Fast Encoder Optimization for Multi-resolution Scalar Quantizer Design

Sorina Dumitrescu

Abstract- The design of optimal multi-resolution scalar quantizers using the generalized Lloyd method was proposed by Brunk and Farvardin for the case of squared error distortion. Since the algorithm details heavily rely on the quadratic expression of the error function, its extension to general error functions faces some challenges, especially at the encoder optimization step. In this work we show how these challenges can be overcome for any convex difference distortion measure, under the assumption that all quantizer cells are convex (i.e., intervals), and present an efficient algorithm for optimal encoder partition computation. The proposed algorithm is faster than the algorithm used by Brunk and Farvardin. Moreover, it can also be applied to channel-optimized and to multiple description scalar quantizer design with squared error distortion, and it outperforms in speed the previous encoder optimization algorithms proposed for these problems.

Keywords- Convex difference distortion, encoder optimization, generalized Lloyd algorithm, Multi-resolution scalar quantizer.

I. Introduction

Successively refinable coders, also known as progressive or rate-scalable, provide embedded or layered bitstreams. The base layer can be decoded to a coarse reconstruction of the source, and by decoding any additional layer the reconstruction quality improves. Such coding strategies have received growing attention lately due to their ability to easily adapt the source rate to varying channel conditions. A successively refinable coder offers a graceful degradation in reconstruction quality when channel conditions deteriorate,

The author is with the Department of Electrical and Computer Engineering McMaster University, Hamilton, ON, Canada L8S 4K1; e-mail: sorina@mail.ece.mcmaster.ca

as opposed to a non-scalable code which may fail abruptly when the effective transmission rate drops below the target bit rate.

A multi-resolution scalar quantizer (MRSQ) is a successively refinable coder based on scalar quantization. Specifically, an MRSQ of L refinement stages is essentially a sequence of L scalar quantizers $\mathbf{Q} = (Q_1, Q_2, \cdots, Q_L)$, whose encoder partitions are embedded. In other words, the encoder partition of Q_i is a refinement of the encoder partition of Q_{i-1}. The problem of optimal MRSQ design can be formulated as the minimization of a weighted sum of the distortions of the L component quantizers, subject to a constraint on each quantizer's rate. When each weight is the probability of the MRSQ to operate at the corresponding refinement stage, the cost function has the meaning of the expected distortion of the signal reconstruction. This problem was first addressed in [2] for the fixed-rate case, and in [16] for the entropy-constrained case. In the aforementioned work, the optimal MRSQ is assumed to have convex cells (i.e., intervals of the real line) and the squared error is considered as a distortion measure. The proposed solution algorithm is a generalization of Lloyd's method for optimal single resolution scalar quantizer design [18], which can only guarantee a local optimum. Such an approach was also used for the design of multi-resolution vector quantizers in [8]. Globally optimal solutions for the MRSQ design problem under the cell convexity constraint were proposed in [19], [24], [3], [4], [20] using combinatorial approaches. The complexity of globally optimal MRSQ design without the cell convexity restriction was also discussed in [11] and fast near-optimal algorithms were proposed in [10], both for the case of discrete sources. An algorithm for adaptive MRSQ design with convex cells was introduced in [15].

The generalized Lloyd method is a popular approach in source coders and joint source-channel coders design. Algorithms based on this approach are generally easier to implement than the combinatorial algorithms. They alternatively optimize the decoder and the encoder, when the other component is fixed. Since the sequence of expected distortions is non-increasing, the algorithm eventually converges to a locally optimal solution. As proved in [5], [7], for fixed-rate MRSQ's with convex cells and convex difference distortion measure, there is a large class of sources (including sources with log-concave probability distribution functions) for which the locally optimal solution is unique, and thus globally optimal.

Motivated by the above observations we revisit the fixed-rate MRSQ design via the generalized Lloyd method. Like most of previous work, we restrict our attention to MRSQ's with convex cells and address the optimal design within this class. While the authors of [2] address only the case of squared distance distortion, we consider any convex difference distortion measure. The use of this more general distortion metric poses some challenges in the encoder optimization step. The algorithm used in [2] to optimize the encoder is borrowed from previous work on multiple description scalar quantizer (MDSQ) design [22], and it heavily relies on the quadratic expression of the distortion. Therefore it is not clear if it can be generalized to the case of any convex difference distortion. We prove that indeed, for the latter case, the characterization of the optimal encoder partition is similar in spirit to that given in [22], and present an efficient algorithm for its computation. Furthermore, we show that the proposed algorithm can be used to solve the encoder optimization step in the design of a wider class of coding systems based on scalar quantization, such as channel-optimized scalar quantizers ([17],[12]) and MDSQ's ([22], [23]), with squared distance distortion, outperforming in speed the existing solutions. The fast algorithm for cell boundaries computation in the optimized encoder partition, proposed in this work, was first presented in [6] for the problem of MDSQ design with squared distance distortion.

The condition of cell convexity is not restrictive in fixed-rate single resolution quantizer design [13]. Unfortunately, it may be so in the case of multi-resolution counterpart. In [9], [20] an example of finite alphabet source with mean squared error (MSE) distortion, was given for which the optimal MRSQ with two refinement stages must have non-convex cells (in other words, non-contiguous cells). Discrete distributions may rend non-convexity of cells in the optimal scalar quantizers even in the case of entropy-constrained single resolution quantization. Specifically, Gyorgy and Linder [14] have shown that there exist discrete distributions and an interval of rates, for which the optimal entropy-constrained scalar quantizer cannot have convex cells. On the other hand, the same work proves that such an example cannot be found when the source distribution is continuous, the number of quantizer cells is finite and the error function is non-decreasing and convex. Unfortunately, the transition from discrete to continuous distributions does not exhibit such a change of behavior with respect to cell convexity in the multi-resolution case as recently proved in [1]. The work [1] constructs a continuous probability density function (pdf) for which the optimal fixed-rate MRSQ with two refinement stages, four cells in the central partition and MSE distortion, cannot have convex cells. However, it is worth to mention that the pdf used as a counterexample in [1] has a non-convex support consisting of four small and disjoint intervals of the real line, and thus, it does not resemble too much continuous distributions commonly encountered in practical signal processing applications.

For MRSQ positive results in favor of convexity are also available. Most notable is the case of uniform distribution for which the optimal MRSQ has convex cells (using uniform partitions in each component quantizer). For general distributions, Effros and Muresan proved in [9], [20] that convexity of cells in the highest resolution partition does not prevent from optimality when the distortion measure is the squared distance. Moreover, in [7] it was shown that convexity of cells is asymptotically optimal for fixed-rate

MRSQ's with the r^{th} power distortion, as the rate of each component quantizer approaches ∞. This result strongly suggests that the class of distributions for which the optimal MRSQ has convex cells, at least when the rates are sufficiently high, might contain more than just the uniform distribution. However, the questions how large this class is and, more importantly, if it contains or not the distributions encountered in practical applications, remain open problems.

The paper is structured as follows. Next section presents the necessary definitions and notations. In Section III we formulate the optimization problems to be solved at each iteration in the design algorithm based on the generalized Lloyd method. Then we present a characterization of the optimal solution for the encoder optimization problem given fixed decoder. In Section IV we propose an efficient algorithm to compute the cell boundaries in the optimal encoder partition, which is faster than existing solutions. Section V presents our approach to deal with empty cells in the optimized encoder. Section VI contains some simulation results to demonstrate the improvement in speed achieved by the proposed algorithm. Finally, Section VII concludes the paper.

II. Definitions, Notations, Problem Formulation

Let X be a random variable over the continuous alphabet $\mathcal{A} = (V, W)$, where V, W are extended real numbers. Let $p(x)$ be the probability density function of the random variable X. We assume that $p(x)$ is continuous and positive.

We consider a distortion function $d(x, y) = \rho(|x-y|)$, where $\rho : [0, \infty) \to [0, \infty)$, is a convex function with its only zero point in 0. Consequently, $\rho(\cdot)$ is continuous and strictly increasing. Additionally, we assume that for any $y \in \mathbb{R}$ the following inequality holds

$$\int_V^W \rho(|y-t|)p(t)dt < +\infty.$$

A fixed-rate scalar quantizer (SQ, for short) Q for the random variable X is a pair of two mappings: the encoder $f_Q : \mathcal{A} \to \{0,1\}^r$, for some positive integer r, and the decoder, which is a one-to-one function $g_Q : \{0,1\}^r \to \mathcal{Y} \subset \mathbb{R}$. For each symbol $x \in \mathcal{A}$, the value $g_Q(f_Q(x))$, also denoted by $Q(x)$, is called the reproduction value or reconstruction value for x, whereas $f_Q(x)$ is called the binary codeword index for x. The set $\mathcal{Y}$ of all reproduction values is called the codebook.

