher its probabilistic assignment to that sample vector. The DA method can construct high performance vector quantizers by avoiding local minima. However, calculation of the association probabilities for each sample vector with *all* the codevectors in each iteration, coupled with the high computational cost of evaluating the Gibbs distribution, and the slowness of the annealing process result in a very high computational complexity that limits DA’s utility for some practical applications.

Although the literature is rich in fast search vector quantization techniques, where the aim is to have low encoding complexity [21], [23], [25], [27] – [36], there is relatively less work toward low complexity VQ design techniques [4], [24], [26].

In this paper we propose a reduced complexity deterministic annealing approach for VQ design by using soft information processing with simplified assignment measures. In communications engineering the definition of soft information is “a reliability measure over the sample space of the investigated random variable.” In other words, soft information provides a measurement of how reliable it is to choose each possible signal choice [7] – [9]. 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. The reduced complexity DA techniques are developed 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 have also derived a theoretical analysis of the performance loss when using a simplified measure instead of the optimal one, and further used 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 appending the SR technique [4] to the SVQ algorithms results in further improvement in performance, where the benefit of the SR technique is less when the performance of the SVQ approaches the optimal.

This paper is organized as follows. Section II briefly introduces the standard DA method for VQ design and points out its computational complexities. In Section III, reduced complexity Gibbs distribution and low complexity soft information measures for VQ design are designed and analyzed. Optimal annealing schedules for the low complexity soft information measures are also derived. Section IV 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 V concludes the paper.

II. Vector Quantizer Design by Deterministic Annealing

In the deterministic annealing algorithm proposed by Rose et al. [6] 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 is represented by a codevector. 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 is the distribution that maximizes the entropy under the constraint (1) [6]:

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

where $|\mathfrak{R}|$ is the cardinality of the cluster set. 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 \mathbf{b} is a term that is inversely proportional to the

temperature in the annealing process. Hence, at infinite temperature, which corresponds to $\mathbf{b} = 0$, the probability associations are uniform: $P(x \in R_j) = 1/|\mathfrak{R}|$, $\forall x, j$. This means that, each sample vector x is equally assigned to all the clusters. As \mathbf{b} gets large, i.e., 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 $\mathbf{b} \rightarrow \infty$, each sample vector gets assigned exactly to one cluster, to the cluster whose representative codevector is closest to the sample vector. 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:

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

Thus, at $\mathbf{b} = 0$ all cluster representatives are at the center of mass of the training set,

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

where K is the number of sample vectors in the training set. Essentially, at $\mathbf{b} = 0$ 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 [6] starts the annealing process with the whole training set as one cluster at $\mathbf{b} = 0$, gradually increases \mathbf{b} , and re-optimizes by solving (3) at each \mathbf{b} . As \mathbf{b} is increased the probability associations start to get harder, 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 [6] is the derivation of the critical values of \mathbf{b} , denoted \mathbf{b}_c , at which these phase transitions occur. These are the optimum splitting temperatures of the clusters and 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. Note that \mathbf{b} does not control the size of the codebook; the system goes through a sequence of phase transitions until the required number of representatives is reached. During the annealing process whenever \mathbf{b} reaches \mathbf{b}_c for an existing cluster, that cluster splits into smaller clusters. In the limit $\mathbf{b} \rightarrow \infty$, the associations become hard and each sample vector is associated with one representative as in the GLA algorithm.

The work by Rose et al. [6] provides the theoretical framework explaining how the DA approach avoids local minima, and shows that through a careful annealing process it can achieve the global minimum. However, for practical applications the algorithm proposed has some drawbacks: in particular,

the annealing of the temperature has to be very slow especially in the vicinity of \mathbf{b}_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.

III. Reduced Complexity Deterministic Annealing

A. Introduction

In the proposed algorithms, called *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 each training vector is allowed to be associated in probability with only a *subset* of the codevectors. These probability associations provide a reliability measure on the set of codevectors that the training vector can be mapped to. The soft associations are functions of the relative distances of the codevectors from 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. Recall that in (2) the term \mathbf{b} determines the level of softness of the assignments, so it acts as a *softness control* factor (as \mathbf{b} increases assignments get harder). 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. Let us define a general “simple” distribution as:

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

where $\mathbf{m}(x_k, c_i)$ 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 (when $N = |\mathcal{C}|$ 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). Using a simple function, $\mathbf{m}(x_k, c_i)$ coupled with $N \ll |\mathcal{C}|$ can result in major computational gains at the expense of some reduction in performance.

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

$$c_i^* = \sum_k x_k p_0(x_k|c_i) = \frac{\sum_k x_k p(x_k) p_0(c_i|x_k)}{\sum_k p(x_k) p_0(c_i|x_k)} = \frac{\sum_k x_k p_0(c_i|x_k)}{\sum_k p_0(c_i|x_k)} \quad (6)$$

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 assignments and codevector locations is shown in Figure 1. Note that this framework is independent of the type of soft assignment used; any type of soft assignment measure that can give soft assignment values for the set of codevectors for a given sample vector can be used in this framework.

Ideally, in any annealing algorithm the annealing temperature should start at a very high temperature (theoretically at infinite) and gradually cool down to zero. However, as we have seen in the standard DA this results in a very slow convergence. In the proposed algorithms (SVQ) the temperature is not infinite at the start; we demonstrate that starting with a low temperature and with fixed (required) number of

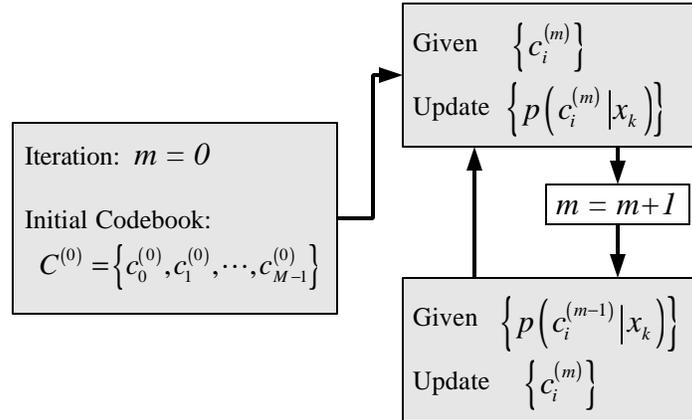


Fig. 1. The iterative procedure showing the updating of the soft assignments and the codevectors.

codevectors, it is possible to achieve near optimal performance. Starting with a low temperature means starting the algorithm with a non-convex energy surface. We show that introduction of controlled randomness as in standard SR into the iterations has the potential to improve the results due to the non-convexity of the energy surface.

