Algorithms for Optimal Multi-resolution Quantization

Sorina Dumitrescu
Department of Electrical and Computer Engineering
McMaster University
Hamilton, ON, Canada
sorina@mail.ece.mcmaster.ca

Xiaolin Wu [*]
Department of Computer and Information Science
Polytechnic University
Brooklyn, New York, NY 11201
xwu@poly.edu

[*] Supported by NSF Grant CCR-0208678.

Abstract– Multi-resolution quantization is a way of constructing a progressively refinable description of a discrete random variable. The underlying discrete optimization problem is to minimize an expected distortion over all refinement levels weighted by the probability or importance of the descriptions of different resolutions. This research is motivated by the application of multimedia communications via variable-rate channels. We propose an $O(rN^2)$ time and $O(N^2 \log N)$ space algorithm to design a minimum-distortion quantizer of r levels for a random variable drawn from an alphabet of size N. It is shown that for a very large class of distortion measures the objective function of optimal multi-resolution quantization satisfies the convex Monge property. The efficiency of the proposed algorithm hinges on the convex Monge property. But our algorithm is simpler (even though of the same asymptotic complexity) than the well-known SMAWK fast matrix search technique, which is the best existing solution to the quantization problem. For exponential random variables our approach leads to a solution of even lower complexity: $O(rN)$ time and $O(N \log N)$ space, which outperforms all the known algorithms for optimal quantization in both multi- and single-resolution cases. We also generalize the multi-resolution quantization problem to a graph problem, for which our algorithm offers an efficient solution.

Key words: *Quantization, multi-resolution signal representation, multimedia communications, convex Monge property, matrix search, dynamic programming.*

1 Introduction

Signal quantization is a subject of fundamental importance to the engineering fields of digital communications and data compression. The problem of optimal quantization is directly motivated by the desire to code and transmit signals as accurately and efficiently as possible. The optimization objective is simple to state: coding of a random variable X to the maximum precision (or minimum distortion) using a given number of bits. Consider a discrete random variable X whose values are drawn from a finite alphabet $\mathcal{A}$, $\mathcal{A} = \{x_1, x_2, \cdots x_N\} \subset \mathbf{R}$, where $x_i < x_{i+1}$, $1 \leq i < N$. Let $p(x_i), 1 \leq i \leq N$, be the probability mass function of the random variable X. We assume without loss of generality that $p(x_i) > 0$ for all $i, 1 \leq i \leq N$. For any positive integer k, denote by $\{0,1\}^k$ the set of all binary words of length k. Let $\mathcal{B}$ be another finite alphabet such that $\mathcal{A} \subseteq \mathcal{B} \subset \mathbf{R}$.

Definition 1. A fixed-rate scalar quantizer Q for the random variable X is a pair of two mappings: the encoder $f_Q : \mathcal{A} \to \{0,1\}^r$, where r is an integer such that $1 \leq r \leq log_2 N$, and the decoder, which is a one-to-one function $g_Q : \{0,1\}^r \to \mathcal{Y}$, $\mathcal{Y} \subset \mathcal{B}$. For each symbol $x \in \mathcal{A}$, the value $g_Q(f_Q(x))$, also denoted by $Q(x)$, is called the reproduction codeword of x, whereas $f_Q(x)$ is called the binary codeword index for x. The set $\mathcal{Y}$ of all reproduction codewords is called a codebook.

The quantizer generates a partition of the input alphabet $\mathcal{A}$: $C_u = \{x \in \mathcal{A} | f_Q(x) = u\}$, $u \in \{0,1\}^r$. The sets C_u of this partition are called the codecells of the quantizer. The quantizer maps all symbols x contained in a codecell C_u to a reproduction codeword $g_Q(u) = Q(x)$. The quantizer mapping function Q induces a distortion $d(x, Q(x))$ between a symbol x and its reproduction $Q(x)$. The overall reproduction quality of quantizer Q is measured by the expected distortion:

$$D(Q) = E\{d(X, Q(X))\} = \sum_{u \in \{0,1\}^r} \sum_{x \in C_u} d(x, g_Q(u)) p(x). \tag{1}$$

Besides its expected distortion $D(Q)$, a quantizer Q is also characterized by its bit rate $R(Q)$ which is the average number of bits per symbol required to label the codewords. Throughout this paper we only consider the case of fixed rate quantizer for which all codewords have the same code length. In the formulation above, the fixed code length is r, hence the quantizer rate is $R(Q) = r$.

Since 1960's the majority of work in the literature on scalar quantization addressed the problem of designing optimal scalar quantizers that minimize $D(Q)$, over all possible quantizers Q, given the random variable X, for a fixed rate $R(Q)$ [3, 5, 13, 16, 18, 19, 20, 21, 22]. We call this class of quantizers single-resolution scalar quantizers. Recently, motivated by the applications in the Internet and wireless communications, researchers turned their attention to the problem of multi-resolution quantization [6, 7, 8, 9, 15, 23], as defined below.

Definition 2. A multi-resolution scalar quantizer of L refinement stages is a sequence of L scalar quantizers $\mathbf{Q} = (Q_1, Q_2, \cdots, Q_L)$ such that $R(Q_1) < R(Q_2) < \cdots < R(Q_L)$, where any rate $R(Q_i)$, $1 \leq i \leq L$, is an integer, and for each $x \in \mathcal{A}$ and $1 \leq i < L$

$$f_{Q_{i+1}}(x) \in f_{Q_i}(x)\{0,1\}^{R(Q_{i+1})-R(Q_i)}. \tag{2}$$

The condition (2) states that the binary codeword index of x for Q_{i+1}, the quantizer at the $(i+1)$-th refinement stage, is obtained by appending exactly $2^{R(Q_{i+1})-R(Q_i)}$ bits to the end of the codeword index of x at the previous refinement stage corresponding to Q_i. It implies that each codecell of Q_i is partitioned into $2^{R(Q_{i+1})-R(Q_i)}$ codecells of Q_{i+1}. In other words, the alphabet partitions formed by the sequence of L quantizers $Q_1, Q_2, \cdots, Q_L$ are successively embedded into each other, and hence multi-resolution quantizer (MRQ) is progressively refinable from Q_1 to Q_2, then to Q_3, and so forth. The description of x can be progressively refined by so-called embedded bit plane coding, which scans the bits of the codeword index $f_{Q_{i+1}}(x)$, from the most significant to the least.

The advantage of multi-resolution quantizer over its single-resolution counterpart is that it facilitates rate-distortion scalable compression of a signal. The rate-distortion scalability is a very important mechanism for maintaining the quality of network service when the bandwidth fluctuates in time due to network congestion and/or channel noise. When the effective transmission rate drops below the target bit rate of a non-scalable code based on single-resolution quantization, the code may fail abruptly, causing sudden outage of network service. In contrast, a scalable embedded code stream of multi-resolution quantizer code offers a graceful degradation in reconstruction quality when channel conditions deteriorate. This is because an embedded bit sequence can be truncated in the middle, and the truncated code segment (a prefix of the sequence) can still be decoded to an overall representation of the coded signal, with a reconstruction quality proportional to the length of the truncated code segment. The effect of successive refinement of a coded image via progressive transmission of embedded MRQ bit stream is illustrated by Figure 1.

One can also define a multi-resolution vector quantizer simply by replacing random

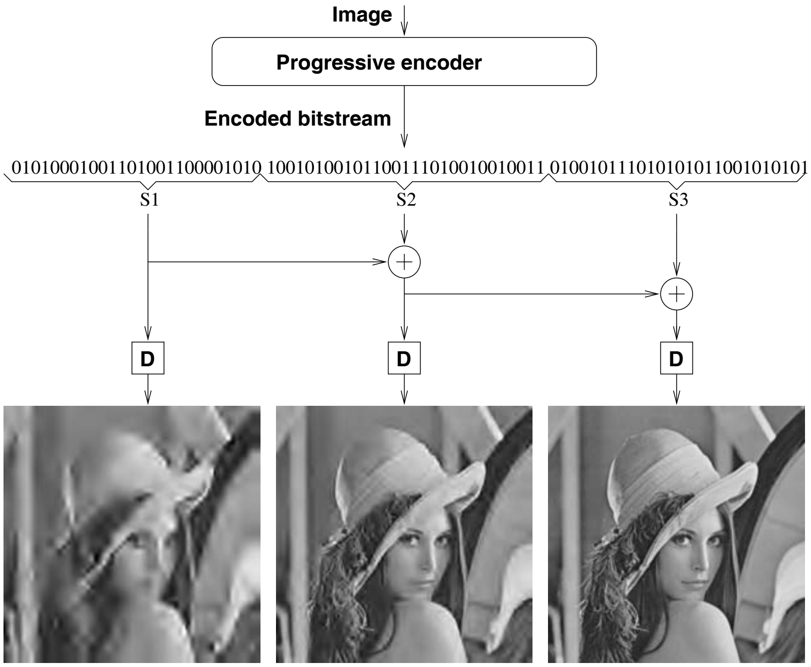

Figure 1: Progressive image reconstruction via scalable embedded bit stream.

variable X with random vector $\mathbf{X}$ in the definition above. However, even single-resolution optimal vector quantization is known to be NP-hard [12], whereas optimal single-resolution scalar quantizers can be computed in $O(KN)$ or $O(N^2)$ time depending on the distortion metric $d(X, Q(X))$, where N is the size of symbol alphabet $\mathcal{A}$ and $K = 2^r$ is the number of codewords [20, 21]. If $d(X, Q(X))$ is mean-square error, then the problem can be solved in $O(N\sqrt{K \log N} + N \log N)$ time [3], or even better in $O(N2^{O(\sqrt{\log K \log \log N})})$ time [18]. Note that $K < N$ in data compression applications. Since efficient algorithms (polynomial in N, and pseudo-polynomial in K) exist for optimal scalar quantizer design but not for optimal vector quantizer design unless P=NP, we restrict ourselves in this paper to the investigation of algorithms for designing optimal scalar multi-resolution quantizers. In the sequel, the terms quantizer and quantization, unless explicitly qualified, all refer to the scalar case.

