Vector Quantization for Entropy Coding of Image Subbands

Takanori Senoo
Image Technology Research Laboratory
Audio Video Research Center
Matsushita Electric Industrial Co.,Ltd.
2-1-61, Shiromi, Chuo-ku, Osaka 540, JAPAN
Tel. x-81-6-947-0505
Fax. x-81-6-949-2218

Bernd Girod
Kunsthochschule für Medien Köln
Peter-Welter-Platz 2
5000 Köln 1, W-GERMANY
tel. x-49-221-201890 Fax. x-49-221-2018917

work carried out at the
Vision and Modeling Group
The Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

May 22, 1990

submitted to
IEEE Transactions on ASSP
May 1990

Abstract

Vector quantization for entropy coding of image subbands is investigated. Rate-distortion curves are computed with mean-square error as a distortion criterion. We show that full-search entropy-constrained vector quantization of image subbands results in the best performance but is computationally expensive. Lattice quantizers yield a coding efficiency almost indistinguishable from optimum full-search entropy-constrained VQ. Orthogonal lattice quantizers were found to perform almost as well as lattice quantizers derived from dense sphere packings. We also show results for binary and ternary tree-structured vector quantizers. A combination of k-D tree VQ and the Hadamard transform was found to be particularly suitable for coding of baseband images. An optimum bit-allocation rule is derived from a Lagrangian multiplier formulation. Coding results are shown for both still and moving images.

I. Introduction

For forty years the field of image coding has been devoted to seeking efficient coding methods for storage or transmission of digital images [1] [2] [3] [4]. Crochiere, et al., introduced subband coding in the context of speech coding in 1976 [5]. Subband coding has also been shown to be an efficient technique for image coding [6]. Subband pyramids, which combine the ideas behind subband coding and pyramid coding [7], can more fully utilize the image correlation in low frequency subbands [8]. The subband pyramid is reported as the most efficient decomposition among several subband decompositions by [9]. The subband pyramid itself does not reduce the overall entropy, but splits an image into a hierarchy of bandpass components. Each of these components has different statistical characteristics, and by designing a quantizer for each subband optimally, quantized data entropy can be drastically reduced without significant quality degradation.

Scalar quantization and independent encoding of the signal samples is computationally simple, but statistical dependencies between samples cannot be removed. On the other hand, vector quantization can approach the rate-distortion limit as the vector dimensionality gets large [10] [11] [12] [13] [14]. A practical code book generation algorithm for vector quantizers was first given by Linde, et al.,

in 1980 [15]. Their so-called LBG algorithm assumes fixed length encoding of the vector quantizer output. Entropy constrained vector quantization (EC-VQ) aims at a minimization of the distortion for a fixed entropy of the quantizer output rather than a fixed number of code vectors. [25] [26]. The LBG algorithm has been extended to incorporate the entropy constraint. Among known vector quantizers, EC-VQ gives the best coding efficiency when followed by entropy coding. The LBG training algorithm works iteratively, and there is the risk of convergence to a local minimum. The resulting code book requires full-search in the quantization process and thus is associated with a very large computational complexity. Several approaches have been suggested to reduce this complexity by introducing structure into the code book.

Tree-structured code books can be used to accelerate both code book generation and quantization, but they are usually slightly less efficient than full search schemes [16] [17] [18] [19]. Lattice quantization, which is an extension of uniform scalar quantization to multiple dimensions, is another approach to ease the computational complexity [20] [21] [22]. Lattice quantizers were conjectured to yield minimum mean squared error at high bit rates [20]. By exploiting the structure inherent to a lattice quantizer, the code book generation and the quantization process can be made very fast [23] [24].

To our knowledge, vector quantization for the encoding of images has only been systematically investigated in the context of fixed length encoding, as reported e.g. in [12] [6]. In this paper, we compare the performance of vector quantizers for baseband and subband images assuming entropy coding of the quantizer output. We measure rate-distortion curves for natural test images using mean squared error as distortion criterion. In section 2, subband pyramid decomposition is briefly reviewed and a new optimum bit allocation rule is derived. Section 3 reviews four kinds of vector quantization (VQ) algorithms, namely full search VQ for fixed length coding, entropy-constrained VQ, k-D tree-structured VQ, and lattice VQ. In section 4, an experimental comparison of these vector quantizers is reported for mathematical model distributions, baseband images, subband still images, and subband moving images. Finally, we present example coding results for still and moving images.

II. Subband Decomposition of Images

A. Pyramid Subband Decomposition by Recursive Bandsplitting

For the simplest possible subband decomposition, a signal is passed through a highpass filter and a lowpass filter, and the outputs of these filters are subsampled by a factor of two as depicted in Fig. 1. For upsampling and reconstruction, again a lowpass and a highpass filter are used, and their outputs are added to yield a reconstructed signal. If the filters are chosen in the right way, the aliasing that is introduced by the 2:1 subsampling cancels out in the reconstruction process.