B. 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,

no matter how small they are, the effect of very small soft associations on (6) and on the converged codebook is negligible. Therefore, for practical considerations, those that are very close to zero can be set to zero. At this point, a logical approach would have been to define a threshold and set all the associations below this threshold to zero. But this would only save us computational cost in (6), we would still need to calculate all the soft associations and compare them with the threshold. In order to further reduce the cost, we decided to fix the N nearest codevectors from a given sample vector, and compute the soft associations only for the closest N codevectors and set the other $|C| - N$ associations to

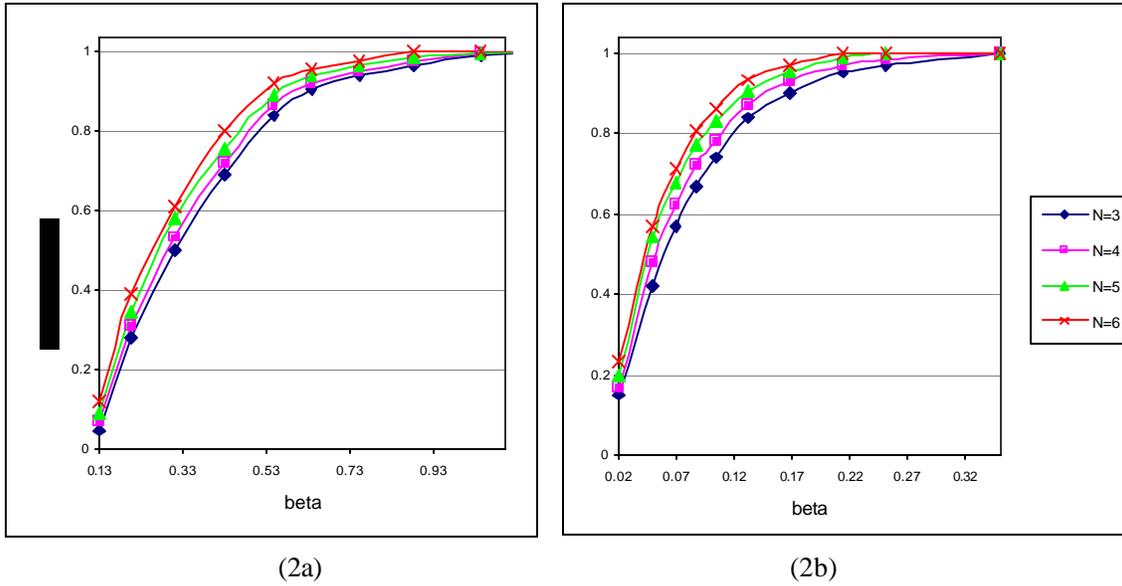


Fig. 2. Probability mass contained in the nearest N codevectors from a randomly selected sample vector, at different beta values. Codebook size = 128, vector dimension = 16. (2a) Zero-mean, unit variance Gaussian source; (2b) Gauss-Markov source, correlation coefficient 0.9.

zero. Note that we are not reducing the size of the codebook, that remains fixed. In this way, only the distances from a given sample vector to the codevectors need to be computed and the N nearest codevectors are determined. 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 are small compared to the effect they will have on the assignment probabilities. On the other hand a periodic update prevents the compounding effect of small displacements to result in 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,

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

where $d(x_k, c_i) = \|x_k - c_i\|^2$. We can assess from Figure 2 that taking all of the codevectors into consideration does not justify the required computational complexity. The figure shows the total probability mass contained in the nearest N codevectors from a randomly chosen sample vector for increasing \mathbf{b} . Each point on the graphs are found by summing the probability mass within the nearest N codevectors from each x_k , and averaging over all x_k ,

$$PM(N) = \frac{1}{K} \sum_{k=0}^{K-1} \sum_{c_i \in N(x_k, N)} p(c_i | x_k), \quad (8)$$

where K again is the size of the training set. We can observe from the figure that the range of \mathbf{b} for convergence is source dependent, but the important point to notice is that at low \mathbf{b} a considerable

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

N	Ave. SNR	Ave. CPU Time
4	3.595 dB	136 sec.
128	3.598 dB	16483 sec.

amount of the probability mass is confined within a small number of codevectors compared to the size of the codebook, and the point of convergence is at $\mathbf{b} \ll \infty$.

a. Fixed Number of Associations

One can first consider an approach with a fixed N . Experimentally we have found $N = 4$ to be a good trade-off 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 I. The loss in performance incurred by considering only the nearest 4 codevectors for each sample vector instead of the whole codebook is only 0.003 dB, which is a negligible performance difference for all practical purposes. In exchange for this negligible loss, a factor of about 120 speed-up in running time is achieved which is a highly significant complexity reduction.

The proposed algorithm is shown in Figure 3, where the iterations start with $N = 4$, $\mathbf{b} = 1/(4\mathbf{s}_x^2)$, where \mathbf{s}_x^2 is the source variance, and an initial codebook C_0 . The initial value of \mathbf{b} , $\mathbf{b} = 1/(4\mathbf{s}_x^2)$, is

found empirically to be a good starting value. At each iteration, we gradually increase \mathbf{b} ($\mathbf{k} > 1.0$), update the soft information according to (7) and re-optimize the codevector locations using (6). We can then apply the codevector perturbation of [4]. As \mathbf{b} increases the *softness* of the codevector associations

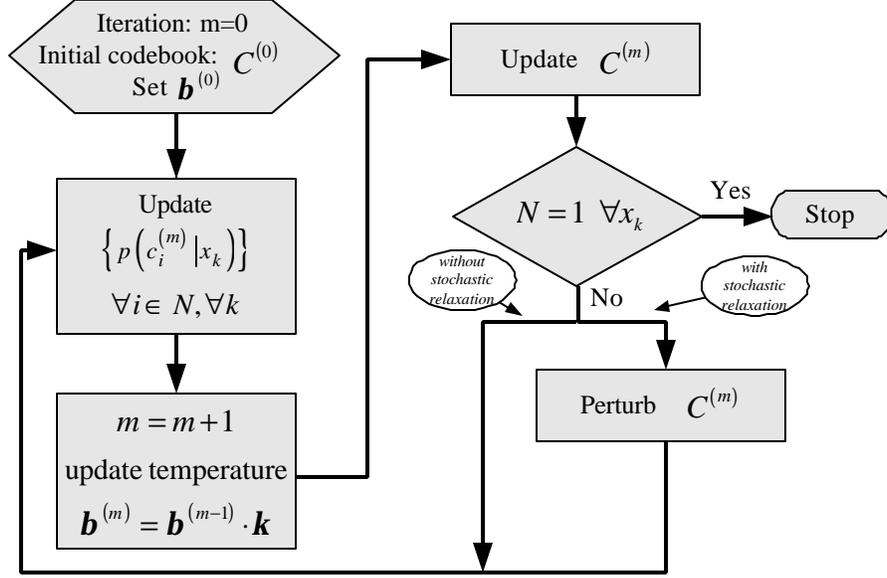


Fig. 3. Flowchart for the reduced complexity Gibbs soft assignment measure algorithm.