The paper is organized as follows. In the next section we formulate the problem of optimal multi-resolution quantization, which aims to minimize the expected distortion over a set of bit rates rather than for a fixed bit rate as in optimal single-resolution quantization. In Section 3 we present an $O(rN^3)$ time dynamic programming algorithm for designing optimal MRQ of L refinement stages for a source alphabet of size N, where r is the rate of the highest refinement level ($r = R(Q_L)$). The time complexity of optimal MRQ design can be reduced to $O(rN^2)$ under a very mild monotonicity condition of the distortion function, which is the topic of Section 4. Section 5 addresses the space complexity of the MRQ design algorithm. We show how the intermediate results of the dynamic programming process, which are required to reconstruct the alphabet partition of the optimal quantizer, can be efficiently stored. In Section 6 we present an $O(MN)$ time and $O(MN)$ space algorithm that computes the distortions of all convex subsets of $\mathcal{A}$, where M is the size of alphabet $\mathcal{B}$. This algorithm performs a necessary preprocessing to facilitate the optimal MRQ design algorithm of Section 4. In practical cases of interest we have $M = O(N)$, hence the preprocessing step does not increase the complexity of the optimization problem. Furthermore, the preprocessing can be completed in $O(N)$ time and $O(N)$ space for the ubiquitous mean-square distortion measure. In Section 7 we show how the time complexity of optimal MRQ design can be further lowered to $O(rN)$ for exponential random variables (commonly encountered in applications of signal compression). This result immediately extends to the design of optimal single-resolution quantizer of exponential random variables, since the problem is a special case of optimal MRQ design. The $O(rN)$ time complexity is the lowest so far in the literature. Section 8 generalizes the problem of optimal MRQ design to a graph problem, which can be solved by using the algorithms presented in this paper.

2 Problem Formulation

Since an MRQ is to operate in a range of bit rates, its distortion should measure the expected reconstruction quality weighted by the probability of its operational bit rates, rather than at a single fixed rate $r = R(Q)$. Let $U(i)$ be the probability that the MRQ operates at the i-th refinement stage, i.e., $R(Q_i)$ bits are used to represent X, $1 \leq i \leq L$. The expected distortion of the MRQ $\mathbf{Q}$ is defined as

$$\bar{D}(\mathbf{Q}) = \sum_{i=1}^{L} U(i) D(Q_i). \tag{3}$$

Now we can state the problem of designing optimal MRQ, the central thesis of this paper, as the following.

Problem 1. Given the discrete random variable X, a sequence of L target rates $R_1 < R_2 < \cdots < R_L$ (all being positive integers) and a probability mass function $U(i)$, $1 \leq i \leq L$, with $U(i)$ being the probability that R_i bits are used to represent X, construct an MRQ with L refinement stages $\mathbf{Q} = (Q_1, Q_2, \cdots, Q_L)$ such that $R(Q_i) = R_i$, $1 \leq i \leq L$, and the expected distortion $\bar{D}(\mathbf{Q})$ is minimal.

A key to the tractability of the underlying optimization problem is the convexity of the codecells. A single-resolution optimal quantizer (for a wide class of distortion measures) has to have convex codecells C_u, i.e., for any two values x and x' contained in C_u with $x < x'$, any symbol $x" \in \mathcal{A}$, $x < x" < x'$, is also contained in C_u [10, 11]. This property permits the use of dynamic programming to design optimal single-resolution quantizers [3, 5, 13, 18, 19, 20, 21, 22]. Unfortunately, pathological cases were found in which an optimal multi-resolution quantizer has non-convex codecells [10]. Also for entropy-constrained scalar quantizers, even in the single-resolution case, codecell convexity might preclude optimality [14]. For tractability, however, codecell convexity was imposed in the development of algorithms for optimal MRQ design [6, 17, 23]. Under this constraint Brunk *et al.* [6] proposed an iterative descent algorithm for MRQ design. However, their algorithm can only find a locally optimal solution. In [23] we presented a dynamic programming algorithm that computes the globally optimal MRQ of convex codecells. The same constraint is also respected in this paper, and should be assumed by the reader in the sequel so that we will not have to state it everywhere. The complexity of the algorithm of [23] is $O(rN^3)$, where $r = R_L$ is the bit rate of the finest refinement level of MRQ.

The main contribution of the present paper is a reduction of the complexity to $O(rN^2)$ for a wide class of distortion functions $d(X, Q(X))$. We call a distortion function $d(x, y)$, $d : \mathbf{R} \times \mathbf{R} \to [0, \infty)$, monotone, if for any real x, y_1 and y_2, if $x \leq y_1 < y_2$ or $x \geq y_1 > y_2$, then $d(x, y_1) \leq d(x, y_2)$, All distortion measures of signal quantization used in practice fall into the class of monotone distortion functions. Based on (1) and (3) the expected distortion of the MRQ $\mathbf{Q}$ can be rewritten as:

$$\bar{D}(\mathbf{Q}) = \sum_{i=1}^{L} U(i) \sum_{u \in \{0,1\}^{R(Q_i)}} \sum_{x \in C_u} d(x, g_{Q_i}(u))p(x). \tag{4}$$

It follows that for each codecell C_u of the optimal MRQ, the associated reproduction codeword $g_{Q_i}(u)$ must satisfy

$$\sum_{x \in C_u} d(x, g_{Q_i}(u))p(x) = \min_{y \in \mathcal{B}} \sum_{x \in C_u} d(x, y)p(x). \tag{5}$$

For each subset C of the input alphabet $\mathcal{A}$, define the distortion of C, $D(C)$ as:

$$D(C) = \min_{y \in \mathcal{B}} \sum_{x \in C} d(x, y)p(x) \tag{6}$$

if $C \neq \emptyset$, and $D(C) = 0$ if $C = \emptyset$. Hence the expected distortion of the MRQ can be expressed as:

$$\bar{D}(\mathbf{Q}) = \sum_{i=1}^{L} U(i) \sum_{u \in \{0,1\}^{R(Q_i)}} D(C_u). \tag{7}$$

For the sake of completeness, we note the necessity of allowing empty MRQ codecells. For single-resolution quantizer the optimal quantizer with given rate r has all 2^r codecells nonempty, $r \leq \log_2 N$, if the distortion function $d(X, Q(X))$ is monotone (this assertion should be understood in a weaker sense: there exists an optimal quantizer with all codecells nonempty). However, in the case of MRQ, imposing the condition that all the $2^{R(Q_i)}$ codecells at each refinement stage Q_i, be nonempty, might preclude the optimality, especially when $R(Q_L)$ is close to $\log_2 N$. We illustrate this by the following example.

Example. Let the two alphabets be $\mathcal{A} = \{20, 40, 60, 140\}$ and $\mathcal{B} = \{y \in \mathbf{N} | 20 \leq y \leq 140\}$. Consider the random variable X whose probability mass function is: $p(20) = \frac{1}{8}, p(40) = \frac{1}{8}, p(60) = \frac{3}{8}$ and $p(140) = \frac{3}{8}$. Let the distortion function be the squared distance: $d(x, y) = (x - y)^2$. Now examine the problem of constructing the optimal MRQ with two refinement stages and target rates $R_1 = 1$ and $R_2 = 2$. If we require all codecells to be nonempty, then the only possible MRQ (up to a reindexing of codecells) must have the codecells: $C_0 = \{20, 40\}$, $C_1 = \{60, 140\}$ at the first refinement stage, and $C_{00} = \{20\}$, $C_{01} = \{40\}$, $C_{10} = \{60\}$, $C_{11} = \{140\}$ at the second refinement stage. The expected distortion of this MRQ is

$$\bar{D}_1 = U(1) \cdot 625 + U(2) \cdot 0. \tag{8}$$

Consider now the MRQ with codecells: $C'_0 = \{20, 40, 60\}$, $C'_1 = \{140\}$ and $C'_{00} = \{20, 40\}$, $C'_{01} = \{60\}$, $C'_{10} = \{140\}$, $C'_{11} = \emptyset$. The expected distortion of this MRQ is

$$\bar{D}_2 = U(1) \cdot 160 + U(2) \cdot 25. \tag{9}$$

For $U(1) = \alpha$ and $U(2) = 1 - \alpha$ such that $\frac{5}{98} < \alpha \leq 1$, we have $\bar{D}_2 < \bar{D}_1$.

