Sample-Adaptive Product Quantization: Asymptotic Analysis and Examples

Dong Sik Kim, *Member, IEEE,* and Ness B. Shroff, *Member, IEEE*

***Abstract*—Vector quantization (VQ) is an efficient data compression technique for low bit rate applications. However, the major disadvantage of VQ is that its encoding complexity increases dramatically with bit rate and vector dimension. Even though one can use a modified VQ, such as the tree-structured VQ, to reduce the encoding complexity, it is practically infeasible to implement such a VQ at a high bit rate or for large vector dimensions because of the huge memory requirement for its codebook and for the very large training sequence requirement. To overcome this difficulty, a structurally constrained VQ called the *sample-adaptive product quantizer* (SAPQ) has recently been proposed. In this paper, we extensively study the SAPQ that is based on scalar quantizers in order to exploit the simplicity of scalar quantization. Through an asymptotic distortion result, we discuss the achievable performance and the relationship between distortion and encoding complexity. We illustrate that even when SAPQ is based on scalar quantizers, it can provide VQ-level performance. We also provide numerical results that show a 2–3 dB improvement over the Lloyd–Max quantizers for data rates above 4 b/point.**

***Index Terms*—Lattice vector quantizer, product quantizer, sample-adaptive product quantizer (SAPQ), vector quantizer.**

I. Introduction

VECTOR quantization (VQ) is an efficient technique for data compression systems with low bit rate applications and for speech recognition/speaker identification systems [17], [42]. By employing VQ, we can achieve high compression gains, especially for image and speech/audio data. Image and speech/audio data are highly correlated and cannot be decorrelated using conventional linear transforms, such as the discrete cosine transform. Depending on the input sources, using a combination of a scalar quantizer and an entropy coder, it is possible to obtain performance up to 1.53 dB worse than the theoretical bound in an asymptotic sense for large codebooks. However, using VQ, one can further improve this performance and come closer to achieving the theoretical lower bound [49].

It is VQ's ability to improve on scalar quantization that has lead to the development of several data coders, especially in the speech/audio coding areas. The prevailing coding algorithms are based on *vector excitation coding*, where an adaptive VQ is employed [18], [45]. From the filterbank analysis or the linear predictive coding analysis, the characteristics of speech signals are represented by a series of spectral vectors. Hence, VQ is very efficient in reducing or representing the spectral data. Because of this efficiency, in addition to VQ's applicability in speech data compression, most speech recognition/speaker identification systems also employ VQ to compare spectral similarity between a pair of vectors [14], [20], [48], [52]. Different clustering techniques are also employed in learning the speech recognition/speaker identification systems [24], [44].

The major problem with VQ is its encoding complexity and storage required for the codebook, which increase dramatically with the vector dimension and bit rate. This is especially problematic in quantizing given spectral vectors to yield a low quantization error. Hence, applying VQ to this low-quantization-error application has the requirement of large storage and high encoding complexity. In order to circumvent this problem, various modified VQ techniques have been proposed [17], e.g., *tree-structured VQ* (TSVQ) [33], *classified VQ*, and *lattice VQ* [8]. However, since such schemes are still based on a VQ structure, and hence, the application areas of these schemes are relatively limited. More recently, the *trellis-coded quantization* (TCQ) schemes [38], [49] have gained popularity for their ability to provide high performance for lower complexity (than traditional VQ schemes). Unfortunately, since TCQ requires special techniques such as the trellis encoder and the Viterbi decoder, implementing the TCQ-based coding scheme is still quite complex.

Recently, we have proposed a feedforward adaptive quantizer called the *sample-adaptive product quantizer* (SAPQ) in order to reduce both the encoding complexity and memory requirements [29]. The SAPQ in [29] is based on k-dimensional VQs for m random vectors, where m and k are integers. This SAPQ is a structurally constrained, km-dimensional VQ. In SAPQ, a *product quantizer* (PQ) [17, p. 430] is selected from a set of candidate PQs. (A block diagram of SAPQ is shown in Fig. 1, where the best quantizer is selected for an input that has a block length of m.) In [29], we also suggested several performance bounds and provided extensive comparisons.

In this paper, however, we will study a special type of SAPQ that accepts m random variables as inputs based on *scalar quantizers* (SQs). Through an asymptotic analytical result on the SQ-based SAPQ, we will describe the achievable gain of SAPQ and discuss the relationship between the distortion and encoding complexity in an asymptotic sense. By introducing several examples, we will show different SAPQ design methods for lattice

Manuscript received January 27, 1999; revised April 19, 2000. The associate editor coordinating the review of this paper and approving it for publication was Dr. Alle-Jan van der Veen.

D. S. Kim was with the Division of Information and Communication Engineering, Hallym University, Chunchon, Kangwon-Do, Korea. He is now with the Department of Electronics Engineering, Hankuk University of Foreign Studies, Yongin, Korea (e-mail: dskim@san.hufs.ac.kr).

N. B. Shroff is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-1285 USA (e-mail: shroff@ecn.purdue.edu).

Publisher Item Identifier S 1053-587X(00)07641-8.

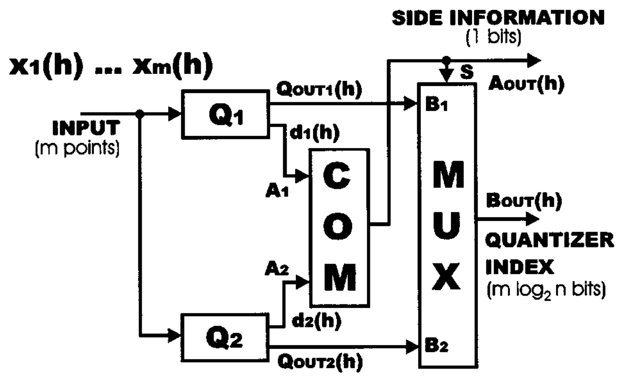


Fig. 1. Block diagram of SAPQ in Example 1 (Lattice $D_m^{\perp}$).

VQs and nonuniform SAPQs. We will illustrate that even when SAPQ is based on SQs, we can obtain VQ-level performance. Further, for high bit rates (greater than 1 b/point), we will obtain high gains while maintaining the low encoding complexity of SQ. This is an important achievement since, as mentioned before, obtaining VQ-level performance for high bit rates is usually quite difficult due to the high encoding complexity and huge codebook requirements. (Note that SAPQ can be extended to the low bit rate cases as well by employing k-dimensional VQs instead of SQs [29].) Further, we will show that SAPQ needs a relatively small *training sequence* (TS) compared with traditional VQs.

The quantization procedure of SAPQ seems similar to that of the *adaptive coding scheme* in [19, p. 371] and *universal source coding scheme* in [11], [12], [13] in some ways. However, it is important to note that SAPQ is very different from the traditional adaptive coding and universal source coding schemes. The adaptive coding scheme produces increased gains by replacing the quantizer, depending on the varying statistical characteristics of a nonstationary source [40], [43]. The universal source coding scheme treats the problems of source coder design for applications where the source statistics are unknown *a priori* [7], [51], [53]. Since the main purpose of the proposed SAPQ is reducing the encoding complexity of VQ, we can in fact apply SAPQ to problems in the adaptive coding and universal source coding schemes by simply substituting their quantizer portions with SAPQ. Note that the problem statement in this paper is quite similar to that of the *scalar-vector quantizer* (SVQ), which has been proposed by Laroia and Farvardin [31], [32]. However, SAPQ can also implement large codebooks (or high bit rates) and achieve even better results than SVQ [15]. Further, SAPQ has a better channel noise characteristics than the SVQ case [30]. Another appealing quality of SAPQ is that the main idea is quite intuitive and simple, compared with TCQ or SVQ. In a coding scheme that does not employ the entropy coder for the quantizer output, the SAPQ quantizer can provide a 2–3 dB improvement over the Lloyd–Max quantizers [35], [39].

This paper is organized as follows. In Section II, we mathematically define the SQ-based SAPQ and analyze the asymptotic performance of the SAPQ in Section III. In Section IV, we introduce several design examples and simulation results with discussions. We then conclude the paper in the last section.

II. Sample-Adaptive Product Quantizer

In this section, we briefly review SAPQ, focus on the case when it is described in terms of scalar quantizers, and lay the foundation for the asymptotic analysis conducted in Section III.

We consider a sequence of random variables $X_1, \cdots, X_m$ taking values in $\mathbb{R}$ as the discrete-time source to be quantized. Here, m is the *block length*. Suppose that $E\{X_i^2\} < \infty$ for $i = 1, \cdots, m$. Let $\mathcal{C}_n$ denote the class of sets that take n points from $\mathbb{R}$, and let the sets in $\mathcal{C}_n$ be called "n-level codebooks," where each such codebook has n codewords. The quantization of X_i is the mapping of a sequence of observations of X_i to a sequence of points of $C(\in \mathcal{C}_n)$ according to a mapping called the quantizer. The average distortion achieved when a random variable X_i is quantized by a codebook $C(\in \mathcal{C}_n)$ is given by $E\{\min_{y\in C}(X_i - y)^2\}$. In this quantization scheme, if fixed length binary codes are used to represent the quantizer outputs, the bit rate (defined as bits per source point in $\mathbb{R}$) required is $\log_2 n$. Note that since $X_i \in \mathbb{R}$ and $C \subset \mathbb{R}$, the quantizer is a *scalar quantizer*. Let an observation of $X_1, \cdots, X_m$ be denoted by $x_1, \cdots, x_m$. Suppose that the codebooks C_i are $C_i \in \mathcal{C}_{n_i}$ for $i = 1, \cdots, m$, where n_i are positive integers. If we quantize this observation by applying scalar quantizers using codebooks C_i to each x_i independently, the overall average distortion D_{PQ} is given by

$$D_{PQ} := E\left\{\frac{1}{m}\sum_{i=1}^{m}\min_{y\in C_i}(X_i - y)^2\right\}. \tag{1}$$

We call this quantization scheme the *product VQ*, or PQ, since the quantizer is a mapping from $\mathbb{R}^m$ to the Cartesian product set $C_1 \times \cdots \times C_m$. The bit rate of the PQ is $(1/m)\sum_{i=1}^{m}\log_2 n_i$. If the random variables are independent (or uncorrelated), then this independence appears to motivate quantizing each random variable independently, as shown in (1). However, even if the input is independent, independently quantizing each of the random variables of $X_1, \cdots, X_m$ is just one of the many possible coding schemes and could be improved upon by appealing to the *block source coding theorem* [49]. If $\boldsymbol{C}$ is a subset of $\mathbb{R}^m$ with $|\boldsymbol{C}| = \nu$, where ν is a positive integer, then the average distortion yielded by using a vector quantizer for $X_1, \cdots, X_m$ is

$$D_{VQ} = E\left\{\min_{\boldsymbol{y}\in\boldsymbol{C}}\frac{1}{m}\sum_{i=1}^{m}(X_i - y_i)^2\right\} \tag{2}$$

where $\boldsymbol{y} = (y_1, \cdots, y_m)(\in \mathbb{R}^m)$, and the bit rate for SAPQ is $(1/m)\log_2 \nu$.

