SEQUENTIAL SCALAR QUANTIZATION OF VECTORS: AN ANALYSIS

Raja Balasubramanian, Charles A. Bouman, Jan P. Allebach

School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

Appeared in*:*

IEEE Transactions on Image Processing, vol. 4, no. 9, pp. 1282-1295, September 1995.

Corresponding Author:
Raja Balasubramanian, Xerox Webster Research Center
800 Phillips Road, Bldg 128-29E
Webster, NY 14580.
Phone: (716) 265 7838
email: raja@wrc.xerox.com

Co-authors:
Charles A. Bouman, School of Electrical Engineering,
Purdue University, West Lafayette, IN 47907.
Phone: (317) 494-0434
email: bouman@ecn.purdue.edu

Jan P. Allebach, School of Electrical Engineering,
Purdue University, West Lafayette, IN 47907.
Phone: (317) 494-3535
email: allebach@ecn.purdue.edu

ABSTRACT

We propose an efficient vector quantization (VQ) technique which we call sequential scalar quantization (SSQ). The scalar components of the vector are individually quantized in a sequence, with the quantization of each component utilizing conditional information from the quantization of previous components. Unlike conventional independent scalar quantization (ISQ), SSQ has the ability to exploit inter-data correlation. At the same time, since quantization is performed on scalar rather than vector variables, SSQ offers a significant computational advantage over conventional VQ techniques, and is easily amenable to a hardware implementation. In order to analyze the performance of SSQ, we appeal to asymptotic quantization theory where the codebook size is assumed to be large. Closed form expressions are derived for the quantizer mean squared error (MSE). These expressions are used to compare the asymptotic performance of SSQ with other VQ techniques. We also demonstrate the use of asymptotic theory in designing SSQ for a practical application, color image quantization, where the codebook size is typically small. Theoretical and experimental results show that SSQ far outperforms ISQ with respect to MSE, while offering a considerable reduction in computation over conventional VQ at the expense of a moderate increase in MSE.

1. INTRODUCTION

Vector quantization (VQ) has found increasing use in data compression applications such as image and speech coding. The technique is an extension of scalar quantization to the vectorial case, and is motivated by the well known result from Shannon's rate distortion theory that superior performance can always be achieved by coding vectors rather than scalars. As with scalar quantization, the objective in VQ design is to minimize a distortion criterion such as mean squared error (MSE). An iterative error minimization algorithm developed by Lloyd for scalar quantization was extended to the vector case by Linde, Buzo, and Gray [1]. The iterative nature of this algorithm makes it computationally very intensive. A host of other suboptimal but computationally simpler VQ techniques have been reported in the literature either as an alternative to the Linde-Buzo-Gray (LBG) algorithm, or as an initial step that may be refined by

the LBG technique. Excellent reviews of these techniques may be found in [2, 3, 4]. Until recently, the general class of VQ techniques has not received much attention for practical implementation because of the high computational cost, and the lack of codebook structures that are amenable to hardware implementation.

In this paper, we propose a VQ technique, which we call sequential scalar quantization (SSQ). Although similar ideas have been alluded to in the literature [3, 4], to our knowledge, this method has not been theoretically analyzed or seriously pursued in any application. As the name implies, the basic idea behind the technique is to sequentially quantize the scalar components of a vector, rather than to quantize the vector as a whole. The main computational savings arises from the fact that we are quantizing along scalar rather than vector dimensions. At the same time, due to its sequential nature, SSQ possesses the ability to exploit the correlation and statistical dependency between scalar components of a vector. As is the case with any other VQ technique, SSQ attempts to minimize a distortion measure. In order to analyze and optimize the performance of SSQ with respect to this measure, we appeal to asymptotic or high-rate quantization theory, where the number of output quantization levels is assumed to be very large. This theory allows us to derive closed form expressions for the distortion resulting from SSQ as a function of the quantizer design parameters, and to find the optimum parameter values that minimize the distortion. It also proves to be a very useful tool in quantizer design even when the number of output levels is small.

While SSQ may be used in any scenario that is amenable to vector quantization, we have investigated its use in the application of color image quantization, where a high quality color image is to be displayed on a low cost display device with a small palette of colors. Several VQ techniques have been applied to this problem. Braudaway [5] and Gentile *et al.* [6] used the LBG iterative algorithm for palette selection; Orchard and Bouman [7] utilized a tree-structured splitting VQ technique; and Balasubramanian and Allebach [8] reported a merging VQ approach. In [9], we describe in detail an algorithm that employs SSQ for color palette design. The algorithm uses the results of the asymptotic analysis developed in this paper to optimize the palette design with respect to a squared error criterion. We show that with the sequential

technique, the palette design is performed very efficiently, while the resulting structure of the palette allows the mapping between image pixels and palette colors to be performed with no computation. In addition, the resulting image quality is comparable with or superior to that obtained from other color quantization algorithms. In this paper, we focus on a theoretical analysis of SSQ within a general VQ context, and include a brief discussion of its application to color quantization. The reader is referred to [9] for details of this application.

2. VECTOR QUANTIZATION

In this section, we formally define the VQ problem and introduce some notation. We will use lower case letters to denote real variables and vectors, while random variables and vectors will be written with upper case letters. Vectors will be represented by boldface notation. We denote by $p(\ )$ the probability density function of a random variable or vector, and by $P(\ )$ the probability of an event. R^k refers to the k-dimensional space of reals.

Let $\boldsymbol{X}$ be a random vector in R^k with probability density function $p(\boldsymbol{x})$. An N point k-dimensional vector quantizer Q: $\mathsf{R}^k \rightarrow \mathsf{R}^k$ is a function whose domain is the set of all possible values of $\boldsymbol{X}$ and whose range is a set of N vectors $C = \{\boldsymbol{y}_1,...,\boldsymbol{y}_N\}$ called a codebook. Such a quantizer defines a partition $S = \{S_1,...,S_N\}$ of N regions in R^k, where $S_i = \{\boldsymbol{x} \in \mathsf{R}^k : Q(\boldsymbol{x}) = \boldsymbol{y}_i\}$. The quantization consists of two steps: the codebook design, which involves an appropriate selection of the output vectors $\boldsymbol{y}_1,...,\boldsymbol{y}_N$; and the mapping of each input vector to one of the output vectors according to the rule $Q(\boldsymbol{x}) = \boldsymbol{y}_i$ if $\boldsymbol{x} \in S_i$. In practice, the vector mapping consists of an encoder which assigns to each input $\boldsymbol{x}$ a channel symbol, and a decoder which maps each channel symbol to a unique output vector in the codebook. Define a distortion measure $d(\boldsymbol{x},\boldsymbol{y})$, d: $\mathsf{R}^k \times \mathsf{R}^k \rightarrow [0, \infty)$. The quantizer is designed to minimize the expected distortion $D_k = E\{d(\boldsymbol{X},Q(\boldsymbol{X}))\}$ between its input and output. Here, $E\{\ \}$ denotes expected value with respect to the input distribution p. The distortion measure that we will be using is the mean-squared error (MSE),

$$D_k = \frac{1}{k} E\left\{\|\boldsymbol{X} - Q(\boldsymbol{X})\|^2\right\}, \tag{1}$$

where || || denotes Euclidean distance. Later, we will also discuss weighted squared error measures. The two necessary conditions for a quantizer to be optimal with respect to the MSE distortion are that (i) the $\boldsymbol{y}_i$ are chosen to be the centroid of $\boldsymbol{x}$ in S_i, *i.e.* $\boldsymbol{y}_i = E\{\boldsymbol{X} \mid \boldsymbol{x} \in S_i\}$; and (ii) each input $\boldsymbol{x}$ is quantized to one of the $\boldsymbol{y}_i$'s according to a nearest neighbor rule, *i.e.* $S_i = \{\boldsymbol{x} \in R^k : \| \boldsymbol{x} - \boldsymbol{y}_i \|^2 \leq \|\boldsymbol{x} - \boldsymbol{y}_j \|^2, j = 1,...,N\}$ [4]. These two conditions are the basis for the iterative codebook design algorithm that was initially proposed by Lloyd for scalar quantization, and then generalized to the vector case by Linde, Buzo and Gray [1]. The nearest neighbor mapping rule, which involves an exhaustive distance calculation between the input vector and each output vector in the codebook, introduces significant computation in the iterative scheme. Moreover, in general, there is no guarantee that the iterations converge to a global minimum in MSE. (In the 1-dimensional case, the iterations will converge to a global minimum if the distribution $p(\boldsymbol{x})$ is known to satisfy certain properties such as log concavity [4]). Other suboptimal but more efficient VQ techniques such as tree-structured VQ have been proposed [4]. Efficient strategies have also been devised to reduce the time taken for nearest neighbor searches [4]. We will use the term conventional VQ to refer to all methods that quantize a vector as a whole entity.

3. SCALAR QUANTIZATION OF VECTORS

As alluded to above, the primary disadvantage of VQ is its associated complexity, which increases rapidly with the dimensionality of the vector and the codebook size. Another suboptimal but computationally simpler approach to quantize a vector $\boldsymbol{X} = [X_1,...,X_k]^t$ is to quantize each of its individual scalar components X_i, $1 \leq i \leq k$. This may be done either independently or in a sequential fashion.