3 Optimal MRQ Design

By its definition an MRQ $\mathbf{Q}$ is completely specified by the encoder at the highest refinement stage and all intermediate rates $R(Q_1), R(Q_2), \cdots, R(Q_{L-1})$. Indeed, the condition (2) is equivalent to the requirement that for each $1 \leq i < L$, and each $u \in \{0,1\}^{R(Q_i)}$, the codecell C_u of the quantizer Q_i is the union of the $2^{R(Q_L)-R(Q_i)}$ codecells of the quantizer Q_L whose indices are binary numbers having the binary word u as their common most significant bits:

$$C_u = \cup_{v \in u\{0,1\}^{R(Q_L)-R(Q_i)}} C_v. \tag{10}$$

where $u\{0,1\}^k$ is the set of all binary words formed by appending to u all possible k-bit binary numbers. In fact the encoder of Q_L defines a multi-resolution quantizer of not only L, but $R(Q_L)$ refinement stages, corresponding to all integer rates from 1 to $R(Q_L)$.

In general, any quantizer Q of fixed rate r is naturally associated with an MRQ of r refinement stages. Let C_v, $v \in \{0,1\}^r$, be the codecells of Q. For each $i, 1 \leq i < r$, and each binary word u of length i, define the set:

$$C_u = \cup_{v \in u\{0,1\}^{r-i}} C_v. \tag{11}$$

Denote by Q_i, $1 \leq i \leq r$, the quantizer of rate i consisting of the codecells C_u, $u \in \{0,1\}^i$, and let $\hat{\mathbf{Q}}$ be the sequence of quantizers $(Q_1, Q_2, \cdots, Q_r)$. Since condition (2) is clearly satisfied, $\hat{\mathbf{Q}}$ is an MRQ. We call $\hat{\mathbf{Q}}$ the multi-resolution quantizer induced by the quantizer Q. Given a probability mass function $W(i)$, $1 \leq i \leq r$, with $W(i)$ being the probability that the first i bits of the quantized random variable X are transmitted via the channel, the expected distortion of the MRQ $\hat{\mathbf{Q}}$ is

$$\bar{D}(\hat{\mathbf{Q}}) = \sum_{i=1}^{r} W(i) \sum_{u \in \{0,1\}^i} D(C_u). \tag{12}$$

The objective of optimal MRQ design is to minimize $\bar{D}(\hat{\mathbf{Q}})$. Note that $\hat{\mathbf{Q}}$ represents an MRQ of the maximum number of refinement stages. However, in the general case of Problem 1 one can obtain the solution by minimizing $\bar{D}(\hat{\mathbf{Q}})$, but letting $W(i) = U(j)$ if there is some j such that $i = R_j$, and $W(i) = 0$ otherwise. Thus we can restate Problem 1 as the following.

Problem 2. Given a random variable X, a positive integer r and the probability mass function $W(i)$, $1 \leq i \leq r$, construct a single-resolution quantizer Q of rate r, such that the multi-resolution quantizer induced by Q has the minimal expected distortion $\bar{D}(\hat{\mathbf{Q}})$.

The convexity of codecells of $\hat{\mathbf{Q}}$ implies that for each nonempty codecell C_u there exist a unique pair of integers (n_1, n_2), $0 \leq n_1 < n_2 \leq N$, such that $C_u = \{x_i | n_1 < i \leq n_2\}$. For all integers $n_1, n_2, 0 \leq n_1 \leq n_2 \leq N$ denote by $c(n_1, n_2]$ the set $\{x_i | n_1 < i \leq n_2\}$ (obviously, $c(n_1, n_1] = \emptyset$). To shorten the notation, the distortion of the set $c(n_1, n_2]$ as defined by (6) is written as $D(n_1, n_2]$ instead of $D(c(n_1, n_2])$.

There is a natural partial order $\prec$ on the set of nonempty convex subsets of $\mathcal{A}$: $c(n_1, n_2] \prec c(n_3, n_4]$ if and only if $n_2 \leq n_3$. This partial order induces a total order on the set of nonempty codecells of each refinemet stage of the MRQ. To simplify the algorithm design we would like to index the codecells of quantizer Q in such a way as to preserve the order of codecells. Specifically, for any two binary words u and u' with equal length and such that C_u and $C_{u'}$ are nonempty, if $u < u'$ in lexicographical sense, then $C_u \prec C_{u'}$. Since the distortion of single-resolution quantizer clearly does not depend on how the codecells are indexed, reindexing the codecells of quantizer Q in lexicographical order does not change $D(Q)$. On the other hand, since the codecells of MRQ $\hat{\mathbf{Q}}$ are structured on the codecells of Q and the structure is labelled by (11), an arbitrary reindexing of the codecells of Q might change the encoder partitions at previous refinement stages of $\hat{\mathbf{Q}}$, thus affecting $\bar{D}(\hat{\mathbf{Q}})$. However, if the codecells at all refinement stages of $\hat{\mathbf{Q}}$ are convex, the following reindexing of the codecells of Q does preserve the order of codecells at each refinement stage Q_i of $\hat{\mathbf{Q}}$, and does not affect the expected distortion of the MRQ, $\bar{D}(\hat{\mathbf{Q}})$. For each integer i, $0 \le i \le r-1$, in increasing order, and each binary word u of length i, test codecells C_{u0} and C_{u1} (on the $(i+1)$-th refinement stage of $\hat{\mathbf{Q}}$). If any of them is empty or $C_{u0} \prec C_{u1}$ do nothing; otherwise reindex all the codecells of Q by interchanging the prefixes $u0$ and $u1$. From now on we may assume that the indexing of codecells preserves the order of codecells.

To find an efficient solution to Problem 2 we exploit the structure of embedded codecells of a quantizer Q: $C_u = \cup_{v \in u\{0,1\}^k} C_v$, with $u \in \{0,1\}^{r-k}$, $1 \le k \le r-1$. Given a codecell C_u, consider all possible partitions of C_u into sub-codecells C_v, $v \in u\{0,1\}^k$. These varying partitions only affect the partial sum of (12):

$$\sum_{i=1}^{k} W(r-k+i) \sum_{v \in u\{0,1\}^i} D(C_v). \tag{13}$$

Furthermore, as long as C_u is fixed, the variation of the other codecells of quantizer Q: C_w, $w \notin u\{0,1\}^k$, does not affect the above expression either. Since the codecells of $\hat{\mathbf{Q}}$ are convex, we have $C_u = c(a,b]$ for some integers a, b, $0 \le a \le b \le N$, and the codecells $C_v \in u\{0,1\}^k$ form a convex partition of $c(a,b]$. Since the indexing of codecells preserves the codecells order, it follows that there is a (2^k+1)-tuple of integers $(s_0, s_1, \cdots s_{2^k-1}, s_{2^k})$ such that $a = s_0 \le s_1 \le \cdots \le s_{2^k-1} \le s_{2^k} = b$ and, for any $j, 0 \le j \le 2^k - 1$,

$$c(s_j, s_{j+1}] = C_{uu'}, \tag{14}$$

where u' is the k-bit binary representation of j. Consequently, the partial sum (13) can be rewritten as

$$D_k(s_0, s_1, \cdots s_{2^k-1}, s_{2^k}) = \sum_{i=1}^{k} W(r-k+i) \sum_{j=0}^{2^i-1} D(s_{j2^{k-i}}, s_{(j+1)2^{k-i}}]. \tag{15}$$

Hence, given $C_u = c(a, b]$ with fixed a and b, minimizing (13) is equivalent to minimizing $D_k(s_0, s_1, \cdots s_{2^k-1}, s_{2^k})$ over all $(2^k + 1)$-tuple of integers $(s_0, s_1, \cdots, s_{2^k-1}, s_{2^k})$ satisfying $a = s_0 \leq s_1 \leq \cdots \leq s_{2^k-1} \leq s_{2^k} = b$. The set of all these $(2^k + 1)$-tuple of integers is denoted by $\mathcal{I}_k(a, b]$. For each pair of a and b, $0 \leq a \leq b \leq N$ and each $1 \leq k \leq r$, define

$$\hat{D}_k(a, b] = \min_{(s_0, s_1, \cdots, s_{2^k-1}, s_{2^k}) \in \mathcal{I}_k(a,b]} \sum_{i=1}^{k} W(r - k + i) \sum_{j=0}^{2^i-1} D(s_{j2^{k-i}}, s_{(j+1)2^{k-i}}]. \tag{16}$$

Obviously, $\hat{D}_k(a, a] = 0$ for all a, $0 \leq a \leq N$. By (16) the minimal expected distortion $\bar{D}(\hat{\mathbf{Q}})$ equals to $\hat{D}_r(0, N]$. The following proposition shows that the values $\hat{D}_k(a, b]$ can be computed recursively. We set by convention $D_0(a, b] = 0$ for all $a, b, 0 \leq a \leq b \leq N$.

Proposition 1. For $1 \leq k \leq r$, $0 \leq a \leq b \leq N$, the following relation holds

$$\hat{D}_k(a, b] = \min_{\xi, a \leq \xi \leq b} \{W(r-k+1)(D(a, \xi] + D(\xi, b]) + \hat{D}_{k-1}(a, \xi] + \hat{D}_{k-1}(\xi, b]\}. \tag{17}$$

Proof.