The quantizer generates a partition of the input alphabet $\mathcal{A}$ into the sets $C_i = \{x \in \mathcal{A} | f_Q(x) = b(i-1)\}$, $1 \le i \le 2^r$, called cells or bins, where $b(i-1)$ denotes the binary r-bit representation of integer $i-1$. The quantizer mapping Q induces a distortion $d(x, Q(x))$ between a symbol x and its reproduction $Q(x)$.

The overall reproduction quality of the quantizer Q is measured by the expected distortion:

$$D(Q) = E\{d(X, Q(X))\} = \sum_{i=1}^{2^r} \int_{C_i} d(x, g_Q(i)) p(x) dx.$$

The rate of the quantizer Q is $R(Q) = r$.

A fixed-rate multi-resolution scalar quantizer (MRSQ) of L refinement stages is a sequence of L fixed-rate scalar quantizers $\mathbf{Q} = (Q_1, Q_2, \cdots, Q_L)$ such that $R(Q_1) < R(Q_2) < \cdots < R(Q_L)$, and for each $x \in \mathcal{A}$ and $1 \le k < L$, the following holds

$$f_{Q_{k+1}}(x) \in f_{Q_k}(x)\{0,1\}^{R(Q_{k+1})-R(Q_k)}.$$

The above condition states that the binary codeword index assigned to x by quantizer Q_{k+1} is obtained by appending exactly $R(Q_{k+1}) - R(Q_k)$ bits to the end of the codeword index assigned to x by Q_k. This condition implies that each cell of Q_k is the union of $n_k = 2^{R(Q_{k+1})-R(Q_k)}$ cells of Q_{k+1}. Specifically, for any $i, 1 \le i \le 2^{R(Q_k)}$, we have:

$$C_i^k = \cup_{\ell=(i-1)n_k+1}^{in_k} C_\ell^{k+1}, \tag{1}$$

where C_i^k denotes the i-th cell of Q_k.

The performance of the MRSQ $\mathbf{Q}$ can be measured in terms of its expected distortion $\bar{D}(\mathbf{Q})$ defined as

$$\bar{D}(\mathbf{Q}) = \sum_{k=1}^{L} \omega_k D(Q_k), \tag{2}$$

where each ω_k denotes the probability that the MRSQ operates at the k-th refinement stage. Then the problem of optimal MRSQ design can be formulated as follows. Given a fixed integer L, a set of positive values ω_k, $1 \le k \le L$ summing up to 1, and L target rates $0 < R_1 < R_2 < \cdots < R_L$, design the L-refinement stage MRSQ which minimizes the expected distortion (2) subject to the constraints

$$R(Q_k) = R_k, \text{ for all } 1 \le k \le L.$$

We refer to the L-tuple of functions $\mathbf{f} = (f_{Q_1}, f_{Q_2}, \cdots, f_{Q_L})$ as the encoder of the MRSQ, and to the L-tuple of functions $\mathbf{g} = (g_{Q_1}, g_{Q_2}, \cdots, g_{Q_L})$, as the decoder of the MRSQ. Clearly, due to relation (1), the encoder is completely specified by the encoder partition at the highest resolution Q_L, which we will also refer to as the central partition. Therefore, in order to describe an MRSQ with the desired target rates, let us make the following notations $M_k = 2^{R_k}$, $M = M_L$, and $m_k = M/M_k$, for all $k, 1 \le k \le L$. We impose the condition that all component quantizers have convex cells (i.e., intervals). Then the central partition consists only of intervals, hence it is completely specified by the interval boundaries, which are called the partition thresholds. Therefore, let us denote by $x_0, x_1, \cdots, x_M$, the partition thresholds, where

$$V = x_0 \leq x_1 \leq x_2 \leq \cdots \leq x_{M-1} \leq x_M = W. \tag{3}$$

Further, let $C_i^L = (x_{i-1}, x_i]$, $1 \leq i \leq M-1$, and $C_M^L = (x_{M-1}, x_M)$. Note that relation (1) implies that for any $k, 1 \leq k \leq L-1$, we have $C_i^k = (x_{(i-1)m_k}, x_{im_k}]$, when $1 \leq i \leq M_k - 1$ and $C_{M_k}^k = (x_{(M_k-1)m_k}, x_{M_k m_k})$. Consequently, the condition of cell convexity is satisfied for any component quantizer Q_k.

Denote by y_i^k the codeword corresponding to cell C_i^k. Then the expected distortion (2) can be rewritten as

$$\bar{D}(\mathbf{Q}) = \sum_{k=1}^{L} \omega_k \sum_{i=1}^{M_k} \int_{x_{(i-1)m_k}}^{x_{im_k}} \rho(|t - y_i^k|) p(t) dt. \tag{4}$$

Note that we allow equality in (3) between consecutive thresholds, in order to account for the case when some cells become empty during the design algorithm. However, as proved in [7], for an optimal fixed-rate MRSQ of convex cells, all cells must be non-empty, therefore, at optimality, the relations in (3) will hold with strict inequalities.

III. Generalized Lloyd Algorithm for Optimal MRSQ Design

The design procedure based on the generalized Lloyd method starts from an initial MRSQ (with all cells non-empty) and proceeds in iterations as follows. Each iteration consists in three steps: I) fix the encoder and optimize the decoder; II) fix the decoder and optimize the encoder; III) resize the partition to make all cells non-empty. Obviously, the third step is necessary only if some cells become empty after step II. Note that most authors unify II) and III) in a single step. However we will treat them separately.

Step I: Decoder optimization given fixed encoder. When the encoder is fixed, the thresholds $x_i, 0 \leq i \leq M$, of the central partition are fixed. The decoder can be optimized by separately optimizing each integral in the summation of (4), i.e., by setting the codewords such that the relation

$$\int_{x_{(i-1)m_k}}^{x_{im_k}} \rho(|t - y_i^k|) p(t) dt = \min_{y \in \mathcal{A}} \int_{x_{(i-1)m_k}}^{x_{im_k}} \rho(|t - y|) p(t) dt,$$

to hold for all $1 \leq k \leq L$, and all $1 \leq i \leq M_k$.

As shown by Trushkin [21], for every $V \leq a < b \leq W$, the function $D_{a,b}(y) = \int_a^b \rho(|t-y|)p(t)dt$, defined for every $y \in (V, W)$, achieves its minimum in some unique point $\mu(a,b)$, situated inside the interval (a, b). This value is called generalized centroid. Therefore for each $1 \leq k \leq L$, and $1 \leq i \leq M_k$, we set

$$y_i^k = \mu(x_{(i-1)m_k}, x_{im_k}).$$

This also ensures that after the decoder optimization step the following property holds

$$x_{(i-1)m_k} < y_i^k < x_{im_k}, \tag{5}$$

for all $1 \leq k \leq L$, and $1 \leq i \leq M_k$.

Because the function $D_{a,b}(\cdot)$ is convex and thus unimodal, $\mu(a, b)$ can be efficiently computed using bisection search. Moreover, since for any $a < b$, $D_{a,b}(\mu(a, b))$ is a strict minimum of the function $D_{a,b}(\cdot)$, and since $p(t) > 0$ for all t, it follows that, if the decoder changes after this step then the expected distortion of the MRSQ strictly decreases.

Finally, notice that in the case of squared error distortion, the optimal reconstruction value $\mu(a, b)$ is the centroid of the interval $[a, b]$, i.e.

$$\mu(a, b) = \frac{\int_a^b tp(t)dt}{\int_a^b p(t)dt}. \tag{6}$$

Step II: Encoder optimization given fixed decoder. At this step the codewords are fixed and the encoder partition is optimized. Since each cell of any Q_k, $1 \leq k \leq L - 1$, is the union of some cells of Q_L, we can write the expected distortion as a summation of integrals over the cells of the highest resolution quantizer. Specifically, because each central partition cell C_i^L is included in the cell $C_{\lceil i/m_k \rceil}^k$ of Q_k, where $\lceil a \rceil$ denotes the smallest integer greater than or equal to a, we obtain

$$\bar{D}(\mathbf{Q}) = \sum_{i=1}^{M} \int_{x_{i-1}}^{x_i} \sum_{k=1}^{L} \omega_k \rho(|t - y_{\lceil i/m_k \rceil}^k|) p(t) dt.$$

Thus, each integer $i, 1 \leq i \leq M$, is associated an L-tuple of codewords $\mathbf{c}_i = (y_{\lceil i/m_k \rceil}^k)_{k=1}^L$. Denote $\phi_i(t) = \sum_{k=1}^L \omega_k \rho(|t - y_{\lceil i/m_k \rceil}^k|)$ for all, $1 \leq i \leq M$ and $t \in (V, W)$. Following [2], [22], notice that the encoder optimization can be performed by mapping each value $t \in (V, W)$ to the L-tuple $\mathbf{c}_i$ for which $\phi_i(t)$ is minimized. For this let us consider the sets $\mathcal{A}_i$ for $1 \leq i \leq M$, defined as follows

$$\mathcal{A}_i = \{t \in (V, W) | \phi_i(t) \leq \phi_{i'}(t), 1 \leq i' \leq M, i' \neq i\}.$$

Then the optimized encoder should satisfy

$$C_i^L =_{ae} \mathcal{A}_i, \tag{7}$$

for all $i, 1 \leq i \leq M$, where the notation $A =_{ae} B$ for two sets A, B of real numbers, represents the fact that the two sets have the characteristic functions equal almost everywhere.