Quadrature mirror filters (QMF) are a particularly appealing possibility for two-band subband decomposition [27]. The frequency responses of the highpass and the lowpass filter are mirror images of each other. Identical highpass and lowpass filters are used for the analysis and the synthesis filter banks. Most QMF's reported in the literature have an even number of taps [28], but filters with an odd number of taps are generally more suitable for image coding [29]. Throughout the following experiments, we have used 9-tap filters with their coefficients listed in Table 1. The frequency responses of the filters are shown in the Fig. 2. This QMF bank yields perfect aliasing cancellation, but not quite perfect reconstruction. The remaining linear frequency distortion is less than -25 dB as shown in Fig. 2.

A comment is in order concerning the use of our odd-length quadrature mirror filters. These filters require that the subsampling points be staggered between the two subbands, i.e., the low band is sampled at positions 0,2,4 ..., and the high-band is sampled at the positions 1,3,5, ... The alternating sampling preserves the information more uniformly and gives better reconstruction than even-length QMF's that require coinciding sampling in both bands.

Among several possible types of subband decomposition schemes, we adopt the four-level subband pyramid illustrated in Fig. 3. On each pyramid level, the low frequency band is divided into four subbands by a concatenation of a horizontal and vertical 2:1 band splitting. The resulting four subbands are horizontal high frequency band, vertical high frequency band, diagonal highs, and a

new low frequency band. As adjacent image samples have a large positive correlation, most of the energy is concentrated in the low frequency band. The energy contained in the other subbands is usually much smaller. The amplitude distribution of the lowest frequency subband is shown in Fig. 4. A typical amplitude distribution of a high frequency subband is shown in Fig. 5. The distribution in Fig. 5 closely resembles a Laplacian distribution. The distributions of the other subbands also have similar shapes to the Laplacian distribution, although their variance differs.

B. Optimum Bit Allocation

Fewer bits can be allocated to the high frequency subbands than to the low frequency subbands as they have less variance, hence less information. How many bits for each subband is optimum? Often, a rule-of-thumb rule is used which postulates equal mean-squared error in each band. The number of bits allocated to each subband is proportional to the logarithm of its variance. If the energy within a band is smaller than the desired mean-squared error value, the band is discarded. This rule can be justified by a rate-distortion analysis of Gauss-Markov source models [30].

A better bit-allocation rule that makes fewer assumptions about the signal statistics is derived as follows. Our goal is to minimize the overall mean squared error for a given overall bit-rate. Mathematically, we write

$$\min_{R_i} \left(\sum_{i=1}^{n} D_i(R_i) \right) \tag{1}$$

under the condition

$$\sum_{i=1}^{n} R_i \leq R_{max} \tag{2}$$

where $D_i(R_i)$ and R_i are the distortion and the rate for each subband, respectively. R_{max} is a given total bit rate. $D_i(R_i)$ is an operational distortion rate function for an individual subband i, which will depend on the quantization and encoding scheme used for the subband. Note that we assume independent encoding of the individual subbands, and thus the rates for the subbands

are combined additively added in (2). We also assume an additive superposition of the distortions $D_i(R_i)$ in (1). This additive superposition is valid for a mean-squared error measure in an orthogonal subband decomposition, given that the individual $D_i(R_i)$ are appropriately scaled.

If we additionally assume that the distortion rate functions $D_i(R_i)$ are continuous and convex, we can find the optimum bit allocation by a Lagrange multiplier method.

$$\min_{R_i} \left(J = \sum_{i=1}^{n} D_i(R_i) + \lambda \sum_{i=1}^{n} R_i \right) \tag{3}$$

where J is the overall cost function, and λ is the Lagrange multiplier. Minimum overall distortion for a given total rate is achieved when the first partial derivatives of the cost function vanish, i.e.,

$$\frac{\partial J}{\partial R_i} = \frac{\partial \sum_{i=1}^{n} D_i(R_i)}{\partial R_i} + \lambda \frac{\partial \sum_{i=1}^{n} R_i}{\partial R_i} = 0 \tag{4}$$

This condition yields

$$\frac{\partial D_i(R_i)}{\partial R_i} = -\lambda \tag{5}$$

for all subbands i. Thus, optimum bit allocation corresponds to the points where the individual distortion rate curves $D_i(R_i)$ have equal slope (In Economics, this condition is called "Pareto-optimality".). This bit allocation rule provides a strong motivation to measure distortion rate curves for the individual subbands.