Consider an arbitrary $(2^k + 1)$-tuple of integers $(s_0, s_1, \cdots, s_{2^k})$ in $\mathcal{I}_k(a, b]$ and let $\xi = s_{2^{k-1}}$. Then

$$D_k(s_0, s_1, \cdots, s_{2^k}) = W(r-k+1)(D(a, \xi]+D(\xi, b])+D_{k-1}(s_0, \cdots, s_{2^{k-1}})+D_{k-1}(s_{2^{k-1}}, \cdots, s_{2^k}).$$

When ξ is fixed, the first term of the above sum is constant, and the quantities $D_{k-1}(s_0, \cdots, s_{2^{k-1}})$ and $D_{k-1}(s_{2^{k-1}}, \cdots, s_{2^k})$ can be minimized independently. Now the conclusion follows. □

Further, for each $1 \leq k \leq r$, $0 \leq a \leq b \leq N$, we define

$$\xi_k(a, b] = argmin_{\xi, a \leq \xi \leq b}\{W(r-k+1)(D(a, \xi] + D(\xi, b]) + \hat{D}_{k-1}(a, \xi] + \hat{D}_{k-1}(\xi, b]\}. \tag{18}$$

In case when the point of minimum of the underlying objective function is not unique, we let $\xi_k(a, b]$ be the largest among these points. Clearly, $\xi_k(a, a] = a$ for all $0 \leq a \leq N$.

Proposition 1 immediately suggests the following dynamic programming algorithm to solve Problem 2.

Algorithm 1. Optimal MRQ Design.

Step 1. For increasing k, $k = 1, 2, \cdots, r$, and for all integers a, b, $0 \leq a \leq b \leq N$, compute and store the values $\hat{D}_k(a, b]$ and $\xi_k(a, b]$ using the recursion (17)(When $k = r$ it suffices to consider only $a = 0$ and $b = N$.)

Step 2. Let $q_0 = 0$ and $q_{2^r} = N$. For decreasing k, $k = r, r-1, \cdots, 1$ and each $i = 0, 1, \cdots, 2^{r-k} - 1$, set

$$q_{(2i+1)2^{k-1}} = \xi_k(q_{i2^k}, q_{(i+1)2^k}].$$

The algorithm outputs the $(2^r + 1)$-tuple of integers $(q_0, q_1, \cdots, q_{2^r})$, which specifies the optimal scalar multi-resolution quantizer. Namely, for each $u \in \{0,1\}^r$, $C_u = c(q_i, q_{i+1}]$, where i is the integer whose r-bit binary representation is u.

The complexity of the above algorithm is dominated by the operations of Step 1. Here we assume that a preprocessing step is taken to compute and store all the values $D(a,b]$, $0 \le a \le b \le N$. The details of this preprocessing are deferred to Section 6 where we will show that the cost of the preprocessing does not affect the complexity of the algorithm. For each triple k, a, b, the computation of $\hat{D}_k(a,b]$ using (17) spends $O(N)$ time if linear search is applied. Since there are $O(rN^2)$ such triples to be considered, the total cost of Step 1 becomes $O(rN^3)$. We will show in the next section that solving (17) does not need linear search, and reduce the time complexity to $O(rN^2)$.

4 Complexity Reduction by Monotonicity

The baseline algorithm for optimal MRQ design as given in the previous section can be improved by exploiting a monotonicity property of $\xi_k(a,b]$ stated below.

Proposition 2. For any integer k, $1 \le k \le r$, and any integers a, a', b, b' such that $0 \le a \le b \le N$, $0 \le a' \le b' \le N$, $a \le a'$ and $b \le b'$, the following inequality holds:

$$\xi_k(a,b] \le \xi_k(a',b'], \tag{19}$$

if the distortion measure $d(X, Q(X))$ between a random variable X and its quantizer reproduction $Q(X)$ is monotone (defined in Section 2).

This proposition says that the search range for $\xi_k(a,b]$ can be reduced from the interval $[a,b]$ to the much smaller one $[\xi_k(a,b-1], \xi_k(a+1,b]]$. Later in this section we will use this property to organize the computations of $\hat{D}_k(a,b]$ and $\xi_k(a,b]$ for all intervals $(a,b]$, $0 \le a < b \le N$, and for a given k in such a way that these $O(N^2)$ values can be computed in $O(N^2)$ time. In fact, the counterpart of Proposition 2 for conventional single-resolution quantization was shown by Wu and Zhang for all monotone distortion measures [21]. In what follows we generalize the results of [21] to the case of optimal multi-resolution quantization and eventually prove Proposition 2.

To proceed we need a few notations. For any integer k, $1 \leq k \leq r$, and any integers a, b, $0 \leq a \leq b \leq N$, let

$$E_k(a, b) = W(r-k+1)D(a, b] + \hat{D}_{k-1}(a, b], \tag{20}$$

and rewrite (17) and (18) respectively as

$$\hat{D}_k(a, b] = \min_{\xi, a \leq \xi \leq b} \{E_k(a, \xi) + E_k(\xi, b)\} \tag{21}$$

and

$$\xi_k(a, b] = argmin_{\xi, a \leq \xi \leq b} \{E_k(a, \xi) + E_k(\xi, b)\} \tag{22}$$

The proof of Proposition 2 relies on the fact that the function $E_k(\cdot, \cdot)$ satisfies the so-called convex Monge condition. A real valued function $A(a, b)$ of integers $a, b, 0 \leq a \leq b \leq N$, is said to satisfy the convex Monge condition if and only if for all integers $0 \leq a < a' \leq b < b' \leq N$, the following relation holds [4]:

$$A(a, b) + A(a', b') \leq A(a, b') + A(a', b). \tag{23}$$

Working toward the proof of Proposition 2, we present two lemmas.

Lemma 1. If $A(a, b)$ and $A'(a, b)$ are two real-valued functions defined on integers $0 \leq a \leq b \leq N$, which satisfy the convex Monge condition, then the function $B(a, b)$ as defined by

$$B(a, b) = \min_{a \leq \mu \leq b} (A(a, \mu) + A'(\mu, b)), \quad 0 \leq a \leq b \leq N, \tag{24}$$

also satisfies the convex Monge condition.

Proof. We want to show that for $0 \leq a < a' \leq b < b' \leq N$

$$B(a, b) + B(a', b') \leq B(a, b') + B(a', b). \tag{25}$$

Consider the integers ξ and ν, $a \leq \xi \leq b'$ and $a' \leq \nu \leq b$, such that by (24)

$$B(a, b') = A(a, \xi) + A'(\xi, b'), \tag{26}$$

$$B(a', b) = A(a', \nu) + A'(\nu, b). \tag{27}$$

There are two cases: $\xi \leq \nu$ and $\xi > \nu$. We present the proof only for the first case; the second case can be treated analogously.

Assuming that $\xi \leq \nu$, then clearly $a \leq \xi \leq b$ and $a' \leq \nu \leq b'$. From (24) it follows that

$$B(a,b) \leq A(a,\xi) + A'(\xi,b), \tag{28}$$

$$B(a',b') \leq A(a',\nu) + A'(\nu,b'). \tag{29}$$

Relations (26), (27), (28) and (29) imply that (25) holds if

$$A(a,\xi) + A'(\xi,b) + A(a',\nu) + A'(\nu,b') \leq A(a,\xi) + A'(\xi,b') + A(a',\nu) + A'(\nu,b), \tag{30}$$

which is equivalent to

$$A'(\xi,b) + A'(\nu,b') \leq A'(\xi,b') + A'(\nu,b). \tag{31}$$

The above inequality is valid because $1 \leq \xi \leq \nu \leq b < b' \leq N$ and the function A' satisfies the convex Monge condition. □

Lemma 2. For each integer $k, 1 \leq k \leq r$, the function $E_k(a,b)$ satisfies the convex Monge condition, if the distortion measure $d(X, Q(X))$ is monotone.

In the proof of Lemma 2 we use the fact that $D(a,b]$, as a function of integers a, b, satisfies the convex Monge condition. We borrow a result of Wu and Zhang [21]. They prove that the function $\epsilon(a,b]$, defined as:

$$\epsilon(a,b] = \min_{y \in \mathbf{R}} \sum_{i=a+1}^{b} d(x_i, y) p(x_i), \tag{32}$$

for all $0 \leq a < b \leq N$, satisfies the convex Monge condition if $d(x,y)$ is monotone. Note that the function $\epsilon(a,b]$ is not identical to $D(a,b]$, because we have according to (6):

$$D(a,b] = \min_{y \in \mathcal{B}} \sum_{i=a+1}^{b} d(x_i, y) p(x_i), \tag{33}$$

for all $0 \leq a < b \leq N$, and $\mathcal{B}$ is strictly included in $\mathbf{R}$. However, an attentive examination of the proof of [21] shows that the result still stands if in the definition (32) the range of y over which the minimum is taken is restricted to a subset of $\mathbf{R}$ which contains all elements of $\mathcal{A}$. Since $\mathcal{A} \subseteq \mathcal{B}$, we conclude that our function $D(a,b]$ satisfies the convex Monge condition, too.

Proof of Lemma 2. The proof is constructed by induction on k. If $k = 1$, then

$$E_1(a,b) = W(r) D(a,b]. \tag{34}$$

Since $D(a, b]$ satisfies the convex Monge condition, clearly $E_1(a, b)$ satisfies the condition, too.

Fix some k, $1 \le k \le r-1$, and assume that the function $E_k(a, b)$ satisfies the convex Monge condition. Applying further the equality (21) and Lemma 1, we conclude that $\hat{D}_k(a, b]$ also satisfies the convex Monge condition. The relation

$$E_{k+1}(a, b) = W(r-k+1)D(a, b] + \hat{D}_k(a, b], \tag{35}$$

implies that $E_{k+1}(a, b)$ is a linear combination of functions satisfying the convex Monge condition, hence clearly it satisfies the condition, too. □

Now we are ready to construct the proof of Proposition 2.