However, we need caution before taking this approach due to our constraint that the MRSQ's cells be convex. In order to maintain the convexity status of the optimized encoder by using this method, the sets $\mathcal{A}_i$ must have certain properties. In the case of squared distance distortion, it is easy to see that all $\mathcal{A}_i$ are convex [2], a fact which ensures the convexity of all cells in the highest resolution partition. Moreover, as shown in [2] the relative position of sets $\mathcal{A}_i$ on the real line agrees with the increasing order of the index i (i.e., for $i < j$, $x \in \mathcal{A}_i$, $x' \in \mathcal{A}_j$, we have $x \leq x'$), a fact which implies that the cells in the lower resolution quantizers will be convex, too. However, when considering a more general error function it is not clear whether these properties will still hold. Fortunately, it can be shown that for convex difference distortion, which is the case of our study, these properties are indeed satisfied. In order to prove this result, let us first define the function $h_{i,i'} : \mathbb{R} \to \mathbb{R}$, for each pair of integers $1 \leq i < i' \leq M$, as follows $h_{i,i'}(t) = \phi_i(t) - \phi_{i'}(t)$. Using the above notation, $\mathcal{A}_i$ becomes

$$\mathcal{A}_i = \cap_{1 \leq \ell < i} \{t \in (V, W) | h_{\ell,i}(t) \geq 0\} \bigcap \cap_{i < i' \leq M} \{t \in (V, W) | h_{i,i'}(t) \leq 0\}. \tag{8}$$

A key result for our development is the following lemma, which is proved in Appendix A.

Lemma 1. For each pair of integers i, i', $1 \leq i < i' \leq M$, the function $h_{i,i'}(\cdot)$ is non-decreasing and has a unique zero point denoted by $t_{i,i'}$. Moreover, $V < t_{i,i'} < W$.

With this result at hand we are able to give a simple characterization of the sets $\mathcal{A}_i$ similar in spirit to that implicitly used in [2] (via the method of [22]) for the case of squared error distortion. For this, let us denote

$$left(\mathcal{A}_i) = \max_{\ell, 1 \leq \ell < i} t_{\ell,i}, \text{ for any } 1 < i \leq M \tag{9}$$

$$right(\mathcal{A}_i) = \min_{j, i < j \leq M} t_{i,j} \text{ for any } 1 \leq i < M. \tag{10}$$

Then equation (8) and Lemma 1 imply that

$$\mathcal{A}_1 = (V, right(\mathcal{A}_1)], \mathcal{A}_M = [left(\mathcal{A}_M), W) \text{ and} \tag{11}$$

$$\mathcal{A}_i = [left(\mathcal{A}_i), right(\mathcal{A}_i)], \text{ for } 1 < i < M, \tag{12}$$

with the convention that $[a, b] = \emptyset$ if $a > b$. Relations (11), (12) prove that each $\mathcal{A}_i$ is an interval (possibly with empty interior) or the empty set. If $\mathcal{A}_i$ is empty or consists of a single point, i.e. if $left(\mathcal{A}_i) \geq right(\mathcal{A}_i)$, we will say that the interval $\mathcal{A}_i$ is *degenerate*. Clearly, in order to construct the optimal encoder partition, only the non-degenerate intervals $\mathcal{A}_i$ are of interest. As shown in Example 1 of Section V, the case when some sets $\mathcal{A}_i$ are degenerate cannot be excluded.

Let us denote by $\mathcal{I}$ the set of integers i with $1 \leq i \leq M$, such that $\mathcal{A}_i$ is non-degenerate. Let M' be the size of $\mathcal{I}$, and let $j_1, j_2, \cdots, j_{M'}$ be the integers in $\mathcal{I}$ ordered in increasing order. It is easy to see that $\mathcal{A}_1$ and $\mathcal{A}_M$ are always non-empty (according to the last conclusion of Lemma 1). Therefore, indices 1 and M are always in $\mathcal{I}$. Specifically, $j_1 = 1$ and $j_{M'} = M$. Also, note that for any $s, m, 1 \leq s < m \leq M'$, we have

$$right(\mathcal{A}_{j_s}) = \min_{j, j_s < j \leq M} t_{j_s,j} \leq t_{j_s,j_m} \leq \max_{\ell, 1 \leq \ell < j_m} t_{\ell,j_m} = left(\mathcal{A}_{j_m}). \tag{13}$$

In other words, the relative position of the non-degenerate intervals $\mathcal{A}_i$ on the real line agrees with the order between their indices. Furthermore, from the fact that $\bigcup_{i \in \mathcal{I}}^{M} \mathcal{A}_i = (V, W)$, combined with (13), it follows that

$$right(\mathcal{A}_{j_s}) = left(\mathcal{A}_{j_{s+1}}) = t_{j_s,j_{s+1}},$$

for every s with $1 \leq s < M'$. Consequently, the partition thresholds in the optimized encoder should be $t_{j_1,j_2}, \cdots, t_{j_s,j_{s+1}}, \cdots, t_{j_{M'-1},j_{M'}}$. Therefore, upon computing these values, in order to complete the encoder optimization step, for each $1 \leq i \leq M-1$, we set

$$x_i = t_{j_s,j_{s+1}},$$

if $j_s \leq i < j_{s+1}$. According to the above discussion this strategy ensures that

$$V = x_0 \leq x_1 \leq x_2 \leq \cdots \leq x_{M-1} \leq x_M = W, \tag{14}$$

and

$$C_i^L = (x_{i-1}, x_i] =_{ae} \mathcal{A}_i,$$

for all $i, 1 \leq i \leq M$. We conclude that this strategy optimizes the encoder and generates an MRSQ with convex cells.

Note that the equality $x_{i-1} = x_i$ may appear in (14) for some values of i if and only if the set $\mathcal{A}_i$ is degenerate. In this case the cell C_i^L obtained after this step will be empty. Such undesirable cases will be handled at step III of the generalized Lloyd algorithm, which is described in Section V.

Because for distinct i and j the sets $\mathcal{A}_i$ and $\mathcal{A}_j$ overlap in at most one point (according to (13)), and since $p(\cdot)$ is non-zero everywhere, it follows that, if the encoder partition is changed after the encoder optimization step then the expected distortion strictly decreases.

The generalized Lloyd method was used in the optimal design for a variety of coding systems based on scalar quantization. Beside the single resolution SQ and MRSQ, other examples of such coding systems are channel-optimized SQ ([17], [12]), and multiple description scalar quantizer (MDSQ) ([22],[23]), both with squared error distortion. In [17] the cell boundaries in the optimized partition, are computed according to an analytical formula derived based on the necessary conditions for locally optimum. Applying this approach to our case is equivalent to setting the boundary between cells C_i^L and C_{i+1}^L to be equal to $t_{i,i+1}$. Unfortunately, due to the possibility of some sets $\mathcal{A}_i$ to be degenerate, this approach does not guarantee that the optimality conditions (7) are satisfied for all i's, as shown in Example 1, in Section V. Farvardin and Vaishampayan pointed out in [12] this limitation of the algorithm of [17]. They further proved that the sets $\mathcal{A}_i$ are intervals and provided a characterization similar in spirit to (9)-(12). In their case, since $\rho(x) = x^2$, the function $h_{i,i'}(\cdot)$ is a linear function. Thus, Lemma 1 trivially holds and a closed form expression is available to compute the values $t_{i,i'}$. Note that a straightforward way to compute the optimal partition is to solve all maximizations and minimizations involved in (9), respectively (10). Such a procedure clearly requires $O(M^2)$ running time, assuming that the values $t_{i,i'}$ are already available. The algorithm proposed in [12] to solve this problem reduces the time complexity to $O(MM')$ based on the observation that if some sets $\mathcal{A}_\ell$ or $\mathcal{A}_j$ have been established to be degenerate, then indices ℓ and j do not need to be further considered in the maximizations and minimizations of (9), respectively (10). In [22], in the context of MDSQ design with squared error distortion, Vaishampayan shows that the optimal cells in the central MDSQ partition have a form similar in spirit to (9)-(12) and uses a different algorithm to determine their boundaries. A tight bound on the running time of this algorithm is more difficult to determine. However, we can easily find a lower bound. Specifically, its running time is $\Omega(M + M'^2)$ as shown in Appendix B. Finally, the same algorithm is used in [2] to find the optimal thresholds in the central partition in the case of MRSQ design with squared error distortion.