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.

b. Variable Number of Associations

In the above scheme we kept N fixed. When we need to design quantizers for very large codebook sizes (e.g., $|C| = 512, 1024, \dots$) it is useful to use a larger N (e.g., 10, 12, 15, ...). 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 append the following simple step to the algorithm: whenever the average probability mass of the nearest $N - 1$ codevectors, $PM(N - 1)$ (8) exceed a certain mass \mathbf{p} (typically $\mathbf{p} = 0.99$), N is reduced by one, $N = N - 1$, i.e.,

$$\text{if } PM(N - 1) = \frac{1}{K} \sum_k \sum_{c_i \in \mathcal{N}(x_k, N-1)} p(c_i | x_k) > \mathbf{p}, \text{ then } N = N - 1. \quad (9)$$

When N is large the cumulative effect of gradually decreasing the number of nearest neighbors to be taken into account results in considerable complexity reduction which was not possible in the fixed N scheme. However, it is important imperative to note that in the case of small N (e.g., $N = 4$) we may not have any computational gain or we may even increase the computational cost by using the gradual reduction scheme. This is a result of the fact that the small gain (from N being small) obtained by gradually decreasing N will be consumed by the computation of $PM(N-1)$. However, when N is large enough the reduction in computational cost obtained by reducing N surpasses the added cost of the computation of $PM(N-1)$. Therefore, when N reaches a small value, e.g., $N = 4$, the process of gradual reduction of N stops.

C. 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 decrease with increasing distance from x_k . “Inverse Euclidean distance” in a soft information measure can be defined as,

$$p(c_i | x_k) = \frac{1/d_i}{\sum_{j=0}^{N-1} 1/d_j} \quad (10)$$

The distances in (10) 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 , where all codevectors closer than R to x_k constitute the N nearest codevectors. The radius R decreases from one iteration to the next, $R^{(m)} = R^{(m-1)} \mathbf{r}$, where $0 < \mathbf{r} < 1.0$.

Another soft information measure can be defined using a triangle function centered on the considered sample vector, x_k as shown in Figure 4. The function with height $h = 1$ and a spread R_x , will contain the N codevectors within an Euclidean distance R_x from x_k . Using the fuzzy systems terminology, we can define this triangle function as the *membership function* of x_k and denote it by m_x . The soft associations are computed by using the *heights* of the membership function corresponding to the Euclidean distances of the codevectors from x_k ,

$$p(c_i | x_k) = \frac{h_i}{\sum_{j=0}^{N-1} h_j}. \quad (11)$$

The spread R_x decreases gradually in each iteration giving more and more importance to the nearer codevectors as the iterations increase. At the limit, when only one codevector stays within the nearest neighbor set, i.e., $N=1$, the soft information measure becomes hard and all the probability mass gets assigned to the nearest codevector. Note that as the spread is decreased, for some sample vectors $N=1$

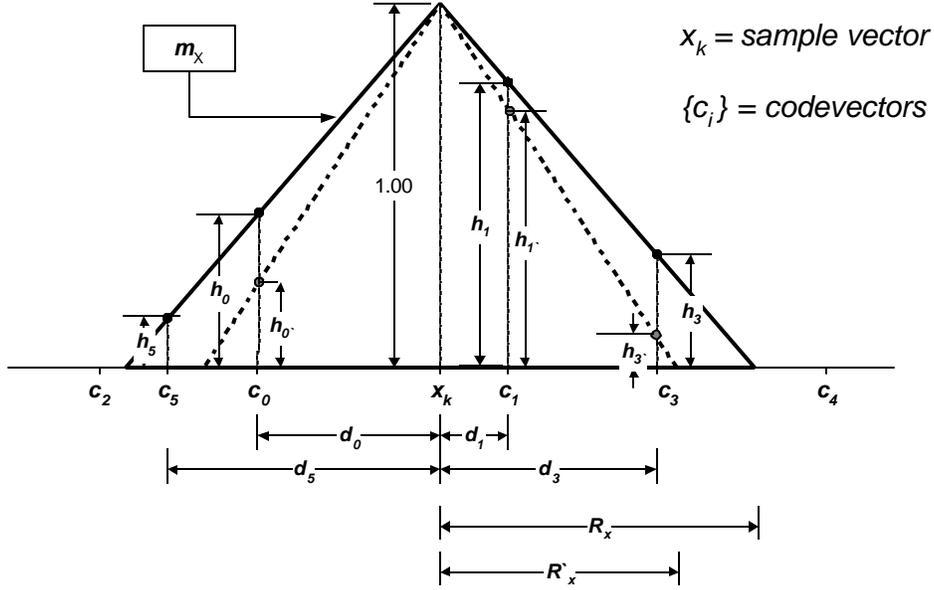


Fig. 4. Triangular membership function used as a soft information measure. Codevectors within the spread of the function comprise the nearest N codevectors for the considered sample vector.

will be reached earlier than the others since the nearest codevector distance cannot be the same for each sample vector. As the spread continues to decrease, at some point for some sample vectors,

$R_x < d(x_k, c_i) \quad \forall i$. In these cases, the algorithm assigns all the probability mass to the nearest

codevector. The spread at the m^{th} iteration is controlled by a geometric schedule as in the Gibbs case:

$$R_x^{(m)} = R_x^{(m-1)} \mathbf{r} \quad (12)$$

where \mathbf{r} is the reduction factor, $0 < \mathbf{r} < 1.0$. The soft information measure in (11) can be defined in

terms of the spread, R_x and the distances, $d_i = d(x_k, c_i)$ using triangular similarities, where the height of the triangle is $h = 1$:

$$\frac{h}{R_x} = \frac{h_i}{R_x - d_i} \Rightarrow h_i = \frac{R_x - d_i}{R_x} \quad (13)$$

Therefore, (11) becomes,

$$p(c_i | x_k) = \frac{R_x - d_i}{NR_x - \sum_{j=0}^{N-1} d_j} \quad (14)$$

This is a better soft information measure than the inverse Euclidean distance measure in (10), because it can mimic the effect of the temperature reduction in the Gibbs distribution much better. With (10) the only time the soft assignments will change as the radius is decreased is when a codevector is left out of the circle of radius R . However, using (11) the *heights* get affected by the reduction in the spread R_x as seen in (14). This is desired in order to approximate the effect of the temperature reduction in the Gibbs distribution; in other words, as the spread decreases the codevectors closer to x_k should increase their share of the soft assignment in conformity with their distances from x_k . Hence, the height-defined soft information measure (14) is a better one than the Euclidean distance-defined measure (10) in terms of mimicking the Gibbs soft measure.

The experimental results will also demonstrate that (14) is in fact a better measure than (10). Note also that the computational cost of computing one soft assignment using (10) requires $5N + 4$ flops, whereas using (14) it requires $N + 7$ flops, counting addition, subtraction and multiplication as one flop and division as four flops (N is the number of codevectors taken into computation). Hence, for $N \geq 1$: $N + 7 < 5N + 4$, implying that (14) is also less costly than (10). Recalling that an exponential computation is 8 times more costly than a basic operation (8 flops compared to one flop of operation time for a basic operation), then (7) takes $N(8+1+1) + 4 = 10N + 4$ flops, which is much larger than $N + 7$. Therefore, the height-defined triangular soft information measure is a computationally less complex distribution than the Gibbs distribution.