3.1 Independent Scalar Quantization (ISQ)

This is the conventional method of scalar quantization. A codebook C_i of scalar outputs is designed independently for each scalar component X_i, $1 \leq i \leq k$, according to its marginal distribution $p(x_i)$. The final codebook is a k-fold Cartesian product of the k scalar codebooks, and is therefore known as a product code. A 2-D example of this scheme is shown in Fig. 1a for a rotated uniform distribution. The symbols **x** denote output vectors, which are taken to be the

centroids within each region. In this example, the codebook size is 25. Vector mapping may be accomplished by independently encoding each X_i to a channel symbol through a set of k lookup tables (LUT's). This is depicted in Fig. 1b for $k = 3$. The outputs of the k LUT's are independently decoded to scalar outputs $Y_1,..,Y_k$, which constitute the output vector $\boldsymbol{Y} = [Y_1,...,Y_k]^t = Q(\boldsymbol{X})$. Since the codebook design only involves quantization of scalar variables, and the encoding operation only entails indexing into LUT's, ISQ involves far less computation than conventional VQ. However, with this scheme, many output vectors are wasted in regions where the input has zero probability of occurrence, as is seen in Fig. 1a.

3.2 Sequential Scalar Quantization (SSQ)

With this approach, the first scalar X_1 is quantized to some predetermined number of levels N_1 based on its marginal distribution $p(x_1)$. Each subsequent X_i, $2 \le i \le k$, is then quantized based on a set of conditional distributions of X_i within regions formed from the quantization of the scalars $X_1,...,X_{i-1}$. A 2-D example of SSQ is shown in Fig. 2a for the same uniform distribution. In this example, first X_1 is quantized to N_1=5 levels. This results in a partition of intervals B_{1j}, $1 \le j \le 5$, in R, or columns B_{2j} in R^2. Next, we quantize X_2 but confine the quantization to the columns formed from the quantization of X_1. This results in a 2-D quantizer with N_2=18 output vectors in R^2. The encoding of input vectors may be performed through a sequential or multistage LUT, as shown in Fig. 2b for 3-D vectors. The input to the first LUT is X_1. The output symbol b_{i-1} of the (i-1)th LUT, $2 \le i \le k$, is then fed to the input of the i-th LUT along with the i-th scalar component X_i. Finally, the output symbol b_k of the last encoder is decoded to one of the output vectors in the codebook C. As is the case with ISQ, the codebook design only involves scalar quantization, while the encoding operation entails no computation, and is easily amenable to a hardware implementation. Hence, there is a significant computational advantage to be gained by using SSQ rather than conventional VQ methods. In addition, SSQ places its output vectors only within the region of support of the input distribution, thus requiring fewer output codevectors than ISQ to achieve the same quality level. In the next section, we compare qualitatively the performance of ISQ and SSQ.

3.3 Comparison of SSQ and ISQ

Makhoul *et al*. [3] provide an excellent qualitative discussion of the advantages of using conventional VQ over ISQ in terms of four properties of the input data: linear dependency; nonlinear dependency; shape of the input distribution; and vector dimensionality. They argue that conventional ISQ can only make use of two of these properties, namely linear dependency and distribution shape, whereas VQ can exploit all four properties. Lookabaugh and Gray [10] quantified the VQ advantage in terms of the four properties. Here, we qualitatively compare ISQ and SSQ in terms of how well they exploit each of these four properties. Some of the examples of input distributions are taken from [3].

(a) Linear Dependency: This refers to the statistical correlation between the vector components. The 2-D distribution in Fig. 1a represents data that are correlated. As was observed previously, ISQ places some vectors in regions of zero probability because it uses only marginal statistics and cannot take into account the inter-data correlation. On the other hand, since SSQ uses a combination of marginal and conditional statistics, or effectively the joint statistics, this technique takes inter-data correlation into account and place all its output vectors within the support of the distribution. Note that if we rotate the uniform distribution to align with the two axes, then the data are both uncorrelated and independent. In this case, ISQ and SSQ offer equivalent performance.

(b) Nonlinear Dependency: This is the residual dependency that remains after the correlation between components has been removed. Consider the distribution in Fig. 3, which has a constant value in the shaded area. It is easily shown that X_1 and X_2 are uncorrelated but not independent, *i.e.*, there exists a nonlinear dependency between them. With the ISQ scheme of Fig. 3a, each scalar is quantized according to its marginal distribution. Since the 2-D codebook is a Cartesian product of the two 1-D codebooks, some output vectors will fall inside the shaded rectangular annulus where the input distribution is zero. The SSQ scheme shown in Fig 3b will, however, place all its vectors in the shaded area, thus again offering performance superior to that of ISQ.

(c) Shape of the Input Distribution: This property refers to the ability of the quantizer to place its output vectors according to the shape of the input distribution in multidimensional space. Ideally, we would expect the output vectors to be more densely spaced where the distribution takes on larger values. Consider two jointly Gaussian random variables with a correlation coefficient of zero. These random variables are uncorrelated and independent. Their distribution is shown schematically in Fig. 4a with ISQ and in Fig 4b with SSQ. Notice that ISQ results in densely spaced output vectors in the regions A, B, C, and D, even though the probability density takes on relatively small values in these areas. SSQ suffers from the same drawback only in regions A and B and not around C and D. This subtle difference in how the quantizers space their codevectors yields an interesting and surprising result: namely that SSQ can outperform ISQ even when the input data is independent! In Sec. 5.3, we will formally show this result for the Gaussian distribution in the asymptotic case where the number of quantization levels becomes large.

(d) Vector Dimensionality: This property refers to the ability of VQ to pack arbitrarily shaped quantization regions in multidimensional space. The nearest neighbor condition for optimality can often only be achieved with non-rectangular polytopal quantization regions. However, ISQ and SSQ are confined to producing cells that are rectangular polytopes; therefore, they are both inferior to conventional VQ in this respect.

In summary, SSQ can be a powerful quantization technique, because while it affords many of the performance advantages of conventional VQ, it also enjoys the computational simplicity of scalar quantization. In the sections below, we will quantify the performance of SSQ and compare it to ISQ and VQ.

4. ASYMPTOTIC QUANTIZATION THEORY

As pointed out earlier, the minimization of MSE is, in general, a nonlinear iterative problem. Thus, the MSE cannot be written in closed form as a function of the number of quantization levels, except for the most trivial uniform distribution. However, if the number of

output quantization levels N is allowed to become asymptotically large, then it is possible to arrive at approximate closed form error expressions [4]. The idea behind asymptotic theory is that the number of output quantization levels is assumed to be large enough (or equivalently, the quantization cells small enough) and the probability distribution of the source is assumed to be locally smooth enough so that within any quantization cell, this distribution is approximately uniform. The MSE's within each quantization region, which are known in closed form for a uniform distribution, are appropriately summed to yield an approximation for the overall MSE. The asymptotic analysis not only provides intuition about the behavior of the quantizer, but also can serve as a valuable guide in the design of the quantizer even for small N. In the section below, we briefly outline basic asymptotic results that have been developed for scalar and vector quantizers. The reader is referred to [4, 11, 12] for details.

4.1 Asymptotic Scalar Quantization

Let X be a random variable with distribution $p(x)$. Consider a 1-D quantizer $Q: \mathsf{R} \rightarrow \mathsf{R}$ with N output points. Let $N(x)dx$ be the number of quantization levels that lie in the interval $[x, x+dx]$. We define the quantizer density function $\lambda(x)$ as

$$\lambda(x) = \lim_{N\rightarrow\infty} \frac{N(x)}{N} \, . \tag{2}$$

Thus for sufficiently large N, the quantity $N\lambda(x)dx$ is approximately the number of quantization levels in the interval $[x, x+dx]$. It follows by definition that integrating $\lambda(x)$ over its entire domain will result in 1. It may be shown [4] that the MSE of the quantizer may be approximated by an integral

$$D = E\{ [X - Y]^2 \} \approx \frac{1}{12N^2} \int \frac{p(x)}{\lambda(x)^2} dx \, . \tag{3}$$

where $Y = Q(X)$ is the quantizer output. Equation (3) is known as Bennett's distortion integral. Using Hölder's inequality or the calculus of variations, it may be shown that the function $\lambda(\cdot)$ that minimizes D is given by

$$\lambda(x) = \frac{p(x)^{1/3}}{\int p(x)^{1/3} dx} \, , \tag{4}$$

and the resulting minimum distortion is given by

$$D \approx \frac{1}{12N^2} \|p(x)\|_{1/3} \ , \tag{5}$$

where we have used the notation

$$\|p(x)\|_m \equiv \left[\int p(x)^m dx \right]^{1/m} \ . \tag{6}$$

If we wish to find the conditional MSE $E\{[X - Y]^2 \mid A\}$ given the event A, we simply replace the marginal distribution $p(x)$ in (3) - (6) with the conditional distribution $p(x \mid A)$.