In the next section we present an $O(M \log M)$ algorithm to compute the optimized central partition for fixed-rate MRSQ with convex error function. We prove that its running time can be further reduced to $O(M)$ in the case of squared distance distortion. Then we show that this algorithm can be applied to channel-optimized SQ and MDSQ design problems with squared distance distortion, thus outperforming in speed the previous solutions.

IV. Fast Computation of Central Partition Thresholds

As pointed out in the previous section, the thresholds in the optimized central partition are the values $t_{j_s,j_{s+1}}$, with $1 \leq s \leq M' - 1$. Recall that $\mathcal{I}$ denotes the set of indices i such that $\mathcal{A}_i$ is a non-degenerate interval, and the elements of $\mathcal{I}$ are denoted $j_1 < j_2 < \cdots < j_{M'}$. Thus, the focus of our algorithm will be to determine the indices of non-degenerate intervals. The next key result, which is proved in Appendix A, allows us to design an efficient algorithm for this purpose.

Lemma 2. For any integers $1 \leq \ell < i < j \leq M$ we have

$$\min\{t_{\ell,i}, t_{i,j}\} \leq t_{\ell,j} \leq \max\{t_{\ell,i}, t_{i,j}\},$$

with equalities only if $t_{\ell,i} = t_{i,j}$.

The main idea of the algorithm is described next. It processes all indices j from 1 to M, in increasing order, and maintains a stack $\mathcal{S}$ of indices i smaller than j, which are candidates to belong to the set $\mathcal{I}$. In other words, if $k < j$ and k is not in the stack, it follows that the interval $\mathcal{A}_k$ is degenerate. The indices in the stack are sorted in increasing order from bottom to top. For each $i \in \mathcal{S}$, a candidate for the left boundary of $\mathcal{A}_i$ is also provided as $left(i) = t_{prev(i),i}$, where $prev(i)$ is the index below i in the stack. The next index to be pushed onto the stack is j, but before doing so, a degeneracy test is performed on the index i on top of $\mathcal{S}$, by checking if $t_{i,j} \leq left(i)$. If this inequality holds, then clearly $left(\mathcal{A}_i) \geq right(\mathcal{A}_i)$, hence $\mathcal{A}_i$ is degenerate and therefore index i is removed from the stack. Next, the value of j is kept unchanged and the degeneracy test is performed on the new index on top of $\mathcal{S}$, i.e. on $prev(i)$. The algorithm continues this way until the top of $\mathcal{S}$ fails the degeneracy check. Only then index j is pushed onto the stack and the algorithm proceeds to process index $j+1$. The algorithm stops when the current value of j equals $M+1$. At this moment the contents of $\mathcal{S}$ equals the contents of $\mathcal{I}$. The pseudocode of the algorithm is provided below.

Algorithm 1.

```
1) Create an empty stack of integers $\mathcal{S}$. Push 1 onto stack $\mathcal{S}$; SET $left(1) \leftarrow V$, and $j \leftarrow 2$.
2) SET $i \leftarrow$ top of stack $\mathcal{S}$. Compute $t_{i,j}$. IF $t_{i,j} \leq left(i)$ GO TO step 4). ELSE push $j$ onto stack
   $\mathcal{S}$ and SET $left(j) \leftarrow t_{i,j}$.
3) SET $j \leftarrow j+1$. IF $j \leq M$, go to step 2). ELSE STOP.
4) Pop $i$ from stack $\mathcal{S}$ and GO TO to 2).
```

The following proposition proves the correctness of the algorithm.

Proposition. At the end of Algorithm 1 the contents of the stack coincides with the set of indices $i, 1 \leq i \leq M$, for which $\mathcal{A}_i$ is a non-degenerate interval.

Proof. Note first that the integer 1 is never popped from the stack because the condition $t_{1,j} > V$ is always satisfied, according to Lemma 1. Therefore the stack is non-empty at the beginning of each iteration and hence the first operation of step 2) is correctly defined. It is clear that any index i in the stack is smaller than the current value of j, therefore $t_{i,j}$ (used at step 2) is correctly defined. Also it is easy to see that at any time, the integers in the stack are ordered in increasing order, from bottom to top.

Let $\mathcal{S}_f$ denote the contents of the stack at the end of the algorithm. Note that index M is on top of $\mathcal{S}_f$ because M is the last item pushed on the stack and it is never popped. In order to prove the conclusion of the proposition, we need to show that $\mathcal{S}_f = \mathcal{I}$. For establishing that $\mathcal{I} \subseteq \mathcal{S}_f$ it is enough to prove that for any $k, 1 \leq k \leq M$, which is not in $\mathcal{S}_f$, $\mathcal{A}_k$ is degenerate. For this note that index k was at some point pushed onto the stack, but later it was removed due to passing the degeneracy test. Hence $\mathcal{A}_k$ is degenerate.

Now we are left to show that $\mathcal{S}_f \subseteq \mathcal{I}$. For this we need to establish the following facts.

F1) For $i \in \mathcal{I}$, in order to compute $left(\mathcal{A}_i)$ and $right(\mathcal{A}_i)$ it is sufficient to perform the maximization in (9) , respectively minimization in (10), only over indices in $\mathcal{S}_f$, in other words, the following relations hold

$$left(\mathcal{A}_i) = \max_{1 \leq \ell < i, \ell \in \mathcal{S}_f} t_{\ell,i}, \text{ for any } i \in \mathcal{I} \cap \{2, \cdots, M\}, \tag{15}$$

$$right(\mathcal{A}_i) = \min_{i < j \leq M, j \in \mathcal{S}_f} t_{i,j}, \text{ for any } i \in \mathcal{I} \cap \{1, \cdots, M-1\}. \tag{16}$$

F2) Let $i_1, i_2, \cdots, i_q$ be the items in $\mathcal{S}_f$ ordered from bottom to top. Clearly, we have $1 = i_1 < i_2 < \cdots < i_q = M$. Then for any integers s, m such that $1 < s < m \leq q$ the following is true

$$t_{i_{s-1},i_s} \leq t_{i_{s-1},i_m} \leq t_{i_s,i_m}. \tag{17}$$

Proof of F1: If $\mathcal{A}_i$ is non-degenerate then its boundaries are thresholds in the central partition. Thus $left(\mathcal{A}_i) = t_{\ell,i}$ and $right(\mathcal{A}_i) = t_{i,j}$ for some $\ell, j \in \mathcal{I}$. Since $\mathcal{I} \subseteq \mathcal{S}_f$, it follows that $\ell, j \in \mathcal{S}_f$, which proves the claim.

Proof of F2: It is clear that for any $s, 1 \leq s < q$, we have $left(i_{s+1}) = t_{i_s,i_{s+1}}$. It is also obvious that for $1 \leq s \leq q-2$, we have $left(i_{s+1}) < t_{i_{s+1},i_{s+2}}$ because this condition had to hold when pushing i_{s+2} onto the stack. Thus, we have

$$t_{i_1,i_2} < t_{i_2,i_3} < \cdots < t_{i_{m-1},i_m} < t_{i_m,i_{m+1}} < \cdots < t_{i_{q-1},i_q} \tag{18}$$

Let us fix some $m, 3 \leq m \leq q$ and prove relation (17) using induction over s, downwards from $s = m-1$ to $s = 2$. The base case is when $s = m-1$. In this case (17) reduces to $t_{i_{m-2},i_{m-1}} \leq t_{i_{m-2},i_m} \leq t_{i_{m-1},i_m}$, which is true by (18) and Lemma 2. Let us proceed now to the inductive step. Assume that the claim holds for some s with $2 < s < m-1$, and and let us prove it for $s-1$. Then we have $t_{i_{s-2},i_{s-1}} \leq t_{i_{s-1},i_s} \leq t_{i_{s-1},i_m}$, where the first inequality is due to (18), and the second is due to the inductive hypothesis. Applying further Lemma 2 we obtain that $t_{i_{s-2},i_{s-1}} \leq t_{i_{s-2},i_m} \leq t_{i_{s-1},i_m}$, and the inductive step is completed, thus completing the proof of F2.