III. Vector Quantization Algorithms

A. VQ for Fixed Length Coding

Vector quantizers map a multidimensional space into a finite or a countable set of reproduction vectors. The set of reproduction vectors is the *code book.* Linde, Buzo, and Gray in 1980 generalized the Lloyd algorithm for the design of nonuniform scalar quantizers to yield a code book for vector

quantizers [15]. Their code book generation scheme is known as LBG algorithm, or generalized Lloyd algorithm. The LBG algorithm yields a minimum distortion quantizer for a given number of reproduction vectors, thus the resulting code book is optimum for successive encoding with fixed code word length. The algorithm starts with an initial set of vectors and then iterates to determine the optimum code book. In the iteration, optimum reproduction vectors are computed for a given subdivision of the vector space (Assuming that we are using mean-squared error as a distortion criterion, the minimum mean squared reproduction vector is the centroid of a given region). Then, a new subdivision of the vector space is calculated that is optimum for the newly computed reproduction vectors. Now, the first step is repeated, i.e., the set of reproduction vectors is optimized for the new subdivision, and so on until convergence. Fig. 6 (b) illustrates an example of two-dimensional vector space divided by LBG algorithm. Figure 6 (a) depicts the training vector space with its distribution density contour map.

The code book resulting from the LBG training procedure is entirely unstructured, and the quantization process requires a full search of the code book, i.e., the input vector has to be compared to each code book entry in order to find the closest reproduction vector. The number of operations necessary for this full search is large. Assuming again a mean squared error distortion criterion, roughly $M \times N$ times multiplications and additions per vector are required, where M is the vector dimensionality and N is size of code book. The decoding process is simply a table look-up.

B. Entropy-Constrained VQ

Further data compression can be achieved if the vector quantizer is followed by an entropy coder that takes advantage of the statistics of the vector quantizer output. The LBG code book generation as described above is not optimum for such a scheme, as it assumes fixed length coding. Recently, the LBG algorithm has been extended for code book design under an entropy constraint [25]. Entropy-constrained vector quantization (EC-VQ) minimizes distortion for a given variable length code word assignment rather than a given code book size. Mathematically,

$$\min(D) \tag{6}$$

under the condition of

$$R \leq R_{max} \tag{7}$$

where, D, R, and R_{max} are average distortion, bit rate, and maximum allowable bit rate, respectively.

Similar to the optimum bit allocation problem above, a Lagrange multiplier formulation is used to design the optimum code book:

$$\min\left(J = D + \lambda R\right) \tag{8}$$

where J is the cost function, λ is the Lagrange multiplier which incorporates the bit rate constraint. Given a training set consisting of N vectors, this expression can be rewritten as

$$\min\left(J = \frac{1}{N}\sum_{j=1}^{N}\rho_j + \lambda\frac{1}{N}\sum_{j=1}^{N}l_j\right) \tag{9}$$

where ρ_j and l_j are the distortion and the code word length associated with the j-th training vector.

The entropy-constrained VQ code book design follows the rules of the LBG algorithm—the only difference is in the subdivision of the vector space for a given set of reproduction vectors. When the reproduction vectors are given, the above expression (9) is satisfied by clustering the training vectors according to

$$\min\left(\rho_j + \lambda l_j\right) \tag{10}$$

Note that any code word assignment rule can be incorporated into (9) and (10). In experiments reported in section IV, we assume

$$l_j = -\log_2(\frac{n_j}{N}) \tag{11}$$

where n_j is the relative frequency of occurrence of the j-th reproduction vector in the set of training vectors. For a given subdivision of the input vector space (or the training set), reproduction vectors are then optimized as for the fixed length version of the LBG algorithm. For a mean squared error distortion criterion, the centroids of each cell are the best reproduction vectors. Again, the design algorithm iterates until convergence.

For EC-VQ, the number of initial reproduction vectors should be sufficiently large because its code book coincides to the LBG code book when λ equals zero. Otherwise, the resulting code book

will not achieve the theoretical minimum distortion for a given rate [25]. Redundant vectors are automatically eliminated in the iterative design process.

As with the LBG-designed code book for fixed length encoding, the entropy-constrained vector quantization process itself is computationally expensive, since it requires full search of the code book to find the closest reproduction vector. Note that "closest reproduction vector" for entropy-constrained VQ means "closest" in the sense of (10). The code word length is part of the cost function. A reproduction vector with a short code word length may be preferred over another vector with a longer code word length, even though the Euclidian distance to the second vector is shorter. The first vector leaves more bits for future vectors to be encoded, and the overall mean squared error is lower. Unlike the fixed code word length VQ, the vector space is not subdivided by hyperplanes with EC-VQ. Rather, the boundaries between cells in EC-VQ are curved shown in Fig. 6 (c).