4.2 Asymptotic Vector Quantization

Extending the analysis to k-dimensional vectors, let $\boldsymbol{X}$ be a random vector in R^k with distribution $p(\boldsymbol{x})$; p: $\mathsf{R}^k \rightarrow \mathsf{R}$, and let Q be an N point quantizer; Q: $\mathsf{R}^k \rightarrow \mathsf{R}^k$. We may define the quantizer density function $\lambda(\boldsymbol{x})$; λ: $\mathsf{R}^k \rightarrow \mathsf{R}$, in a manner analogous to (2), so that $N\lambda(\boldsymbol{x})d\boldsymbol{x}$ is the number of quantization levels in an incremental volume $d\boldsymbol{x}$ around $\boldsymbol{x}$. Na and Neuhoff [12] extend Bennett's distortion integral (3) to the vector case yielding

$$D_k \approx \frac{1}{N^{2/k}} \int \frac{m_k(\boldsymbol{x})}{\lambda(\boldsymbol{x})^{2/k}} p(\boldsymbol{x}) d\boldsymbol{x} \ , \tag{7}$$

where D_k is the MSE as defined in (1), and $m_k(\boldsymbol{x})$ is a function characterized by the shapes of the cells in the vicinity of $\boldsymbol{x}$, known as the inertial profile function. Gersho [11] conjectures that for large N, the optimal quantizer is one whose quantization cells are tessellations of a congruent polytope. This result implies that $m_k(\boldsymbol{x}) = M_k$ is a constant with respect to $\boldsymbol{x}$ and may be taken outside the integral in (7). As with the scalar case, (7) can be minimized with respect to $\lambda(\boldsymbol{x})$, yielding

$$\lambda(\boldsymbol{x}) = \frac{p(\boldsymbol{x})^{k/(k+2)}}{\int p(\boldsymbol{x})^{k/(k+2)} d\boldsymbol{x}} \ , \tag{8a}$$

and

$$D_k \approx \frac{M_k}{N^{2/k}} \|p(\boldsymbol{x})\|_{k/(k+2)} , \tag{8b}$$

where $\|p(\boldsymbol{x})\|_m$ is defined in a manner analogous to (6). Equation (8b) serves as a lower bound on MSE performance for an N point VQ in k-dimensional space. It is easily shown that the inertial profile M_1 for an interval in R is equal to 1/12, so that (8b) reduces to (5) for the case $k = 1$.

5. ASYMPTOTIC ANALYSIS OF SSQ

We now analyze the asymptotic behavior of SSQ in a manner similar to that used to derive the results in Sec. 4. Since SSQ is fundamentally different from either ISQ or conventional VQ, we cannot simply use the results of Sec. 4; rather, we will have to rederive the theory for the sequential structure. The basic results will provide us with the tools necessary for such a derivation. We begin in Sec. 5.1 by precisely defining the SSQ structure and making explicit the variables that we are free to choose in the design of the quantizer.

5.1 The SSQ Design Problem

Given an input random vector $\boldsymbol{X} = [X_1,...,X_k]^t$ in R^k with distribution $p(\boldsymbol{x})$, we wish to design a codebook of N output vectors $C = \{\boldsymbol{y}_1,...,\boldsymbol{y}_N\}$ by sequentially quantizing the scalar components $X_1,...,X_k$. To simplify the notation, we will assume that the components are quantized in the order $X_1, X_2,..., X_k$. The quantization of a scalar coordinate to a fixed number of levels requires some rule Φ to place the decision and reconstruction levels along that coordinate. This rule will be treated as a design variable.

We begin by designing an N_1 level quantizer for X_1 using the marginal distribution $p(x_1)$. N_1 is a design variable which we shall momentarily assume to be fixed. This quantization creates a partition $P'_1 = \{B'_{11},..., B'_{1N_1}\}$ of intervals in R, or a partition $P_2 = \{B_{21},..., B_{2N_1}\}$ of cylinder sets in R^2. We denote by B'_{1j} the j-th quantization region in R, and by B_{2j} the cylinder set in R^2 given by $B_{2j} = B'_{1j} \times \mathsf{R}$. Next, we design a quantizer for X_2, which creates a refinement $P'_2 = \{B'_{21},..., B'_{2N_2}\}$ of P_2 containing N_2 quantization regions. Here N_2 is again a design variable. The refinement is obtained by quantizing X_2 within each B_{2j} to some predetermined number of levels n_{2j} based on its conditional distribution $p(x_2 | B_{2j})$. Referring to Fig 2a, we have $n_{21} = 3$, $n_{22} = 4$, *etc*. Note that the n_{2j}'s must sum to N_2.

We may generalize this discussion to quantization of the i-th scalar component X_i, $2 \le i \le k$. The result of quantization along the dimension x_{i-1} is a partition P'_{i-1} of N_{i-1} regions in R^{i-1}, or a partition P_i of N_{i-1} cylinder sets in R^i. The i-th quantizer produces a refinement P'_i of P_i with N_i regions, where $N_i \ge N_{i-1}$. This is accomplished by quantizing along x_i within each of the

cylinder sets B_{ij}, $1 \le j \le N_{i-1}$, to n_{ij} levels according to the conditional distributions $p(x_i \mid B_{ij})$. The N_i's are design variables that must satisfy the constraint $1 \le N_1 \le N_2 \le ... \le N_k = N$, where N is the desired number of codewords in R^k. Furthermore, for each i, $2 \le i \le k$, the n_{ij}'s are also design variables that must satisfy the constraint

$$\sum_{j=1}^{N_{i-1}} n_{ij} = N_i \quad . \tag{9}$$

Our objective is to pick the design variables N_i, n_{ij}, and the quantization rule Φ along each coordinate x_i to minimize the overall MSE D_k given in (1) for a fixed codebook size $N_k = N$. Since the MSE is a separable distortion measure, we may write

$$D_k = \frac{1}{k}\sum_{i=1}^{k} d_i \quad , \tag{10}$$

where $d_i = E\{[X_i - Y_i]^2\}$ is the MSE along x_i resulting from the quantization $Y_i = Q(X_i)$. In the following section, we will obtain approximate analytical expressions for the d_i's in the asymptotic case.

5.2 Asymptotic Optimization of SSQ

We now derive an asymptotic theory for SSQ that provides us with closed form expressions for the MSE of the quantizer in terms of the aforementioned design variables. This allows the optimization of the quantizer with respect to these variables. For the subsequent analysis, we make the following assumptions: 1) the number of quantization levels N_i, $1 \le i \le k$, and the relative allocations n_{ij} are large; 2) the probability distribution $p(\boldsymbol{x})$ of the source is relatively smooth; 3) the rule Φ for placing decision and reconstruction levels along a scalar dimension x is completely specified by the quantizer density function $\lambda(x)$; 4) quantization along x_i will attempt to minimize only the MSE d_i along that dimension, and in particular will not affect the MSE's $d_1,...,d_{i-1}$ due to previous quantizations; 5) the probability of quantizer overload is assumed to be negligible [4].

We will derive in detail the error expressions for the case where $\boldsymbol{X}$ is a 3-D vector. The results generalize to k dimensions in a straightforward manner. We begin by quantizing X_1 to N_1

levels using the marginal distribution $p(x_1)$. We obtain from (3) the asymptotic approximation to the MSE along x_1

$$d_1 \approx \frac{1}{12 N_1^2} \int \frac{p(x_1)}{\lambda(x_1)^2} dx_1 \quad . \tag{11}$$

The optimal quantizer spacing $\lambda(x_1)$ that minimizes d_1 is given by (4) with $p(x)$ replaced by $p(x_1)$, and from (5), the resulting minimum d_1 is

$$d_1 \approx \frac{1}{12 N_1^2} \left\| p(x_1) \right\|_{1/3} \quad . \tag{12}$$

Next, we derive an expression for d_2, the MSE along x_2, by conditioning on the cylinder sets B_{2j} formed from quantizing X_1

$$\begin{aligned} d_2 &= E\{ [X_2 - Y_2]^2 \} , \\ &= \sum_{j=1}^{N_1} E\{ [X_2 - Y_2]^2 \mid B_{2j} \} P(B_{2j}), \end{aligned} \tag{13}$$

where for brevity, we have used $P(B_{2j})$ to denote $P([X_1, X_2]^t \in B_{2j})$. Note that within each B_{2j}, the random variable X_2 has conditional distribution $p(x_2 \mid B_{2j})$ and is quantized to n_{2j} levels. We may use (3) to approximate the conditional distortion within each B_{2j} for large n_{2j}

$$E\{ [X_2 - Y_2]^2 \mid B_{2j} \} \approx \frac{1}{12 n_{2j}^2} \int \frac{p(x_2 \mid B_{2j})}{\lambda(x_2 \mid B_{2j})^2} dx_2 , \tag{14}$$

where $\lambda(x_2 \mid B_{2j})$ denotes the relative density of quantization levels along x_2 within B_{2j}. We know from (4) that the optimal $\lambda(x_2 \mid B_{2j})$ that minimizes (14) is

$$\lambda(x_2 \mid B_{2j}) = \frac{p(x_2 \mid B_{2j})^{1/3}}{\int p(x_2 \mid B_{2j})^{1/3} dx_2} \quad . \tag{15}$$

Substituting (15) into (14), we get a minimum error expression similar to (5), which can then be substituted into (13) to yield

$$d_2 \approx \frac{1}{12} \sum_{j=1}^{N_1} \frac{1}{n_{2j}^2} \left\| p(x_2 \mid B_{2j}) \right\|_{1/3} P(B_{2j}) . \tag{16}$$