F2 implies that for any $m, 1 < m \leq q$, we have $t_{i_{m-1},i_m} \geq t_{i_p,i_m}$ for all $p, 1 \leq p < m$. Consequently, we have that $t_{i_{m-1},i_m} = \max_{1 \leq \ell < i_m, \ell \in \mathcal{S}_f} t_{\ell,i_m}$. Equality (15) further implies that, if $i_m \in \mathcal{I}$, then we have

$$t_{i_{m-1},i_m} = left(\mathcal{A}_{i_m}). \tag{19}$$

On the other hand, by F2 the inequality $t_{i_{m-1},i_m} \leq t_{i_{m-1},i_n}$ holds for all $n, m \leq n \leq q$, fact which leads to $t_{i_{m-1},i_m} = \min_{i_{m-1} < j \leq M, j \in \mathcal{S}_f} t_{i_{m-1},j}$. Further, by (16) it follows that, if $i_{m-1} \in \mathcal{I}$, then the following equality is valid

$$t_{i_{m-1},i_m} = right(\mathcal{A}_{i_{m-1}}). \tag{20}$$

Clearly $i_1, i_q \in \mathcal{I}$, hence we are left to prove that $i_m \in \mathcal{I}$ for all $1 < m < q$. For this let us define the set $\mathcal{B} = (V, t_{i_1,i_2}] \cup \bigcup_{m, 1 < m < q, i_m \in \mathcal{I}} [t_{i_{m-1},i_m}, t_{i_m,i_{m+1}}] \cup [t_{i_{q-1},i_q}, W)$. Notice that, from the strict inequalities in (18) it follows that set $\mathcal{B}$ coincides with the whole interval (V, W) if and only if $i_m \in \mathcal{I}$ for all $1 < m < q$. On the other hand, by (19), (20), and $\mathcal{I} \subseteq \mathcal{S}_f$, we have $\mathcal{B} = \cup_{i \in \mathcal{I} \cap \mathcal{S}_f} \mathcal{A}_i = \cup_{i \in \mathcal{I}} \mathcal{A}_i = (V, W)$, consequently the proof is completed. $\square$

Let us evaluate now the running time of Algorithm 1. In order to calculate the number of iterations over the steps 2)-4), note that at each iteration either a push or a pop is performed. The number of pushes is M since each index $j, 1 \leq j \leq M$, is pushed onto the stack exactly once. On the other hand, the number of pops is $M - M'$, since only indices i corresponding to degenerate sets $\mathcal{A}_i$ are popped, and each such index is popped only once. Consequently, the number of iterations is $2M - M'$. Note that the time complexity of an iteration is dominated by the time requirement to compute $t_{i,j}$. For a general error function $\rho(\cdot)$, a closed form may not be available for $t_{i,j}$, but since $t_{i,j}$ is the unique zero point of the non-decreasing function $h_{i,j}(\cdot)$, it can be found by using the bisection search technique. The number of iterations of the bisection search is in the worst case $\lceil \log_2((W - V)N) \rceil$, where $1/N$ is the required precision. Each iteration takes $O(L)$ time, since $O(L)$ operations are necessary to compute $h_{i,j}(t)$ for some trial value t. Assuming N to be constant, the time complexity of Algorithm 1 thus becomes $O(LM) = O(M \log M)$, where we have used the fact that $M \geq 2^L$.

The number of operations can be further reduced by replacing the degeneracy test $t_{i,j} \leq left(i)$ by the equivalent test $h_{i,j}(left(i)) \geq 0$. The equivalence follows from Lemma 1. Specifically, step 2) is replaced by 2') presented next:

2') SET $i \leftarrow$ top of stack $\mathcal{S}$. IF $h_{i,j}(left(i)) \geq 0$ GO TO step 4). ELSE compute $t_{i,j}$, push j onto stack $\mathcal{S}$ and SET $left(j) \leftarrow t_{i,j}$.

In the case when $\rho(x) = x^2$, the computation of values $t_{i,j}$ is easier. According to [2], [22], the following relation holds $h_{i,j}(t) = (\alpha_j - \alpha_i)t + \beta_i - \beta_j$ where

$$\alpha_i = \sum_{k=1}^{L} \omega_k y^k_{\lceil i/m_k \rceil}, \beta_i = \sum_{k=1}^{L} \omega_k (y^k_{\lceil i/m_k \rceil})^2, \tag{21}$$

for all $1 \le i \le M$. Thus,

$$t_{i,j} = \frac{\beta_j - \beta_i}{2(\alpha_j - \alpha_i)}. \tag{22}$$

If the quantities α_i and β_i are precomputed then the evaluation of each $t_{i,j}$ requires only a constant number of operations. The computation of all α_i and β_i using relations (21) amounts to $O(LM)$ time. However, we can discard the factor L by doing the computation recursively as follows. Denote $\alpha^k_i = \sum_{k'=1}^{k} \omega_{k'} y^{k'}_{\lceil i/m_k \rceil}$ and $\beta^k_i = \sum_{k'=1}^{k} \omega_{k'} (y^{k'}_{\lceil i/m_k \rceil})^2$, for all $1 \le k \le L$ and $1 \le i \le M_k$. Then the following equalities are valid

$$\alpha^k_i = \alpha^{k-1}_{\lceil iM_{k-1}/M_k \rceil} + \omega_k y^k_i, \beta^k_i = \beta^{k-1}_{\lceil iM_{k-1}/M_k \rceil} + \omega_k (y^k_i)^2, \tag{23}$$

for all $1 \le k \le L$ and $1 \le i \le M_k$. Employing (23), the computation of all α_i (i.e., α^L_i) and β_i (i.e., β^L_i), $1 \le i \le M$ requires $O(M)$ time overall. We conclude that when the distortion function is the squared distance, Algorithm 1 runs in $O(M)$ time.

Next we show that Algorithm 1 can also be applied to compute the optimal encoder partition in the case of channel-optimized SQ and MDSQ design, with squared error distortion. Specifically, in these cases we have $h_{i,i'}(t) = (\alpha'_{i'} - \alpha'_i)t + \beta'_i - \beta'_{i'}$, for some constants α'_i, β'_i which are computed based on the fixed decoder [12], [22]. As shown in [12], [22], [23] if $\alpha'_{i'} = \alpha'_i$ and $\beta'_i < \beta'_{i'}$, then the interval $\mathcal{A}_{i'}$ is degenerate, thus index i' can be eliminated. Moreover, if $\alpha'_{i'} = \alpha'_i$ and $\beta'_i = \beta'_{i'}$, assigning some source sample t to index i is equivalent to assigning it to index i'. Therefore, nothing is lost if all elements of the cell $C_{i'}$ are transferred to cell C_i, leaving $C_{i'}$ empty. Consequently, these indices i' can be eliminated as well. After performing these eliminations, the remaining M_0 indices are relabeled from 1 to M_0 such that the values α_i to be strictly increasing as i increases. Then Lemma 1 will be satisfied. From the proof of Lemma 2 it follows that its conclusion holds when the conclusion of Lemma 1 holds. Thus, Lemma 2 is satisfied as well. Consequently, we can apply Algorithm 1 to compute the thresholds in the optimized encoder partition, thus improving in speed the existing solution.

V. Generating An All Non-empty Cells Partition

Empty cells may appear in practical applications of Lloyd's method even in the case of single resolution quantizer design due to the finiteness of the training set. On the other hand, in MRSQ design empty cells may appear however even when working on a continuous alphabet, which is the case of our study. This is due to the possibility of having $left(\mathcal{A}_i) \geq right(\mathcal{A}_i)$ for some index $i, 1 < i < L$. We illustrate this possibility by the following example.