C. k-D Tree-Structured VQ

The weak point of full-search vector quantization algorithms is the number of computations required for the code book generation and especially for the quantization process itself. Tree-structured vector quantization is a powerful alternative to accelerate both code book generation and encoding speed. As tree-structured VQ encodes an input vector by tracing a binary (or n-ary) tree, the computational complexity reduces the order of complexity from MN (for full search methods) to $M \log_n(N)$. Among several tree-structured vector quantizers, we investigate the k-D tree algorithm in this paper [16] [17] [18] [19]. The k-D tree is a binary tree. It is extremely fast because it considers the distortion of only one vector component at each tree node, i.e. at each node, a single vector component is compared to a threshold. The resulting subdivision of the input vector space is made up of hyperplanes orthogonal to a coordinate axis. Because of this constraint, the performance of a k-D tree VQ is bound to be lower than an LBG-designed full search vector quantizer. k-D tree vector quantizers for variable code word length can be designed by growing an unbalanced binary tree where the depth of the tree is variable.

In the experiments reported in Section IV, the k-D tree is built by recursively splitting cells in

vector space along the coordinate which currently has the widest range. The range is the interval between the minimum value and maximum value actually occurring in training set. The split threshold is the mid-point of this range, i.e. half way between minimum and maximum. This split criterion was adopted as it yielded the best coding efficiency for subband image among the other criterions. This method roughly simulates an uniform vector quantization. Fig. 6 (d) shows a k-D tree built in two-dimensional vector space. Finally, each cell is represented by a vector which is the centroid of the cell.

k-D trees do not perform well for highly correlated data, such as baseband images. In a baseband image, vectors made of neighboring pixels cluster around a 45 degree diagonal in the vector space. The k-D tree cannot divide the vector space in any way other than orthogonal to one of the axes. If the ridge of the distribution is not along an axis (which is the case most of the time) performance is poor. If we can rotate the vector space coordinate system to align one of the coordinate axes with the vector distribution ridge, the performance will be greatly improved. A particularly simple way to achieve such a rotation is a Hadamard transform [30]. It requires only additions, subtractions, and shift operations, and fast transform algorithms are known. Other transforms, as for example the Haar transform [30] or the Discrete Cosine transform [30] could be used as well. We denote the concatenation of the a Hadamard transform and a k-D tree vector quantizer as *Hadamard k-D tree vector quantizer.* An example of Hadamard k-D tree is illustrated in Fig. 6 (e).

Bandpass images resulting from a subband decomposition possess a much lower correlation, and the rotation by a Hadamard transform is not necessary. The multidimensional probability density function of subband images is usually sharply peaked around the zero-vector. The binary k-D tree usually splits peak of the distribution if data are distributed symmetrically. Splitting a peak sacrifices coding efficiency because the new reproduction vectors after the split are still close to the peak. Hence, average distortion does not decrease while the bit rate increases. We experimented with a ternary k-D tree to overcome this problem. The ternary k-D tree divides a cell into three equal-sized sub-cells along one of the vector coordinates. Consequently, a center peak of the former cell is preserved in one of the sub-cells without split. Fig. 6 (f) shows a ternary tree.

D. Lattice VQ

Assuming a smooth one-dimensional probability density function and a sufficiently fine quantizer, it has been shown that the uniform scalar quantizer approaches the minimum mean squared quantization error for a given entropy [20]. It has been conjectured that this finding can be extended to multiple dimensions, and that lattice quantizers that correspond to the densest multidimensional sphere packing would play the same role as the uniform quantizer in 1-D. For example, in 2-dimensional case, it is well known that a hexagonal partitioning of the space yields a lower mean squared error than a rectangular partitioning. The corresponding hexagonal lattice is called A_2-lattice. In 3-dimensional space, the A_3-lattice is known to yield the best partitioning [20]. The regions in vector space with the smallest Euclidean distance to a lattice point are called Voronoi regions, Dirichlet regions, or nearest neighbor regions. For example, the Voronoi regions of an A_3-lattice have the shape of a truncated octahedron. All input vectors of a Voronoi region are reproduced by a vector that is the centroid of the region rather than by the corresponding lattice point itself. In higher dimensions, lattices corresponding to densest sphere packings are currently known only in 8-D (the E_8-lattice) and in 24-D (the Leech lattice) [22].

For several lattice quantizers, fast algorithms have been developed to find the lattice point closest (in the mean squared sense) to an arbitrary vector. We have used vector quantizers based on the lattices A_2, D_4, E_8, and the orthogonal lattices Z^2, Z^4, and Z^8 for the experiments in Section IV. In Fig. 6 (g) and (h), A_2 and Z^2 are shown. The quantizers and their fast algorithms are summarized in the Appendix. More details can be found in [23] [24].

