EFFICIENT QUANTIZATION OF STATISTICALLY NORMALIZED VECTORS USING MULTI-OPTION PARTIAL-ORDER BIT-ASSIGNMENT SCHEMES

Sean A. Ramprashad

DoCoMo Communications Labs, USA
San Jose, CA, USA.

ABSTRACT

In this paper we focus on new options for the efficient quantization of statistically normalized target vectors at low bitrates. This problem is fundamental to many low-rate speech and audio coder designs. Here many such coders follow a general principle of taking a structured speech or audio signal, applying a process of redundancy removal and then quantizing each of the resulting statistically normalized targets to a relevant distortion level.

We look at this latter problem when some of these targets are to be quantized at very low bitrates ($\leq$ 1 bit/target-scalar). The approach we take is to efficiently communicate a target-adaptive pattern of unequal bit-assignments (noise allocations) across each target. This can increase performance over an approach that has a constant noise allocation even when target vectors consist of independent and identically distributed (i.i.d.) scalars. We extend these schemes to multi-option schemes allowing further options to adapt and improve performance.

1. INTRODUCTION

In this paper we are interested in the problem of quantizing statistically normalized target vectors at very low bitrates. This problem is of underlying interest to many low-rate speech and audio coder designs. Here many such coder designs follow the general principle of taking an input signal frame "$\mathbf{s}$" and first applying a process of redundancy removal and possibly a joint or subsequent process for irrelevancy removal. In the final coding stage the encoder quantizes the resulting statistically normalized targets "$\mathbf{x}$", each with a given number of bits and/or to a given distortion level. See for example the general illustration of the process in [1].

For good reasons speech and audio coding research has tended to be dominated by advancements in the former processes of redundancy and irrelevancy removal. With these advancements well-known classic quantization techniques [2] are often used to quantize the resulting statistically normalized targets. However, to further improve coder performance it is becoming increasingly important to revisit the latter process and to possibly define new quantization techniques [3][4]. Of particular interest are techniques that optimize not only objective performance, such as minimizing the mean square error (MSE) as done with many classic techniques, but techniques which also have important practical attributes such as an inherent match to perceptual coding principles, low complexity, the ability to control the allocation of noise, etc.

For example in [3] an approach is presented to quantize Modified Discrete Cosine Transform (MDCT) coefficients which exploits both simplifications to high-dimensional vector quantizers (VQs) and the statistical nature of such coefficients as seen with audio. In [4] trade-offs in the relative quantization of phase and magnitude in Fourier representations are explored thus addressing a link to auditory perception.

In this paper we focus on a different approach and problem. The approach can be used to quantize frequency domain coefficients (as in the prior work just mentioned) and/or residuals in linear predictive coders. However the problem we will address is the case in which our target vectors of interest "$\mathbf{x}$" have little or no structure. The assumption is that most if not all of the non-i.i.d. properties of "$\mathbf{s}$"have been already exploited and removed by early redundancy removal steps (e.g. by an analysis stage in linear prediction and/or gain normalization of transform coefficients in subbands). In the results of this paper we assume an extreme case where $\mathbf{x}$ consists of i.i.d. real scalars, though our earlier work does look at designs for speech and audio that relax this assumption[1].

Our approach is to divide targets into sub-vectors and to use and communicate (to the decoder) an explicit signal-adaptive pattern of unequal bit-assignments (noise-allocations) to these sub-vectors. The use of an explicit adaptive allocation is an interesting feature of our approach that does distinguish it from other approaches. Unlike many high-dimensional VQ approaches where bits are essentially assigned to the entire vector in a single unit, or assigned in a specific non-adaptive pattern, the use of an explicit signal-adaptive bit-allocation to sub-vectors allows the encoder and decoder to control and know the noise-allocation across the vector. The key issue addressed in our approach is how to communicate this adaptation efficiently. Also in contrast to variable-length quantization schemes that often use a uniform noise allocation, e.g. through use of a uniform scalar quantizer combined with Huffman coding of quantization indices, our scheme fixes the total bit-allocation and uses an unequal noise allocation within the vector. These features can allow the encoder to select bit-assignment patterns based on a variety of criteria besides MSE, e.g. considering masking effects within the vector.