The algorithm for the low complexity soft information is similar to the one shown in Figure 3. In this case the temperature control is done by the spread of the triangle function. The initial spread $R_x^{(0)} = 4\mathbf{s}_x^2$ was empirically found to give good performance, where \mathbf{s}_x^2 is the variance of the training set components.

D. Optimal Temperature Schedule

In the previous section we have proposed a low complexity soft assignment measure, namely, the triangular soft information measure as a simplified way of computing the soft assignments. Although this measure will significantly reduce the computational cost of the soft assignments compared to the Gibbs soft measure, the reduction in computational cost will be at the expense of some loss in performance, since Gibbs is the optimal soft measure. However, the loss in performance can be minimized if we can find temperature reduction schedules for the low complexity measure that can follow the Gibbs \mathbf{b}

schedule such that the “distance” between the two distributions is minimized. In other words, for a given codevector the difference in the probability mass assigned by the low complexity (non-optimal) measure and the optimal Gibbs measure is as small as possible. By definition this is the minimization of the L_1 distance between the two distributions [20],

$$\|p_0(c|x) - p_G(c|x)\|_1 = \sum_i |p_0(c_i|x) - p_G(c_i|x)|. \quad (15)$$

We would like to find the spread reduction schedule R_x for a given Gibbs \mathbf{b} schedule that minimizes (15). But note that minimizing (15) is equivalent 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 [20] that,

$$D(p_0(c|x)\|p_G(c|x)) \geq \frac{1}{2\ln 2} \|p_0(c|x) - p_G(c|x)\|_1^2 \quad (16)$$

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, Appendix A provides 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 them, hence, minimizing the relative entropy minimizes 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):

$$\begin{aligned} \text{var}_G &= \mathbb{E}_G\{z^2\} - (\mathbb{E}_G\{z\})^2 \\ &= \frac{\int_{z=0}^{\infty} z^2 e^{-bz^2} dz}{\int_{z=0}^{\infty} e^{-bz^2} dz} - \left(\frac{\int_{z=0}^{\infty} z e^{-bz^2} dz}{\int_{z=0}^{\infty} e^{-bz^2} dz} \right)^2 = \frac{1}{2b} - \left(\sqrt{\frac{1}{p \cdot b}} \right)^2 = \frac{p-2}{2p \cdot b} \end{aligned} \quad (17)$$

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

Equating (17) and (18), and solving for R_x , we get,

$$R_x = \sqrt{\frac{9(\mathbf{p} - 2)}{\mathbf{p} \cdot \mathbf{b}}}. \quad (19)$$

Hence, using (19) we can obtain a schedule for R_x given a schedule for \mathbf{b} . We need to verify that the relationship in (19) minimizes the relative entropy.

We have used the set up in Figure 5 to show that for a given \mathbf{b} for the Gibbs distribution, the spread R_x obtained by (19) for the triangle distribution minimizes the relative entropy. In the figure there are a set of L codevectors at increasing distances from a sample vector x . For each \mathbf{b} in a set $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_m\}$, the soft Gibbs assignments of the codevectors are computed using the Gibbs soft information measure (7) with $N = L$. Then, through an exhaustive search, the spread R_x that gives the

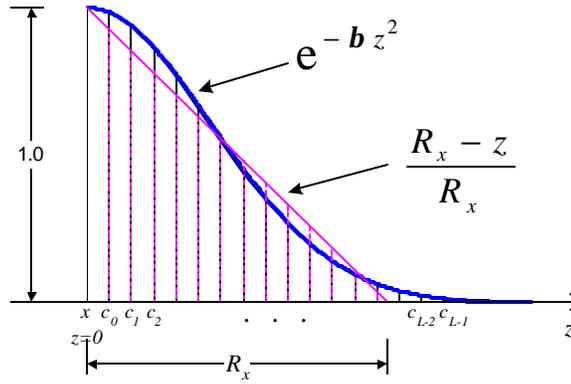


Fig. 5. An instance of the Gibbs function with parameter β and an instance of the triangle function with parameter R_x is shown. There are L codevectors at increasing distances from sample vector x .

soft assignments using (14) 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 6 by the solid line. This is compared with the result obtained using (19) by the dashed curve. We can see that the derived relation in (19) can approximate well the minimum relative entropy curve, and hence the best R_x schedule for a given \mathbf{b} 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 size 128 and 256 (for details on experiments see the Experimental Results section). The results are shown in Table II. Of the two schedules for triangular soft information measure, the first one is the geometric spread reduction given in (12), and the second one is obtained using (19) and the Gibbs schedule (referred to as Gibbs guided spread reduction in Table II). Observe from the results in the Table

It that the performance of the triangular soft information measure using the Gibbs guided spread reduction schedule outperformed the geometric spread reduction in same number of iterations (running time). Therefore, for a given \mathbf{b} – schedule the relation in (19) provides a better R_x – schedule than the

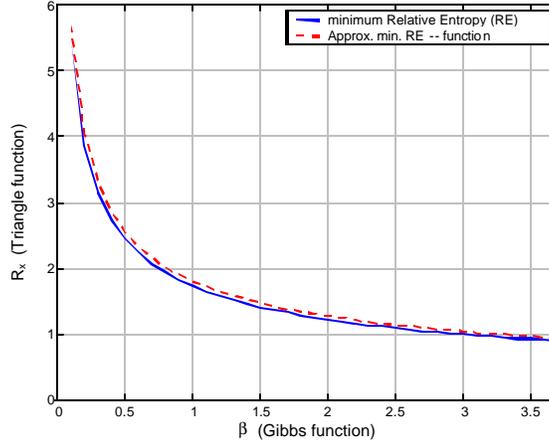


Fig. 6. Plot shows the minimum relative entropy between the triangular soft measure and the Gibbs soft measure at various spread R_x and β pairs. The solid curve is obtained by an exhaustive search for R_x that gives the minimum relative entropy for a given value of β . The dashed curve is obtained using the derived relationship between R_x and β to give the minimum relative entropy.

geometric reduction. Note that \mathbf{b} – schedule is itself geometric, $\mathbf{b}^{(m)} = \mathbf{b}^{(m-1)} \cdot \mathbf{k}$. But since the Gibbs soft information measure is the optimal measure, following the \mathbf{b} – schedule in a simple soft information

TABLE II: Comparing the geometric and 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 dimensions 16.

Reduced Complexity Gibbs Soft Information Measure ($\mathbf{k} = 1.005$).		Low Complexity Soft Information Measure – Triangular. Geometric spread reduction ($\mathbf{r} = 0.995$).		Low Complexity Soft Information Measure – Triangular. Gibbs guided spread reduction.		
Codebook Size	Ave. SNR	Ave. CPU time	Ave. SNR	Ave. CPU time	Ave. SNR	Ave. CPU time
128	3.595 dB	136 sec.	3.392 dB	91 sec.	3.411 dB	91 sec.
256	5.210 dB	329 sec.	4.919 dB	232 sec.	4.952 dB	232 sec.

measure that approximates the Gibbs measure, increases the simple soft information measure’s performance as demonstrated above. Note also that to obtain the \mathbf{b} – schedule the Gibbs algorithm need not be run, it can be obtained using $\mathbf{b}^{(m)} = \mathbf{b}^{(m-1)} \cdot \mathbf{k}$, $\mathbf{k} > 1.0$.