Now we are left with the relative allocations n_{2j}, $1 \le j \le N_1$ which may be chosen to minimize (16) subject to the constraint (9) that they sum to N_2. This is a straightforward constrained minimization problem that is easily solved by a Lagrangian method to yield the optimal allocations

$$n_{2j} = N_2 \frac{\left[\left\| p(x_2 \mid B_{2j}) \right\|_{1/3} P(B_{2j})\right]^{1/3}}{\sum_{l=1}^{N_1}\left[\left\| p(x_2 \mid B_{2l}) \right\|_{1/3} P(B_{2l})\right]^{1/3}} \quad . \tag{17}$$

The resulting minimum d_2 is given by

$$d_2 \approx \frac{1}{12 N_2^{\,2}} \left\{ \sum_{j=1}^{N_1} \left[\left\| p(x_2 \mid B_{2j}) \right\|_{1/3} P(B_{2j}) \right]^{1/3} \right\}^3 \quad . \tag{18}$$

For large N_1, we use the smoothness restriction on $p(\mathbf{x})$ to rewrite (18) as a Riemann sum and approximate it by an integral (see Lemma 1 of Appendix)

$$d_2 \approx \frac{N_1^{\,2}}{12 N_2^{\,2}} \left\{ \int \left[\left\| p(x_2 \mid x_1) \right\|_{1/3} p(x_1) \lambda^2(x_1) \right]^{1/3} dx_1 \right\}^3 , \tag{19a}$$

$$= \frac{N_1^{\,2}}{12 N_2^{\,2}} \left(\frac{\left\| \left\| p(x_2 \mid x_1) \right\|_{1/3} p(x_1)^{5/3} \right\|_{1/3}}{\left\{ \int p(x_1)^{1/3} dx_1 \right\}^2} \right) , \tag{19b}$$

where (19b) is obtained by substituting for $\lambda(x_1)$ from (4) into (19a). Looking at (12) and (19b), we see that the asymptotic assumptions allow us to express the MSE along each coordinate as a product of a term that depends on the number of quantization levels and a constant term that depends on the input statistics. This factorization is an important result as will be seen shortly.

We can carry through a similar analysis to obtain the MSE d_3 along x_3. This time we condition on the cylinder sets B_{3j}, $1 \le j \le N_2$ formed from quantization of X_2

$$d_3 = E\{ [X_3 - Y_3]^2 \} ,$$

$$= \sum_{j=1}^{N_2} E\{ [X_3 - Y_3]^2 \mid B_{3j} \} P(B_{3j}) ,$$

$$\approx \frac{1}{12} \sum_{j=1}^{N_2} \frac{1}{n_{3j}^{\,2}} \left\| p(x_3 \mid B_{3j}) \right\|_{1/3} P(B_{3j}) \quad , \tag{20a}$$

$$\approx \frac{1}{12 N_3^{\,2}} \left\{ \sum_{j=1}^{N_2} \left[\left\| p(x_3 \mid B_{3j}) \right\|_{1/3} P(B_{3j}) \right]^{1/3} \right\}^3 , \tag{20b}$$

where (20b) is obtained by minimizing (20a) with respect to the n_{3j} subject to the constraint (9) that they sum to N_3. Assuming that the joint distribution $p(x_1,x_2)$ is locally smooth, we may once again approximate (20b) by an integral

$$d_3 \approx \frac{N_2^{\,2}}{12 N_3^{\,2}} \left\{ \int\int \left[\left\| p(x_3 \mid x_1, x_2) \right\|_{1/3} p(x_1, x_2) \lambda^2(x_1, x_2) \right]^{1/3} dx_2 dx_1 \right\}^3 , \tag{21}$$

where $\lambda(x_1, x_2)$ is the 2-dimensional quantizer density function that defines the relative spacing of output vectors in 2-D space. We may derive an approximate expression for $\lambda(x_1, x_2)$ in terms of our known 1-D marginal and conditional quantizer density functions $\lambda(x_1)$, $\lambda(x_2|x_1)$ (see Lemma 2 of Appendix), and substitute this into (21) to yield

$$d_3 \approx \frac{N_2^{\,2}}{12\,N_3^{\,2}} \left(\frac{\left\| \left\| p(x_3 \mid x_1, x_2) \right\|_{1/3} p(x_1, x_2)^{5/3}\, p(x_1)^{4/9} \right\|_{1/3}}{\left\{ \int\int p(x_1, x_2)^{1/3}\, p(x_1)^{2/9}\, dx_1 dx_2 \right\}^2} \right) . \tag{22}$$

Once again, we have factored the error as the product of a constant and a term that depends only on the number of quantization levels. Substituting (12), (19b), and (22) into (10), we see that the overall 3-D MSE D_3 is of the form

$$D_3 \approx \frac{1}{36} \left(\frac{1}{N_1^{\,2}} \alpha_1 + \frac{N_1^{\,2}}{N_2^{\,2}} \alpha_2 + \frac{N_2^{\,2}}{N_3^{\,2}} \alpha_3 \right) , \tag{23}$$

where α_i are the constant terms associated with d_i, $1 \le i \le 3$. Recall that $N_3 = N$ is the known codebook size. The expression (23) may be successively minimized with respect to N_1 and N_2 to yield the optimal allocations

$$N_1 = N_2^{\,1/2} \left(\frac{\alpha_1}{\alpha_2} \right)^{1/4} \quad ; \quad N_2 = N_3^{\,2/3} \left(\frac{\sqrt{\alpha_1 \alpha_2}}{\alpha_3} \right)^{1/3} . \tag{24}$$

The resulting distortion is

$$D_3 \approx A_3 \frac{\left(\alpha_1 \alpha_2 \alpha_3 \right)^{1/3}}{N_3^{\,2/3}} , \tag{25}$$

where $A_3 = (2^{1/3} + (1/2)^{2/3})/18^{2/3} \approx 0.275$.

Generalizing to k-dimensional vectors, it may be shown using arguments similar to the one presented above, that the overall MSE D_k for an N point SSQ is given by

$$D_k \approx \frac{1}{12k} \sum_{i=1}^{k} \frac{N_{i-1}^{\,2}}{N_i^{\,2}} \alpha_i , \tag{26}$$

where $N_0 \equiv 1$ and $N_k = \mathrm{N}$. The term α_i is a function of the conditional distribution $p(x_i \mid x_1,\ldots,x_{i-1})$ and all lower order distributions. We may optimize D_k with respect to the N_i's and obtain expressions similar to (24), where each N_i is defined successively in terms of N_{i+1}. The resulting overall MSE is of the form

$$D_k \approx \frac{A_k}{N^{2/k}} \prod_{i=1}^{k} \alpha_i^{1/k} \quad , \tag{27}$$

where A_k is a dimension dependent number.

Several points are worth noting here. First of all, (27) does not, in general, give the globally minimum MSE over all sequential quantizers. This is due to our simplifying assumption that the quantization rule λ along each coordinate is chosen to minimize the MSE only along that coordinate, *i.e.* the optimization is performed in a greedy rather than joint fashion. Secondly, we have only considered the case where the scalar components are quantized in the order $X_1, X_2, ..., X_k$. In general, since the terms α_i depend on conditional densities, they will be different for different quantization orderings. Hence, in theory, we must evaluate (27) for all $k!$ possible orderings, and pick the one that yields the minimum D_k. In practice, the specific application may impose a natural order of quantization (as we will see later). Failing this, some simple criterion such as variance along each coordinate may be used to determine an order that yields a possibly suboptimal but acceptable solution.

Thirdly, comparing (27) with (8b), the lower bound on the MSE that is achieved by the optimal vector quantizer, we see that asymptotically, the MSE for both SSQ and optimal VQ in k dimensions decreases as $1/N^{2/k}$. This implies that a fixed percentage loss in performance, independent of N, will be incurred by using SSQ instead of the optimal VQ method. The loss will depend on the relative values of the constants in (8b) and (27), which in turn depend on the joint, conditional, and marginal statistics of the source $\boldsymbol{X}$. The suboptimal performance of SSQ is explained by the rectangular cell shapes resulting from this technique, and the fact that SSQ may not achieve the optimal density of quantization points in k-dimensional space given by (8a). Finally, we remark that the analysis developed above can be generalized to any r-th power distortion measure for $r \geq 1$, the only restriction being that the measure be separable along the scalar coordinates.