The size of a lattice VQ code book is in principle infinitely large. In practice, however, we limit the range of the input vectors and the code book size becomes finite. Still, compared to the LBG designed code books, our lattice code books are typically an order of magnitude larger at a given bit rate. This size is an advantage with respect to picture quality, since quantizer overload never occurs and even very unlikely input vectors can be represented well, although they require a large number of bits. However, the large code book requires a lot of memory space. As a remedy to the memory

problem, we adopted a hybrid code book method. In this method, probabilities and centroids for Voronoi cells are measured and stored for a code book of feasible size, based on the training set. This code book now contains the "most popular" code vectors. When an image that is not part of the training set is encoded, lattice cells may be encountered without pre-measured probability and centroid. For these cells the lattice point is used as a reproduction vector and a fixed length code word is assigned. The length of this code word is sufficiently large to cover the entire range of lattice cells. Although the total lattice space is huge when the vector dimensionality is large, the number of lattice points used frequently are limited. Consequently, it suffices to only build an explicit code book for the popular Voronoi regions. The code word length outside the explicit code book is long, but the frequency of its occurrence is very small, so its effect on the average code word length is negligibly small.

IV. Experimental Comparison of Vector Quantizers

In this section, we experimentally compare the performances of the vector quantizers introduced in the previous section for two source models, still images, and moving images. The quantizers compared are the uniform memoryless scalar quantizer, LBG-designed full search VQ with fixed-length coding, entropy-constrained VQ, k-D tree search VQ, Hadamard k-D tree VQ, ternary k-D tree VQ, and several lattice vector quantizers. The mean squared error performance of these quantizers is compared over a wide range of bit rates. Finally, we apply our results to form a still image coder and a moving image coder.

A. Performance for Model Distributions

In order to obtain a general idea about the various vector quantizers, we have studied their performance for two well-defined source models. The first model is a first order Gauss-Markov model with a correlation coefficient $r = 0.95$. The correlation between adjacent samples in a baseband image is high, and the Gauss-Markov model can be helpful in interpreting VQ results for baseband images. The second model is a memoryless random variable with a Laplacian probability density

function corresponding roughly to the zeroth-order statistics found in bandpass or high frequency subbands. The training set are the test data consisted of 65,536 samples each. Except for the full search vector quantizer designed by the LBG algorithm with a fixed code word length constraint (Section III. A), entropy was measured for all quantizers.

Gauss-Markov Model

Fig. 7 shows the performance of six quantizers for the first-order Gauss-Markov model with correlation coefficient $r = 0.95$. The solid line is the Shannon Lower Bound (SLB) of the distortion rate function. The SLB for a first-order Gauss-Markov model is

$$D(R) = 20R\log_{10}(2) - 10\log_{10}(1 - r^2)\ (dB). \tag{12}$$

where D and R are the SNR in dB and the bit rate in bpp, respectively. r is the correlation coefficient.

For highly correlated data, vector quantizers perform much better than the scalar quantizer (SCALAR) because they take advantage of the correlation between pixels. All vector quantizers shown in Fig. 7 work in eight-dimensional space. Among the vector quantizers, full-search algorithms generally performed better than those with the structured code books for fast search. Interestingly, entropy constrained full-search VQ (ECVQ8) does not perform significantly better than full-search VQ with fixed length coding (LBG8). The entropy-constrained quantizer design improves performance only for very peaky probability density functions (PDF's), as we will see later on. For less peaky PDF's like Gaussians, the entropy constraint is not effective. In fact, if we follow the fixed length full-search VQ (LBG8) by entropy coding, the bit rate will not be significantly reduced since all code words are roughly equally likely.

The k-D tree (KD8) performed about 1 dB worse than full search algorithms. As explained in Section III.C, the k-D tree can only split the vector space along one of the coordinate axes, and not along the distribution ridge of the highly correlated data. By aligning one of the coordinate axes with the distribution ridge, the Hadamard k-D tree (HKD8) performs as well as the full-search VQ's at much lower computational complexity. At low bit rates, the HKD8 performs even slightly

better than the full-search VQ's at a much smaller computational cost. However, its performance saturates/deteriorates at higher bit rates because of the insufficient quantity of the training images. This problem also occurs for conventional k-D tree VQ. Surprisingly, the E_8-lattice VQ performs even worse than the k-D tree in the range of bit rates shown in Fig. 7.

These experiments show that for a highly correlated Gauss-Markov source a Hadamard k-D tree VQ is very attractive at low bit rates. Very little is gained by incorporating an entropy constraint into the quantizer.