Now, we introduce a feedforward adaptive quantization scheme, which is based on a new concept of adaptation to each observation of $X_1, \cdots, X_m$. Let $C_{i,j}(\subset \mathbb{R})$ denote the ith codebook for each X_i, where $j \in \{1, 2, \cdots, 2^\eta\}$, and η is a non-negative integer. The sample adaptive scheme quantizes each observation $x_1, \cdots, x_m$ using the codebooks $C_{1,j}, \cdots, C_{m,j}$ to form the 2^η candidates of distances

$$\frac{1}{m}\sum_{i=1}^{m}\min_{y\in C_{i,j}}(x_i - y)^2, \quad \text{for } j = 1, 2, \cdots, 2^\eta \tag{3}$$

and choose the smallest distance. Hence, the average distortion of SAPQ is given by

$$D_{\text{SAPQ}} := E\left\{\min_j \frac{1}{m}\sum_{i=1}^{m}\min_{y\in C_{i,j}}(X_i - y)^2\right\}. \tag{4}$$

Here, we suppose that $C_{i,j} \in \mathcal{C}_{n'_i}$, for $j = 1, 2, \cdots, 2^\eta$, where $n'_i \in \mathbb{N}$. We call this quantization scheme, the *sample-adaptive product quantizer*, since the quantization scheme is selecting a PQ from a set of 2^η candidate PQs. For each sample, SAPQ produces the bit streams for a codebook index and the m quantized element indices, in the form of a feed-forward adaptive coding scheme [17, p. 602]. This makes it possible to replace different codebooks for each sample of $X_1, \cdots, X_m$. Therefore, the total bit rate is given by $(1/m)\sum_{i=1}^m \log_2 n'_i + \eta/m$, where η are the additional bits (side information) required in our scheme to indicate which codebook is employed. Note that throughout this paper, we will suppose that $\log_2 n_i$ and $\log_2 n'_i$ can be nonintegers for the quantizer performance comparisons. Although our discussion in this paper will focus on m random variables, we can also consider m k-dimensional random vectors as the input to the SAPQ. This generalization of SAPQ is described in [29]. As discussed in [29], the SAPQ in (4) is a *structurally constrained VQ* in m dimensions. Hence, the average distortion of SAPQ is between those of the scalar quantizer (or PQ) and full search VQ. However, SAPQ can asymptotically achieve the full search VQ distortion.

III. Performance of SAPQ

In this section, through asymptotic analysis, we will formally study the performance of SAPQ.

A. Asymptotic Analysis

To simplify the notation, let $\boldsymbol{X} := (X_1, \cdots, X_m)$ denote an m-dimensional random vector. We assume an absolutely continuous distribution function for $\boldsymbol{X}$. Now, consider root lattices [21]. Let the points of an m-dimensional lattice $\mathcal{L}_m (\subset \mathbb{R}^m)$ be denoted by $\boldsymbol{y}_\upsilon$, where $\upsilon \in \mathbb{Z}$. The closure of the υth Voronoi region of the lattice $\mathcal{L}_m$ is the convex polytope H_υ, which is defined as

$$H_\upsilon := \{\boldsymbol{x} \in \mathbb{R}^m : \|\boldsymbol{x} - \boldsymbol{y}_\upsilon\|^2 \le \|\boldsymbol{x} - \boldsymbol{y}_{\upsilon'}\|^2, \text{ for all } \upsilon' \in \mathbb{Z}\} \quad \text{for } \upsilon \in \mathbb{Z} \tag{5}$$

where $\|\boldsymbol{x}\| = \sqrt{x_1^2 + \cdots + x_m^2}$, and $\boldsymbol{x} = (x_1, \cdots, x_m) \in \mathbb{R}^m$. In (5), we let $\boldsymbol{y}_1 = (0, \cdots, 0)$; thus, H_1 includes the origin $\boldsymbol{y}_1$. Now, $G(\mathcal{L}_m)$, which is the *normalized second moment of* H_υ, is defined as

$$G(\mathcal{L}_m) := \frac{1}{m} \frac{\int_{H_\upsilon} \|\boldsymbol{x} - \boldsymbol{y}_\upsilon\|^2 \, d\boldsymbol{x}}{V(H_\upsilon)^{1/\rho}} \tag{6}$$

where $\rho := m/(m+2)$, and $V(H_\upsilon) := \int_{H_\upsilon} d\boldsymbol{x}$ is the volume of H_υ [16]. Note that all H_υ, where $\upsilon \in \mathbb{Z}$, have the same shape. Thus, the normalized second moments and the volumes of H_υ are all the same. The quantity $G(\mathcal{L}_m)$ is the normalized quantizer distortion per codeword having the Voronoi region for uniformly distributed data [9]. Conway and Sloane have conducted extensive research on various lattices [8]–[10]. For the definitions of the lattices and discussions, see [8] and [29]. In order to explain the achievable performance from SAPQ in an asymptotic aspect, we now use a variation of a result derived from the asymptotic result in [29, Th. 1]. In this variation, which is summarized in the following theorem, the SAPQ is based on SQs, and the asymptotic result is obtained for an arbitrary fixed codebook of size n'. In the theorem, we will assume that $C_{i,j} \in \mathcal{C}_{n'}$ and $\mathcal{C}_{\text{SAPQ}} = \bigcup_{j=1}^{2^\eta} (C_{1,j} \times \cdots \times C_{m,j})$ for a fixed codebook size n'.

Theorem 1: Suppose that $\boldsymbol{X}$ has a joint density function f with $E\{\|\boldsymbol{X}\|^{2+\epsilon}\} < \infty$ for some $\epsilon > 0$, and f is bounded on $\mathbb{R}^m$. Then

$$\limsup_{\eta \to \infty} [(n')^m 2^\eta]^{2/m} \inf_{\mathcal{C}_{\text{SAPQ}}} D_{\text{SAPQ}} \le G(\mathcal{L}_m) \|f\|_\rho \tag{7}$$

where the functional $\|\cdot\|_\rho$ is given by

$$\|f\|_\rho := \left[\int f^\rho(\boldsymbol{x}) \, d\boldsymbol{x}\right]^{1/\rho}. \tag{8}$$

Proof of Theorem 1: In [29, App. C], the relationships (C1) and (C2) are also satisfied for an arbitrary fixed codebook size $n_\eta = n' (\in \mathbb{N})$. Hence, for the SQ case, i.e., $k = 1$ in [29, th. 1], we obtain (7). □

From Theorem 1, we can obtain the asymptotic result $\limsup_{\eta \to \infty} [(n')^m 2^\eta]^{2/m} \inf_{\mathcal{C}_{\text{SAPQ}}} D_{\text{SAPQ}} \le J_m \|f\|_\rho$, where $J_m := \inf_{\mathcal{L}_m} G(\mathcal{L}_m)$. It is clear from [26] that the optimal m-dimensional VQ is such that $\limsup_{\nu \to \infty} \nu^{2/m} \inf_{\boldsymbol{C}} D_{VQ} \le J_m \|f\|_\rho$, where $|\boldsymbol{C}| = \nu$. From [6] and [50], we know that the sequence on the left-hand side converges. Further, Gersho's conjecture tells us that the asymptotically optimal quantizer is a function of J_m [16]. In other words

$$\lim_{\nu \to \infty} \nu^{2/m} \inf_{\boldsymbol{C}} D_{VQ} = J_m \|f\|_\rho. \tag{9}$$

Therefore, if this conjecture were true (as is typically assumed), then from (7) and (9), SAPQ would achieve the asymptotically optimal m-dimensional VQ performance $J_m \|f\|_\rho$ [36]. Hence, the advantages of SAPQ over PQ are the same as the VQ case over the scalar quantizer [36], [37]. We will now try to obtain more insight by studying the performance of SAPQ based on the two factors $G(\mathcal{L}_m)$ (or J_m) and $\|f\|_\rho$.

B. Voronoi Region Shape: $G(\mathcal{L}_m)$ (or J_m)

From (6) and Theorem 1, we can conclude that the factor J_m is concerned with the shape of the Voronoi region of a quantizer. The gain achieved by this factor is called the *space-filling advantage* [36]. Since $J_1 = 1/12$ and $\inf J_m = 1/2\pi e$, the achievable maximal gain through the shape of the Voronoi region is less than or equal to $10 \log(J_1 / \inf J_m) \cong 1.53$ dB.

In fact, in the literature, lattice VQs have been used to exploit this space-filling advantage. Several important lattices can be described as the union of the cosets of a set [8]. Based on this fact, various encoding/decoding algorithms for lattice VQs have been proposed [9, eq. (8)]. Such lattice VQs can be described by SAPQ. For example, the hexagonal lattice A_2, which yields the minimum $J_2 = G(A_2) \cong 0.0802$ in 2-D, can be defined as

$$A_2 := \bigcup_{j=1}^{2} \left(\boldsymbol{r}_j + \mathbb{Z} \times \left\{\cdots, -\sqrt{3}, -\frac{\sqrt{3}}{2}, 0, \frac{\sqrt{3}}{2}, \sqrt{3}, \cdots\right\}\right). \tag{10}$$

Here, the coset representatives $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are $\boldsymbol{r}_1 = (0, 0)$ and $\boldsymbol{r}_2 = (-1/2, \sqrt{3}/2)$. Since a coset of a product set is also a product set, A_2 is a union of two product sets. Hence, a truncated lattice of A_2 has the same structure as the SAPQ codebook in m-dimensions ($m = 2$). Therefore, a truncated lattice of A_2 can be implemented by SAPQ with $\eta = 1$ since we have two representatives.