Example 1. Consider the uniform distribution over the interval $[0, 26]$. Consider the design of an MRSQ with 2 refinement stages at rates $R_1 = 1$ and $R_2 = 3$, and squared error distortion. Hence, the quantizer Q_1 has 2 cells. Each of them is partitioned into 4 cells of quantizer Q_2. Assume that the thresholds of the initial central partition are: $x_1 = 2, x_2 = 4, x_3 = 6, x_4 = 8,\ x_5 = 10, x_6 = 12, x_7 = 18$. Then the optimized decoder has the following codewords for Q_2: $y_1^2 = 1, y_2^2 = 3$, $y_3^2 = 5, y_4^2 = 7$, $y_5^2 = 9, y_6^2 = 11$, $y_7^2 = 15, y_8^2 = 22$, and for Q_1: $y_1^1 = 4, y_2^1 = 17$. Assume that $\omega_1 = \omega_2 = 0.5$. Then, by using (22), we obtain

$$t_{4,5} = \frac{1}{2}\frac{\omega_1((y_2^1)^2 - (y_1^1)^2) + \omega_2((y_5^2)^2 - (y_4^2)^2)}{\omega_1(y_2^1 - y_1^1) + \omega_2(y_5^2 - y_4^2)} = \frac{61}{6},$$

$$t_{5,6} = \frac{1}{2}\frac{\omega_1((y_2^1)^2 - (y_2^1)^2) + \omega_2((y_6^2)^2 - (y_5^2)^2)}{\omega_1(y_2^1 - y_2^1) + \omega_2(y_6^2 - y_5^2)} = 10.$$

Since $t_{4,5} > t_{5,6}$, it follows that $\mathcal{A}_5 = \emptyset$, hence C_5^1 becomes empty after the encoder optimization step.

It is common procedure in quantization design to split some non-empty cells in order to generate a partition with all cells non-empty. The approach taken for single resolution quantizer is to split the non-empty cell with the highest partial expected distortion [13]. As realized in [2] this approach cannot be directly used in MRSQ design, due to the more complex structure of the MRSQ. The technique proposed by the authors of [2] resizes the partitions Q_k, proceeding in increasing order of k, as follows. If some cell C_i^k is empty and the cell of Q_{k-1} containing it, i.e., C_ℓ^{k-1}, where $\ell = \lceil (im_k)/m_{k-1} \rceil$, is non-empty, then C_ℓ^{k-1} is partitioned uniformly into m_{k-1}/m_k intervals which become the new cells C_j^k, $\ell m_{k-1}/m_k \leq j \leq (\ell+1)m_{k-1}/m_k - 1$. We note that this procedure does not guarantee that the expected distortion will not increase.

Therefore we propose a different technique. We operate directly at the highest resolution partition. Note first that cells C_1^L and C_M^L cannot be empty because $\mathcal{A}_1$ and $\mathcal{A}_M$ are non-degenerate as justified in Section III. Assume now that we have $x_{i-1} < x_i = x_{i+1} = \cdots = x_j < x_{j+1}$ for some $1 \leq i < j \leq M - 1$. In other words, all cells $C_{i+1}^L, C_{i+2}^L, \cdots C_j^L$ are empty, while C_i^L and C_{j+1}^L are not empty. Let ξ denote the common value of $x_i, \cdots, x_j$. First we find the lowest resolution level k_0 such that ξ is a threshold in Q_{k_0}. Note that k_0 is the smallest integer in the range 1 to L such that $\lceil i/m_{k_0} \rceil < \lceil (j+1)/m_{k_0} \rceil$, and let $i_0 = \lceil i/m_{k_0} \rceil m_{k_0}$. Then $i_0 < j + 1$ since otherwise we would have $\lceil i/m_{k_0} \rceil = i_0/m_k \geq \lceil (j+1)/m_{k_0} \rceil$, leading to a contradiction. Clearly, $i \leq i_0$. Then we set the new thresholds $x'_i, \cdots, x'_j$ such that the following to be satisfied:

$$x_{i-1} < x'_i < x'_{i-1} < \cdots < x'_{i_0} = \xi < x'_{i_0+1} < \cdots < x'_j < x_{j+1}. \tag{24}$$

In other words the old cell C^L_i is split into the new non-empty cells $C^L_i, \cdots, C^L_{i_0}$, and the old cell C^L_{j+1} is split to generate new non-empty cells $C^L_{i_0+1}, \cdots, C^L_{j+1}$. We will prove that this technique followed by a decoder optimization step leads to a strict decrease of the MRSQ's expected distortion.

It is known that by splitting a quantizer cell into two non-empty cells and then optimizing the decoder, the quantizer expected distortion strictly decreases (because $p(\cdot)$ is positive everywhere). By a simple inductive argument it follows that our procedure leads to a strict decrease of $D(Q_L)$. We will show that this is true for the lower resolutions, too. Notice that by the definition of k_0, for $k < k_0$, none of $x'_i, \cdots x'_j$ is a threshold in Q_k, hence the quantizers at resolutions lower than k_0 are not affected. Let us fix now some $k \geq k_0$, and let $\ell_1 = i_0/m_k$. Clearly ℓ_1 is an integer. Further consider the positive integers $\ell_0 = \lceil i/m_k \rceil$ and $\ell_2 = \lceil (j+1)/m_k \rceil - 1$. Then we have

$$(\ell_0 - 1)m_k \leq i - 1 < i \leq \ell_0 m_k \leq \ell_2 m_k \leq j < j + 1 \leq (\ell_2 + 1)m_k.$$

It is obvious that $\ell_0 \leq \ell_1 \leq \ell_2$. Then the effect of (24) on Q_k is that the old non-empty cell $C^k_{\ell_0}$ is split into the new non-empty cells $C^k_{\ell_0}, \cdots, C^k_{\ell_1}$, and the old non-empty cell $C^k_{\ell_2+1}$ is split into the new non-empty cells $C^k_{\ell_1+1}, \cdots, C^k_{\ell_2+1}$. Therefore, $D(Q_k)$ is strictly decreased, too. Thus, the proof of our claim is completed.

VI. Experimental Results

This section presents experimental results to demonstrate the computational savings offered by the proposed algorithm for the encoder optimization step. We consider a memoryless Gaussian source of 0 mean and variance 1, truncated to the interval $[-3, 3]$. The generalized Lloyd algorithm is used for the optimal design of MRSQ's with 2 refinement stages and squared distance distortion. Notice that, although this work is concerned with the MRSQ design for continuous distributions, in practice some form of discretization is necessary in order to compute the integrals needed for the calculation of the centroids at the decoder optimization step according to (6). Therefore, the Gaussian source is discretized by applying a fine uniform quantizer of step size 3×10^{-6}. Thus, the alphabet of the discrete distribution, $\mathcal{C} = \{c_1, \cdots, c_N\}$, contains $N = 2 \times 10^6$ symbols. Let us denote by $q(c_i)$ the probability of c_i, $1 \leq i \leq N$, and assume that $c_1 < c_2 < \cdots < c_N$.

We have implemented two variants of the design algorithm: variant A uses the proposed fast algorithm for computing the central partition thresholds, while variant B uses the algorithm of [22] for the same purpose. All the other steps are identical. Both variants contain a preprocessing step which computes and stores the cumulative probabilities $cum(i)$ and first moments $mom(i)$, $0 \leq i \leq N$. Specifically, $cum(i) = \sum_{j=1}^{i} q(c_j)$ and $mom(i) = \sum_{j=1}^{i} c_j q(c_j)$ for all $0 \leq i \leq N$. Having these quantities available, the centroid of any cell $C = \{c_i, c_{i+1}, \cdots, c_{i+j}\}$ can be efficiently computed at the decoder optimization step as $\mu(C) = \frac{mom(i+j)-mom(i-1)}{cum(i+j)-cum(i-1)}$. Thus, the time complexity of the decoder optimization step becomes $O(M)$. Corroborating with the observations in the previous sections, it follows that the asymptotical time complexity of variant A is $O(\lambda M)$, and that of variant B is $\Omega(\lambda M^2)$ (assuming that $M' = O(M)$), where λ denotes the number of iterations. We mention that we have excluded the time complexity of the preprocessing step in the aforementioned evaluation since it becomes negligible as λ increases.

The design algorithm, in both variants, stops at convergence, in other words, when the central partition thresholds become identical with those at the previous iteration. Notice that convergence is guaranteed to occur after a finite number of iterations since the number of possible different central partitions is finite (as a result of the discretization) and since any change in the central partition after some iteration generates a strict decrease in the expected distortion, according to the discussion in Section III.

In our simulations we have considered three values for R_2: $R_2 = 8, 9, 10$, and in each case all possible integer values of R_1: $R_1 = 1, \cdots, R_2 - 1$. For each MRSQ we have set $\omega_1 = \omega_2 = 0.5$. The tests were run on an Intel Core 2 Duo Processor P8400 with clock speed of 2.26 GHz. The results of our experiments are recorded in Tables I, II and III corresponding to $R_2 = 8, 9, 10$, respectively. Each table presents for each value of R_1, the number of iterations λ until convergence, and the running time in seconds of variants A (T_A) and B (T_B).

As our results show, variant A is faster than variant B in all the cases as expected. Moreover, the advantage in speed heightens as R_2 and λ increase. An interesting observation is that, for fixed R_2, the number of iterations until convergence does not depend monotonically on R_1. It rather tends to increase as the absolute difference between R_1 and $R_2 - R_1$ increases.