IV. 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 were two cases of first order Gauss-Markov sources, one with correlation coefficient $\mathbf{a}_0 = 0.0$ (uncorrelated source) and the other with $\mathbf{a}_0 = 0.9$ (correlated source). We blocked 16384 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 [22] 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 from 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 the appended “a” means without stochastic perturbation (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 [6].
5. SR-D: Vector quantizer design using the reduced complexity decoder perturbation algorithm [4].
6. GLA: Vector quantizer design using the generalized Lloyd algorithm [16].

In the cases where PNN initialization is not used, for each algorithm, except VQ-DA, the average performances are computed for 20 different initial codebooks, where for each codebook design the same set of initial codebooks are used, allowing us to compare the average performances of the different algorithms. 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:

$PSNR = 10 \cdot \log_{10}(255/D)$ 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 [4], and the GLA was run until convergence.

A. 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 7 and 8. 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

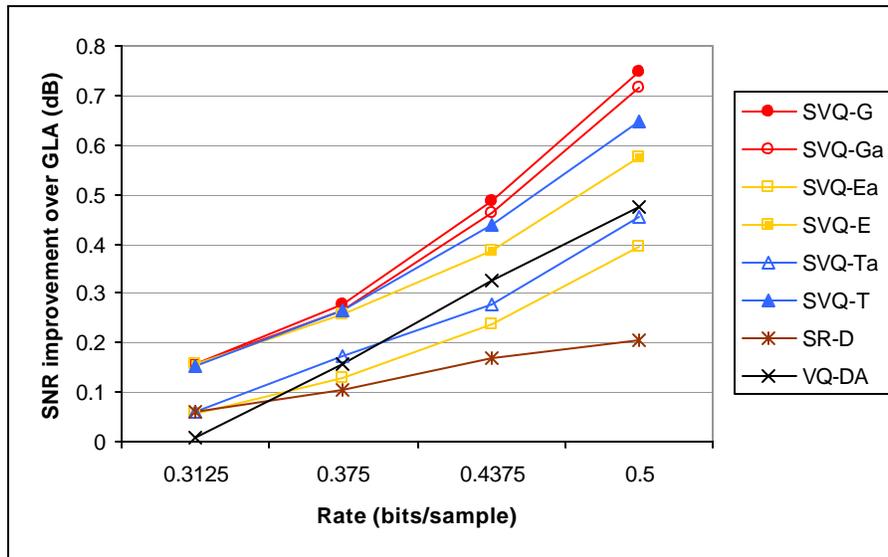


Fig. 7. Improvements over GLA for Gaussian source; vector dimension = 16 samples/vector.

[4]. 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 non-stochastic 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.

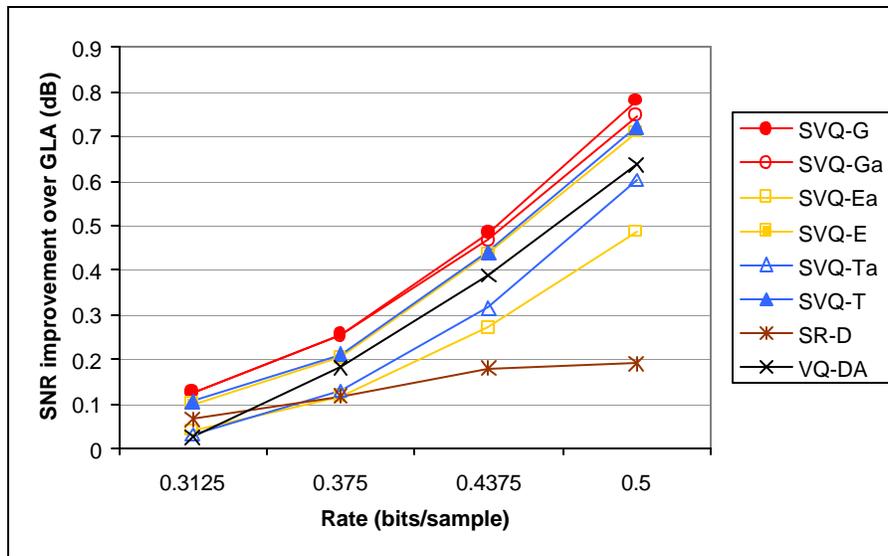


Fig. 8. Improvements over GLA for Gauss-Markov source; vector dimension = 16 samples/vector.

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 optimal as shown in [6], 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.

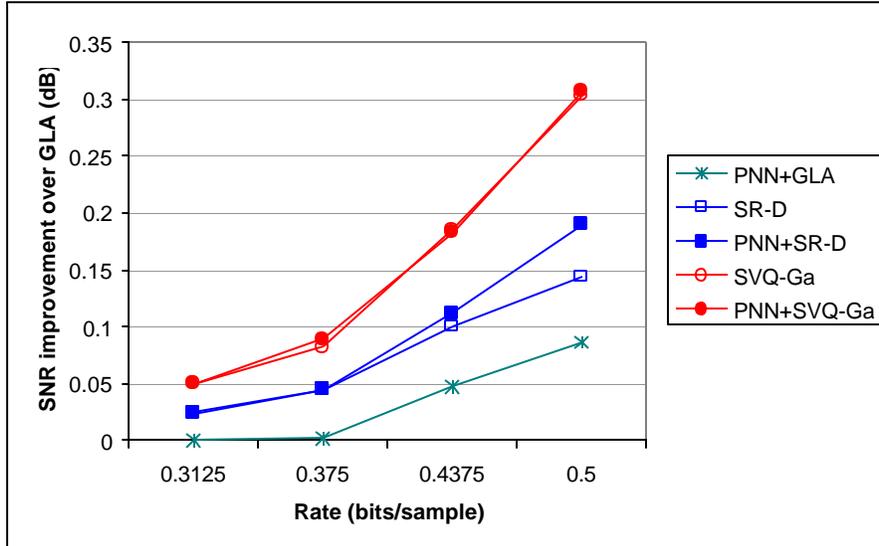


Fig. 9. Showing the effect of PNN initialization as improvement over GLA. Source is Gaussian, vector dimension = 16 samples/vector.

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 [22]. In Figure 9 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 7. 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 [22]. The results presented in Figure 9 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 [22] that the fast-PNN algorithm increased the coding error by 0.4 – 0.6dB 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 were 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. And since SVQ-Ga performs the same with and without the initialization, then the SVQ-Ga algorithm outperforms GLA and SR-D in less running time.