Proof of Proposition 2. Assume that inequality (19) is not satisfied, in other words

$$\xi_k(a, b] > \xi_k(a', b']. \tag{36}$$

Let $\xi' = \xi_k(a, b]$ and $\xi = \xi_k(a', b']$. It follows that $a \le a' \le \xi < \xi' \le b \le b'$. The definition of $\xi_k(a, b]$ implies that

$$E_k(a, \xi) + E_k(\xi, b) \ge E_k(a, \xi') + E_k(\xi', b) \tag{37}$$

Since the function $E_k(\cdot, \cdot)$ satisfies the convex Monge condition (Lemma 2), we have

$$\begin{aligned} E_k(a, \xi') + E_k(a', \xi) &\ge E_k(a, \xi) + E_k(a', \xi'), \\ E_k(\xi, b') + E_k(\xi', b) &\ge E_k(\xi, b) + E_k(\xi', b'). \end{aligned} \tag{38}$$

Summing up the above inequalities yields:

$$\begin{aligned} &E_k(a, \xi') + E_k(\xi', b) + E_k(a', \xi) + E_k(\xi, b') \ge \\ &\quad E_k(a, \xi) + E_k(\xi, b) + E_k(a', \xi') + E_k(\xi', b'). \end{aligned} \tag{39}$$

Relations (37) and (39) imply that

$$E_k(a', \xi) + E_k(\xi, b') \ge E_k(a', \xi') + E_k(\xi', b'), \tag{40}$$

which contradicts the definition of $\xi_k(a', b']$ (recall that $\xi = \xi_k(a', b']$). □

Proposition 2 implies that (21) is equivalent to

$$\hat{D}_k(a, b] = \min_{\xi, \xi_k(a, b-1] \le \xi \le \xi_k(a+1, b]} (E_k(a, \xi) + E_k(\xi, b)) \tag{41}$$

and (22) is equivalent to

$$\xi_k(a,b] = argmin_{\xi,\xi_k(a,b-1]\leq\xi\leq\xi_k(a+1,b]}\{E_k(a,\xi)+E_k(\xi,b)\}. \tag{42}$$

Now we see that the search range for $\xi_k(a,b]$ can be reduced from the interval $[a,b]$ to a much smaller one $[\xi_k(a,b-1],\xi_k(a+1,b]]$. In order to take advantage of this, the computation of $\xi_k(a,b-1]$ and $\xi_k(a+1,b]$ has to be completed before the computation of $\xi_k(a,b]$ starts. The required sequencing can be achieved if the entries of the upper triangular matrices $\hat{D}_k$ and ξ_k (whose entries are $\hat{D}_k(a,b]$ and $\xi_k(a,b]$, $0\leq a\leq b\leq N$, where a is the row index and b is the column index) are computed advancing from the leftmost column to the rightmost one, and inside each column advancing from the bottom to the top. This strategy leads to a surprisingly simple algorithm of computing all $\hat{D}_k(a,b]$ for a given k as described by the following pseudocode.

Algorithm 2.

```
for $a = 0$ to $N$ do
    $\xi_k(a,a] := a$; $\hat{D}_k(a,a] := 0$;
    for $i = 1$ to $a$ do
        $\hat{D}_k(a-i,a] := \min_{\xi,\xi_k(a-i,a-1]\leq\xi\leq\xi_k(a-i+1,a]}(E_k(a-i,\xi)+E_k(\xi,a))$;
        $\xi_k(a-i,a] := argmin_{\xi,\xi_k(a-i,a-1]\leq\xi\leq\xi_k(a-i+1,a]}(E_k(a-i,\xi)+E_k(\xi,a))$;
```

Note that for each pair of integers a, and b, $E_k(a,b]$ is evaluated in constant time using formula (20). Consequently, the evaluation of both $\hat{D}_k(a-i,a]$ and $\xi_k(a-i,a]$ takes $O(\xi_k(a-i+1,a]-\xi_k(a-i,a-1])$ operations if $i>0$, or $O(1)$ operations if $i=0$. The time complexity T_k of Algorithm 2 is:

$$\begin{aligned} T_k &= O(N)+O(\sum_{a=1}^{N}\sum_{i=1}^{a}(\xi_k(a-i+1,a]-\xi_k(a-i,a-1])) \\ &= O(N)+O(\sum_{a=1}^{N}\sum_{i=1}^{a}\xi_k(a-i+1,a]-\sum_{a=0}^{N-1}\sum_{i=1}^{a+1}\xi_k(a-i+1,a]) \\ &= O(N)+O(\sum_{i=1}^{N}\xi_k(N-i+1,N]-\sum_{a=0}^{N-1}\xi_k(0,a])=O(N^2). \end{aligned} \tag{43}$$

Consequently, replacing Step 1 of Algorithm 1 by r invocations of Algorithm 2 (one for each $k=1,2,\cdots,r$) reduces the time complexity of optimal MRQ design from $O(rN^3)$ to $O(rN^2)$.

Before ending this section we mention in passing that the same $O(N^2)$ time complexity for the computation of all values $\hat{D}_k(a,b]$ for a given k, can also be achieved by the fast matrix search technique proposed by Aggarwal *et al.* [1]. Indeed, it can be easily shown that this problem is equivalent to the problem of tube minima in a totally monotone three-dimensional array [2]. Even though our algorithm achieves the same asymptotic complexity, it is much simpler in structure.

But can this simpler algorithm be applied to optimal single-resolution quantization (SRQ) as well? Clearly, SRQ is a special case of MRQ, where the finest resolution has probability 1 ($W(r) = 1$) and all intermediate resolutions have probability 0 ($W(k) = 0$, $1 \leq k < r$). Thus the algorithm presented in this section offers a new solution to designing optimal SRQ of $K = 2^r$ codecells, without using the SMAWK matrix reduction technique introduced by Aggarwal *et al.* [1] for fast matrix search. But its time complexity is higher than SMAWK: $O(rN^2)$ vs. $O(N^2)$ for general monotone distortion function, and vs. $O(KN)$ when the distortion function is the squared Euclidean distance [20, 21]. Interestingly though, if the random variable to be quantized has exponential distribution, applying the simple idea developed in this section in conjunction with the properties of exponential distribution yields an $O(rN)$ time algorithm for optimal MRQ design, and for optimal $K = 2^r$-codecells SRQ design, too. In this case our simple algorithm also has a lower asymptotical time complexity than all previous algorithms of optimal SRQ design. We present this algorithm in Section 7.

5 Space Complexity

Now we discuss the space complexity. For each k, the matrix $\hat{D}_k$, which is formed by the k-th invocation of Algorithm 2, has to be stored until the completion of the $(k+1)$-th invocation. To store all the $O(N^2)$ entries of the matrix, at least $O(N^2 \log_2 N)$ bits are required. In order to reconstruct the underlying partition of the resulting optimal MRQ (Step 2 of Algorithm 1) the algorithm also needs to keep all intermediate values of $\xi_k(a,b]$. If we simply stored all matrices ξ_k, $1 \leq k \leq r$, we would need an additional $O(rN^2 \log_2 N)$ bits, which dominates and determines the space complexity. We can reduce this space complexity by storing the information about $\xi_k(a,b]$ via a compact encoding scheme. Of course this adds an extra time of decoding to Step 2 of Algorithm 1. But since only $2^{r+1} < 2N$ values of $\xi_k(\cdot,\cdot]$ need to be back traced in Step 2, the decoding time is only $O(rN)$, as we will show below, being negligible comparing to the time complexity of $O(rN^2)$ for Step 1.

By Proposition 2, for each k, any row a of the matrix ξ_k has the entries in nondecreasing order. Hence any entry $\xi_k(a,b]$ is either equal to or larger than the previous entry on the row. We use one bit to encode which is the case, and create an $N \times N$ upper triangular matrix Z_k of binary entries $Z_k(a,b)$, $0 \le a \le N-1$, $a+1 \le b \le N$. Namely, $Z_k(a,b) = 0$ if $\xi_k(a,b] = \xi_k(a,b-1]$ and $Z_k(a,b) = 1$ otherwise. Since the value of $\xi_k(a,a+t]$ remains a constant in a range $t = 0, 1, \cdots$, we can compactly encode these ranges. To this end we use another $N \times N$ upper triangular matrix Z'_k of binary entries $Z'_k(a,\xi)$, $0 \le a \le N-1$, $a+1 \le \xi \le N$, such that $Z'_k(a,\xi) = 1$ if $\xi = \xi_k(a,b]$ for some b, $a \le b \le N$, otherwise $Z'_k(a,\xi) = 0$.

Computing the two matrices Z_k and Z'_k incurs almost no cost. Algorithm 2 first initializes all the binary entries to 0. Upon obtaining each $\xi_k(a,b]$, the algorithm sets both $Z_k(a,b)$ and $Z'_k(a,\xi_k(a,b])$ to 1 if $\xi_k(a,b] \ne \xi_k(a,b-1]$. (The value $\xi_k(a,b]$ is not discarded immediately, but only after the computation of $\xi_k(a,b+1]$.)