An important lattice listed in [47] is the $D_m^{\perp}$ lattice. For $m \geq 2$, $D_m^{\perp}$ is the dual of the lattice D_m, which is defined as

$$D_m^{\perp} := \mathbb{Z}^m \bigcup ((\tfrac{1}{2}, \tfrac{1}{2}, \cdots, \tfrac{1}{2}) + \mathbb{Z}^m). \tag{11}$$

In a similar manner as in (10), it is clear that SAPQ can construct a truncated lattice of $D_m^{\perp}$, for $m = 1, 2, \cdots$, with only $\eta = 1$. Hence, SAPQ with $\eta = 1$ can construct the optimal lattice in 3-D since the $D_3^{\perp}$ lattice (or equivalently the lattice $A_3^{\perp}$) is a body-centered cubic lattice and optimal in 3-D [3]. The minimum value of $G(D_m^{\perp})$ is about 0.0747 at $m = 9$ [8]. Hence, the maximum gain is asymptotically $10 \log(J_1/G(D_9^{\perp})) \cong 0.475$ dB if we use lattice $D_m^{\perp}$.

Now, we consider the lattice E_8. The lattice E_8 can be rewritten as $E_8 = \{\boldsymbol{x} | \boldsymbol{x} = U_{E_8}\boldsymbol{p}, \boldsymbol{p} \in \mathbb{Z}^8\}$, where $\boldsymbol{p}$ is written as a column vector, and U_{E_8} is the generator matrix of E_8 given by

$$U_{E_8} = \frac{1}{2}\begin{pmatrix} 2 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 2 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{12}$$

Define $E_{8,j}$ as the set

$$\begin{aligned} E_{8,j} := \{\boldsymbol{x} | \boldsymbol{x} \in U_{E_8}\boldsymbol{p}' \\ \boldsymbol{p}' := (p_0, p_1, p_2, p_3, 2p_4 + j_0, 2p_5 \\ + j_1, 2p_6 + j_2, 2p_7 + j_3) \\ p_0, \cdots, p_7 \in \mathbb{Z}\}. \end{aligned} \tag{13}$$

Here, $j_0, \cdots, j_3 \in \{0, 1\}$ are given by $j = 1 + j_0 2^0 + j_1 2^1 + j_2 2^2 + j_3 2^3$, for $j = 1, 2, \cdots, 2^4$. Then

$$\begin{aligned} E_8 &= \bigcup_{j=1}^{16} E_{8,j} \\ &= \bigcup_{j=1}^{16} [U_{E_8}(0, 0, 0, 0, j_0, j_1, j_2, j_3) \\ &\quad + \{\boldsymbol{x} | \boldsymbol{x} = U_{E_8}(p_0, p_1, p_2, p_3, 2p_4, 2p_5, 2p_6, 2p_7) \\ &\quad p_0, \cdots, p_7 \in \mathbb{Z}\}] \\ &= \bigcup_{j=1}^{16} (\boldsymbol{r}_j + \mathbb{Z}^8) \end{aligned} \tag{14}$$

where $\boldsymbol{r}_j := U_{E_8}(0, 0, 0, 0, j_0, j_1, j_2, j_3)$ for $j = 1, \cdots, 16$. Hence, a truncated lattice of E_8 can be implemented by the SAPQ with $\eta = 4$. Note that in this case, the gain will be $10 \log(J_1/G(E_8)) \cong 0.654$ dB. We can construct examples of SAPQ for several other types of lattices, such as A_3, D_m, and E_7 in a similar manner.

C. Distribution Shape: $\|f\|_\rho$

The factor $\|f\|_\rho$, which is defined in (8), is concerned with the joint density function f. From this factor, we could potentially obtain a large gain in SAPQ based on the *constrained-distortion quantizer* [25]. From Hölder's inequality [4], $(\|f\|_\rho)_m$ is a nonincreasing sequence. Hence, depending on f, we can expect some gain by increasing m. Suppose that an i.i.d. input has a uniform density function f; then $\|f\|_\rho = 12\sigma^2$, where σ^2 is the variance of the input. Hence, for the uniformly distributed input case, we cannot expect any gain through this factor. However, suppose that f is a joint Gaussian density function given by

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{m/2}(\det \boldsymbol{S})^{1/2}} \exp\left(\frac{-\boldsymbol{x}^t \boldsymbol{S}^{-1}\boldsymbol{x}}{2}\right) \tag{15}$$

where $\boldsymbol{S}$ is the auto-covariance matrix of $\boldsymbol{X}$. Then

$$\|f\|_\rho = 2\pi\rho^{-(m+2)/2}(\det \boldsymbol{S})^{1/m}. \tag{16}$$

It is well known that if there is correlation between $X_1, \cdots, X_m$, then we can reduce the distortion through the factor $(\det \boldsymbol{S})^{1/m}$ since $(\det \boldsymbol{S})^{1/m} \leq (\text{tr}\boldsymbol{S})/m$ [5]. The gain from this factor is known as the *memory advantage* [36]. The well-known example that exploits this advantage is the Karhunen–Loéve transform and the discrete cosine transform [19]. (Note that the term $(\det \boldsymbol{S})^{1/m}$ can be derived for other type of density function if we employ $\|\cdot\|^2$ as a distortion measure [28].)

Now, focus on the factor $\rho^{-(m+2)/2}$ in (16), which is dependent on the shape of f. This factor $\rho^{-(m+2)/2}$ is equal to $3^{3/2} \cong 5.20$ for $m = 1$ and monotonically decreases to $e \cong 2.72$ as $m \to \infty$. Hence, the achievable gain through the factor $\rho^{-(m+2)/2}$ is $10 \log(3^{3/2}/e) \cong 2.81$ dB. Furthermore, for the Laplacian density case, since

$$\|f\|_\rho = 2\rho^{-(m+2)}(\det \boldsymbol{S})^{1/m} \tag{17}$$

the potential improvement is about 5.63 dB. The gain from the shape of the density function is called the *shape advantage* [36].

From the space-filling advantage and the shape advantage, even when the input $X_1, \cdots, X_m$ is i.i.d. (or uncorrelated), we have the potential for an improvement of up to about 4.35 dB over PQ and up to about 7.16 dB over PQ for the Gaussian and the Laplacian density cases, respectively. *Note that these maximum gains are the same as those obtainable from the corresponding theoretical bounds in an asymptotic sense for large codebooks* [46].

IV. Examples in Designing SAPQ

From the asymptotic analysis in Section III, we can estimate the achievable performance of SAPQ when η is large. However, the asymptotic analysis does not provide any explicit value of η that yields good performance. This appropriate value of η is quite dependent on the SAPQ codebook design method. In order to provide an appropriate range of η for SAPQs, in this section, we provide several different SAPQ design examples based on the theoretical observations made in Section III. Through several examples, we explicitly demonstrate the performance of SAPQ and its encoding complexity in conjunction with codebook sizes and η.

A. Uniform SAPQ Based on Lattices

In order to demonstrate the space-filling advantage gained from SAPQ, we now consider a uniformly distributed input as follows. Suppose that $X_1, \cdots, X_m$ are i.i.d. and X_i has a uniform distribution. In other words, we suppose that $\boldsymbol{X} = (X_1, \cdots, X_m)$ has a uniform density function f given by

$$f(\boldsymbol{x}) = \begin{cases} 1/a^m, & \boldsymbol{x} \in ([-a/2, a/2])^m \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where a is a positive constant. As shown in Theorem 1, for this uniform density function, the gain comes only from the space-filling advantage. Hence, using the uniform density function, we can numerically observe the achievable gain from SAPQ by changing the shape of the Voronoi region.

If we use the same m Lloyd–Max scalar quantizers for the quantization of $\boldsymbol{X}$ based on the PQ in (1), then the Lloyd–Max quantizer is the uniform quantizer given by output points

$$-\tfrac{1}{2}(a+b) + b \cdot \ell, \quad \text{for } \ell = 1, \cdots, n \tag{19}$$

where $b = a/n$ is the step size. Hence, the average distortion in (1) can be rewritten as

$$D_{PQ} = \sigma^2 2^{-2R} \tag{20}$$

where $\sigma^2 = a^2/12$ is the variance of X_i with bit rate $R = \log_2 n$. Note that Shannon's lower bound (SLB) for the uniform density input is given by

$$D_{\text{SLB}} = \frac{6}{\pi e} \sigma^2 2^{-2R} \tag{21}$$

which is less than D_{PQ} of (20) by 10 $\log(\pi e/6) \cong 1.53$ dB. Hence, there is a potential of about 1.53 dB improvement, which is the same as the gain 10 $\log(J_1/\inf J_m)$ from the space-filling advantage. For a finite dimension m, VQ can achieve a fraction of this potential improvement (i.e., better than D_{PQ}). It will be shown that SAPQ can also obtain this gain (without resorting to the complexities involved in the other schemes).

Now, for the uniform density function in (18), we provide several SAPQ design examples and numerical results based on different lattices in Section III as follows. Note that η is dependent on the lattice type.