B. 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 10 and 11. In Figure 10 the performance improvement over GLA and in Figure 11 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 bits/sample),

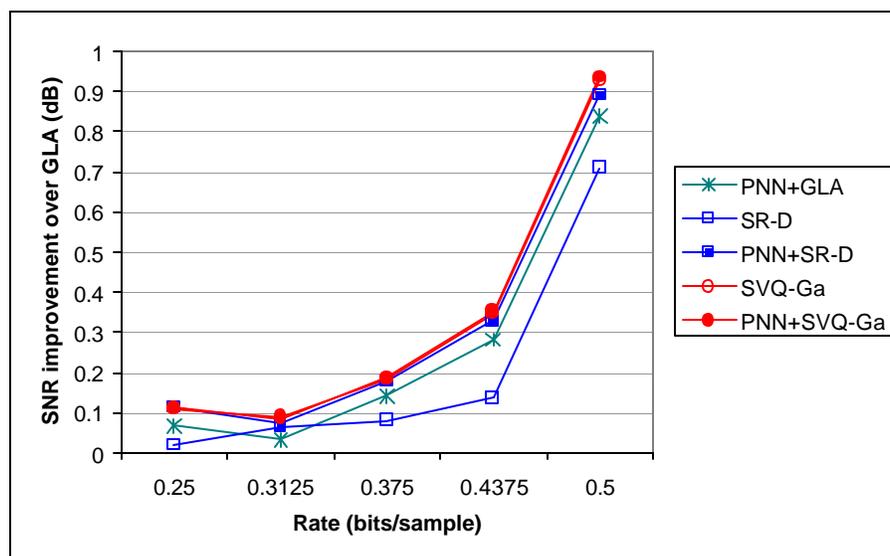


Fig. 10. Improvements over GLA for human speech source sampled at 8kHz; vector dimensions = 16 samples/vector.

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 bits/sample for GLA and SR-D, respectively, after initialization. Therefore, as in the Gaussian source, the SVQ-Ga renders the initialization unnecessary.

C. Image Source

The last source considered was the image source, where the results are shown in Figure 12 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

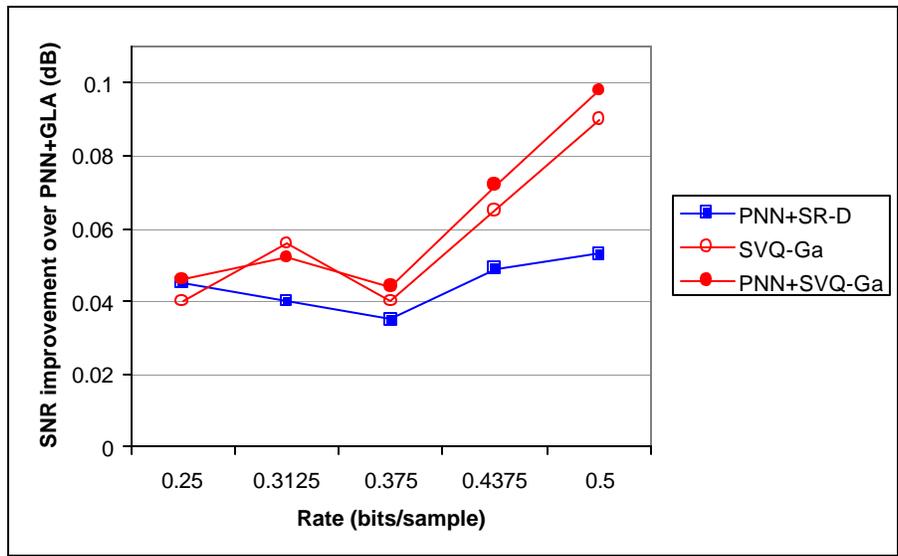


Fig. 11. Improvements over PNN initialized GLA for human speech source sampled at 8kHz; vector dimensions = 16 samples/vector.

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.

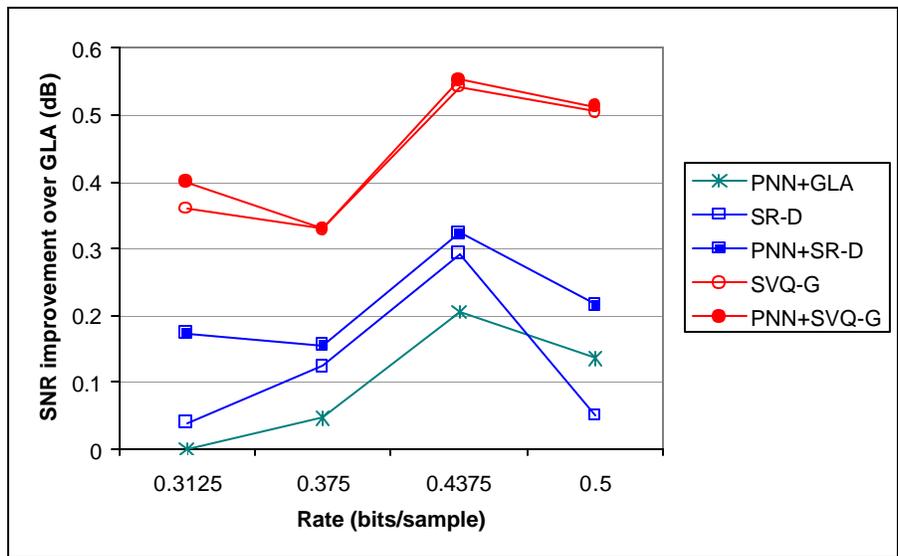


Fig. 12. Improvements over GLA for the coding of image source "Lena." Vector dimension = 16 samples/vector (corresponding to 4x4 blocks).

V. Conclusion

In this paper we have designed reduced/low complexity methods for deterministic annealing (DA) for the vector quantizer design problem, which we named as *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 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 are shown to be drastically reduced. Using the same annealing temperature the SVQ algorithms run by over 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 A

We have proposed two low complexity soft assignment measures, the triangular soft information measure and the multi-triangular soft information measure, as simplified ways of computing the soft assignments for the VQ design problem using deterministic annealing. Although these measures significantly reduce the computational cost of the soft assignments compared to the optimal Gibbs soft measure, this improvement in computational cost comes in exchange for some loss in performance since Gibbs is the optimal soft measure. In this section we derive the penalty paid in distortion for using the simplified soft measures instead of the optimal one at a given system entropy (softness).