Memoryless Source with Laplacian Distribution

Fig. 8 shows the performance of various quantizers for a memoryless source with Laplacian PDF. The solid line is Shannon lower bound (SLB)

$$D(R) = 20R\log_{10}(2) - 10\log_{10}(\frac{e}{\pi})\ (dB). \tag{13}$$

The results are dramatically different from the first-order Gauss-Markov source results. The uniform scalar quantizer (SCALAR) performed better than the tree-search vector quantizers such as k-D tree (KD8), Hadamard k-D tree (HKD8), and the ternary k-D tree (Tri-KD8). In fact, the scalar quantizer performance is close to the full-search vector quantizers. Of course, there is no coding gain due to a statistical dependency between the vector components. Any gain over scalar quantization is due to a better multi-dimensional packing of the vector quantizer cells.

The ternary k-D tree algorithm (Tri-KD8) performed relatively well at low bit rates, but its advantage is soon lost as the bit rate increased. Among all quantizers, entropy-constrained VQ performed best. The E_8-lattice VQ (E8-LTC) performed almost as well as the entropy-constrained VQ (ECVQ8) at a much lower computational cost.

It is important to note that the memoryless Laplacian source and the highly-correlated Gauss-Markov source lead to a very different ranking of the merits of the various quantization schemes.

B. Direct Vector Quantization of Baseband Images

In this section, vector quantizers are compared for a natural test image (Lenna, 512×512 pixels). The image is quantized directly, without subband decomposition. A block of 2 adjacent pixels in 4 successive lines formed an 8-D vector for quantization. The code book was made from a training set consisting of four face images as shown in Fig. 9. Note that the test image Lenna is not contained in the training set.

Fig. 10 compares the performances of the quantizers. It resembles curves obtained for the Gauss-Markov model. As expected, vector quantization is greatly superior compared to scalar quantization. Among the vector quantizers, EC-VQ showed the best performance. The E_8-lattice VQ, the Hadamard k-D tree VQ, and the full-search fixed length VQ (LBG8) follow closely. The conventional k-D tree VQ performed up to 4dB worse to other vector quantizers, but the Hadamard rotation greatly improved its coding efficiency as shown by the HKD8. Scalar quantization is worst because it does not exploit the spatial correlation. The lattice quantizer performed better than it did for the Gauss-Markov model.

C. Vector Quantization of Image Subbands

In order to improve coding efficiency further, one would have to make the vector dimensionality larger. Large vector dimensionality unfortunately implies exploding code book and training set sizes if we desire a meaningful results at high bit rate. A more feasible approach to improve the coding efficiency is to combine vector quantizers with some other decorrelating method. We adopted subband pyramid coding because of its high coding efficiency and the absence of block artifacts. Specifically, we used the four level subband pyramid illustrated in the Fig. 3.

With each pyramid level, the image area decreases by a factor of 4. This implies that the size of our training set for that particular pyramid level shrinks accordingly and becomes too small rapidly. We could use a larger number of images in our training set to compensate this effect, but then

we would have to process a lot more data for the high frequency pyramid levels also, where the training set in principle was large enough. Instead of using more images in the training set, we use a trick to extend our training set where needed. Consider the pyramid level n that has a quarter of the samples of pyramid level $n-1$. First, we decompose the level $n-1$ image into high band and low band horizontally and vertically as usual. The resulting horizontal highs, vertical highs, and diagonals on level n are arranged into vectors and entered into the training set. Now, we shift the level $n-1$ data by one sample horizontally, and we repeat the subband decomposition. The one-sample shift corresponds to a half-sample shift after subsampling, and the resulting vectors are sufficiently independent from the previous decomposition to extend the training set. The procedure is repeated for a vertical one-sample-shift and a diagonal shift of $(1,1)$. The resulting increase in training set size exactly compensates for decrease due to subsampling in the pyramid, and a training set size independent from the pyramid level results.

Figures 11 to 13 show typical distortion rate curves for selected subbands. We show mean-squared error rather than SNR to ease a later application of the bit-allocation rule derived in Section II. B. Fig. 11 depicts the performance on a horizontally high subband of the first pyramid level (H0). The results correspond to the Laplacian distribution model discussed above. Binary tree methods like k-D tree VQ, Hadamard k-D tree VQ, and the LBG-designed fixed-length VQ perform poorly, since they tend to divide the distribution peak. The ternary k-D tree performs relatively well at low bit rates only. E_8-lattice quantization (E8-LTC) performs well, almost as good as EC-VQ at low rates. At higher rates, the E_8-lattice quantizer performs even better than EC-VQ. This result, which can also been seen in some other figures in this paper, is actually quite typical although it might be surprising initially. At the higher rates, the LBG-trained quantizer has too many degrees of freedom and overadapts itself to the training set data. Consequently, its performance drops for data outside of the training set. The lattice quantizer is much more constrained and consequently suffers from this problem to a lesser extent.