Aided with Z_k and Z'_k, Algorithm 1 can reconstruct the value $\xi_k(a,b]$ as follows. Given k and a, keep an ordered list of the nonzero entries of matrix Z'_k on row a in increasing column indices. Then $\xi_k(a,b]$ equals the j-th element of this list such that $j = \sum_{t=a+1}^{b} Z_k(a,t)$ (if $j = 0$, then $\xi_k(a,b] = a$). To determine $\xi_k(a,b)$, the algorithm firstly computes the associated j value, which takes $b-a$ additions. Then it finds the j-th nonzero entry on row a of matrix Z_k, which requires at most $b-a$ comparisons (since $\xi_k(a,b] \le b$, only the entries up to $Z_k(a,b)$ are checked). Thus, the time spent to reconstruct any $\xi_k(a,b]$ is only $O(b-a)$.

In Step 2 of Algorithm there are 2^{r+1} quantizer end points to be reconstructed, namely, $\xi_k(q_{i2^k}, q_{(i+1)2^k}]$ for each $k = r, r-1, \cdots, 1$ and each $i = 0, 1, \cdots, 2^{r-k}-1$. Consequently, the total extra time required by the compact encoding scheme to save space is

$$O(\sum_{k=1}^{r} \sum_{i=0}^{2^{r-k}-1} (q_{(i+1)2^k} - q_{i2^k})) = O(\sum_{k=1}^{r} N) = O(rN). \tag{44}$$

This is negligible comparing to the time complexity of Step 1. Hence, the total time complexity of $O(rN^2)$ is not affected.

With the proposed compact encoding scheme we only need to store the two $N \times N$ upper triangular binary matrices Z_k and Z'_k, $1 \le k \le r$, to facilitate step 2 of the algorithm. This space requirement is clearly only $O(rN^2)$ in bits. In addition to the space requirement of $O(N^2 \log_2 N)$ for matrix $\hat{D}_k$, the total space complexity is $O(N^2 \log_2 N + rN^2)$, or $O(N^2 \log_2 N)$ since $r = O(\log N)$.

6 The Preprocessing Step

In this section we consider the computation of the distortions of all convex subsets of alphabet $\mathcal{A}$, i.e. all $D(a, b]$, $0 \leq a \leq b \leq N$. The same task was addressed in [21] and was solved with $O(MN)$ time and space requirements, where M denotes the size of alphabet $\mathcal{B}$. We present here a different method, which achieves the same asymptotic complexity, but is much simpler.

Recall that

$$D(a, b] = \min_{y \in \mathcal{B}} \sum_{i=a+1}^{b} d(x_i, y)p(x_i), \tag{45}$$

for all $0 \leq a \leq b \leq N$. In [21] it is shown that the above minimum is achieved for some $y \in \mathcal{B} \cap [x_{a+1}, x_b]$ (see remarks after Lemma 2). Denote this by $\mu(a, b]$ (in case of multiple points, the largest one is picked). It is also shown in [21] that the function $\mu(a, b]$ of integers a and b is monotone in both a and b, i.e. for any integers a, a', b, b', such that $0 \leq a < b \leq N$, $0 \leq a' < b' \leq N$, $a \leq a'$ and $b \leq b'$, the following inequality holds:

$$\mu(a, b] \leq \mu(a', b']. \tag{46}$$

The above property allows us to compute the values $D(a, b]$ and $\mu(a, b]$ for all $0 \leq a < b \leq N$, in $O(MN)$ time (by using the same idea as in Section 4), provided that the expression $\sum_{i=a+1}^{b} d(x_i, y)p(x_i)$ can be evaluated in constant time for any integers $a, b, 0 \leq a < b \leq N$, and any $y \in \mathcal{B} \cap [x_{a+1}, x_b]$. Let $y_1, y_2, \cdots y_M$ denote the elements of $\mathcal{B}$, listed in increasing order.

Instead of precomputing and storing all the values $\sum_{i=a+1}^{b} d(x_i, y)p(x_i)$ we use the approach of [21] to save time and space. Namely, we compute and store the $(N + 1) \times M$ matrix S with entries $S(b, j)$, $0 \leq b \leq N$, $1 \leq j \leq M$, defined as:

$$S(b, j) = \sum_{i=1}^{b} d(x_i, y_j)p(x_i) \tag{47}$$

if $b > 0$, and $S(0, j) = 0$. Then $\sum_{i=a+1}^{b} d(x_i, y_j)p(x_i)$ can be computed in constant time according to

$$\sum_{i=a+1}^{b} d(x_i, y_j)p(x_i) = S(b, j) - S(a, j), \tag{48}$$

for all $a, b, 0 \leq a < b \leq N$, and $y_j \in \mathcal{B} \cap [x_{a+1}, x_b]$. Mention that the matrix S can be built in $O(MN)$ time since each column j can be incrementally computed in $O(N)$ time.

The method of [21] consists of computing the distortions $D(a, b]$ for fixed a and all $b, a < b \leq N$, by applying the fast matrix search technique. Indeed, this problem is equivalent to the problem of finding all the row minima of an $(N-a) \times M_a$ matrix, which is totally monotone (due to the property of the distortion function $d(x,y)$), where $M_a = M - |\mathcal{B} \cap [x_1, x_{a+1}]|$. Note that $M_a \geq N-a$ since $\mathcal{A} \subseteq \mathcal{B}$. As proved in [1], this problem can be solved in $O(M_a)$ time. Applying this method for each $a, 0 \leq a < N$, the total complexity becomes $O(MN)$ time.

We achieve the same time complexity but in a much simpler way using the idea exposed in Section 4. The inequality (46) implies that

$$D(a,b] = \min_{j, \mu(a,b-1] \leq y_j \leq \mu(a+1,b]} (S(b,j) - S(a,j)), \tag{49}$$

for all $a, b, 0 \leq a < b-1 \leq N-1$. Obviously, $\mu(a,a+1] = a+1$ and $D(a,a+1] = 0$ for any $a, 0 \leq a \leq N-1$. The pseudocode for computing all $D(a,b]$ and $\mu(a,b]$ is the following:

Algorithm 3.

```
for $b = 0$ to $N-1$ do
    $\mu(b,b+1] := b+1$; $D(b,b+1] := 0$;
    for $i = 1$ to $b$ do
        $D(b-i,b+1] := \min_{j,\mu(b-i,b] \leq y_j \leq \mu(b-i+1,b+1]}(S(b+1,j) - S(b-i,j))$;
        $\mu(b-i,b+1] := \max argmin_{j,\mu(b-i,b] \leq y_j \leq \mu(b-i+1,b+1]}(S(b+1,j) - S(b-i,j))$;
```

The time complexity of this algorithm, via a similar analysis as that of Algorithm 2 in Section 4, is $O(MN)$. In most cases of interest $M = O(N)$, hence the preprocessing step does not increase the complexity of the algorithm for optimal MRQ design.

The most widely used distortion function in data compression is the square distance $d(x,y) = (x-y)^2$. In this case, the preprocessing step is even faster. Instead of computing all $O(N^2)$ values $D(a,b]$, only some $O(N)$ quantities are computed in $O(N)$ time. These quantities allow the evaluation of $D(a,b]$ in constant time, every time it is needed [20].

7 Optimal MRQ for Exponential Random Variable

In this section we treat the special case of quantizing an exponential random variable. We will show that for a family of distortion measures, which includes the ubiquitous squared Euclidean distance, the time complexity of the algorithm for optimal MRQ design can be reduced to $O(rN)$. This result presents complexity reduction for optimal SRQ design as well. Indeed, it means that the optimal K-codecells SRQ design for an exponential random variable can be effected in $O(N\log K)$ time, which is the fastest among all known solutions for optimal SRQ design [3, 5, 13, 16, 18, 19, 20, 21]. (The algorithm presented in this section works for the case when the number of codecells is a power of two, but it can be easily extended to the other cases, too.)

We assume the symbols of alphabet $\mathcal{A}$ to be $x_i = \alpha + i\delta$, $1 \le i \le N$, for some real values α, δ, $\delta > 0$. The probability mass function is $p(x_i) = ce^{\lambda i}$ for all $i, 1 \le i \le N$, where λ is a real value, $\lambda \neq 0$, and c is a constant such that $\sum_{i=1}^{N} p(x_i) = 1$.

We also assume that the symbols of alphabet $\mathcal{B}$ are $y_j = \alpha + j\frac{\delta}{m}$, $1 \le j \le M$, where m is a positive integer. The distortion function $d(x, y)$ is assumed to be a nondecreasing function of the absolute distance $|x - y|$, in other words, there is a nondecreasing function $f : \mathbf{R} \to [0, \infty)$, such that $d(x, y) = f(|x - y|)$, for all real x and y. Note that under this assumption the distortion function $d(x, y)$ is monotone, hence all the results obtained in the previous sections hold.

The reduction in complexity of optimal MRQ design from $O(rN^2)$ to $O(rN)$ follows from the observation that there is no longer the need to evaluate the quantities $\hat{D}_k(a, b]$ for all pairs of integers a, b, $0 \le a \le b \le N$, but only for the pairs with $a = 0$. This property is a consequence of the following proposition.

Proposition 3. For any integers k, a, b, $1 \le k \le r, 0 \le a \le b \le N$, the following equalities hold:

$$\hat{D}_k(a, b] = e^{\lambda a}\hat{D}_k(0, b - a], \tag{50}$$

$$\xi_k(a, b] = a + \xi_k(0, b - a]. \tag{51}$$

Proof.