For a given soft assignment measure (conditional probability), $p(c|x)$ we have the expected distortion and the average mutual information in (20) and (21), respectively.

$$D(p(c|x)) = \sum_x p(x) \sum_c p(c|x) d(x,c), \quad (20)$$

$$I(X;C) = \sum_x p(x) \sum_c p(c|x) \log \frac{p(c|x)}{p(c)}. \quad (21)$$

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

$$P_I = \{p(c|x) : I(X;C) \leq I\}, \quad (22)$$

and hence, for fixed I ,

$$D(R) = \min_{p(c|x) \in P_I} D(p(c|x)). \quad (23)$$

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, $p_0(c|x)$ to be defined as,

$$p_0(c|x) = p_G(c|x) + \Delta p(c|x) \quad \forall c, x. \quad (24)$$

We require two conditions to be satisfied on (24):

$$\sum_c \Delta p(c|x) = 0 \quad \forall x, \quad (25)$$

$$\Delta I = I_0(X;C) - I_G(X;C) = 0. \quad (26)$$

The condition in (25) is required so that $p_0(c|x)$ in (24) is a valid pmf, $\sum_c p_0(c|x) = 1$, and the condition in (26), the difference in the mutual information to be zero, is required so that $p_0(c|x) \in P_I$. 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 (25) and (26). The situation is depicted in Figure A.1, and a real simulation result is shown in Figure A.2. We will start by expanding, $\Delta I = I_0(X;C) - I_G(X;C)$,

$$\Delta I = \sum_x p(x) \sum_c p_0(c|x) \log \frac{p_0(c|x)}{p_0(c)} - \sum_x p(x) \sum_c p_G(c|x) \log \frac{p_G(c|x)}{p_G(c)}. \quad (27)$$

Substitute $p_G(c|x) = p_0(c|x) - \Delta p(c|x)$ from (24) into (27):

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

Simplifying the above expression, we get,

$$\Delta I = \sum_x p(x) \sum_c \Delta p(c|x) \log \frac{p_G(c|x)}{p_G(c)} + D(p_0(c|x) \| p_G(c|x)) - D(p_0(c) \| p_G(c)). \quad (28)$$

Since $p_G(c|x)$ is the optimal distribution, $p_G(c|x) = (p_G(c) e^{-b d(x,c)}) / (\sum_{c'} p_G(c') e^{-b d(x,c')})$, where with $p_G(c)$ uniform we get the Gibbs distribution, we substitute it in the first term in (28),

$$\begin{aligned} \sum_x p(x) \sum_c \Delta p(c|x) \log \frac{p_G(c|x)}{p_G(c)} &= \sum_x p(x) \sum_c \Delta p(c|x) \log \left(\frac{1}{p_G(c)} \frac{p_G(c) e^{-b d(x,c)}}{\sum_{c'} p_G(c') e^{-b d(x,c')}} \right) \\ &= \sum_x p(x) \sum_c \Delta p(c|x) \left(-\frac{b d(x,c)}{\ln 2} - \log \sum_{c'} p_G(c') e^{-b d(x,c')} \right) \end{aligned}$$

Therefore,

$$\begin{aligned} &\sum_x p(x) \sum_c \Delta p(c|x) \log \frac{p_G(c|x)}{p_G(c)} \\ &= \frac{-b}{\ln 2} \underbrace{\sum_x p(x) \sum_c \Delta p(c|x) d(x,c)}_{=\Delta D} - \sum_x p(x) \left(\log \sum_{c'} p_G(c') e^{-b d(x,c')} \right) \underbrace{\sum_c \Delta p(c|x)}_{=0 \quad \forall x} \quad (29) \\ &= \frac{-b}{\ln 2} \Delta D \end{aligned}$$

Substituting (29) into (28), and using the condition in (26) that $\Delta I = 0$, we get,

$$\Delta I = \frac{-\mathbf{b}}{\ln 2} \Delta D + D(p_0(c|x) \| p_G(c|x)) - D(p_0(c) \| p_G(c)) = 0. \quad (30)$$

Finally, the difference in the expected distortion is,

$$\Delta D = \frac{\ln 2}{\mathbf{b}} \left[D(p_0(c|x) \| p_G(c|x)) - D(p_0(c) \| p_G(c)) \right]. \quad (31)$$

Note that for large vector dimensions [20], $-D(p_0(c) \| p_G(c)) \cong D(p_G(c) \| p_0(c))$. Hence, for large dimensions the penalty paid in terms of distortion at a given system entropy for using the non-optimal soft assignment measure, $p_0(c|x)$ instead of the optimal one, $p_G(c|x)$, is,

$$\Delta D = \frac{\ln 2}{\mathbf{b}} \left[D(p_G(c) \| p_0(c)) + D(p_0(c|x) \| p_G(c|x)) \right]. \quad (32)$$

Note also that $p_G(c)$ and $p_0(c)$ are dependent on $p_G(c|x)$ and $p_0(c|x)$, respectively. Hence, minimizing the conditional relative entropy, $D(p_0(c|x) \| p_G(c|x))$ in (32) minimizes ΔD .

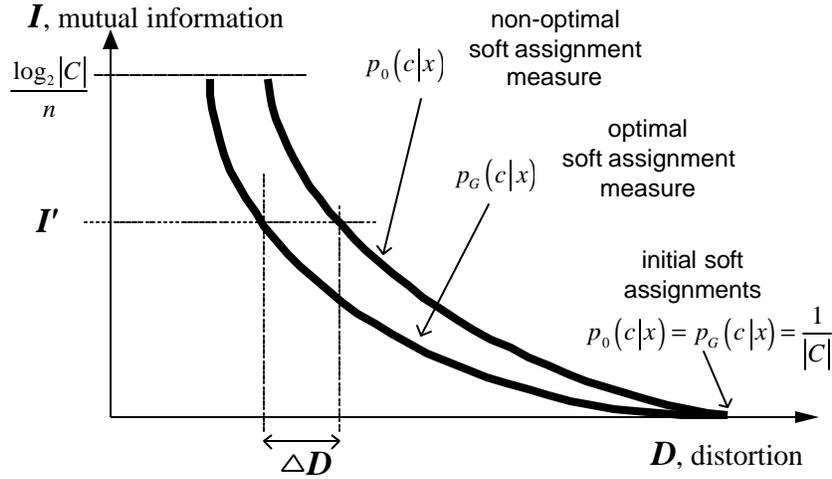


Fig. A1. Convergence of the optimal and a non-optimal soft assignment measures (distributions). Starting with equal, uniform soft assignments, the optimal soft assignment measure achieves a lower distortion than the non-optimal soft measure. At a given system entropy level, I' the difference in distortion is shown as ΔD . The term n is the vector dimension and $|C|$ is the size of the codebook.

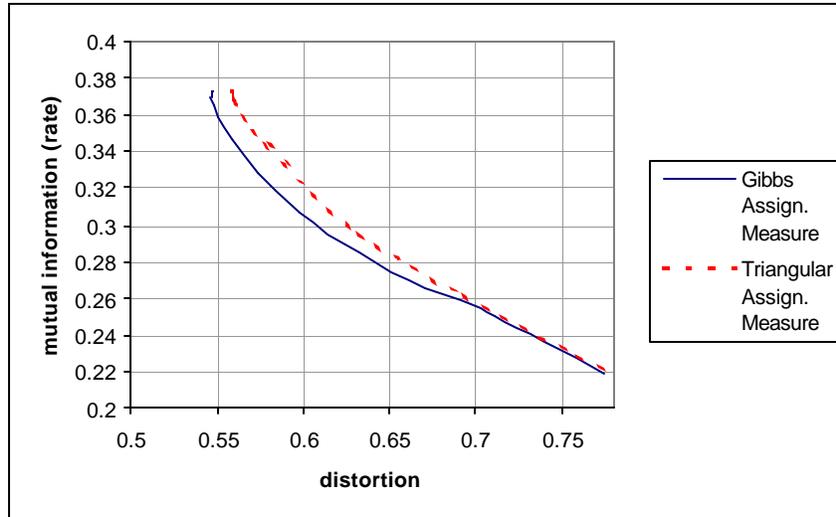


Fig. A2. Plot showing the convergence of two experiments, one with Gibbs soft assignment measure and the other with Triangular soft assignment measure, for the design of a codebook of size 64, vector dimension 16, and the source type zero-mean, unit variance Gaussian.