In Fig. 11 we also included a Z^8 lattice quantizer (UNIF8). The Z^8-lattice is orthogonal and thus completely separable, and its Voronoi regions are 8-D hypercubes. The subdivision of the 8-D vector space is conveniently done by 8 independent scalar quantizers. Different from straightforward scalar

quantization, however, the quantizer output is jointly encoded in 8-D, and the each 8-D Voronoi cell has its individual reproduction vector. Surprisingly, the Z^8-lattice performs close to the E_8-lattice VQ, and is actually closer to the E8-LTC curve than to the conventional scalar quantization curve with independent 1-D encoding and centroids (SCALAR) in Fig. 11. Most of the lattice VQ gain over scalar quantization is due to joint encoding and joint centroids, while the denser sphere packing property of an E_8-lattice seems to be less important.

Fig. 12 shows similar results for the vertical high subband for the second pyramid level (V1). We show results for 4-D vector quantizers rather than for 8-D here. Higher-dimensional vector quantizers perform better at low bit rate, but the number of degrees of freedom gets too large for high bit rates, and performance deteriorates due to overtraining. In fact, at higher bit rates, we were not able to train 8-D quantizers to outperform the corresponding 4-D quantizers. Note that with our subband pyramid subband scheme, distortion is weighted heavier than bit rate at higher pyramid levels. For the overall image, the mean squared error figures associated with the subbands simply add up, while the contributions of the individual bit rates (in bits/sample) decrease in proportion to the number of samples in the pyramid level.

Fig. 13 shows a comparison of 2-D quantizers for the lowest frequency subband on the fourth pyramid level (L3). The result is as one would expect from the Gauss-Markov model 7 or the baseband image curves 10. For the high bit rates that we are interested in for the highest level of a subband pyramid, higher-dimensional vector quantizers did not perform as well as 2-D vector quantizers due to overtraining. Entropy constrained VQ (ECVQ2) performs best but appears to be overtrained at the very high rates. The A_2 lattice quantizer (A2-LTC) is second and outperforms even EC-VQ at the very high rates. k-D tree VQ (KD2), Hadamard k-D tree VQ (HKD2), LBG-designed fixed word length VQ (LBG2), and the scalar quantizer (SCALAR) follow with lower performance.

D. A Still Image Coder

We used the distortion-rate curves discussed in the previous section to determine the optimum bit allocation according to (5) in the subband pyramid. Each subband is quantized at a point on

the distortion-rate curves of some fixed slope. Of course, the bit rate axes were scaled beforehand according to the number of samples on the pyramid level. The best quantizer was selected individually for each subband taking also into account the computational cost associated with the scheme. More specifically, lattice quantizers were chosen as the best quantizer for each subband. The E_8 lattice was selected for the first level of the pyramid, the D_4 lattice for the second level, and the A_2 lattice for the third and the fourth level.

Fig. 14 shows the original test picture "Lenna". Fig. 15 shows an encoded example, with a bit rate of 0.136bpp and an SNR of 30.9 dB. SNR is measured relatively to the peak-to-peak range of the video signal. The overall quality is very good for a compression ratio of approximately 60:1 (starting with 8 bpp). Picture quality is slightly impaired by blur and jagged diagonal edges. For the comparison with the direct vector quantization, an encoded Lenna with the entropy-constrained 8-dimensional vector quantizer is shown in Fig. 16. The bit rate is 0.137bpp and peak SNR was 19.2dB.

Fig. 17 illustrates the bit allocation among subbands and the contribution of each subband to the overall distortion. Interestingly, the mean square error (MSE) contribution of the various subbands is by no means equal. It ranges between MSE = 1.07 for subband $L3$ and MSE = 10.46 for subband $H0$. If we use the rule-of-thumb bit allocation rule of equal distortion in all subband, the total distortion increases by 0.8dB at the same bit rate. We also encoded the image with orthogonal lattice quantizers, and it was confirmed that these quantizers perform only slightly worse than the denser lattices.

E. Encoding of Moving Image

The results for still picture coding can be easily extended to moving images. One of the applications that we are interested in is the encoding of moving video at rates around 1 Mbit/s for distribution on a medium the optical compact disc. Our strategy for moving image coding is slightly different from the still image coding presented above in that we compute an optimized code book for each $\frac{1}{3}$ second of the signal and store the code book as side information. Thus, the code book information has to be included in the total bit rate. We used the test sequence *Alley*. The luminance

signal of *Alley* was sampled at $(480 lines) \times (640 pixels/line)$, and the frame rate was 24 frames per second. The sequence was encoded with a 3-dimensional subband coding scheme and orthogonal lattice quantizers. We adopted an 8-point Discrete Cosine Transform (DCT) along the temporal axis instead of a temporal subband decomposition with overlapping filter impulse responses. The DCT can be regarded as a subband filter bank with short filter lengths. Consequently, memory requirement for the filtering process are relatively low, although the temporal band separation is not very good. The temporal subbands have equal bandwidth and do not form a pyramid.