Example 1 (Lattice $D_m^\perp$): Let $C_j = \{y_{1,j}, \cdots, y_{n',j}\} (\in \mathcal{C}_{n'})$ for $j = 1$ and 2, denote the codebooks of an SAPQ, where n' is a constant such that $\log_2 n' \in \mathbb{N}$. Let the codewords based on the lattice $D_m^\perp$ be defined as

$$y_{\ell,j} = -\frac{1}{2}\left(a + \frac{3b}{2}\right) + b \cdot \ell + b_j, \quad \text{for all } i, j, \text{ and } \ell \tag{22}$$

where $b = a/n'$, $b_1 = 0$, and $b_2 = b/2$. Note that *this quantizer is the SAPQ with* $\eta = 1$. Therefore, the average distortion in (4) can be rewritten as

$$D_{\text{SAPQ}} = E\left\{ \min_{j \in \{1,2\}} \frac{1}{m} \sum_{i=1}^{m} \min_{\ell \in \{1, \cdots, n'\}} (X_i - y_{\ell,j})^2 \right\}. \tag{23}$$

Here, the bit rate is $R = \log_2 n' + 1/m$. In Fig. 1, a block diagram of the SAPQ of this example is depicted. We have two different quantizers, (Q_1 and Q_2) for $j = 1$ and 2, respectively. Each quantizer has two outputs; in Fig. 1, $Q_{\text{out}j}(h)$ implies m quantizer outputs, which is represented by a fixed length $\log_2 n$ bits in input h, and $d_j(h) = (1/m) \sum_{i=1}^{m} \min_{\ell \in \{1, \cdots, n\}} (x_i(h) - y_{\ell,j})^2$. The comparator (**COM**) then compares the two values $d_1(h)$ and $d_2(h)$ and selects the index that has the lowest value. The output $A_{\text{out}}(h)$ has 1 bit to indicate Q_1 or Q_2 for the multiplexer **MUX**.

TABLE I
SAPQ BASED ON LATTICE $D_m^\perp$ IN (22) (EXAMPLE 1). DISTORTIONS (IN DECIBELS) FOR THE UNIFORM I.I.D. (18), $m = 8$, AND $\eta = 1$

Bit Rates: R	PQ (SQ)	SAPQ
1.125	-6.77	-6.07
2.125	-12.79	-12.69
3.125	-18.81	-19.00
4.125	-24.84	-25.18
5.125	-30.86	-31.26
6.125	-36.88	-37.33
7.125	-42.90	-43.37
8.125	-48.92	-49.38

In Table I, we compare the nonadaptive product quantizer with the adaptive quantizer of (22) for a uniform input with the variance $\sigma^2 = 1$. In this table, D_{PQ} is calculated from the nonadaptive quantizer in (20). As shown in Table I, at low bit rates, D_{PQ} is lower than D_{SAPQ} due to the codewords of the sides of the uniform pdf. However, as the bit rate increases, D_{SAPQ} is lower than D_{PQ} [as was discussed in Section III below the $D_m^\perp$ lattice in (11)]. Note that we should expect a gain (from our asymptotic analysis) to be 10 $\log(J_1/G(D_8^\perp)) \cong 0.47$ dB, which appears to be consistent with the results in Table I.

Example 2 (Lattice E_8): Let $C_{i,j} = \{y_{1,i,j}, \cdots, y_{n',i,j}\}$ $(\in \mathcal{C}_{n'})$, for $i = 1, \cdots, 8$ and $j = 1, \cdots, 16$ denote the codebooks of an SAPQ, where n' is a constant such that $\log_2 n' \in \mathbb{N}$. Let the codewords based on the lattice E_8 be defined as

$$y_{\ell,i,j} = -\frac{1}{2}\left(a + \frac{3b}{2}\right) + b \cdot \ell + b_{i,j}, \quad \text{for all } i, j, \text{ and } \ell \tag{24}$$

where $b = a/n'$, $b_{i,j} := br'_{i,j}$, and $r'_{i,j}$ are given by the coset representatives $\boldsymbol{r}_j (\in \mathbb{R}^8)$ as follows. Let $r_{i,j}$ be defined as

$$\begin{aligned} \boldsymbol{r}_j &= U_{E_8}(0, 0, 0, 0, j_0, j_1, j_2, j_3) \\ &:= (r_{1,j}, \cdots, r_{8,j}) \end{aligned} \tag{25}$$

where U_{E_8} is given in (12), $j_0, \cdots, j_3 \in \{0, 1\}$, and $j := 1 + j_0 2^0 + j_1 2^1 + j_2 2^2 + j_3 2^3$ for $j = 1, \cdots, 16$ from (14). Then $\boldsymbol{r}'_j := (r'_{1,j}, \cdots, r'_{8,j})$ is defined as

$$r'_{i,j} := r_{i,j} - \lfloor r_{i,j} \rfloor \tag{26}$$

where $\lfloor c \rfloor$, $c \in \mathbb{R}$ is the largest integer less than or equal to c. Hence, the $\boldsymbol{r}'_j$ are given as follows.

$$\begin{aligned} \boldsymbol{r}'_1 &= (0, 0, 0, 0, 0, 0, 0, 0) \\ \boldsymbol{r}'_2 &= 1/2(1, 1, 1, 0, 1, 0, 0, 0) \\ \boldsymbol{r}'_3 &= 1/2(0, 1, 1, 1, 0, 1, 0, 0) \\ \boldsymbol{r}'_4 &= 1/2(1, 0, 0, 1, 1, 1, 0, 0) \\ \boldsymbol{r}'_5 &= 1/2(0, 0, 1, 1, 1, 0, 1, 0) \\ \boldsymbol{r}'_6 &= 1/2(1, 1, 0, 1, 0, 0, 1, 0) \\ \boldsymbol{r}'_7 &= 1/2(0, 1, 0, 0, 1, 1, 1, 0) \\ \boldsymbol{r}'_8 &= 1/2(1, 0, 1, 0, 0, 1, 1, 0) \end{aligned}$$

TABLE II
SAPQ BASED ON LATTICE E_8 IN (24) (EXAMPLE 2). DISTORTIONS (IN DECIBELS) FOR THE UNIFORM I.I.D. (18), $m = 8$, AND $\eta = 4$

Bit Rates: R	PQ (SQ)	SAPQ
1.5	-9.03	-9.05
2.5	-15.05	-15.42
3.5	-21.07	-21.58
4.5	-27.09	-27.67
5.5	-33.11	-33.73
6.5	-39.13	-39.77
7.5	-45.15	-45.78
8.5	-51.18	-51.83

$$\begin{aligned}\boldsymbol{r}'_9 &= 1/2(1, 1, 1, 1, 1, 1, 1, 1)\\ \boldsymbol{r}'_{10} &= 1/2(0, 0, 0, 1, 0, 1, 1, 1)\\ \boldsymbol{r}'_{11} &= 1/2(1, 0, 0, 0, 1, 0, 1, 1)\\ \boldsymbol{r}'_{12} &= 1/2(0, 1, 1, 0, 0, 0, 1, 1)\\ \boldsymbol{r}'_{13} &= 1/2(1, 1, 0, 0, 0, 1, 0, 1)\\ \boldsymbol{r}'_{14} &= 1/2(0, 0, 1, 0, 1, 1, 0, 1)\\ \boldsymbol{r}'_{15} &= 1/2(1, 0, 1, 1, 0, 0, 0, 1)\\ \boldsymbol{r}'_{16} &= 1/2(0, 1, 0, 1, 1, 0, 0, 1).\end{aligned} \tag{27}$$

Therefore, *this quantizer is an SAPQ with* $\eta = 4$, and the bit rate is $\log n' + 1/2$. (Note that the quantizer in Example 1 has only $\boldsymbol{r}'_1$ and $\boldsymbol{r}'_9$.) The SAPQ distortion of this example is given by

$$D_{\text{SAPQ}} = E\left\{\min_{j\in\{1,\cdots,16\}} \frac{1}{8}\sum_{i=1}^{8} \min_{\ell\in\{1,\cdots,n'\}} (X_i - y_{\ell,i,j})^2\right\}. \tag{28}$$

The results are summarized in Table II. Note that from our asymptotic analysis, we would expect a gain of approximately 0.65 dB on lattice E_8, which is consistent with our results in Table II. In a similar manner, we can design an SAPQ based on the lattice E_7 with $\eta = 3$. Furthermore, some experimental results for the cases of $\eta = 2$ and 3, by choosing several $\boldsymbol{r}'_j$ from (27), are compared in [30].

B. Uniform SAPQ and Output Entropy

If we do not use the entropy coder for the quantized output, then we can obtain a large gain using SAPQ, as shown in Example 5. However, if we employ the entropy coder, then we still achieve a nontrivial gain from SAPQ, albeit one that is not as large. The next example shows an entropy-constrained SAPQ. Consider a midtread uniform quantizer in $\mathbb{R}$ with codebook given by

$$\mathcal{C}u := \{0, \pm b, \pm 2b, \cdots\}. \tag{29}$$

Here, $b(> 0)$ is the step size. Let $\boldsymbol{H}_U$, which is the entropy of the quantizer, be defined as

$$\boldsymbol{H}_U := \sum_{\ell=1}^{L} P_\ell \log_2 P_\ell \tag{30}$$

where P_ℓ is the nonzero probability that the quantizer output is the ℓth codeword in $\mathcal{C}_U$, and suppose that $P_\ell \neq 0$ for $\ell = 1, \cdots, L$, and zero otherwise. In the case of high entropies of $\boldsymbol{H}_U$, the PQ that employs the uniform quantizer will yield the minimum entropy, and this minimum is higher than the rate distortion bound by only about one fourth of a bit, which corresponds to about 1.53 dB. The next example will show that using SAPQ can reduce this 1.53 dB gap.