Our prior work[1] looked at some initial schemes and potential benefits of the approaches when applied to MDCT-based coding of speech and audio. The focus was mainly on the sparse nature of bit assignments at low bitrates for an MSE criteria. It is shown that schemes with good MSE performance often concentrated bit-assignments into a few sub-vectors in each target vector. This effect can be seen later in Table 2. In this paper we extend the schemes to multi-option schemes where the bit-pattern has further degrees of freedom to adapt to targets "$\mathbf{x}$". In this paper we focus on the MSE criterion simply to illustrate the general principle. However, other criteria such as minimizing the MSE subject to a constraint on how sparse an allocation is can also be considered in the framework.

The paper is organized as follows. In Section 2 we review the basic concept of a partial-order scheme, showing how partial orders are specified and used in quantization. In Section 3 we show how to design single-class schemes and then move on to multi-class schemes in Section 4. Examples of multi-class designs and benchmark designs are presented and compared in Section 5. The paper concludes with a discussion in Section 6.

[1] Under review by the *IEEE Trans. on Speech and Audio Processing*

2. A REVIEW OF PARTIAL ORDERING SCHEMES

We focus on the underlying problem of quantizing a target vector $\mathbf{x}$ of dimension N given a total bit budget of B bits for quantization. Here, for example, $\mathbf{x}$ could consist of MDCT coefficients from a single scale-factor band in a transform or transform-predictive speech/audio coder [1][6]. Such cases that underly the motivation of this work.

It is assumed that B is known both to the encoder and decoder and determined by earlier stages in the coding algorithm. The vector $\mathbf{x}$ is assumed to be a statistically normalized with little or no additional structure that can be leveraged by techniques such as linear predictive modelling, transforms, etc. We will focus on the illustrative case of vectors of i.i.d. Laplace random scalars. Often normalized MDCT coefficients of audio can have marginal distributions very close to that of Laplace distributions [1].

The underlying premise of Partial Ordering Schemes is that within a limited sampling (group) of random vectors or scalars there can be enough statistical variation to motivate an unequal assignment of quantization resources across the group at low bit-rates. This is true even when scalars/vectors are i.i.d. We look at the case where a group of q such i.i.d. p-dimensional sub-vectors $\mathbf{x} = \{\mathbf{x}_0, \ldots, \mathbf{x}_{q-1}\}$, with $N = qp$, is to be quantized with a total of B bits. The approach we will take is to quantize each sub-vector $\mathbf{x}_k$ with a p-dimensional fixed-rate VQ given a signal adaptive assignment of $b_k(\mathbf{x}) \leq B$ bits to that sub-vector.

Here it is worth noting that if we simply wanted to minimize the average MSE the best approach would be to design a straight-forward single-stage B-bit N-dimensional VQ. However such an approach may not be practical. For example if $N = 32$ and $B = 24$ (0.75 bits/scalar) the codebook required consists of 2^{29} scalar entries which can be prohibitively large to store or search. Also, even if parameterized VQs, such as multi-stage VQs or trellis structured VQs, are of interest we have broader goals of wanting to know, control and communicate to the decoder the distribution of bits (noise) across the vector $\mathbf{x}$.

If sub-vectors $\mathbf{x}_0, \ldots, \mathbf{x}_{q-1}$ are i.i.d. then one simple "Benchmark" strategy to quantize the sub-vectors with respect to an MSE criterion is to divide the B bits up as equally as possible among vectors with e_k bits assigned to $\mathbf{x}_k$:

$$\text{Benchmark:} \quad e_k = \lceil B/q \rceil \text{ or } \lfloor B/q \rfloor \text{ and } \textstyle\sum_{k=0}^{q-1} e_k = B \tag{1}$$

The $\lceil\ \ \rceil$ and $\lfloor\ \ \rfloor$ allow for consideration of cases where B/q is not an integer.