5.3 Analysis of SSQ for a Gaussian Source Distribution

We will use the analysis above to evaluate and compare the asymptotic performance of SSQ, ISQ, and VQ for the case where the input is a 2-D Gaussian random vector. Let $X = [X_1,$

$X_2]^t$ be characterized by the jointly Gaussian density $\mathbf{N}[\mu_1, \mu_2, \sigma_1, \sigma_2, \rho]$. We have used the usual convention that μ_1 and μ_2 are the means along x_1 and x_2; σ_1 and σ_2 are the standard deviations, and ρ is the correlation coefficient. Since μ_1 and μ_2 do not affect the analysis in any way, we will assume them to be zero. We note first of all that if $p(x)$ is a 1-D Gaussian density with variance σ^2, then $p(x)^m$ when properly normalized is also Gaussian with variance σ^2/m. Secondly, if $p(x_1,x_2)$ is jointly Gaussian, then the conditional density $p(x_2|x_1)$ is also Gaussian with mean $\rho\sigma_2 x_1/\sigma_1$ and variance $\sigma_2{}^2(1-\rho^2)$. These two facts allow us to easily evaluate the integrals in (12) and (19b) and obtain the constants α_1 and α_2. Noting that $N_2 = N$ is fixed, we may find the optimal N_1 from (24), substitute this into (12) and (19b), and average the two 1-D MSE's to yield a surprisingly simple expression for the 2-D MSE

$$D_2^{SSQ} \approx \frac{0.777\pi}{N}\sigma_1\sigma_2\sqrt{1-\rho^2}\,. \tag{28}$$

Equation (28) tells us that the error is inversely related to the correlation coefficient ρ. This makes sense because the sequential scheme, which uses the conditional distribution $p(x_2|x_1)$ to quantize X_2, will offer improved performance when X_1 provides more information about X_2, *i.e.* when ρ is larger. Note also that the performance of SSQ is independent of the order in which the scalar components are quantized.

We may compare SSQ with the lower bound given in (8b) for our example distribution. We will use $M_2 = 0.08$, as this corresponds to a hexagonal packing of cells, which is the optimal scheme in 2-D [9]. This yields the lower bound

$$D_2^{OPT} \approx \frac{0.642\pi}{N}\sigma_1\sigma_2\sqrt{1-\rho^2}\,. \tag{29}$$

Since SSQ is restricted to rectangular cell shapes, we would obtain a tighter lower bound on its performance if we let $M_2 = 1/12$ in (8b), which corresponds to VQ with square cell shapes. This results in the fraction 0.642 in (29) being replaced by 0.667. Hence, there is a 16.5% loss in using SSQ over a VQ scheme with square cells. Finally, if we assume that X_1 and X_2 are quantized independently according to their marginal distributions, with optimal bit allocations among the two coordinates, then we obtain the following asymptotic error [4]

$$D_2^{IND} \approx \frac{0.886\pi}{N}\sigma_1\sigma_2. \tag{30}$$

As expected, the performance of ISQ does not depend on the inter-data correlation coefficient ρ. As ρ increases, we see the increased advantage of using either SSQ or VQ rather than ISQ. If we let ρ = 0 (*i.e.*, X_1 and X_2 are both uncorrelated and independent), and compare (28) and (30), we arrive at the interesting result that SSQ outperforms ISQ even when the components are independent! This result was alluded to earlier, and is due to the ability of SSQ to exploit the shape of the input distribution.

5.4 Weighted Distortion Measures

Although the mean squared error is a mathematically tractable metric, it may not be an appropriate measure of distortion for a given application. This is certainly true in image processing problems, where MSE often does not correlate well with visually perceived error. Variations of the squared error metric have been sought [3] to better reflect the application dependent distortion criterion. An example of this is the weighted squared error defined as

$$D_k^W = \frac{1}{k}E\left\{[\boldsymbol{X}-\boldsymbol{Y}]^t W[\boldsymbol{X}-\boldsymbol{Y}]\right\} \ , \tag{31}$$

where W is a $k \times k$ positive definite weighting matrix. In general, (31) is not a separable distortion measure, and therefore does not readily lend itself to the foregoing analysis. However, without loss of generality, we may assume that W is symmetric, in which case we may factor it into the form $W = P^t W' P$, where W' is a diagonal matrix and P is a $k \times k$ orthogonal matrix [3]. Letting $\boldsymbol{X}' = P^t \boldsymbol{X}$, and $\boldsymbol{Y}' = P^t \boldsymbol{Y}$, it is easily seen that D_k^W is given by

$$D_k^W = \frac{1}{k}\sum_{i=1}^{k} w_{ii} d_i \ , \tag{32}$$

where w_{ii}' is the i-th diagonal element of W', and $d_i' = E\{[X_i' - Y_i']^2\}$. Hence, the distortion measure is made separable by an orthogonal transformation of the data; and the foregoing analysis is valid with α_i being replaced by $\alpha_i' = w_{ii}' \alpha_i$ in Equations (23) - (27).

5.5 Entropy Constrained SSQ

So far we have looked at the problem of minimizing the distortion while keeping the codebook size N fixed. If VQ is the only compression scheme in a given application, then the

rate of the coder is given by $R = \log_2(N/k)$ bits per source letter. In coding applications, however, the output of VQ is often additionally compressed with a lossless coder. From Shannon's noiseless source coding theorem, we know that the rate of a lossless coding scheme is bounded from below by the entropy of the input to the coder, which in our case is the output of the VQ. Hence, it would be in our interest to design the VQ to minimize the distortion while constraining the entropy of the output, rather than the codebook size.

It has been shown [11] that the minimum entropy VQ scheme for asymptotically large N is one that achieves a uniform distribution of output points, *i.e.* $\lambda(\boldsymbol{x})$ is a constant in k-dimensional space. SSQ achieves this uniform distribution in the trivial case where all marginal and conditional quantizer density functions are identically uniform. This is equivalent to quantizing each component independently with a uniform quantizer. Hence, SSQ offers no performance advantage over ISQ if we wish to perform entropy constrained quantization. On the other hand, VQ has an advantage over both ISQ and SSQ, because for the same uniform distribution $\lambda(\boldsymbol{x})$ of output vectors, VQ with its arbitrary cell shapes can pack its cells in a more efficient way into a finite volume in k-dimensional space than is possible with either SSQ or ISQ.

6. APPLICATION TO COLOR IMAGE QUANTIZATION

We will briefly describe how the analysis developed above may be used in a practical application, where the assumptions that the number of quantization levels is large and the input distribution is smooth may no longer be valid. The reader is referred to [9] for details. The problem we address is that of quantizing a color image to a small palette of colors. Such quantization is often necessitated by hardware limitations with low cost display and printing devices. We will focus on the display application, where the user is often allowed to choose a palette of (usually 256) colors from a much larger set of $2^{24} \approx 16$ million colors.

In the present context, the input to the quantizer is a 3-D RGB color vector $\boldsymbol{X}$, and the output codebook is the desired palette. We wish to design the quantizer to minimize the MSE for a fixed palette size N. In order to perform the quantization in a visually meaningful space, we transform the image from RGB to YC_rC_b luminance-chrominance coordinates. A color

histogram of the image is generated, which serves as the input probability distribution $p(\mathbf{x})$, and quantization is performed first on the chrominance components, followed by luminance. This ordering is motivated by the intuitively appealing idea of first assigning hues to the various objects in the image and then providing fine luminance shading to each hue.

In general, we have noticed that the functional form in (23) is a fairly good model for the experimental MSE with color image data. However, when the number of quantization vectors becomes small, the constant terms α_i are not well modeled by the integral expressions given in (12), (19b), and (22). To overcome this problem, we perform a preliminary quantization using some arbitrary N_1, N_2; measure the experimental MSE's d_1, d_2, d_3 along the 3 coordinates; and obtain empirical estimates of α_i by equating each d_i, $1 \le i \le 3$ to the corresponding term in (23) [9]. Substituting these estimates into (24), we obtain our optimal N_1 and N_2, which are then used for the final quantization.

In order to quantize X_1 to N_1 levels, we first compute $\lambda(x_1)$ from (4). We then partition the x_1 axis into N_1 intervals such that the area under $\lambda(x_1)$ within each interval is equal to $1/N_1$. It may be shown that with such a scheme, the fraction of quantization levels in an interval $[x,x+dx]$ will approach $\lambda(x)dx$ as N_1 becomes large [9]. The centroid of the data within each interval is then chosen to be the output level for that interval. The same procedure is used to quantize along the subsequent scalar dimensions x_i, $i = 2,3$ using the conditional quantizer density functions $\lambda(x_i|B_{ij})$. This quantization rule works well when the marginal and conditional distributions are relatively smooth. However, it can result in a non-optimal positioning of decision and output levels when there are severe discontinuities in the 1-D histograms. We have developed techniques to detect and correct for such cases [9].

A common artifact that appears in color quantization is false contouring, where smooth color transitions in the original image are represented by a small number of palette colors. The contouring tends to be more visually objectionable along the luminance coordinate than along chrominance. To alleviate this problem, we use a simple weighted MSE measure, where the MSE along luminance is weighted K times as much as that along chrominance, for some $K>1$.

Finally, the mapping between input image pixels and output palette colors is performed through the sequential LUT shown in Fig. 2b. In practice, the LUT structure is easily implemented in hardware, thus enabling the pixel mapping to be performed in real time.

7. EXPERIMENTAL RESULTS

The results presented in this section are intended to verify the theoretical analysis developed in Sec. 5, and to assess the performance of SSQ on simulated and experimental data.

7.1 Verification of SSQ Analysis

We performed quantization using SSQ on 2-D simulated Gaussian data and compared the resulting experimental MSE with the theoretical expression given in (28). Figure 5 shows a plot of the percentage difference between experimental and theoretical MSE as a function of the number of quantization levels N for two examples of jointly Gaussian distributions. We see that in both cases, as N becomes large, the experimental error converges to the formula (28), thus confirming the validity of the analysis, at least in 2 dimensions.