Example 3 (SAPQ Based on Lattice $D_m^\perp$ and Output Entropy): Suppose that $r_i = (b/40)\cdot i (\in \mathbb{R})$ for $i = 0, \cdots, 20$. For a given i, consider two cosets $C_1 = r_i + \mathcal{C}_U$ and $C_2 = -r_i + \mathcal{C}_U$ as the codebooks in an SAPQ with $\eta = 1$. Note that if $i = 0$, then C_1 is a midtread codebook, and $C_1 = C_2$; if $i = 20$, then C_1 is a midrise codebook and $C_1 = C_2$, and if $i = 10$, then the set $\boldsymbol{C} = C_1 \times C_2$ is a lattice that is equivalent to $D_m^\perp$. For the $i \geq 21$ case, the set $\boldsymbol{C}$ is equal to one of the cases for $i = 0, \cdots, 20$. Hence, we will consider only the cases for $i = 0, \cdots, 20$, i.e., $0 \leq r_i \leq b/2$. Let $\boldsymbol{H}_{\text{SAPQ}}$ denote an entropy of the SAPQ using $\boldsymbol{C}$ as a codebook in m-dimensions, where $\boldsymbol{H}_{\text{SAPQ}}$ is defined as

$$\begin{aligned}\boldsymbol{H}_{\text{SAPQ}} := &-\Pr\{C_1 \text{ used for } \boldsymbol{X}\}\\ &\cdot \sum_{\ell=1}^{L_1} P_{1,\ell}\log_2 P_{1,\ell} - \Pr\{C_2 \text{ used for } \boldsymbol{X}\}\\ &\cdot \sum_{\ell=1}^{L_2} P_{2,\ell}\log_2 P_{2,\ell} + \frac{1}{m}.\end{aligned} \tag{31}$$

Here, $1/m$ is the side information given by the SAPQ with $\eta = 1$

$$P_{j,\ell} := \Pr\{\text{output} = \ell\text{th codeword in } C_j | C_j \text{ used for } \boldsymbol{X}\} \tag{32}$$

and it is assumed that $P_{j,\ell} \neq 0$ for $\ell = 1, \cdots, L_j$ and zero otherwise. Several numerical results are summarized in Tables III and IV for Gaussian and Laplacian density functions, respectively. In these tables, the step size of the SAPQ codebook $\boldsymbol{C}$ is denoted by b_{SAPQ}. Note that the rates of both the PQ and SAPQ satisfy $\boldsymbol{H}_U \leq R$ and $\boldsymbol{H}_{\text{SAPQ}} \leq R$, respectively. In this simulation, we conducted the SAPQ for the 21 values of r_i and found r_i that yields the minimum distortion. (Variation of $\boldsymbol{H}_{\text{SAPQ}}$ is very small for the various values of r_i, especially for the high entropy case.) For the Gaussian density function case (see Table III), most of the results show the minimum distortion at $r_{10} = b_{\text{SAPQ}}/4$. Since $D_{\text{SLB}} = \sigma^2 2^{-2R}$, where the variance $\sigma^2 = 1$, the distortion of PQ at $\boldsymbol{H}_U = 4.170$ has about a 1.53-dB difference from the SLB as shown in Table III. In this case, SAPQ achieves about 0.49 dB gain over the PQ, as expected from the discussion following (11). However, this gain decreases as the entropy $\boldsymbol{H}_{\text{SAPQ}}$ decreases. This fact can be explained in a similar manner to Example 1. Table IV shows the numerical result for a Laplacian density function. At $\boldsymbol{H}_U = 4.073$, the difference between D_{PQ} and the SLB is about 1.56 dB, and the gain from the SAPQ is about 0.46 dB. (Note that $D_{\text{SLB}} = \sigma^2 e/\pi 2^{-2R}$.) For the Laplacian case, we again note that for large step sizes, r_i is less than $r_{10} = b_{\text{SAPQ}}/4$, and the performance of SAPQ can even be worse than that of the PQ. However, in both the cases, if the step size is smaller than or equal to the standard deviation (in these cases, 1), then, as shown in Tables III and IV, there is a reasonable gain. Note that these gains come from the space-filling advantage.

TABLE III
SAPQ BASED ON $D_m^\perp$ AND OUTPUT ENTROPY (EXAMPLE 3). DISTORTIONS (IN DECIBELS) FOR GAUSSIAN I.I.D. WITH VARIANCE 1, $m = 8$, AND $\eta = 1$

Step Size: b_{SAPQ}	$\mathbf{H}_{\text{SAPQ}}$ ($\mathbf{H}_{\text{U}}$)	PQ (SQ)	SAPQ
0.25	4.170	-23.58	-24.07
0.5	3.179	-17.56	-18.04
1	2.213	-11.55	-12.01
2	1.331	-5.44	-6.01
4	0.458	-1.06	-1.14

TABLE IV
SAPQ BASED ON $D_m^\perp$ AND OUTPUT ENTROPY (EXAMPLE 3). DISTORTIONS (IN DECIBELS) FOR LAPLACIAN I.I.D. WITH VARIANCE 1, $m = 8$, AND $\eta = 1$

Step Size: b_{SAPQ}	$\mathbf{H}_{\text{SAPQ}}$ ($\mathbf{H}_{\text{U}}$)	PQ (SQ)	SAPQ
0.25	4.073	-23.59	-24.05
0.5	3.088	-17.62	-18.03
1	2.134	-11.77	-12.05
2	1.255	-6.43	-6.41
4	0.509	-2.38	-2.23

Suppose that an entropy coding scheme is employed in the SAPQ of Example 6, and let $\overline{R}$ denote the resultant bit rate. The optimal entropy coding, where $\overline{R} = \boldsymbol{H}_{\text{SAPQ}}$, can only be reached if the probability $P_{j,\ell}$ satisfies the Shannon–Fano integral constraint. Otherwise, the bit rate $\overline{R}$ that results from entropy coding will be slightly higher than $\boldsymbol{H}_{\text{SAPQ}}$. It is useful to use entropy coding on vector (rather than on single outputs) in order to reduce the difference between $\overline{R}$ and $\boldsymbol{H}_{\text{SAPQ}}$. For example, a 3-D variable-length coding scheme is employed for the DCT coefficient coding in ITU-T, H.263 [22].

We can also use a nonuniform quantizer with entropy coding in order to obtain a higher gain than does the uniform quantizer with entropy coding. For example, an application of SAPQ to the quantizers for the very low bit rate video coding scheme based on H.263 is studied in [30].

C. *Nonuniform SAPQ*

We now introduce several examples to demonstrate the gain from the space-filling and shape-advantages for nonuniform sources for different codebook sizes and η in conjunction with the encoding complexity.

The design problem of SAPQ is to find an optimal codebook that achieves the distortion $\inf_{\mathcal{C}_{\text{SAPQ}}} D_{\text{SAPQ}}$, for a fixed rate R. However, finding such an optimal codebook is not easy for the nonuniformly distributed inputs. In order to find (sub)optimal codebooks, we have developed a clustering algorithm that uses a large number of samples as a TS for given values of m, n_i', and η, but this TS size is still substantially less than that of traditional VQ or modified schemes since the total number of codewords to be designed is smaller than those of VQ. Let $x_{1,\ell}, \cdots, x_{m,\ell}$ denote the ℓth training sample in a given TS that has M samples, where a sample has m *training points*. The clustering algorithm has two parts, which are from two necessary conditions, respectively, for an optimal SAPQ. The first part of our algorithm quantizes m training points in each sample using 2^η different codebooks and then selects a codebook that yields the minimal distance [given in (3)] for the sample. The second part of the algorithm updates the codebooks using the partitioned TS in the quantization process of the first part. (Regarding the second part, see the Appendix.) These two parts are then iteratively applied to the given TS. The clustering algorithm is described below.

Clustering Algorithm (SAPQ):

0) Initialization ($\gamma = 0$): Given codebook sizes n_i', $i = 1, \cdots, m$, sample size m, side bits η, distortion threshold $\epsilon \geq 0$, initial codebook $\boldsymbol{C}_0$, and TS $((x_{1,\ell}, \cdots, x_{m,\ell}))_{\ell=1}^M$, set $D_{-1} = \infty$.

1) Given codebook $\boldsymbol{C}_\gamma = \bigcup_{j=1}^{2^\eta}(C_{1,j} \times \cdots \times C_{m,j})$, where $C_{i,j} \in \mathcal{C}_{n_i'}$, find $2^\eta \sum_{i=1}^m n_i'$ partitions of each training points in the TS for the corresponding $2^\eta \sum_{i=1}^m n_i'$ codewords, where each training point's codeword is determined by the following quantization:

$$d_\ell := \min_j \frac{1}{m} \sum_{i=1}^m \min_{y \in C_{i,j}} (x_{j,\ell} - y)^2, \quad \text{for } \ell = 1, \cdots, M. \tag{33}$$

Next, we compute the average distortion D_γ for the γth iteration, which is given by

$$D_\gamma := \frac{1}{M} \sum_{\ell=1}^M d_\ell. \tag{34}$$

2) If $(D_{\gamma-1} - D_\gamma)/D_\gamma \leq \epsilon$, stop. $\boldsymbol{C}_\gamma$ is the final codebook. Otherwise, continue.

3) Compute centroids for each of the $2^\eta \sum_{i=1}^m n_i'$ partitions and replace the codewords in $\boldsymbol{C}_\gamma$ by the new $2^\eta \sum_{i=1}^m n_i'$ centroids. Increase γ by 1. Go to Step 1.

It can be shown using similar techniques as in the case of the Lloyd–Max algorithm (Lloyd's Method II) [34], [35] or the K-means algorithm [2] that D_γ is a decreasing sequence. Thus, D_γ converges to a (local) minimum, which depends on the initial codebook $\boldsymbol{C}_0$. The next example shows an effect of the initial codebook in the clustering algorithm.