The proof proceeds in two steps. The first step is to show that the equality

$$D(a, b] = e^{\lambda a} D(0, b - a] \tag{52}$$

is valid for all integers $0 \le a \le b \le N$. The second step is to prove by induction on k, that (50) and (51) hold, too.

Starting from the definition of $D(a, b]$ (45) and the observation mentioned in Section 6 that $\mu(a, b] \in \mathcal{B} \cap [x_{a+1}, x_b]$, the following sequence of equalities follows:

$$\begin{aligned} D(a, b] &= min_{y \in \mathcal{B} \cap [x_{a+1}, x_b]} \sum_{i=a+1}^{b} f(|y - \alpha - i\delta|) \cdot ce^{\lambda i} \\ &= e^{\lambda a} \cdot min_{y \in \mathcal{B} \cap [x_{a+1}, x_b]} \sum_{j=1}^{b-a} f(|y - \alpha - a\delta - j\delta|) \cdot ce^{\lambda j}. \end{aligned} \tag{53}$$

¿From the way the symbols of alphabets $\mathcal{A}$ and $\mathcal{B}$ were defined it follows that $y \in \mathcal{B} \cap [x_{a+1}, x_b]$ if and only if $y - a\delta \in \mathcal{B} \cap [x_1, x_{b-a}]$. By a change of variable $y' = y - a\delta$, we have

$$\begin{aligned} D(a,b] &= e^{\lambda a} \cdot min_{y' \in \mathcal{B} \cap [x_1, x_{b-a}]} \sum_{j=1}^{b-a} f(|y' - \alpha - j\delta|) \cdot ce^{\lambda j} \\ &= e^{\lambda a} D(0, b-a], \end{aligned} \tag{54}$$

concluding the first step of the proof.

We now prove Proposition 3 by induction on k. Let $k = 1$ and a, b be arbitrary integers such that $0 \leq a \leq b \leq N$. Then, from Proposition 1 it follows that

$$\hat{D}_1(a,b] = \min_{\xi, a \leq \xi \leq b} W(r)(D(a,\xi] + D(\xi, b]). \tag{55}$$

Replacing $D(a,\xi]$ and $D(\xi, b]$ according to (52) yields

$$\begin{aligned} \hat{D}_1(a,b] &= \min_{\xi, a \leq \xi \leq b} W(r)(e^{\lambda a} D(0, \xi - a] + e^{\lambda \xi} D(0, b - \xi]) \\ &= \min_{\xi, a \leq \xi \leq b} W(r)(e^{\lambda a} D(0, \xi - a] + e^{\lambda a} e^{\lambda(\xi - a)} D(0, b - \xi]) \\ &= e^{\lambda a} \min_{\xi, a \leq \xi \leq b} W(r)(D(0, \xi - a] + D(\xi - a, b - a]) \\ &= e^{\lambda a} \min_{\mu, 0 \leq \mu \leq b-a} W(r)(D(0, \mu] + D(\mu, b - a]) \\ &= e^{\lambda a} \hat{D}_1(0, b-a]. \end{aligned} \tag{56}$$

The second last equality in the above sequence is obtained by replacing $\xi - a$ by μ, which also implies that

$$\xi_1(a,b] = a + \xi_1(0, b-a]. \tag{57}$$

Thus the verification step of the inductive proof is completed. The inductive step $k \to k+1$ follows easily using the same idea and we omit the proof. □

A direct consequence of Propositions 1 and 3 is the following recursive formula:

$$\begin{aligned} \hat{D}_k(0,a] = \min_{\xi, 0 \leq \xi \leq a} \{ & W(r-k+1)(D(0,\xi] + e^{\lambda \xi} D(0, a - \xi]) \\ & + \hat{D}_{k-1}(0,\xi] + e^{\lambda \xi} \hat{D}_{k-1}(0, a-\xi] \} \end{aligned} \tag{58}$$

for all $1 \leq k \leq r$ and $0 \leq a \leq N$. On the other hand, Proposition 2 implies that

$$\xi_k(0, a-1] \leq \xi_k(0,a] \leq \xi_k(1,a] \tag{59}$$

for all $1 \leq k \leq r$ and $1 \leq a \leq N$. Moreover, Proposition 3 implies $\xi_k(1,a] = 1 + \xi_k(0, a-1]$. Summing up the above observations leads to the following recursion

$$\begin{aligned} \hat{D}_k(0,a] = \min_{\xi \in \{\xi_k(0,a-1], 1+\xi_k(0,a-1]} \{ & W(r-k+1)(D(0,\xi] + e^{\lambda \xi} D(0, a - \xi]) \\ & + \hat{D}_{k-1}(0,\xi] + e^{\lambda \xi} \hat{D}_{k-1}(0, a-\xi] \} \end{aligned} \tag{60}$$

for all $1 \leq k \leq r$ and $1 \leq a \leq N$.

Using the recursion above the minimal expected distortion $\bar{D}(\hat{\mathbf{Q}})$, or $\hat{D}_r(0, N]$, can be obtained by recursively computing all values $\hat{D}_k(0, a]$ and $\xi_k(0, a]$, $1 \leq k \leq r$ and $0 \leq a \leq N$, in increasing order of k and a. According to (60) the computation of each such value requires constant time, hence the whole process takes $O(rN)$ time.

The space complexity of this algorithm for exponential random variable is also decreased by a factor of N comparing to the general algorithm. Indeed, the matrix $\hat{D}_k$ which has to be stored at each current value of k has the dimension $1 \times (N+1)$. Also for each $k, 1 \leq k \leq r$, the matrix Z_k with binary entries which encodes the information about the values $\xi_k(0, a]$, has only N entries: $Z_k(0, a)$, $1 \leq a \leq N$. Mention that the matrix Z'_k is no longer needed (since $\xi_k(0, a]$ is either $\xi_k(0, a-1]$ or $1 + \xi_k(0, a-1]$, and $\xi_k(0, a]$ is increasing in a, it follows that $\xi_k(0, a] = \sum_{i=1}^{a} Z_k(0, a)$, $1 \leq a \leq N$). Hence the space requirement amounts to $O(N \log_2 N + rN) = O(N \log_2 N)$ bits.

8 Generalization to A Graph Problem

The design of optimal K-codecells single-resolution quantizer is an instance of the problem of finding a minimum-weight K-link path in a directed acyclic graph (DAG) [3]. Conversely, we can generalize optimal MRQ design to a graph problem. First, let us introduce a so-called multi-edge-sets weighted directed acyclic graph (MEWDAG), denoted by $\mathcal{G} = (V, E_1, \omega_1, \cdots E_r, \omega_r)$, where for each k, $1 \leq k \leq r$, $G_k = (V, E_k, \omega_k)$ is a weighted directed acyclic graph, with the set V of vertices, the set E_k of edges, and the function ω_k assigning weights to edges, $\omega_k : E_k \rightarrow \mathbf{R}$. Moreover, the topological order of the set V is the same in all component graphs $G_k, 1 \leq k \leq r$. V is called the vertex set of the MEWDAG $\mathcal{G}$. Let $v_0, v_1, \cdots, v_n$ be the vertices of the graph, in topological order. We call an r-layered embedded path in $\mathcal{G}$ any sequence of paths $\mathcal{P} = (P_1, P_2, \cdots, P_r)$, where each P_k is a path from v_0 to v_n in graph G_k, and for any k, $1 \leq k \leq r-1$, and any link (v_i, v_j) of P_k, there is a subpath of P_{k+1} from node v_i to node v_j (this subpath is called the expansion of the link (v_i, v_j)). We define the weight of the r-layered embedded path $\mathcal{P}$, denoted by $\omega(\mathcal{P})$, to be the sum of the weights $\omega_k(P_k)$ of the component paths:

An interesting problem associated with $\mathcal{G}$ is the r-layered embedded path of minimum weight, called the **minimum-weight r-layered embedded path** problem. Some examples of the applications of the minimum-weight layered embedded path are optimal multi-resolution piecewise approximation of a discrete signal, and optimal entropy-constrained multi-resolution quantization [7]. The $O(rN^3)$ dynamic programming algorithm of [7] for the latter problem can be generalized to solve the graph problem of the minimum-weight layered embedded path. Of close relevance to optimal fixed-rate MRQ of r refinement levels is a more restrictive variant of minimum-weight layered embedded path, as stated below.

Problem 3 (bifurcate minimum-weight r-layered embedded path). Let $\mathcal{G} = (V, E_1, \omega_1, \cdots E_r, \omega_r)$ be an MEWDAG. Find the r-layered embedded path $\mathcal{P} = (P_1, P_2, \cdots, P_r)$ of minimum weight, which satisfies the additional constraint that the path P_1 contains at most two links and for any $k, 1 \leq k \leq r-1$, the expansion of any link (v_i, v_j) of path P_k has at most two links, too.

Clearly, optimal MRQ design (Problem 2) is an instance of the bifurcate minimum-weight r-layered embedded path problem. The corresponding MEWDAG is $\mathcal{G} = (V, E_1, \omega_1, \cdots E_r, \omega_r)$, where $V = \{v_0, v_1, \cdots, v_N\}$, $E_k = \{(v_i, v_j) | 0 \leq i < j \leq N\}$ and $\omega_k(v_i, v_j) = W(k)D(i, j]$, for all $0 \leq i < j \leq N$ and $1 \leq k \leq r$. Note that the r component graphs $G_k = (V_k, E_k, \omega_k)$ share not only the same vertex set, but also the same edge set, which is the maximal possible given the topological order (i.e., all these r DAG's are complete and the corresponding MEWDAG $\mathcal{G}$ is also said to be complete).