7.2 Scalar Quantization According to $\lambda(\mathbf{x})$

In Sec. 6, we described an asymptotically optimal quantization rule to partition a scalar dimension into N intervals. Namely, the decision levels are positioned so that the area under the function $\lambda(x)$ given in (4) is equal within each interval. We compared this method, which we call the lambda technique, with the scalar version of the binary splitting (BSP) algorithm [7] and the Lloyd-Max (LM) quantizer [4] for a simulated Gaussian source distribution. Since the Gaussian density is log concave, the LM algorithm yields a globally minimum solution and serves as a lower bound on quantizer performance. The lambda technique was used as a starting point for the LM iterations. Figure 6 compares the performance of the three quantizers for various N. In order to obtain a meaningful comparison, we normalized the MSE of each algorithm by that of the LM algorithm. For small N, the MSE due to the lambda technique is only about 5 % higher than the minimum MSE. As N increases, the lambda technique converges to the minimum MSE solution, verifying that this technique is indeed asymptotically optimal. The BSP algorithm results in a consistently higher relative MSE, suggesting that at least for 1-D Gaussian data, this

is not an optimal strategy even in the asymptotic sense. We have also compared the lambda technique and BSP as scalar quantization strategies in the SSQ algorithm applied to 3-D color image data, and have observed that the lambda technique results in superior image quality.

7.3 Performance of SSQ on Experimental Data

We now present some results on 3-D color image data that lend support to the qualitative comparison of ISQ, SSQ, and VQ offered in Sec. 3.3. The data was derived from a 512×512 color image, a monochrome version of which is shown in Fig. 9. ISQ was designed to quantize each of the three color components independently according to its marginal distribution using the lambda technique. The relative number of quantization levels along each coordinate was chosen in proportion to the variance along that coordinate [3]. SSQ was implemented as described in Sec. 6; and a fast BSP algorithm [13] was chosen as the conventional VQ technique. Once again, we wished to evaluate these algorithms relative to some lower bound on MSE performance. Since an attempt to find the globally minimum MSE quantizer would be impractical, we used the LBG algorithm [1] to arrive at a local minimum in MSE. We ran the LBG algorithm with ISQ, SSQ, and BSP as starting points, and chose the minimum of the three resulting MSE's as a lower bound. Figure 7 shows a plot of the MSE's of ISQ, SSQ, and BSP normalized by the minimum MSE from LBG for various N. As anticipated, the performance of SSQ is far superior to that of ISQ, while BSP offers a moderate improvement over SSQ. We deduce that the superior performance of SSQ over ISQ is a result of the exploitation by SSQ of linear and nonlinear dependencies in the data; while the improved performance of BSP over SSQ is due mainly to the fact that the former is not constrained to cells that are rectangular parallelepipeds.

Figure 8 shows the execution times on a SUN SPARCstation 2 of ISQ, SSQ, and BSP on the same image data set for various N. We have broken down the execution time of each algorithm into that required for codebook design and that needed to map input vectors (or in this case pixels) to codebook vectors. The latter is often the quantity of greater interest in practical applications. In terms of codebook design, SSQ involves more computation than ISQ, but is considerably faster than BSP. Moreover, as is characteristic of conventional VQ techniques, the

required time for BSP increases with N, whereas this is not the case with SSQ and ISQ. In terms of the mapping step, both SSQ and ISQ use LUT's, and thus require no computation. In contrast, the BSP technique uses a binary tree to perform mapping [7], hence requiring a considerable amount of computation that increases with the codebook size. These results are somewhat indicative of the asymptotic complexities of the three algorithms. If N_i is the number of input training vectors, then the complexity of the codebook design is $O(N_i \log_2 N)$ for BSP [7, 13], and $O(N_i)$ for both SSQ and ISQ [9]. If N_t is the number of test vectors to be quantized, then for the mapping, BSP requires $O(N_t \log_2 N)$ computation, while SSQ and ISQ require $O(N_t)$ operations.

Figure 9 compares the original image with an image that was quantized to 256 colors using the SSQ algorithm described in Sec. 6. A luminance weighting of K = 4 was used. This weighting has the effect of considerably reducing the visibility of contouring artifacts, hence yielding a visual quality that is superior to that achieved by other techniques [9].

8. CONCLUSIONS

We have proposed an efficient technique for quantizing vectors which we call sequential scalar quantization. A theoretical analysis of SSQ in the asymptotic case yields intuitive and useful results that allow us both to compare this method with other quantization strategies and to design a practical quantizer. We have theoretically and experimentally demonstrated the fact that SSQ yields considerably improved performance over conventional independent scalar quantization, while offering a significant computational advantage over conventional VQ. Moreover the resulting sequential structure of this technique lends itself very easily to a hardware embodiment. While we have successfully applied SSQ to the color image quantization problem, this technique is of potential use in any vector quantization application where computational cost is an important consideration, and a moderate loss in quantitative performance can be tolerated.

9. ACKNOWLEDGEMENTS

This work was partially supported by an NEC Faculty Fellowship (C. A. B.) and an Eastman Kodak Company Fellowship (R. B.). The authors are grateful to Eastman Kodak Company for the use of their image "balloon" in the experiments.

10. REFERENCES

[1] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, Jan. 1980.

[2] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: A review," *IEEE Trans. Commun*, vol. COM-36, pp. 957-971, August 1988.

[3] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, 73, 1551-1588 (1985).

[4] A. Gersho, R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Norwell, 1991.

[5] G. Braudaway, "A procedure for optimum choice of a small number of colors from a large color palette for color imaging," *Electronic Imaging '87*, San Francisco, CA, 1987.

[6] R. S. Gentile, J. P. Allebach, and E. Walowit, "Quantization of color images based on uniform color spaces," *Journal of Imaging Technology*, vol. 16, no. 1, pp. 12-21, Feb. 1990.

[7] M. T. Orchard and C. A. Bouman , "Color quantization of images," *IEEE Trans. Signal Processing*, vol. 39, no. 12, pp. 2677-2690, Dec. 1991.

[8] R. Balasubramanian and J. P. Allebach, "A new approach to palette selection for color images," *Journal of Imaging Technology*, vol. 17, no. 6, pp. 284-290, Dec. 1991.

[9] R. Balasubramanian, C. A. Bouman and J. P. Allebach, "Sequential scalar quantization of color images," to be submitted to *Journal of Electronic Imaging*.

[10] T. D. Lookabaugh and R. M. Gray, "High-resolution quantization theory and the vector quantizer advantage," *IEEE Trans. Inform. Thy.*, vol. 35, no. 5, pp. 1020-1033, Sept. 1989.

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

[12] S. Na and D. L. Neuhoff, "Bennett's integral for vector quantizers, and applications," *1990 IEEE Int'l Symposium on Information Theory*, Jan. 1990.

[13] R. Balasubramanian. C. A. Bouman and J. P. Allebach, "New results in color image quantization," *Proceedings of the 1992 SPIE/SPSE Symposium on Electronic Imaging - Science and Technology*, San Jose, CA, February 10 - 13, 1992.

11. APPENDIX

***Lemma 1*:** Let $\boldsymbol{X}$ be a random vector in R^k with probability density function $p(\boldsymbol{x})$. For $p(\boldsymbol{x})$ sufficiently smooth and N_1 sufficiently large, the following approximation holds

$$\sum_{j=1}^{N_1}\left[\left\|p(x_2 \mid B_{2j})\right\|_{1/3} P(B_{2j})\right]^{1/3} \approx N_1^{2/3}\int\left[\left\|p(x_2 \mid x_1)\right\|_{1/3} p(x_1)\lambda(x_1)^2\right]^{1/3} dx_1 \ , \qquad \text{(i)}$$

where B_{2j} and $\lambda(x_1)$ are as defined in Sec. 5.

Proof: Recall that quantization along x_1 divides this dimension into N_1 intervals B'_{1j}, $1 \le j \le N_1$. Let Δ_j be the width of interval B'_{1j}, and let y_j be any point within B'_{1j}. Given that N_1 is large and $p(\boldsymbol{x})$ is smooth, we may assume that the marginal density $p(x_1)$ is approximately constant within any interval Δ_j so that

$$P(B_{2j}) = P([X_1, X_2]^t \in B_{2j}) = P(X_1 \in B_{1j}) \approx p(y_j)\Delta_j \ . \qquad \text{(ii)}$$

Consider an interval $[y_j, y_j+dy_j]$ around the point y_j, where $dy_j >> \Delta_j$. We have

$$\begin{aligned}\Delta_j &= (\text{length of interval } [y_j, y_j+dy_j]) \,/\, (\#\text{ of quantization levels in } [y_j, y_j+dy_j]) \ , \\ &= \frac{dy_j}{N_1\lambda(y_j)dy_j} = \frac{1}{N_1\lambda(y_j)} \ . \end{aligned} \qquad \text{(iii)}$$

Now

$$P(B_{2j})^{1/3} \approx \left[p(y_j)\Delta_j\right]^{1/3} \qquad \text{(from (ii))} \ , \qquad \text{(iva)}$$

$$= \left[\frac{p(y_j)}{\Delta_j^2}\right]^{1/3}\Delta_j = \left[N_1^2 p(y_j)\lambda^2(y_j)\right]^{1/3}\Delta_j \qquad \text{(from (iii))} \ . \qquad \text{(ivb)}$$

Also, for small Δ_j we have $p(x_2 \mid B_{2j}) = p(x_2 \mid X_1 \in B_{1j}) \approx p(x_2 \mid y_j)$. (v)

Substituting (ivb) and (v) into the left hand side of (i), we have

$$\sum_{j=1}^{N_1}\left[\left\|p(x_2 \mid B_{2j})\right\|_{1/3} P(B_{2j})\right]^{1/3} \approx N_1^{\,2/3}\sum_{j=1}^{N_1}\left[\left\|p(x_2 \mid y_j)\right\|_{1/3} p(y_j)\lambda(y_j)^2\right]^{1/3}\Delta_j \ . \qquad \text{(vi)}$$

The right hand side of (vi) is a Riemann sum which may be approximated by the integral on the right hand side of (i).