Example 4 (Initial Guess in Clustering Algorithm): The clustering algorithm can be used to effectively design the SAPQ codebook using the TS that has an underlying distribution function. However, the performance of the designed SAPQ is quite dependent on choosing the initial codebook $\boldsymbol{C}_0$. An example of the different choices of the initial guess is illustrated in Fig. 2 by plotting the codebook of the SAPQ in m-dimensions, where $m = 2$. (Note that in Fig. 2, each figure has a codebook $\boldsymbol{C}$ that is the union of two product codebooks ($2^\eta = 2$), and each product codebook has four codewords [$(n_i')^m = 4$, and $C_{1,j} = C_{2,j}$]. Fig. 2(a) and (b) show the converged codebooks in the clustering algorithm. However, the corresponding initial codebooks also have similar arrangements to the converged codebooks. This fact implies that the designed SAPQ codebook is quite dependent on the initial codebook $\boldsymbol{C}_0$. Furthermore, if n_i' and η are large, then we have many choices of the initial codebooks. Hence, finding a globally optimal codebook for an input is quite difficult, except for several trivial cases. In Fig. 2(a), we have employed a simple *split method*, which will be introduced in this example [17], to determine the initial codebook $\boldsymbol{C}_0$. The split method doubles the number of the product codebooks by adding and subtracting a small constant

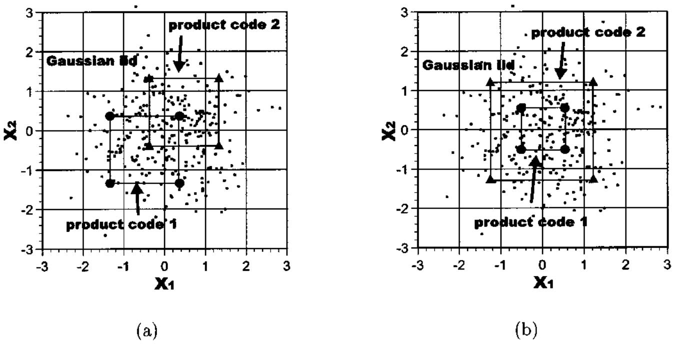


Fig. 2. Codebooks of SAPQ in m-dimensions for different initial guesses (Gaussian i.i.d. input with the variance 1, $n = 2$, $m = 2$, and $\eta = 1$. Note that each codebook $\boldsymbol{C}$ is the union of two product codebooks.). (a) Initial guess 1 (distortion: −6.93 dB). (b) Initial guess 2 (distortion: −6.14 dB).

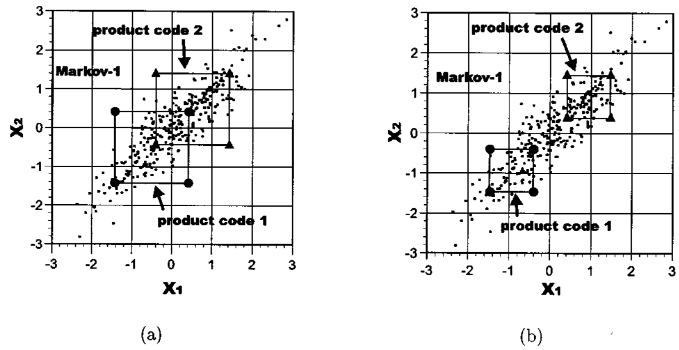


Fig. 3. Codebooks of SAPQ in m-dimensions for different initial guesses (Gaussian Markov-1 source with the variance 1 and the correlation coefficient 0.9, $n = 2$, $m = 2$, and $\eta = 1$). (a) Split method with $\varepsilon = 0.01$ (distortion: −8.07 dB). (b) Split method with $\varepsilon = 1$ (distortion: −8.73 dB).

$\varepsilon (\in \mathbb{R})$. For the generation of an initial codebook $\boldsymbol{C}_0$ from the split method, we need a start codebook that is denoted by $\boldsymbol{C}_0^0$ in $\mathbb{R}^m$. The start codebook $\boldsymbol{C}_0^0 = C_{1,1}^0 \times \cdots \times C_{m,1}^0$ contains codebooks $C_{i,1}^0$ that belong to $\mathcal{C}_{n_i'}$, where $C_{i,1}^0$ is the Lloyd–Max quantizer that is optimal for X_i.

Initial Codebook Guess (Split Method for SAPQ):

0) Initialization ($\gamma = 0$): Given codebook size n_i', sample size m, side bits η, split constant $\varepsilon \geq 0$, start codebook $\boldsymbol{C}_0^0 \subset \mathbb{R}^m$, and TS $((x_{1,\ell}, \cdots, x_{m,\ell}))_{\ell=1}^M$.

1) If $\gamma \geq \eta$, stop. $\boldsymbol{C}_0^\gamma$ is the initial codebook $\boldsymbol{C}_0$ for the clustering algorithm. Otherwise continue.

2) Increase γ by 1. Construct a new codebook $\boldsymbol{C}_0^\gamma = \bigcup_{j=1}^{2^\gamma} (C_{1,j}^\gamma \times \cdots \times C_{m,j}^\gamma)$ by doubling the number of codebooks from $\boldsymbol{C}_0^{\gamma-1} = \bigcup_{j=1}^{2^{\gamma-1}} (C_{1,j}^{\gamma-1} \times \cdots \times C_{m,j}^{\gamma-1})$ as follows.

$$C_{i,j}^\gamma = -\varepsilon + C_{i,j}^{\gamma-1} \text{ and } C_{i,2^{\gamma-1}+j}^\gamma = \varepsilon + C_{i,j}^{\gamma-1} \qquad (35)$$

for $i = 1, \cdots, m$ and $j = 1, 2, \cdots, 2^{\gamma-1}$.

3) Given $\boldsymbol{C}_0^\gamma$, find $2^\gamma \sum_{i=1}^m n_i'$ partitions of training points according the quantization

$$\min_{j \in \{1, 2, \cdots, 2^\gamma\}} \frac{1}{m} \sum_{i=1}^m \min_{y \in C_{i,j}^\gamma} (x_{i,\ell} - y)^2, \quad \text{for } \ell = 1, \cdots, M. \qquad (36)$$

Compute the centroids for each of the $2^\gamma \sum_{i=1}^m n_i'$ partitions, and replace the codewords in $\boldsymbol{C}_0^\gamma$ by the new $2^\gamma \sum_{i=1}^m n_i'$ centroids. Go to Step 1.

Fig. 3 illustrates an example of the constant ε in the split method for a correlated input. We note that depending on ε, we can obtain different converged SAPQs as shown in Fig. 3(a) and (b), respectively. In this split method, we will set $\varepsilon = 0.001$ during the simulation.

D. Comparison of SAPQ with Other Quantizers

Note that the SAPQ in (4) requires at most $m2^\eta$ different codebooks. Hence, if m is large, the decoder needs a large memory for the codebooks, and the codebook design complexity may be high. In order to reduce the required number of codebooks, one possibility is to use the same codebooks in calculating the distance of (3) under an assumption that the random variables $X_1, \cdots, X_m$ are identically distributed. In other words, $C_{i,j}$ are set equal for $i = 1, \cdots, m$. We can regard this scheme as a *codebook-constrained SAPQ* and in this case, the average distortion is given as

$$D_{\text{SAPQ}} = E\left\{ \min_j \frac{1}{m} \sum_{i=1}^m \min_{y \in C_j} (X_i - y)^2 \right\}. \qquad (37)$$

Here, the index i is omitted in the codebook notation, i.e., C_j. Note that the number of required codebooks is reduced to 2^η, and the bit rate is given by $\log_2 n' + \eta/m$, if $C_j \in \mathcal{C}_{n'}$, for all j. As shown in the asymptotic analysis of Section III, increasing m for a fixed value of n yields more gain over PQ in the SAPQ of (4). However, for the SAPQ in (37), the decrease in distortion can be seen to diminish for large values of m, and the distortion will eventually increase and converge to that of the n'-level quantizer [29, Prop. 2]. Therefore, to obtain gains in the SAPQ, it is important to use as large a value for m (and n') as possible while keeping the ratio m/n' small (note that since increasing n' increases the total bit rate, this implies that for a given bit rate, the side information η should be accordingly decreased). Furthermore, if we employ the split method as an initial guess in the clustering algorithm, the distortion of the SAPQ in (37) is nearly the same as that of the SAPQ in (4) [29]. Therefore, for a relatively large n' (compared with η) and a fixed ratio of m/n', it is advantageous to use the SAPQ in (37) since its performance will closely approximate that of SAPQ, and the number of required codebooks is 2^η.

Since the SAPQ in (37) has a scalar quantizer structure, i.e., the codewords of SAPQ belong to $\mathbb{R}$, we can easily apply to the current quantization schemes. For example, we can implement SAPQ based on lookup tables, and apply SAPQ to the *differential PCM* schemes [23] that use traditional scalar quantizers and predictors [30].

Example 5 (Encoding Complexity and Asymptotic Distortion): In full-search VQ, suppose that ν denotes the size of codebook. The number of multiplications required for encoding are then ν and $2^{\eta+1} \log_2 n'$ for the full-search VQ and the codebook-constrained SAPQ, respectively. If the bit rates of VQ and SAPQ are the same, i.e., $\nu = (n')^m 2^\eta$, then $\nu > 2^{\eta+1} \log_2 n'$ from $2 \log_2 n' < (n')^m$ for $m = 3, 4, \cdots$. This implies that the number of multiplications for VQ is always greater than that of the SAPQ. Hence, for $m \geq 3$, the

TABLE V
COMPARISON OF SAPQ (EXAMPLE 6). DISTORTIONS (IN DECIBELS) AT BIT RATE 4.5

	VQ	TSVQ	MSVQ		SAPQ		
Block Length (m)	2	2	2	4	2	4	8
Codebook Size	$\nu = 512$		$n'' = 8$	$n'' = 64$	$n' = 16$	$n' = 16$	$n' = 16$
Breadth (b)		2					
Depth (d)		9					
Stages (g)			3	3			
Side Bits (η)					1	2	4
Multiplications	512	18	24	192	16	32	128
Memory	1024	2044	48	768	32	64	256
Gaussian i.i.d	-23.8	-23.3	-19.0	-21.1	-23.7	-24.2	-24.7
Laplacian i.i.d	-22.9	-22.1	-16.7	-19.9	-22.0	-23.0	-23.9

encoding complexity of SAPQ is always less than that of VQ. Further, since SAPQ has a structurally constraint codebook compared with the arbitrary codebooks of full-search VQs, the distortion of the SAPQ is always less than or equal to that of the full-search VQ. In a similar manner for the memory size case, we have $m\nu > 2^{\eta}n'$ at the same bit rates. Hence, we can design an SAPQ, which requires a smaller codebook than the traditional VQ.