In each component graph G_k, each edge (v_i, v_j) corresponds to a subset of alphabet $\mathcal{A}$, namely $c(i, j]$. Hence to each path in the graph G_k corresponds a partition of the alphabet $\mathcal{A}$ into nonempty convex sets, and the correspondence is one-to-one. From the discussion in Section 3 it follows that any quantizer can be identified with the partition of alphabet $\mathcal{A}$ consisting of the quantizer's nonempty codecells (because the distortion of the quantizer depends only on its nonempty codecells and not on the way they are indexed; also recall that the codecells are convex sets). It follows that there is a one-to-one correspondence between the paths of the graph G_k and the quantizers, and the weight of each path is equal to the distortion of the corresponding quantizer, multiplied by $W(k)$.

Let now Q be a quantizer of rate r (i.e. with at most 2^r nonempty codecells), and let $\hat{\mathbf{Q}} = (Q_1, Q_2, \cdots, Q_r)$ be the MRQ induced by Q. Let P_k be the path in the graph G_k corresponding to quantizer Q_k, for each $k, 1 \leq k \leq r$. Each path P_k has as many links as nonempty codecells of quantizer Q_k, i.e. at most 2^k (recall that the rate of Q_k equals k). The condition (2) is equivalent to the condition that for any $k, 1 \leq k \leq r-1$, and any link (v_i, v_j) of path P_k, there is a subpath of P_{k+1} between nodes v_i and v_j and it has at most two links. Thus, it follows that $\mathcal{P} = (P_1, P_2, \cdots, P_r)$ is an r-layered embedded path satisfying the constraint in Problem 3. Moreover, the weight of this r-layered embedded path, $\omega(\mathcal{P})$, equals the expected distortion of MRQ $\hat{\mathbf{Q}}$:

$$\omega(\mathcal{P}) = \sum_{k=1}^{r} \omega_k(P_k) = \sum_{k=1}^{r} W(k)D(Q_k) = \bar{D}(\hat{\mathbf{Q}}), \tag{61}$$

Conversely, if $\mathcal{P} = (P_1, P_2, \cdots, P_r)$ is an r-layered embedded path satisfying the conditions in Problem 3, and Q_k is the quantizer corresponding to each path P_k, $1 \le k \le r-1$, it follows that the sequence of quantizers $(Q_1, Q_2, \cdots, Q_r)$ is an MRQ, namely the MRQ induced by quantizer Q_r.

The equivalence we have shown between the problem of optimal MRQ design (Problem 2) and the bifurcate minimum-weight r-layered embedded path problem for the MEWDAG $\mathcal{G}$, allows the generalization of the algorithms presented in this paper to solve Problem 3. Namely, the following proposition holds.

Proposition 4. Let $\mathcal{G} = (V, E_1, \omega_1, \cdots E_r, \omega_r)$ be an r-edge set WDAG, with the vertex set $V = \{v_0, v_1, \cdots v_N\}$, the nodes being indexed in topological order. Then Problem 3 can be solved in $O(rN^3)$ time. Moreover, if all the component WDAG's G_k, $1 \le k \le r$, are complete and satisfy the convex Monge condition, i.e.

$$\omega_k(i,j) + \omega_k(i',j') \le \omega_k(i,j') + \omega_k(i',j), \text{ for all } 0 \le i < i' \le j < j' \le N \text{ and all } k, \tag{62}$$

then Problem 3 can be solved in $O(rN^2)$ time. (In the above relations $\omega_k(i,j)$ is a shortened notation for $\omega_k(v_i, v_j)$ if $i < j$, and $\omega_k(i,j) = 0$ if $i = j$.)

Proof. Note first that the MEWDAG $\mathcal{G}$ may be assumed to be complete (otherwise it can be extended to a complete one simply by assigning the infinite value to the weights $\omega_k(i,j)$ for the pairs (v_i, v_j) which are not edges of G_k, without changing the solution of Problem 3). Then Algorithm 1 can be applied to solve Problem 3, where recursion (17) is replaced by:

$$\hat{D}_k(a,b] = \min_{\xi, a \le \xi \le b} \{\omega_{r-k+1}(a,\xi) + \omega_{r-k+1}(\xi,b) + \hat{D}_{k-1}(a,\xi] + \hat{D}_{k-1}(\xi,b]\}. \tag{63}$$

Note that the whole development of Section 4, which leads to the complexity reduction of optimal MRQ design, henges on the fact that the function $D(a,b]$ satisfies the convex Monge condition. In order to extend this to the algorithm for the graph problem, each weighting function $\omega_k(\cdot,\cdot)$, $1 \le k \le r$, must satisfy the convex Monge condition. Hence, if condition (62) is fulfilled, then the idea of Section 4 can be applied to solve Problem 3 in $O(rN^2)$ time, too. □

9 Conclusion

We present a simple algorithm of $O(rN^2)$ time and $O(N^2 \log N)$ space complexity to design an optimal multi-resolution quantizer of r refinement levels (or of bit rate r in case of data compression) for a very general class of distortion measures, where N is the size of alphabet of the input discrete random variable. The simplicity and relatively high efficiency of the proposed algorithm hinge on the convex Monge property of the underlying objective function. Our algorithm is simpler than the SMAWK matrix search technique, which is the best existing solution to the quantization problem. Moreover, in the case of exponential random variable, the time and space complexity of optimal MRQ design can be reduced to $O(rN)$ and $O(N \log N)$ respectively.

The proposed algorithm also offers a new simple solution to the conventional problem of designing optimal single-resolution quantizer. In the case of exponential random variable, this solution has lower time and space complexity than the best existing algorithms.

We also generalize the problem of optimal multi-resolution quantization to a new graph problem, which can be solved by our $O(rN^2)$ time algorithm.

Acknowledgement

We sincerely thank the reviewers for their constructive comments and suggestions that significantly enriched the contents and improved the presentation of this paper.

References

[1] A. Aggarval, M. Klave, S. Moran, P. Shor, and R. Wilber, " Geometric applications of a matrix-searching algorithm", *Algorithmica*, 2(1987), pp.195-208.

[2] A. Aggarwal and J. Park, "Notes on Searching in Multidimensional Monotone Arrays", *Proc. of FOCS'88*, pp.497-512.

[3] A. Aggarwal, B. Schieber and T. Tokuyama, "Finding a minimum-weight k-link path in graphs with the concave Monge property and applications", *Discrete and Computational Geometry*, vol. 12, pp. 263-280, 1994.

[4] A. Apostolico and Z. Galil(eds.), *Pattern Matching Algorithms*, New York 1997.

[5] J. D. Bruce, "Optimum quantization", Sc. D. thesis, M. I. T., May 14, 1964.

[6] H. Brunk and N. Farvardin, "Fixed-rate successively refinable scalar quantizers," *Proc. IEEE Data Compression Conference*, pp. 250-259, Apr.1996.

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

[8] M. Effros, "Practical Multi-Resolution Source Coding: TSVQ Revisited", *Proc. DCC'98*, Snowbird, Utah, March 1998.

[9] M. Effros, ”Distortion-Rate Bounds for Fixed- and Variable-Rate Multiresolution Source Codes”, *IEEE Trans. Inform. Theory*, vol. 45, pp. 1887-1910, Sept. 1999.

[10] M. Effros and D. Muresan, ”Codecell Contiguity in Optimal Fixed-Rate and Entropy-Constrained Network Scalar Quantizers”, *Proc. IEEE Data Compression Conference*, pp. 312-321, April 2002.

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

[12] M. R. Garey, D. S. Johnson, and H. S. Witsenhausen, ”The complexity of the generalized Lloyd-Max problem”, *IEEE Trans. Inform. Theory*, vol. 28, no. 2, pp. 255-256, 1982.

[13] D. Greene, F. Yao, and T. Zhang, ”A linear algorithm for optimal context clustering with application to bi-level image coding” *IEEE ICIP98*, pp. 508-511.

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

[16] J. Max, ”Quantizing for minimum distortion”, *IRE Trans. Inform. Theory*, vol. IT-6, pp. 7-12, Jan. 1960.

[17] D. Muresan and M. Effros, ”Quantization as Histogram Segmentation: Globally Optimal Scalar Quantizer Design in Network Systems”, *Proc. IEEE Data Compression Conference*, pp. 302-311, April 2002.

[18] B. Schieber, ”Computing a minimum-weight k-link path in graphs with the concave Monge property”, *Proc. ACM-SIAM Symp. on Algorithms’95*, pp. 405-411.

[19] D. K. Sharma, ”Design of absolutely optimal quantizers for a wide class of distortion measures”, *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 693-702, Nov. 1978.

[20] X. Wu, ”Optimal Quantization by Matrix Searching”, *Journal of Algorithms* 12(1991), pp. 663-673.

[21] X. Wu and K. Zhang, ”Quantizer monotonicities and globally optimal scalar quantizer design”, *IEEE Trans. Inform. Theory*, vol. 39, pp. 1049-1053, May 1993.

[22] X. Wu, P. Chou and X. Xue, “Minimum Conditional Entropy Context Quantization”, *Proc. of the 2000 IEEE International Symposium on Information Theory*, Naples, Italy, June 2000.

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