Each sub-vector is then quantized with an appropriate e_k-bit VQ codebook, and the noise-allocation across $\mathbf{x}$ is essentially constant. In fact, it is shown that given random scalars $y_0, \ldots, y_{n-1}$ under a *high resolution approximation*, the optimum bit-allocation for an MSE criterion has the form [2, Chapter 8, equation 8.3.4]:

$$\begin{aligned} b_k &= \bar{b} + \delta_k - \frac{1}{2} log_2\left(\rho^2 H\right) \quad \text{with } \delta_k = \tfrac{1}{2} log_2\left(\sigma_k^2 h_k\right) \\ h_k &= (1/12)\left(\int_{-\infty}^{\infty} [f_k(y)]^{1/3}\, dy\right)^3 \end{aligned} \tag{2}$$

Here b_k is the assignment to y_k, σ_k^2 is the variance of y_k, $f_k(y)$ is the PDF of y_k/σ_k, ρ^2 is the geometric mean of values $\sigma_0^2, \ldots, \sigma_{n-1}^2$, and H is the geometric mean of values $h_0, \ldots, h_{n-1}$.

If $y_0, \ldots, y_{n-1}$ are i.i.d., then $b_j = b_k$ for all j, k motivating the equal allocation of (1). However, at low bitrates (which violates the high resolution approximation) an equal assignment may not be the best. Also, (2) assumes a *fixed* assignment that does not change for any realization of the random scalars. Making the assignment adaptive to the actual values of $y_0, \ldots, y_{n-1}$ (similarly to $\mathbf{x}_0, \ldots, \mathbf{x}_{q-1}$) may improve performance.

2.1. Defining a Partial Order

Partial ordering schemes allow for an adaptive unequal bit assignment by arranging target vectors into groups and defining the assignment as a function of this grouping. To do this each realization of targets $\mathbf{x}_0, \ldots, \mathbf{x}_{q-1}$ are arranged into a number of *ordered* non-overlapping groups. If we have s ordered groups $\mathcal{G}_0, \ldots, \mathcal{G}_{s-1}$ with n_j sub-vectors in $\mathcal{G}_j$ (with $\sum_{k=0}^{s-1} n_j = q$) one can specify this ordered arrangement with $\hat{b}_{over}$ bits of side information. Here:

$$\hat{b}_{over} \quad = \quad \lceil log_2(q!) - \textstyle\sum_{j=0}^{s-1} log_2(n_j!) \rceil \tag{3}$$

Since only the membership in groups are defined, not the internal order inside of the groups, we have an incomplete ordering of the targets $\mathbf{x}_{k_0}, \ldots, \mathbf{x}_{k_{q-1}}$, hence the term "Partial Ordering". Specifying a partial order requires less bits than a full ordering which would require and overhead of $\lceil log_2(q!) \rceil$ bits.

To further reduce the overhead of defining a partial order, a general two-stage scheme to ordering is used. In one such approach the first-stage arranges sub-vectors into u non-overlapping groups of c sub-vectors each. Since our sub-vectors are i.i.d. we use in this paper a simple scheme of grouping consecutive sub-vectors where a first-stage group $\mathcal{F}_k = \{\mathbf{x}_{ck}, \ldots, \mathbf{x}_{ck+c-1}\}$. In general any deterministic grouping (which is known to, and thus can be inverted at, the decoder) can be used. The second stage then partially orders these first-stage groups.

Define a full ordering of the groups as $\mathcal{F}_{j_0}, \mathcal{F}_{j_1}, \ldots, \mathcal{F}_{j_{u-1}}$. With n_j/c an integer for all j and $q/c = u = \sum_{j=0}^{s-1} n_j/c$, one can specify a partial order for the two-stage approach with only

$$b_{over} \quad = \quad \lceil log_2((q/c)!) - \textstyle\sum_{j=0}^{s-1} log_2((n_j/c)!) \rceil \text{ bits} \tag{4}$$

For example, a grouping of these fully ordered first-stage groups into 3 second-stage groups could be:

$$\overset{\textbf{\textit{There is no internal order within each group}}}{\underbrace{\{\mathcal{F}_{j_0}, \ldots, \mathcal{F}_{j_{n_0/c-1}}\}}_{\mathcal{G}_0} \underbrace{\{\mathcal{F}_{j_{n_0/c}}, \ldots\}}_{\mathcal{G}_1}, \underbrace{\{\mathcal{F}_{j_{\frac{n_0+n_1}{c}}}, \ldots, \mathcal{F}_{j_{u-1}}\}}_{\mathcal{G}_2}} \tag{5}$$