We now increase the block length of SAPQ from m to $m'(> m)$ and keep the bit rates the same, i.e., $\nu^{1/m'} = (n')2^{\eta/m}$. Then, there is an integer θ such that $\eta > \theta$ implies that $\nu > 2^{\eta+1}\log_2 n'$, i.e., the encoding complexity of m'-dimensional SAPQ is less than that of full-search m-dimensional VQ. From Section III-C, $\|f\|_{\rho'} < \|f\|_{\rho}$, where $\rho' := m'/(m'+2)$. Further, from [10], since there is a lower bound for J_m, and for appropriate values of m and m', we have a relationship $J_{m'} < J_m$ [9], [10, Fig. 1]. Therefore, from Theorem 1, there exist block lengths m and m' such that

$$\limsup_{\eta\to\infty}[(n')^{m'}2^{\eta}]^{2/m'} \inf_{\mathcal{C}_{\text{SAPQ}}} D_{\text{SAPQ}} \le J_{m'}\|f\|_{\rho'} < J_m\|f\|_{\rho}. \tag{38}$$

In other words, we can design a better SAPQ than the traditional VQ in an asymptotic sense while obtaining a lower encoding complexity. A numerical result on this fact will be introduced in Example 6.

Example 6 (Numerical Comparison of SAPQ): An extensive comparison on SAPQ in terms of the average distortion, encoding complexity, and memory requirement is shown in our early work [29], where the SAPQ is based on k-dimensional vector quantizers. More results on the SAPQ of (37), where the SAPQ is based on scalar quantizers, are summarized in Table V. Note that in this simulation, the full-search VQ was designed by the generalized Lloyd algorithm (GLA) [17, p. 362]. We also compared the SAPQ with the multistage VQ (MSVQ) [17, p. 451] since MSVQ is one of the quantization schemes that can reduce both the encoding complexity and memory requirement. However, as we can see in Table V, the average distortion of MSVQ is significantly worse than the average distortion of the other quantizers. As discussed in Example 5, we can design an SAPQ whose distortion and complexities in terms of encoding and memory requirement are better than the full-search VQ, as shown in the SAPQ cases of $m = 4$ and $m = 8$ in Table V). This fact implies that even though the asymptotic analysis in Section III only shows the converged results without any results about the convergence speed, we can design a good SAPQ for

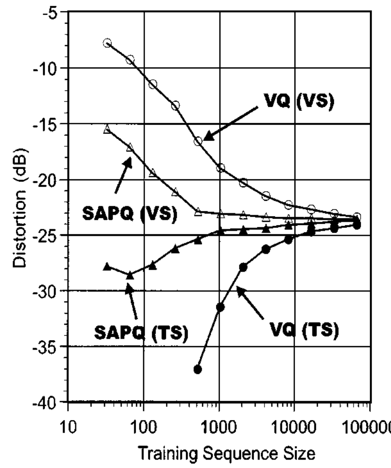


Fig. 4. SAPQ and VQ trained on finite TS's (VQ: $m = 2, \nu = 512$, and SAPQ: $m = 2, n' = 16, \eta = 1$ for Gaussian i.i.d. with variance 1 at bit rate 4.5).

TABLE VI
COMPARISON OF SAPQ TRAINED FINITE IN TS (EXAMPLE 7). DISTORTIONS (IN DECIBELS) FOR GAUSSIAN I.I.D., VS SIZE: 65 536, AT BIT RATE 4.5

TS Size	128	1,024	8,192	65,536	$> 5,242,880$
VQ ($m = 2$, $\nu = 256$)	-11.5	-19.0	-22.3	-23.4	-23.8
TSVQ ($m = 2$, $b = 2$, $d = 9$)	-10.6	-17.6	-21.7	-23.0	-23.3
SAPQ ($m = 2$, $n' = 16$, $\eta = 1$)	-19.4	-23.1	-23.5	-23.6	-23.7
SAPQ ($m = 4$, $n' = 16$, $\eta = 2$)	-19.2	-23.4	-23.9	-24.1	-24.2
SAPQ ($m = 8$, $n' = 16$, $\eta = 4$)	-16.8	-22.9	-24.1	-24.4	-24.7

small (and hence implementable) parameters m, n', and η. For correlated sources, such as the Markov-1 sequences [23, p. 62], several numerical results for various quantization schemes, including predictive VQ, are also shown in [30].

Example 7 (SAPQ Trained on Finite TS): Since VQs are usually designed by clustering training sequences, the average distortion of VQ is dependent on the choice of the TS and its size. The size of the TS is especially important in designing a good codebook for an underlying distribution function. In general, the training ratio, which is defined as the ratio of the TS size to the codebook size [17, p. 364], indicates how close the trained codebook is to an optimal one for the distribution function [26], [27]. From [1] and [41], it is known that a large TS ensures a good codebook for the distribution function. However, the size of a TS could be quite different, depending on the quantization schemes. For a similar bit rate and quantizer distortion, a quantization scheme, which requires a smaller TS, is obviously better. In Fig. 4, the distortions of trained codebooks of VQ and SAPQ are tested on a validating sequence (VS). (In testing a codebook using a VS, there is no need to use a large VS [27]. In this simulation, we used 65 536 elements for the VS.) As we can see in Fig. 4, SAPQ requires much smaller TS sizes, and SAPQ (for VS) always shows better performance than the VQ cases. In other words, for the TS sizes as in Fig. 4, SAPQ is even better than the full-search VQ. Further, in Table VI, the trained codebooks of several quantization schemes on finite TS's are compared. Through this table, we can infer that the SAPQ yields less distortion than the full-search VQ if the codebooks are designed by using finite TS's.

V. CONCLUSION

In this paper, we have studied our newly introduced sample-adaptive product quantizer (SAPQ) [29] from an asymptotic aspect and with several examples. The SAPQ scheme that is considered in this paper is based on m scalar quantizers. This SAPQ is, hence, very appealing from a practical implementation point of view. Through an asymptotic analysis based on lattices, we have designed lattice VQs by applying SAPQ and numerically compared their performance. We have also shown that SAPQ can achieve better performance than the full-search VQ in an asymptotic sense, while maintaining lower encoding complexities and memory requirement than the full-search VQ. This asymptotic result is also numerically observed in this paper. In designing regular VQ by clustering a TS since the size of TS is limited to some finite values, the trained codebook performance is quite dependent on TS sizes. However, for relatively small sizes of TSs, we show that the average distortion of SAPQ is better than full-search VQs. Further, SAPQ can even be applied for high bit rates, where conventional VQ (or even modified VQ) techniques are very difficult to use. The scalar quantizer structure of SAPQ also allows us to easily apply it to current coding systems and generate VQ-level performance.

APPENDIX
SECOND NECESSARY CONDITION

For an SAPQ codebook $\bigcup_{j=1}^{J}(C_{1,j}\times\cdots\times C_{m,j})$, denote a codebook $C_{i,j}(\in\mathcal{C}_{n'}\subset\mathbb{R})$ as

$$C_{i,j}:=\{y_{i,1}(j),\cdots,y_{i,n'}(j)\} \tag{A1}$$

where $y_{i,1}(j),\cdots,y_{i,n'}(j)\in\mathbb{R}$. For a fixed j, the product codebook is given by

$$\begin{aligned}(C_{1,j}&\times\cdots\times C_{m,j})\\ &=\{(y_{1,1}(j),y_{2,1}(j),\cdots,y_{m,1}(j))\\ &\quad(y_{1,1}(j),y_{2,1}(j),\cdots,y_{m,2}(j)),\cdots\\ &\quad(y_{1,n'}(j),y_{2,n'}(j),\cdots,y_{m,n'}(j))\}.\end{aligned} \tag{A2}$$

Note that the SAPQ codebook is the unions of such product codebooks (A2) for $j=1,\cdots,2^{\eta}$. Hence, the SAPQ codebook has $(n')^m 2^{\eta}$ codewords in $\mathbb{R}^m$. Let $S_{\ell_1,\ldots,\ell_m}(j)(\subset\mathbb{R}^m)$ denote the quantizer region corresponding to the codeword $(y_{1,\ell_1}(j),\cdots,y_{m,\ell_m}(j)) =: \boldsymbol{y}_{\ell_1,\ldots,\ell_m}(j)$ for $\ell_1,\cdots,\ell_m = 1,\cdots,n'$. The SAPQ average distortion in (4) can be rewritten as

$$\begin{aligned}D_{\mathrm{SAPQ}}=\sum_{j=1}^{2^{\eta}}\sum_{\ell_1=1}^{n'}\cdots\sum_{\ell_m=1}^{n'}&\int_{S_{\ell_1,\ldots,\ell_m}(j)}\\ &\cdot\|\boldsymbol{x}-\boldsymbol{y}_{\ell_1,\ldots,\ell_m}(j)\|^2\,dF(\boldsymbol{x})\end{aligned} \tag{A3}$$

where $\boldsymbol{x}:=(x_1,\cdots,x_m)\in\mathbb{R}^m$, and F is the distribution function of the input. We now obtain the second necessary condition by differentiating D_{SAPQ} with respect to the $y_{i,\ell}(j)$ s and setting derivatives equal to zero:

$$\begin{aligned}&\int_{S_{i,\ell}(j)}[x_i-y_{i,\ell}(j)]\,dF(\boldsymbol{x})=0,\\ &\quad\text{for } i=1,\cdots,m,\ j=1,\cdots,J,\ \ell=1,\cdots,n'\end{aligned} \tag{A4}$$

where $S_{i,\ell(j)}:=\bigcup_{\ell_1=1}^{n'}\cdots\bigcup_{\ell_{i-1}=1}^{n'}\bigcup_{\ell_{i+1}=1}^{n'}\cdots\bigcup_{\ell_m=1}^{n'} S_{\ell_1,\cdots,\ell_{i-1},\ell,\ell_{i+1},\cdots,\ell_m}(j)$. In other words, the necessary condition is given by

$$\begin{aligned}y_{i,\ell}(j)=\int_{S_{i,\ell}(j)}x_i\,dF(\boldsymbol{x})\int_{S_{i,\ell}(j)}dF,&\\ \text{for } i=1,\cdots,m,\ j=1,\cdots,J,\ \ell=1,\cdots,n'.&\end{aligned} \tag{A5}$$

REFERENCES

[1] E. A. Abaya and G. L. Wise, "Convergence of vector quantizers with applications to optimal quantization," *SIAM J. Appl. Math.*, vol. 44, pp. 183–189, 1984.