Based on our simulation results we conclude that the proposed algorithm for the central partition computation can provide a significant speed up of the MRSQ design when the number of cells in the central partition is sufficiently high.

VII. Conclusion

In this work we address the problem of optimal design of multi-resolution scalar quantizer (MRSQ) via the generalized Lloyd method. Such a design procedure was used previously in [2] for the case of squared error distortion. The algorithm employed to compute the optimal partition thresholds in the

R_1	# iterations	T_A in sec.	T_B in sec.
1	20419	0.7	2.3
2	7668	0.38	1.0
3	2560	0.25	0.47
4	949	0.2	0.28
5	1456	0.22	0.34
6	4123	0.31	0.63
7	10858	0.52	1.36

TABLE I
RUNNING TIMES OF VARIANTS A AND B FOR $R_2 = 8$ AND DIFFERENT VALUES OF R_1.

R_1	# iterations	T_A in sec.	T_B in sec.
1	43629	2.2	16.25
2	21331	1.25	7.98
3	7833	0.61	3.05
4	2634	0.33	1.16
5	1713	0.28	0.81
6	4302	0.45	1.77
7	12036	0.88	4.73
8	27916	1.77	10.5

TABLE II
RUNNING TIMES OF VARIANTS A AND B FOR $R_2 = 9$ AND DIFFERENT VALUES OF R_1.

encoder optimization step was borrowed from the work of [22] on multiple description scalar quantizer design. As in [2], we assume that the MRSQ cells are intervals, but generalize the distortion metric to any convex difference distortion. This generalization poses some challenges at the encoder optimization step. We show how these challenges can be overcome and propose an efficient algorithm for computing the optimized encoder partition. Our algorithm is more efficient than the algorithm of [2]. Moreover, it can be applied to solve the similar task in channel-optimized scalar quantizer and in multiple description scalar quantizer design, outperforming in speed the existing solutions.

R_1	# iterations	T_A in sec.	T_B in sec.
1	28225	2.59	38.92
2	46289	4.25	63.92
3	21472	2.2	29.94
4	7700	1.0	10.88
5	3048	0.55	4.36
6	4565	0.69	6.61
7	12519	1.53	17.64
8	30749	3.33	44.34
9	48190	5.11	67.09

TABLE III

RUNNING TIMES OF VARIANTS A AND B FOR $R_2 = 10$ AND DIFFERENT VALUES OF R_1.

APPENDIX A
PROOF OF LEMMAS

In this appendix we prove Lemma 1 and Lemma 2.

Lemma 1. For each pair of integers i, i', $1 \leq i < i' \leq M$, the function $h_{i,i'}(\cdot)$ is non-decreasing and has a unique zero point denoted by $t_{i,i'}$. Moreover, $V < t_{i,i'} < W$.

Proof. Fix some pair $i < i'$. Let K be the smallest element of the set $\{k | 1 \leq k \leq L, \lceil i/m_k \rceil \neq \lceil i'/m_k \rceil\}$. In other words, K is the coarsest resolution level at which the sets C_i^L and $C_{i'}^L$ are not included in the same quantization cell. Because $i < i'$, we have $\lceil i/m_K \rceil < \lceil i'/m_K \rceil$, hence the interval $C^K_{\lceil i/m_K \rceil}$ is situated to the left of $C^K_{\lceil i'/m_K \rceil}$. Moreover, $C^K_{\lceil i/m_K \rceil}$ contains the sets $C^k_{\lceil i/m_k \rceil}$ for all $k, K \leq k \leq L$. Likewise, $C^K_{\lceil i'/m_K \rceil}$ contains the sets $C^k_{\lceil i'/m_k \rceil}$ for all $k, K \leq k \leq L$. Using further the fact that each codeword is contained in the interior of the corresponding cell (as a consequence of the previous decoder optimization step (5)), it follows that

$$y^k_{\lceil i/m_k \rceil} < x_{\lceil i/m_K \rceil m_K} < y^k_{\lceil i'/m_k \rceil} \text{ for all } K \leq k \leq L. \tag{25}$$

To obtain the above relations we used the fact that $x_{\lceil i/m_K \rceil m_K}$ is the upper boundary of $C^K_{\lceil i/m_K \rceil}$.

For each $k, K \leq k \leq L$, consider the function $h^{(k)}(t)$ defined as follows:

$$h^{(k)}(t) = \rho(|t - y^k_{\lceil i/m_k \rceil}|) - \rho(|t - y^k_{\lceil i'/m_k \rceil}|).$$

Then

$$h_{i,i'}(t) = \sum_{k=K}^{L} \omega_k h^{(k)}(t). \tag{26}$$

As proved in [14], (Lemma 1), the inequality $y^k_{\lceil i/m_k \rceil} < y^k_{\lceil i'/m_k \rceil}$ implies that $h^{(k)}(\cdot)$ is a non-decreasing function, for all $k, K \le k \le L$. Then, using (26) and the fact that ω_k are positive, it follows that $h_{i,i'}(\cdot)$ is non-decreasing too.

Note that for $K \le k \le L$, and $t \le y^k_{\lceil i/m_k \rceil}$ we have $|t - y^k_{\lceil i/m_k \rceil}| < |t - y^k_{\lceil i'/m_k \rceil}|$ due to (25). Since $\rho(\cdot)$ is strictly increasing it follows that $h^{(k)}(y^k_{\lceil i/m_k \rceil}) < 0$. Further, using the positivity of ω_k it follows that

$$h_{i,i'}(t) < 0 \text{ for all } t \le \min\nolimits_{k,K\le k\le L} y^k_{\lceil i/m_k \rceil}. \tag{27}$$

Likewise, for $K \le k \le L$, and $t \ge y^k_{\lceil i'/m_k \rceil}$ we obtain that $h^{(k)}(y^k_{\lceil i'/m_k \rceil}) > 0$, and further that

$$h_{i,i'}(t) > 0 \text{ for all } t \ge \max\nolimits_{k,K\le k\le L} y^k_{\lceil i'/m_k \rceil}. \tag{28}$$

Relations (27) and (28) together with the continuity of $h_{i,i'}(\cdot)$ imply that $h_{i,i'}(\cdot)$ has at least one zero point $t_{i,i'}$ satisfying

$$t_{i,i'} \in (\min_{K\le k\le L} y^k_{\lceil i/m_k \rceil}, \max_{K\le k\le L} y^k_{\lceil i'/m_k \rceil}). \tag{29}$$

We next show that this point is unique using a proof by contradiction.

Denote by $null$ the set of zero points of the function $h_{i,i'}(\cdot)$, i.e., $null = \{t | h_{i,i'}(t) = 0\}$ and assume that $null$ contains two distinct values $t_1 < t_2$. Since each $h^{(k)}(\cdot)$ is non-decreasing, it follows that $h_k(t_1) \le h_k(t_2)$, for all $k, K \le k \le L$. If at least for one value of k the previous inequality were strict then we would have $h_{i,i'}(t_1) < h_{i,i'}(t_2)$ (because $\omega_k > 0$ for each k), which contradicts the definition of $null$. Therefore, we conclude that each $h^{(k)}(\cdot)$ must be constant on the set $null$.

Furthermore, note that each $h^{(k)}(\cdot)$ is strictly increasing on the interval $[y^k_{\lceil i/m_k \rceil}, y^k_{\lceil i'/m_k \rceil}]$. This claim follows easily from the fact that ρ is strictly increasing. Therefore, in order for each $h^{(k)}(\cdot)$ to be constant on $null$, the following condition must be satisfied

$$null \subseteq \bigcap_{k=K}^{L} ((-\infty, y^k_{\lceil i/m_k \rceil}] \cup [y^k_{\lceil i'/m_k \rceil}, \infty)). \tag{30}$$

Relation (25) implies that $\max_{k,K\le k\le L} y^k_{\lceil i/m_k \rceil} < \min_{k,K\le k\le L} y^k_{\lceil i'/m_k \rceil}$, leading to the conclusion that

$$\begin{aligned} \bigcap_{k=K}^{L} ((-\infty, y^k_{\lceil i/m_k \rceil}] \cup [y^k_{\lceil i'/m_k \rceil}, \infty)) &= \bigcap_{k=K}^{L} (-\infty, y^k_{\lceil i/m_k \rceil}] \cup \bigcap_{k=K}^{L} [y^k_{\lceil i'/m_k \rceil}, \infty) \\ &= (-\infty, \min_{K\le k\le L} y^k_{\lceil i/m_k \rceil}] \cup [\max_{K\le k\le L} y^k_{\lceil i'/m_k \rceil}, \infty). \end{aligned} \tag{31}$$

Relations (30), (31), (27), and (28) imply that $h_{i,i'}(t) \neq 0$ for any $t \in null$, thus contradicting the definition of $null$.