2.2. Using a Partial Order for Quantization

A partial order scheme uses b_{over} bits to specify the order and then takes the remaining $R = B - b_{over}$ bits and assigns them to each second-stage group with group $\mathcal{G}_k$ getting B_k bits. Here $\sum_k B_k = R$ and $B_k \geq 0$. The B_k bits are then divided as equally as possible within a group with each $\mathbf{x}_i \in \mathcal{G}_k$ being assigned either

$$\hat{b}_i = \lceil B_k/n_k \rceil \text{ or } \lfloor B_k/n_k \rfloor \text{ bits with } \textstyle\sum_{\mathbf{x}_i \in \mathcal{G}_k} \hat{b}_i = B_k \tag{6}$$

The effective adaptation in assignment to $\mathbf{x}$ is through the adaptation in the grouping communicated through the overhead b_{over}.

It can be shown that at some bitrates (e.g. sufficiently small B) a good selection of B_k, s, c, and $n_0, \ldots, n_{s-1}$ can achieve an average MSE performance that is better than the equal assignment in (1). To understand why such a partial order is often sufficient to provide some gain, let us take the case of $c = 1$ ($\mathcal{F}_k = \mathbf{x}_k$) where the *full* ordering is defined by energy with

$$\mathbf{z}_i \quad = \quad \mathbf{x}_{k_i} \quad \text{where} \quad \|\mathbf{x}_{k_i}\|^2 \leq \|\mathbf{x}_{k_{i+1}}\|^2 \quad \forall i \tag{7}$$

We now look at the examples of δ_k, σ_k, and h_k as defined in (2) for these sorted variables $\mathbf{z}_k$. To estimate these values 10^6 realizations of $\mathbf{x}_0, \ldots, \mathbf{x}_{q-1}$ (i.i.d. Laplace random scalars) are generated in Matlab, sorted to make $\mathbf{z}$ and then used to estimate the values. The resulting δ_k values, the effective difference in predicted bit-assignments $\mathbf{b}_k$ in (2), are shown in Figure 1. The figure shows that the predicted assignment increases with k.

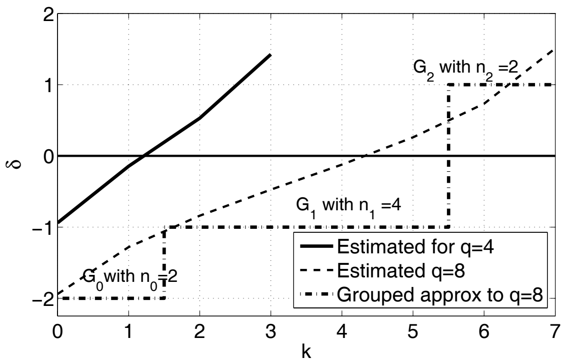

Figure 1: $\check{\delta}_k$ for $\mathbf{z}_k$ being sorted scalar ($p=1$) Laplace rv's.

One way to think of an unequal assignment based on groupings (partial orderings) is as a piecewise linear approximation to assignments as in (2). See the Figure 1. Another is as assignments that define how "Typical" a sub-vector is relative to an average behavior. See [5] for a different (though related) discussion on "Typical" sets. Both views are illustrative only and not entirely the case. Also, for low bitrates one also has to consider setting lower allocations to zero as done in reverse water-filling [5, Section 13.3.3]. The next section will describe how to select good groupings and bit-assignment patterns.

A final point to mention is the criterion/algorithm defining the groupings, i.e. which first-stage groups get put in which second-stage groups. For a bit-assignment as in (6) one could in fact check every possible group-assignment for a given $\mathbf{x}$ for the one with the best performance. Following however the observation in (7), the full ordering of $\mathcal{F}_{j_0}, \ldots, \mathcal{F}_{j_{u-1}}$ is defined in terms of increasing energy, where $\|\mathcal{F}_k\|^2 = \sum_{\mathbf{x}_i \in \mathcal{F}_k} \|\mathbf{x}_i\|^2$ and $\|\mathcal{F}_{j_k}\|^2 \leq \|\mathcal{F}_{j_{k+1}}\|^2$. The $\mathcal{F}_k$'s are then assigned to the $\{\mathcal{G}_k\}$ in this order as in (5), and it is assumed that $\mathcal{G}_k$ gets proportionally more (or equal) bits than $\mathcal{G}_j$ if $k > j$.