[2] M. R. Anderberg, *Clustering Analysis for Applications*. New York: Academic, 1973.

[3] E. S. Barnes and N. J. A. Sloane, "The optimal lattice quantizer in three dimensions," *SIAM J. Alg. Discrete Methods*, vol. 4, no. 1, pp. 30–41, Mar. 1983.

[4] E. F. Beckenbach and R. Bellman, *Inequalities*. New York: Springer-Verlag, 1961.

[5] R. Bhatia, *Matrix Analysis*. New York: Springer-Verlag, 1997.

[6] J. A. Bucklew and G. L. Wise, "Multidimensional asymptotic quantization theory with rth power distortion measures," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 239–247, Mar. 1982.

[7] P. A. Chou, "Codebook clustering for weighted universal VQ and other applications," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Budapest, Hungary, June 1991, p. 253.

[8] J. H. Conway and N. J. A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 211–226, Mar. 1982.

[9] ——, "On the Voronoi regions of certain lattices," *SIAM J. Alg. Discrete Methods*, vol. 5, no. 3, pp. 294–305, Sep. 1984.

[10] ——, "A lower bound on the average error of vector quantizers," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 106–109, Jan. 1985.

[11] M. Effros and P. A. Chou, "Weighted universal bit allocation: Optimal multiple quantization matrix coding," in *Proc. IEEE ICASSP*, Detroit, MI, May 1995, pp. 2343–2346.

[12] ——, "Weighted universal transform coding: Universal image compression with the Karhunen–Loève transform," in *Proc. IEEE ICASSP*, Detroit, MI, May 1995, pp. 2343–2346.

[13] M. Effros, P. A. Chou, and R. M. Gray, "Universal image compression," *IEEE Trans. Image Processing*, vol. 8, pp. 1317–1329, Oct. 1999.

[14] K. R. Farrell, R. J. Mammone, and K. T. Assaleh, "Speaker recognition using neural networks and conventional classifier," *IEEE Trans. Speech Audio Processing*, pt. II, vol. 2, pp. 194–205, Jan. 1994.

[15] N. Farvardin and J. W. Modestino, "Optimal quantizer performance for a class of non-Gaussian memoryless sources," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 485–497, May 1984.

[16] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 373–380, July 1979.

[17] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.

[18] I. A. Gerson and M. A. Jaiuk, "Vector sum excited linear prediction (VSELP) speech coding at 8 Kbps," in *Proc. IEEE ICASSP*, Albuquerque, NM, Apr. 1990, pp. 461–464.

[19] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. New York: Addison Wesley, 1992.

[20] J. He, L. Liu, and G. Palm, "A discriminative training algorithm for VQ-based speaker identification," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 353–356, May 1999.

[21] J. E. Humphreys, *Introduction to Lie Algebras and Representation Theory*. New York: Springer-Verlag, 1972.

[22] ITU-T, "Video coding for low bit rate communication," ITU-T Recommendation H.263, Jan. 1998.

[23] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[24] S. Katagiri and C.-H. Lee, "A new hybrid algorithm for speech recognition based on HMM segmentation and leaning vector quantization," *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 421–430, Oct. 1993.

[25] J. C. Kieffer, "A survey of the theory of source coding with a fidelity criterion," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1473–1490, Sep. 1993.

[26] D. S. Kim and M. R. Bell, "Bounds on the trained vector quantizer distortion measured using training data," Purdue Univ., West Lafayette, Tech. Rep. TR-ECE 98-6, Apr. 1998.

[27] D. S. Kim, T. Kim, and S. U. Lee, "On testing trained vector quantizer codebooks," *IEEE Trans. Image Processing*, vol. 6, pp. 398–406, Mar. 1997.

[28] D. S. Kim and S. U. Lee, "Image vector quantizer based on a classification in the DCT domain," *IEEE Trans. Commun.*, vol. 39, pp. 549–556, Apr. 1991.

[29] D. S. Kim and N. B. Shroff, "Quantization based on a novel sample-adaptive product quantizer (SAPQ)," *IEEE Trans. Inform. Theory*, vol. 45, pp. 2306–2320, Nov. 1999.

[30] ——, "Novel quantization scheme for very low bit rate video coding based on sample adaptation," Purdue Univ., West Lafayette, IN, Tech. Rep. TR-ECE 99-9, June 1999.

[31] R. Laroia and N. Farvardin, "A structured fixed-rate vector quantizer derived from a variable-length scalar quantizer: Part I—Memoryless sources," *IEEE Trans. Inform. Theory*, vol. 39, pp. 851–867, May 1993.

[32] ——, "A structured fixed-rate vector quantizer derived from a variable-length scalar quantizer: Part II—Vector sources," *IEEE Trans. Inform. Theory*, vol. 39, pp. 868–876, May 1993.

[33] W. P. LeBlanc, B. Bhattacharya, S. A. Mahnound, and V. Cuperman, "Efficient search and design procedures for robust multi-stage VQ of LPC parameters for 4 kb/s speech coding," *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 373–385, Oct. 1993.

[34] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COMM-28, no. 1, pp. 84–95, Jan. 1980.

[35] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 128–137, Mar. 1982.

[36] T. D. Lookabaugh and R. M. Gray, "High-resolution quantization theory and the vector quantizer advantage," *IEEE Trans. Inform. Theory*, vol. 35, pp. 1020–1033, Sep. 1989.

[37] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, pp. 1551–1588, Nov. 1985.

[38] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memotuless and Gaussian–Markov source," *IEEE Trans. Commun.*, vol. 38, no. 1, pp. 82–93, Jan. 1990.

[39] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, vol. IT-6, pp. 7–12, Mar. 1960.

[40] A. Ortega and M. Vetterli, "Adaptive scalar quantization without side information," *IEEE Trans. Image Processing*, vol. 6, pp. 665–676, May 1997.

[41] D. Pollard, "Quantization and the method of k-means," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 199–205, Mar. 1982.

[42] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[43] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

[44] R. P. Ramachandran, M. S. Zilovic, and R. J. Mammone, "A comparative study of robust linear predictive analysis methods with applications to speaker identification," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 117–125, Jan. 1995.

[45] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (CELP): High-quality speech at very low bit rates," in *Proc. IEEE ICASSP*, Tampa, FL, Mar. 1985, pp. 937–940.

[46] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in *IRE Nat. Conv. Rec.*, Mar. 1959, pp. 142–163.

[47] N. J. A. Sloane, "Tables of sphere packings and spherical codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 327–338, May 1981.

[48] K.-Y. Su and C.-H. Lee, "Speech recognition using weighted HMM and subspace projection approaches," *IEEE Trans. Speech Audio Processing*, pt. I, vol. 2, pp. 69–79, Jan. 1994.

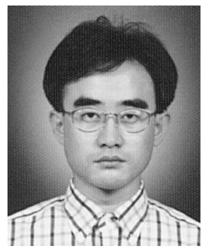
[49] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.

[50] P. Zador, "Asymptotic quantization error of continuous signals and the quantization dimension," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 139–149, Mar. 1982.

[51] K. Zeger, A. Bist, and T. Linder, "Universal source coding with codebook transmission," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 336–346, Feb./Mar./Apr. 1994.

[52] Y. Zhang, R. Toogneri, and M. Alder, "Phoneme-based vector quantization in a discrete HMM speech recognizer," *IEEE Trans. Speech Audio Processing*, vol. 5, pp. 26–32, Jan. 1997.

[53] Z. Zhang and V. K. Wei, "An on-line universal lossy data compression algorithm via continuous codebook refinement—Part I: Basic Results," *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 803–821, May 1996.



Dong Sik Kim (S'86–M'95) received the B.S., M.A., and Ph.D. degrees from Seoul National University (SNU), Seoul, Korea, in 1986, 1988, and 1994, respectively, all in electrical engineering.

From 1993 to 1996, he was a Special Researcher with the Institute of New Media and Communications and the Engineering Research Center for Advanced Control and Instrumentation, SNU. From 1996 to 1998, he was a Visiting Scholar and, from 1998 to 1999, a Visiting Assistant Professor with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. Since 1986, he has been a Research Director with Automan Co., Ltd., Seoul. With Automan, he has conducted radio frequency (RF) circuit design projects and has several patents. He joined Hallym University, Kangwon-Do, Korea, in 1999, where he was an Assistant Professor with the Division of Information and Communication Engineering. He is now with the Hankuk University of Foreign Studies, Yongin, Korea. His research interests are the theory of quantization, postprocessing and error concealment of coded images, multimedia in networks, video data compression for (wireless) network environments, and RF circuit analysis and design.

Ness B. Shroff (S'90–M'94) received the B.S. degree in from the University of Southern California, Los Angeles, the M.S.E. degree from the University of Pennsylvania, Philadelphia, and the M.Phil. and Ph.D. degrees from Columbia University, New York, NY.

He is currently an Assistant Professor with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. During his doctoral study, he worked at AT&T Bell Labs in 1991 and Bell Communications Research in 1992, on problems involving fault management in telephone networks. His current research interests are in high speed broadband and wireless communication networks. He is especially interested in studying issues related to performance modeling and analysis, routing, network management, scheduling, and control in such networks. He has received research and equipment grants to conduct fundamental work in broadband and wireless networks and quantization from the National Science Foundation, AT&T, Hewlett Packard, Intel, LG Electronics, and the Purdue Research Foundation.

Dr. Shroff received the NSF CAREER award from the National Science Foundation in 1996. He has served on the Technical Program Committees of various conferences and on NSF review panels. He was the Conference Chair for the 14th Annual IEEE Computer Communications Workshop (CCW) and is Program Co-Chair for the High-Speed Networking Symposium at Globecomm 2000. He is also Editor of IEEE COMMUNICATION LETTERS and *Computer Networks*.