Lemma 2: Let the quantizer density function $\lambda(x_1)$ along x_1 be given by (4) with $p(x)$ replaced by $p(x_1)$. For each fixed x_1, let the quantizer density function $\lambda(x_2 \mid x_1)$ along x_2, be given by (4) with $p(x)$ replaced by $p(x_2 \mid x_1)$. Then the 2-D quantizer density function $\lambda(x_1, x_2)$ is given by

$$\lambda(x_1, x_2) \approx \frac{p(x_1, x_2)^{1/3}\, p(x_1)^{2/9}}{\iint p(x_1, x_2)^{1/3}\, p(x_1)^{2/9}\, dx_2 dx_1} \ . \qquad \text{(vii)}$$

Proof: Consider an incremental rectangular area in 2-D space $\Delta(y_{1j}, y_{2k}) = \Delta(y_{1j})\Delta(y_{2k})$ surrounding a quantization value $[y_{1j}, y_{2k}]^t$, $1 \le j \le N_1$, $1 \le k \le n_{2j}$. By definition of $\lambda()$, we may make the following approximations:

(1) the number of quantization levels along x_1 within this interval is $N_1\lambda(y_{1j})\Delta(y_{1j})$;

(2) for each quantization level y_{1j} along x_1 within this interval, the number of quantization levels along x_2 is approximately $n_{2j}\,\lambda(y_{2k} \mid y_{1j})\Delta(y_{2k})$, where n_{2j} is as defined in Sec. 5.1;

(3) the total number of quantization levels within $\Delta(y_{1j}, y_{2k})$ is $N_2\lambda(y_{1j}, y_{2k})\Delta(y_{1j}, y_{2k})$.

We assume that the quantity in (2) is constant for each y_{1j} within a small enough interval $\Delta(y_{1j})$. Therefore we can approximate the quantity in (3) by a product of (1) and (2)

$$N_2\lambda(y_{1j}, y_{2k})\Delta(y_{1j}, y_{2k}) \approx \left[N_1\lambda(y_{1j})\Delta(y_{1j})\right]\left[n_{2j}\lambda(y_{2k} \mid y_{1j})\Delta(y_{2k})\right]$$

$$\Rightarrow \qquad \lambda(y_{1j}, y_{2k}) \approx \frac{N_1}{N_2} n_{2j}\lambda(y_{2k} \mid y_{1j})\lambda(y_{1j}) \ . \qquad \text{(viii)}$$

We may substitute for n_{2j} with the optimum choice given in (17)

$$\lambda(y_{1j}, y_{2k}) \approx N_1 \frac{\left[\left\|p(x_2 \mid B_{2j})\right\|_{1/3} P(B_{2j})\right]^{1/3}}{\sum_{l=1}^{N_1}\left[\left\|p(x_2 \mid B_{2l})\right\|_{1/3} P(B_{2l})\right]^{1/3}} \lambda(y_{2k} \mid y_{1j})\lambda(y_{1j}). \qquad \text{(ix)}$$

We now approximate the discrete probabilities $P(B_{2j})$ and $P(B_{2l})$ by continuous probability density functions. Substituting (iii) into (ii) to approximate $P(B_{2j})$ in the numerator above, and using (ivb) to approximate $P(B_{2l})^{1/3}$ in the denominator , we have

$$\lambda(y_{1j}, y_{2k}) \approx \frac{\left[\left\| p(x_2 \mid B_{2j}) \right\|_{1/3} p(y_{1j})\right]^{1/3} \lambda(y_{1j})^{-1/3}}{\sum_{l=1}^{N_1} \left[\left\| p(x_2 \mid B_{2l}) \right\|_{1/3} p(y_{1l}) \lambda(y_{1l})^2\right]^{1/3} \Delta_l} \lambda(y_{2k} \mid y_{1j}) \lambda(y_{1j}) \quad . \qquad \text{(x)}$$

As was done in *Lemma 1*, we argue that for small $\Delta(y_{1j})$ and Δ_l, we may replace B_{2j} and B_{2l} above with y_{1j} and y_{1l} respectively. Finally, we replace all discrete quantization variables y_{1j}, y_{2k}, with continuous variables x_1, x_2, and approximate the Riemann sum in the denominator of (x) by an integral to obtain

$$\lambda(x_1, x_2) \approx \frac{\left[\left\| p(x_2 \mid x_1) \right\|_{1/3} p(x_1)\right]^{1/3}}{\int \left[\left\| p(x_2 \mid x_1) \right\|_{1/3} p(x_1) \lambda(x_1)^2\right]^{1/3} dx_1} \lambda(x_2 \mid x_1) \lambda(x_1)^{2/3} \quad . \qquad \text{(xi)}$$

If we now substitute the appropriate expressions for $\lambda(x_1)$ and $\lambda(x_2 \mid x_1)$ according to the hypothesis of the lemma, we obtain the desired result (vii).

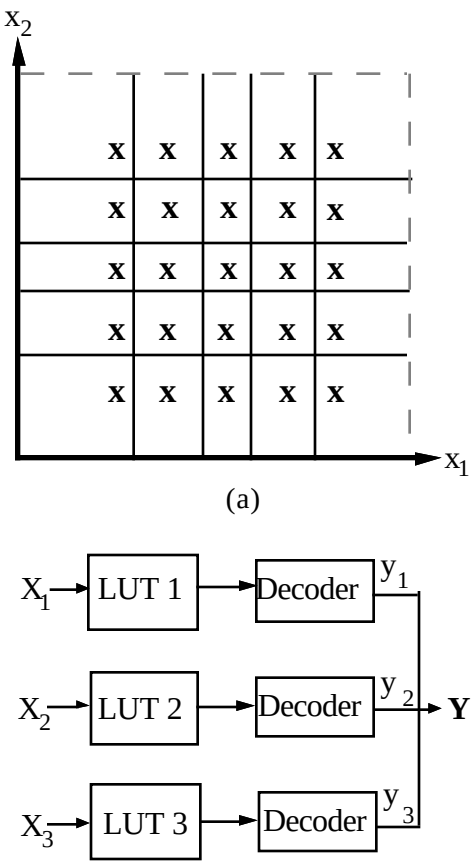


Fig. 1 (a) 2-D example and (b) encoder-decoder operation in independent scalar quantization.

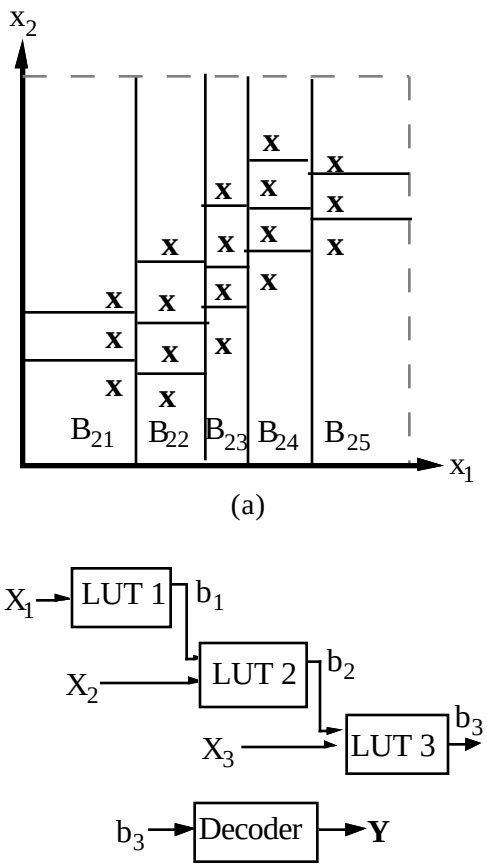
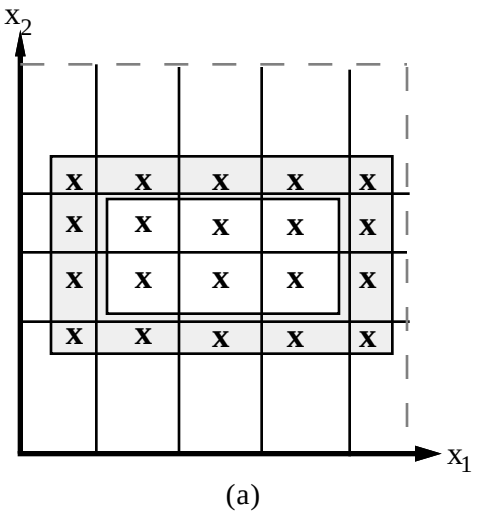


Fig. 2 (a) 2-D example and (b) encoder-decoder operation in sequential scalar quantization.