3. DESIGNING AND USING SINGLE-OPTION SCHEMES

For a given grouping option (s, c, and $n_0, \ldots, n_{s-1}$), a given dimension q and a selected number of bits B, the designer selects (offline) a fixed pattern of assignments $B_0, \ldots, B_{s-1}$. This pattern does not adapt and is known both to the encoder and decoder. What does adapt, of course, for each $\mathbf{x}$ is the membership of each group (i.e. mappings of targets $\mathbf{x}_i \in \mathcal{G}_k$). In selecting the assignment pattern the only constraint the designer has is $\sum_j B_j = B - b_{over} = R$, and there are

$$\sum_{k_1=0}^{B} \sum_{k_2=0}^{B-k_1} \cdots \sum_{k_{s-1}=0}^{B-k_{s-2}} 1 \tag{8}$$

possible combinations of such assignments that meet this constraint. For a given criterion (e.g. MSE) finding the best bit assignment with a fixed s, c and $n_0, \ldots, n_{s-1}$ can be done for small B by an exhaustive search checking the performance for each option over a test set. Selection of the best scheme then considers a search over all n, c and $n_0, \ldots, n_{s-1}$.

To speed the search some bit-assignments are of less interest. For example, based on the full-ordering of first-stage groups in the end of the previous section it is expected that $\mathcal{G}_k$ should receive proportionally more (or equal) bits than $\mathcal{G}_j$ for $k > j$. Also note, as shown in (6), that within a group G_k the allocations to sub-vectors in the group, i.e. $\{\hat{b}_i : x_i \in G_k\}$, can at times differ by 1 bit. However, as in (5), the internal order within groups is not defined. To handle the differentiations (without loss in generality), in each group we simply arrange sub-vectors in increasing j_k. The internal (possible 1-bit) differentiations can then be assigned at will (in a known pattern) to members $x_i \in G_k$ without a performance loss. In the experiments to follow random patterns are used.

4. MULTI-CLASS EXTENSIONS

It has been shown that a simple dual-option scheme[1] can improve performance. This scheme is a fail-safe variation where an extra 1 bit is used to decide between an assignment that is in the spirit of (1), or one that is based on a partial-order. If the bit selects an assignment as in (1) the remaining $B-1$ bits are spread as equally as possible among the q targets with $\hat{e}_k = \lceil (B-1)/q \rceil$ or $\lfloor (B-1)/q \rfloor$ assigned to $\mathbf{x}_k$ and $\sum_{k=0}^{q-1} \hat{e}_k = B-1$. If the bit selects a non-trivial pattern of assignments based on a partial order, then the order is communicated with b_{over} bits and the $\hat{R} = B - 1 - b_{over}$ bits is divided between the groups as previously described.

We now extend this idea to general multi-class schemes. Here we allow the quantizer to select between L such schemes for each target $\mathbf{x}$, and $b_{ch} = \lceil log_2(L) \rceil$ bits of overhead is used to communicate the selection for each target. Each of the L schemes can differ by s, c, $n_0, \ldots, n_{s-1}$, and b_{over} and/or its pattern of bit-allocation $\{\tilde{B}_k\}$. For each scheme "$\mathcal{S}$" we have the constraint:

$$\sum_{i=0}^{s(\mathcal{S})-1} \tilde{B}_i(\mathcal{S}) = \tilde{R} = B - b_{ch} - b_{over}(\mathcal{S}) \tag{9}$$

Designing such schemes can be complex. For single-option and fail-safe schemes it is often practical to simply perform an exhaustive search of all the possibilities as described in Section 3, i.e. all possibilities of schemes and bit-assignments. However for multi-option schemes, considering all these individual possibilities $\{\tilde{B}_k\}$, s and $n_0, \ldots, n_{s-1}$ with all L multi-class combinations of such schemes, an exhaustive search is not necessarily practical.