Acknowledgments

We would like to thank Prof. Z. Zhang and Prof. K. M. Chugg for their feedback in the derivation of the performance bound for the low complexity measures.

References

- [1] S. Kirkpatrick, C.D. Gellat, Jr., and M.P. Vecchi, "Optimization by simulated annealing," *Science*, Vol. 220, pp. 671-680, 1983.
- [2] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, Vol. 21, pp. 1087 – 1092, 1953.
- [3] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images" *IEEE Trans. Patt. Anal.Mach. Intel.*, Vol. 6, pp. 721 – 741, 1984.
- [4] K. Zeger, J. Vaisey and A. Gersho "Globally optimal vector quantizer design by stochastic relaxation," *IEEE Trans. on Signal Processing*, vol. 40, No.2, Feb. 1992.
- [5] K. Rose, E. Gurewitz and G. C. Fox, "Statistical mechanics and phase transitions in clustering," *Phys. Rev. Lett.* Vol. 65, no. 8, pp. 945-948, 1990.
- [6] K. Rose, E. Gurewitz, and G. C. Fox, "Vector quantization by Deterministic Annealing," *Trans. on Inform. Theory*, vol. 38, no. 4, pp. 1249-1257, July 1992.

- [7] K. M. Chugg, A. Anastasopoulos and X. Chen, *Iterative Detection: Adaptivity, Complexity Reduction, and Applications*, Kluwer Academic Press, 2001.
- [8] K. M. Chugg, K. Demirciler and A. Ortega “Soft information for source channel coding and decoding,” *Proc. of thirty-third International Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, October 1999.
- [9] 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 Trans. Commun.*, March/April 1998.
- [10] K. Rose, “Deterministic annealing for clustering, compression, classification, regression, and related optimization problems,” *Proc. IEEE*, vol. 86, no. 11, Nov. 1998.
- [11] A. Gersho, R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer International Series in Engineering and Computer Science, 1992.
- [12] R. L. Joshi, H. Jafarkhani, J. H. Kasner, T. R. Fischer, N. Farvardin, M. W. Marcellin, and R. H. Bamberger, “Comparison of different methods of classification in subband coding of images,” *IEEE Trans. Image Processing*, vol. 6, pp. 1473-1486, Nov. 1997.
- [13] E. Simoncelli and J. Portilla, “Texture characterization via joint statistics of wavelet coefficient magnitudes,” *Proc. 5th IEEE Int. Conf. Image Processing*, vol. I, pp. 4-7, Chicago, IL, Oct. 1998.
- [14] R. R. Coifman and M. V. Wickerhauser, “Entropy-based algorithms for best basis selection,” *IEEE Trans. Inform. Theory*, vol. 38, pp. 713-718, Mar. 1992.
- [15] B.-H. Juang, W. Chou and C.-H. Lee, “Minimum classification error rate methods for speech recognition,” *IEEE Trans. Speech Audio Processing*, vol. 5, pp. 257-265, May 1997.
- [16] Y. Linde, A. Buzo and R. M. Gray, “An algorithm for vector quantizer design,” *IEEE Trans. on Communications*, vol. COM-28, pp. 84-95, Jan. 1980.
- [17] L.A. Rastrigin, *Random Search in Optimization Problems for Multiparameter Systems*, Riga Latvia. “Zinatne” Publishing House, 1965.
- [18] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Boston, D. Reidel Publishing, 1987.
- [19] U. Moller, M. Galicki, E. Baresova and H. Witte, “An efficient vector quantizer providing globally optimal solutions,” *IEEE Trans. on Signal Processing*, vol. 46, No. 9, Sept. 1998.
- [20] Thomas M. Cover and Joy A Thomas, *Elements of Information Theory*, Wiley, 1991.
- [21] K. Motoishi and T. Misumi, “Fast vector quantization algorithm by using an adaptive searching technique,” *Abstracts of the IEEE Int’l Symp. Inform. Theory*, San Diego, CA Jan. 1990.
- [22] W.H. Equitz, “A new vector quantization clustering algorithm,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 1568-1575, October 1989.

- [23] K. T. Lo and W. K. Cham, "Subcodebook searching algorithm for efficient VQ encoding of images," *IEE Proc.-Vis. Image Signal Process.*, vol 140, pp. 327-330, Oct. 1993.
- [24] K. Zeger and A. Gersho "A stochastic relaxation algorithm for improved vector quantizer design," *Electronics Letters*, vol 25, pp. 896-898, July 1989.
- [25] C.-H. Lee and L.-H. Chen, "A fast search algorithm for vector quantization using mean pyramids of codewords," *IEEE Trans. Commun.*, vol. 43, pp. 1697-1702, Feb./Mar./Apr. 1995.
- [26] C. H. Hsieh, P. C. Lu and J. C. Chang, "Fast codebook generation algorithm for vector quantization of images," *Pattern Recognition Lett.*, vol. 12, pp.605-609, 1991.
- [27] X. Wu and L. Guam, "Acceleration of the LBG algorithm," *IEEE Trans. Commun.*, vol. 42, pp. 1518-1523, Feb./Mar./Apr. 1994.
- [28] 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, pp. 1623-1628, Dec. 1996.
- [29] L. Torres and J. Huhuet, "An improvement on codebook search for vector quantization," *IEEE Trans. Commun.*, vol. 42, pp. 208-210, Feb./Mar./Apr. 1994.
- [30] N. Moayeri, D. L. Neuhoff and W. E. Stark, "Fine-coarse vector quantization," *IEEE Trans. Signal Process.*, vol. 39, pp. 1503-1515, July 1991.
- [31] J. H. Conway and N. J. A. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. 29, pp. 820-824, Nov. 1983.
- [32] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform Theory.*, vol. 32, pp. 568-583, July 1986.
- [33] R. M. Gray and Y. Linde, "Vector quantizers and predictive quantizers for Gauss-Markov sources," *IEEE Trans. Comm.*, vol. 30, pp. 381-389, Feb. 1982.
- [34] A. B. Nobel and R. A. Olshen, "Termination and continuity of greedy growing for tree-structured vector quantizers," *IEEE Trans. Inform. Theory.*, vol. 42, pp. 191-205, Jan. 1996.
- [35] C. F. Barnes, S. A. Rizvi and N. M. Nasrabadi, "Advances in residual vector quantization: a review," *IEEE Trans. Image Processing*, vol. 5, pp. 226-262, Feb. 1996.
- [36] P. Sriram and M. Marcellin, "Image coding using wavelet transforms and entropy-constrained trellis-coded quantization," *IEEE Trans. Image Processing*, vol. 4, pp. 725-733, June. 1995.