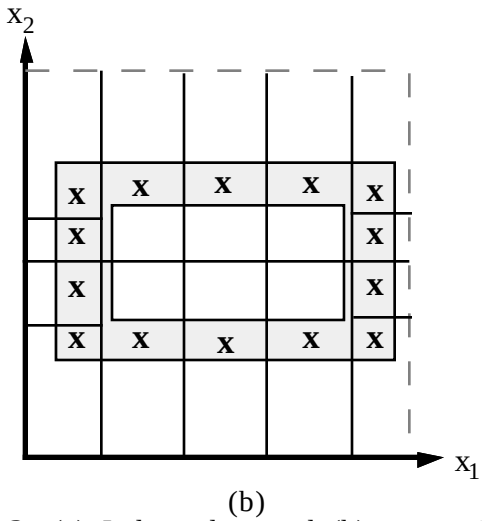
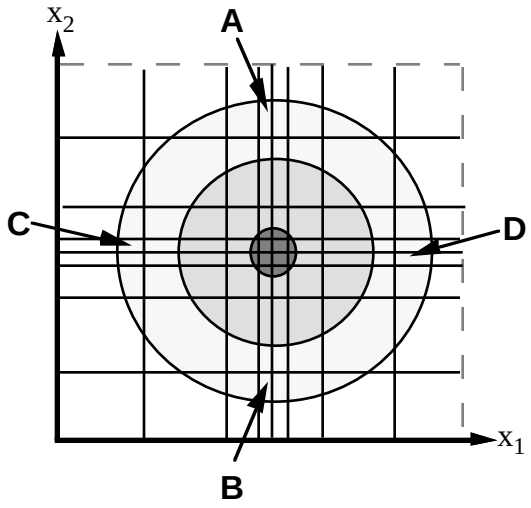
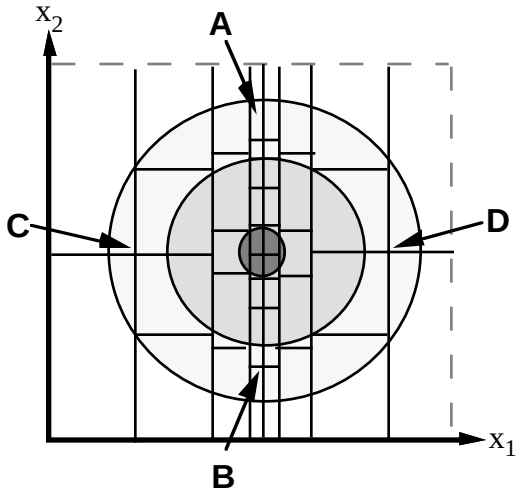


Fig. 3 (a) Independent and (b) sequential scalar quantization on uncorrelated data



(a)



(b)

Fig. 4 (a) Independent and (b) sequential scalar quantization of 2-D jointly Gaussian independent data.

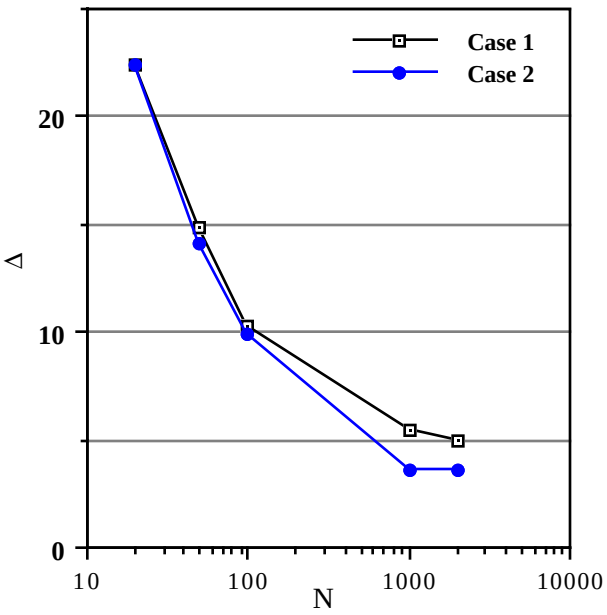


Fig. 5 Percentage discrepancy Δ between experimental and theoretical MSE's as a function of the codebook size N for 2-D Gaussian data (Case 1: $\sigma_1^2 = 1/3$, $\sigma_2^2 = 1/6$, $\rho = 0.8$; Case 2: $\sigma_1^2 = \sigma_2^2 = 1/4$, $\rho = 0$).

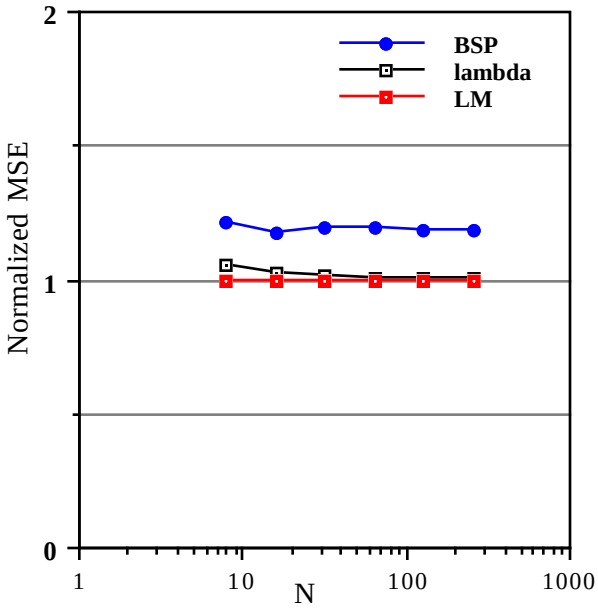


Fig. 6 Comparison of MSE performance of the lambda technique and binary splitting (BSP) relative to the Lloyd Max (LM) quantizer as a function of the codebook size N for 1-D Gaussian data.

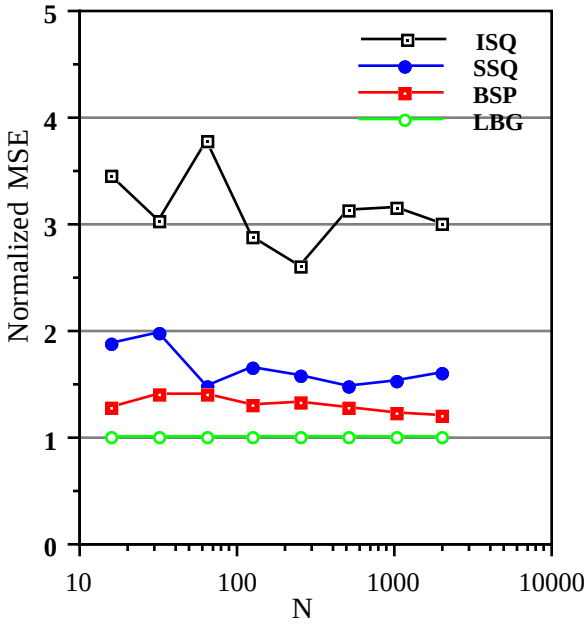


Fig. 7 Comparison of MSE performance of independent scalar quantization (ISQ), sequential scalar quantization (SSQ), and binary splitting (BSP) relative to the Linde-Buzo-Gray (LBG) quantizer as a function of the codebook size N for 3-D color image data.

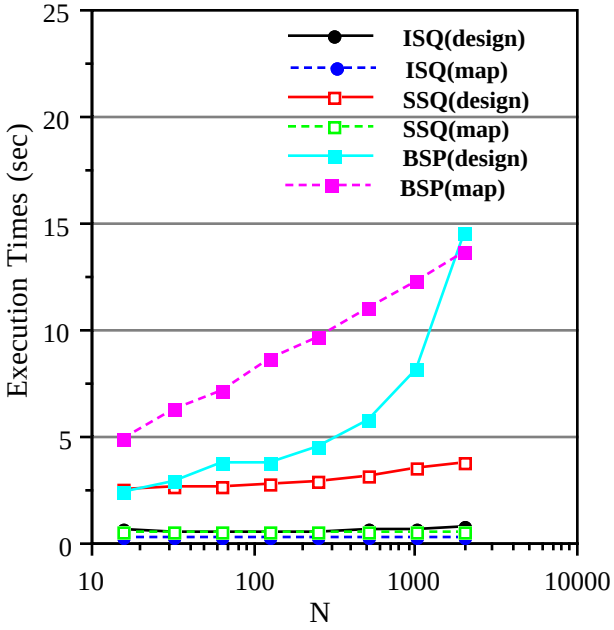


Fig. 8 Comparison of execution times for codebook design and pixel mapping on a SUN SPARCstation 2 for independent scalar quantization (ISQ), sequential scalar quantization (SSQ), and binary splitting (BSP) as a function of the codebook size N for 3-D color image data.

Fig. 9 Comparison of monochrome versions of original color image (left) and image quantized to 256 colors using SSQ (right).