In our investigations we do a limited search. We first find the D best single-option pairs of ordering schemes and bit-assignments that are designed with the constraint of (9). Always included is the equal assignment scheme (n_0=q, c=1) with b_{over}=0. For each scheme the MSE obtained for each training vector $\mathbf{x}$ is noted. For each of the $D!/(D-L)!L!$ possible L-class combinations the MSE for each target vector is simply the minimum MSE over the L schemes being considered. The best L-class combination is the one with the minimum average MSE over all vectors $\mathbf{x}$.

5. AN EXAMPLE WITH PERFORMANCE COMPARISONS

We focus on an example with $N=16$ and sub-vectors $\mathbf{x}_k$ with p=4. Other N and p have been tested and will be discussed later and in the presentation. We limit our investigation to the following schemes (options) in Table 1. Here $q=4$ and we consider c=1, 2.

Table 1: Schemes Considered

Label	s	n_0/c	n_1/c	n_2/c	n_3/c	n_4/c
A	2	$u-1$	1			
B	3	1	$u-2$	1		
C	3	2	$u-4$	2		
D	5	1	1	$u-4$	1	1
E	4	2	$u-4$	1	1	
F	4	1	1	$u-4$	2	
G	3	$u-2$	1	1		
H	2	$u-2$	2			
Equal	1	u				

For the experiment p-dimensional fixed-rate VQs are trained with a training set of 80×10^6 zero-mean unit-variance i.i.d. Laplace random scalars. Sub-vectors are non-overlapping p vectors from this sequence. Codebooks trained are of size $0, 1, \ldots, 7$ bits, limiting $0 \leq \hat{b}_i \leq 7$. Another set of 80×10^6 zero-mean unit-variance i.i.d. Laplace random scalars forms the test set. Test vectors $\mathbf{x}$ are non-overlapping 16-dimensional vectors from this set. This set will be used to search

Table 2: Performance of various schemes with $N = 16$, $p = 4$ and $B = 12$.

Design type	**Bitrate** (bits/scalar)	B (bits)	b_{ch} (bits)	**MSE** (dB)	**Scheme** $\mathcal{S}$	s	c	$n_1, \ldots, n_s$	b_{order} (bits)	$\{b_i : \mathbf{x}_i \in \mathcal{G}_0\}, \ldots, \{b_i : \mathbf{x}_i \in \mathcal{G}_{s-1}\}$ (bits)
Benchmark (1)	0.75	12	0	-3.40	Equal	1	1	4	0	$\{3,3,3,3\}$
Single-opt	0.75	12	0	-3.68	H	2	1	2,2	3	$\{0,0\}$, $\{5,4\}$
Fail-Safe	0.75	12	1	-3.88	Equal	1	1	4	0	$\{3,3,3,2\}$
					A	2	1	3,1	2	$\{1,1,0\}$,$\{7\}$
2-option (equal to Fail-Safe)	0.75	12	1	-3.88	Equal	1	1	4	0	$\{3,3,3,2\}$
					A	2	1	3,1	2	$\{1,1,0\}$,$\{7\}$
4-option	0.75	12	2	-4.15	Equal	1	1	4	0	$\{3,3,2,2\}$
					A	2	1	3,1	2	$\{1,0,0\}$,$\{7\}$
					B	3	2	2,0,2	1	$\{0,0\}$,$\{-\}$,$\{5,4\}$
					H	2	1	2,2	3	$\{0,0\}$,$\{4,3\}$
A single-stage 16-dimensional 12-bit VQ has an MSE of -4.3 dB										

and select the schemes+bit-assignments (for various B and b_{ch}) as well to calculate the final MSE performance. Calculating final MSE performance on a second test set did not change the results.

With the codebooks and test-set we designed single-option schemes ($b_{ch} = 0$), Fail-Safe schemes ($b_{ch} = 1$ with one scheme being the benchmark scheme), 2-option schemes ($L = 2$, $b_{ch} = 1$), and 4-option schemes ($L = 4$, $b_{ch} = 2$). For each value $B - b_{ch}$ we test all possible bit-assignments with all schemes and pre-select the $D = 20$ schemes+assignments that give the best MSE performance. We then search all possible L multi-class combinations of these D schemes as described in Section 4.