We conclude that the function $h_{i,i'}(\cdot)$ has a unique zero point $t_{i,i'}$. Moreover, the inequalities (29) imply that $t_{i,i'} \in (V, W)$, thus completing the proof. □

Lemma 2. For any integers $1 \leq \ell < i < j \leq M$ we have

$$\min\{t_{\ell,i}, t_{i,j}\} \leq t_{\ell,j} \leq \max\{t_{\ell,i}, t_{i,j}\}, \tag{32}$$

with equalities only if $t_{\ell,i} = t_{i,j}$.

Proof. The following equality is immediate

$$h_{\ell,j}(t) = h_{\ell,i}(t) + h_{i,j}(t). \tag{33}$$

The function $h_{\ell,j}(\cdot)$ is continuous and, according to Lemma 1, its unique zero point is $t_{\ell,j}$. Therefore, in order to prove the relation (32) it suffices to show that the quantities $h_{\ell,j}(t_{\ell,i})$ and $h_{\ell,j}(t_{i,j})$ have different signs. Since $h_{\ell,i}(t_{\ell,i}) = 0$ and $h_{i,j}(t_{i,j}) = 0$, we obtain from (33) that

$$h_{\ell,j}(t_{\ell,i}) = h_{i,j}(t_{\ell,i}) \text{ and } h_{\ell,j}(t_{i,j}) = h_{\ell,i}(t_{i,j}). \tag{34}$$

The functions $h_{i,j}$ and $h_{\ell,i}$ are non-decreasing, therefore each of them is negative to the left of its zero point and positive to the right. Therefore, a moment of thought reveals that, when $t_{\ell,i} \neq t_{i,j}$, we obtain that $h_{i,j}(t_{\ell,i})h_{\ell,i}(t_{i,j}) < 0$, leading to $h_{\ell,j}(t_{\ell,i})h_{\ell,j}(t_{i,j}) < 0$ via (34), which further implies that (32) holds with strict inequalities. Moreover, in the case when $t_{\ell,i} = t_{i,j}$ we have $h_{\ell,j}(t_{\ell,i}) = 0$ by (34), hence $t_{\ell,j} = t_{\ell,i} = t_{i,j}$. Thus, the proof is completed. □

Appendix B

Complexity Analysis of Algorithm of [22] for Central Partition Optimization

In this appendix we analyze the time complexity of Vaishampayan's algorithm for encoder partition optimization [22]. The pseudocode of the algorithm is presented below.

```
1) SET $m \leftarrow 1$, $p \leftarrow (m+1)$, $left(m) = V$, $n \leftarrow M$.
2) Compute $t_{m,p}$ and SET $t \leftarrow t_{m,p}$.
3) Evaluate $\phi_l(t)$ for $m < l \leq n$. SET $p' \leftarrow \arg\min_{m<l\leq n} \phi_l(t)$. If this minimum is achieved by
   several indexes then choose the highest index.
4) IF $p' \neq p$, SET $p \leftarrow p'$, $n \leftarrow p'$ and return to step 2); ELSE, SET $left(p) \leftarrow t$, $right(m) \leftarrow t$ and
   GO TO step 5).
```

5) IF $p = M$, STOP, ELSE, SET $m \leftarrow p$, $p \leftarrow (m+1)$, $n \leftarrow M$ and return to step 2).

Note that the most computationally intensive is step 3). As justified in [22], the values taken by m are all indices in $\mathcal{I} \setminus \{M\}$. Moreover, for each such m, the first time Step 3) is executed, the value of n equals to M, consequently this execution of step 3) requires $M - m$ comparisons to solve the underlying maximization. Thus, a lower bound on the total number of operations is

$$\sum_{s=1}^{M'-1} (M - j_s) \geq (M-1) + (M'-2) + (M'-1) + \cdots + 1 + 0, \tag{35}$$

which implies that the running time is $\Omega(M + M'^2)$.

References

[1] A. Antos, "On codecell convexity of optimal multi-resolution scalar quantizers for continuous sources", submitted for publication.

[2] H. Brunk and N. Farvardin, "Fixed-rate successively refinable scalar quantizers", *Proc. DCC'96*, Snowbird, Utah, March 1996, pp.250-259.

[3] S. Dumitrescu, X. Wu, "Optimal multiresolution quantization for scalable multimedia coding", Proc. IEEE Information Theory Workshop 2002, pp. 139 - 142, Oct. 2002.

[4] S. Dumitrescu and X. Wu, "Algorithms for optimal multi-resolution quantization", *J. Algorithms*, 50(2004), pp. 1-22.

[5] S. Dumitrescu and X. Wu, "On global optimality of gradient descent algorithms for fixed-rate scalar multiple description quantizer design", *Proc. IEEE DCC'05*, pp. 388-397, March 2005.

[6] S. Dumitrescu, "Speed-up of encoder optimization step in multiple description scalar quantizer design", *Proc. of IEEE Data Compression Conference*, pp. 382-391, March 2008, Snowbird, UT.

[7] S. Dumitrescu, X. Wu, "On Properties of Locally Optimal Multiple Description Scalar Quantizers with Convex Cells", *IEEE Trans. on Inform. Theory*, vol. 55, no. 12, pp. 5591-5606, Dec. 2009.

[8] M. Effros, D. Dugatkin, "Multiresolution Vector Quantization", *IEEE Trans. Inform. Theory*, vol. 50, no. 12, pp. 3130-3145, Dec. 2004.

[9] M. Effros and D. Muresan, "Cell Contiguity in Optimal Fixed-Rate and Entropy-Constrained Network Scalar Quantizers", *Proc. DCC'2002*, pp. 312-321, April 2002.

[10] M. Effros, and L. Schulman, "Rapid Near-Optimal VQ Design with a Deterministic Data Net", *Proc. ISIT'04*, pp. 298, July 2004.

[11] M. Effros, "Optimal multiple description and multiresolution scalar quantizer design", *Information Theory and Applications Workshop'08*, Univesity of California, San Diego, 27 January-1 February 2008.

[12] N. Farvardin and V. Vaishampayan, "Optimal Quantizer Design for Noisy Channels: an Approach to Combined Source-Channel Coding", *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 827-838, Nov. 1987.

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

[14] A. Gyorgy, and T. Linder, "On the structure of optimal entropy-constrained scalar quantizers", *IEEE Transactions on Information Theory*, vol. 48, pp. 416-427, Feb. 2002.

[15] A. Gyorgy, T. Linder, G. Lugosi,"Tracking the Best Quantizer" *IEEE Trans. Inform. Theory*, vol. 54, no. 4, pp. 1604-1625, April 2008.

[16] H. Jafarkhani, H. Brunk, and N. Farvardin, "Entropy-constrained successively refinable scalar quantization," *Proc. IEEE Data Compression Conference*, pp. 337-346, Mar.1997.

[17] A. Kurtenbach and P. Wintz, "Quantizing for noisy channels", *IEEE Trans. Comm. Techn.*, vol. COM-17, pp. 291-302, Apr. 1969.

[18] S. P. Lloyd, "Least squares quantization in PCM", *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129-137, Mar. 1982.

[19] D. Muresan and M. Effros, "Quantization as histogram segmentation: globally optimal scalar quantizer design in network systems", *Proc. DCC'2002*, pp. 302-311, April 2002.

[20] D. Muresan and M. Effros, "Quantization as histogram segmentation: optimal scalar quantizer design in network systems", *IEEE Trans. Inform. Th.*, vol. 54, no. 1, pp. 344-366, Jan. 2008.

[21] A. V. Trushkin, "Sufficient conditions for uniqueness of a locally optimal quantizer for a class of convex error weighting functions", *IEEE Trans. Inform. Th.*, vol. 28, no. 2, pp. 187-198, March 1982.

[22] V. A. Vaishampayan, "Design of multiple-description scalar quantizers", *IEEE Trans. Inform. Th.*, vol. 39, no. 3, pp. 821-834, May 1993.

[23] V. A. Vaishampayan, J. Domaszewicz, "Design of entropy-constrained multiple-description scalar quantizers", *IEEE Trans. Inform. Th.*, vol. 40, no. 1, pp. 245-250, Jan. 1994.

[24] X. Wu and S. Dumitrescu, "On optimal multi-resolution scalar quantization", *Proc. IEEE Data Compression Conference'02*, pp. 322-331, April 2002.