Table 2 shows the results for the case B=12, i.e. a bitrate of B/N $= 0.75$ bits/scalar. For comparison we also designed a single-stage 16-dimensional 12-bit VQ and obtained an MSE of -4.3 dB. The results clearly show that despite the overhead of the partial ordering schemes, the ordering allows such schemes to make net gains through the bit-assignments. The 4-option scheme has a gain over the benchmark of about 0.75 dB and comes close to the performance of the 16-dimensional VQ. This shows that it is possible to build effective high-dimensional VQs out of lower dimensional VQs (in this case dimension $p = 4$ VQs) using the new approach. Also interesting is that the best 2-option scheme found is in fact the Fail-Safe scheme.

Similar result are seen for other N, e.g. $N = 12$, 24, 32, and other B ranging from bitrates of 0.25 to 1.0 bits/scalar. In these other cases some of the schemes mentioned in Table 1 and not used in Table 2 are selected. A general trend is that the larger the B (or larger N for a given bits/scalar) the more likely a scheme with a larger b_{order} is selected as the one with the best MSE performance. Another trend is that as the VQ and sub-vector dimension p gets smaller the gain in performance (in dB) over the corresponding benchmark of (1) increases.

6. CLOSING REMARKS

The paper presented a description of partial order quantization schemes and described a multi-option extension to these schemes. The schemes are motivated by quantization challenges that arise in a number of transform and transform-predictive coder designs [1] [6] . All schemes work by using the ordering to direct an unequal allocation of bits across sub-vectors. Doing so allows one to improve the average MSE performance on target vectors relative to a benchmark equal bit assignment. The schemes have a useful property of being able to communicate to a decoder an explicit bit-assignment (noise-allocation) across each target. In addition, the schemes are less complex both in terms of search complexity and codebook size when compared to that of the corresponding single-stage full-dimension VQ, e.g. schemes in Table 2 are less complex than a single-stage 12-bit 16-dimension VQ.

An interesting issue to consider is that of variable-length coding. Here the quantization indices are further compressed with variable-length codes. Using this approach one could try to take the benchmark scheme in Table 2, or any of the schemes, and further reduce the average bitrate thus improving the rate-distortion tradeoff. However for the examples in Table 2, and often at the rates B/N we are considering, this is not the case. In particular for the benchmark scheme in Table 2 the calculated entropy of the quantization indices for the 3-bit codebook used is practically equal to 3 bits/sub-vector (0.75 bits/scalar), i.e. all eight codewords are used with equal frequency. Similar results are seen for cases $B/N < 0.75$. The entropy of quantization indices was also investigated for some of the partial-order schemes. Here for multi-option schemes the entropy estimates are first conditioned on the scheme used and then the average entropy (considering the frequency of use of each constituent scheme) is calculated. For the few cases tested, the result is the same as in the benchmark scheme, i.e. the average entropy is practically equal to B/N bits/scalar.

Finally, there are further extensions to these schemes[1]. In one such extension different sub-vector VQ codebooks are used for each scheme in a multi-option scheme. For example, in the Fail-Safe scheme of Table 2 one set of codebooks are trained specifically for the equal bit-allocation and another set for the unequal allocation. Here a multi-option scheme therefore looks like a classified VQ scheme [2]. Training codebooks for this extension is an iterative process of parsing the training data using a given set of schemes each with its own codebook designs. Data that was best coded with a given scheme is then used to re-train the codebooks for that scheme, and the data is then re-parsed. Doing this with the Fail-Safe scheme of Table 2 can reduce the MSE to -3.96 dB. Future work will further explore this classified approach as well as variable-length vector coding of quantization indices.

REFERENCES

[1] S. A. Ramprashad, "Understanding the quality losses of embedded speech and audio coders," in *IEEE Workshop on Speech Coding*, Ibaraki, Japan, October 2002, pp. 11–13.

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

[3] F. Norden and P. Hedelin, "Companded quantization of speech MDCT coefficients," *IEEE Trans. on Speech and Audio Proc*, vol. 13, no. 2, pp. 163–173, March 2005.

[4] R. Vafin and W. B. Kleijn, "Entropy-constrained polar quantization and its application to audio coding," *IEEE Trans on Speech and Audio Proc.*, vol. 13, no. 2, March 2005.

[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley and Sons, New York, 1991.

[6] S. A. Ramprashad, "The multimode transform predictive coding paradigm," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 2, pp. 117–129, March 2003.