Fig. 18 depicts the 3-dimensional subband scheme for the moving image coding. The diagonally high subbands were generally discarded before the quantization. The horizontal highs (LLH) and vertical highs (LHL) were further subsampled by factor 2 vertically or horizontally. These frequency components usually have small energy and are less visible to the human eye. Spatial bandwidth of temporally high subbands is reduced by half. According to this spatio-temporal subband scheme, 60 subbands have to be quantized separately. We evaluated full search LBG algorithms, entropy-constrained VQ's, orthogonal lattice quantizers, and k-D tree VQ's of different dimensionality for these image subbands. Among these quantizers, the orthogonal lattice VQ's were chosen as the best quantizer. In our coding system for moving image, as the code books are renewed every 8 frames, small vector dimensionality yielded better results. The Z^3 lattice quantizer was chosen for subband LLH, and Z^2 was chosen for the subbands LHL,LH, and HL. The vectors were aligned to image pattern orientation of each subband. Z^1 was optimum for the other subbands.

Figs. 19 and 20 show the first frame of original and the encoded sequence, respectively. Subjectively, the encoded sequence is almost indistinguishable from the original. The bit rate including the code book was 0.146bpp, or 1.08Mbps at a peak SNR = 39.4dB. As the temporal correlation in the moving image sequence is usually high, higher coding efficiency was achieved compared to the still image. Fig. 21 depicts the peak SNR fluctuation as a function of frame number. The color components of the sequence Alley were also encoded, requiring 82.1 kbps for a peak SNR of 41.1dB, but they are not shown here.

V. Conclusions

In this paper, optimization of vector quantization for entropy coding of subband images has been investigated. Our comparison was based on mean squared error performance over a wide range of bit rate. It included full-search entropy-constrained vector quantizers, lattice quantizer, and tree-structured vector quantizers. We have experimented with source models, both with memory and without memory, still images and moving images and gained several new experimental insights useful for vector quantization of images.

Several vector quantizers were applied directly to a baseband image. Given the limited vector dimensionality (= 8), coding efficiency was generally rather poor. Much better results were obtained by combining VQ with a subband pyramid. For high frequency subbands, which have low spatial correlation and a peaky distribution, it was shown that EC-VQ gave the best results, but is computationally very expensive. We prefer lattice VQ which is almost as efficient and much cheaper. Orthogonal lattices perform almost as well as lattices derived from densest sphere packings.

An optimum rule for bit allocation among the subbands has been derived, that concludes equal slope of the individual distortion rate curves. We applied this rule to design a coder for still images as well as a coder for moving images. Using lattice quantizers in the subband pyramid, Lenna was encoded with a peak SNR = 30.9 dB at bit rate = 0.136 bpp. The image sequence Alley was encoded with peak SNR = 39.4dB at 0.146 bpp.

In many situations, vector quantization optimized for variable length coding can improve performance greatly over conventional vector quantizers designed for fixed length coding. It is a very satisfying result that the simple and elegant lattice quantizers yield an almost optimum performance when combined with a subband pyramid. The important aspect of vector quantizers for entropy coding of image subbands is not the exact subdivision of the multidimensional space. For that purpose an orthogonal lattice seems sufficient. The major coding gains are due to joint encoding of several samples and the use of individual reproduction vectors in each Voronoi region of a multidimensional lattice.

Acknowledgement

The authors wish to thank everybody in the Vision & Modeling Group and in the Movies of the Future Group of the MIT Media Laboratory. Especially, we thank Prof. E. H. Adelson and Mr. E. Simoncelli for the design of the QMF filters.

Appendix

A_2 lattice

An A_2 lattice is generated by the matrix

$$U = \begin{bmatrix} \mathbf{u_1} \\ \mathbf{u_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \tag{14}$$

Any A_2 lattice point $\mathbf{x}$ is generated by the integer combination of the row vectors $\mathbf{u_1}$ and $\mathbf{u_2}$.

$$\mathbf{x} = a_1\mathbf{u_1} + a_2\mathbf{u_2} \tag{15}$$

where a_1 and a_2 are arbitrary integers.

A fast A_2 lattice quantization algorithm is as follows:

Step 1: Scale input vector $\mathbf{x}$ by multiplying the scaling matrix M and fit the lattice points on integer points.

$$\mathbf{x}' = M\mathbf{x} = \begin{bmatrix} \frac{1}{step} & 0 \\ 0 & \frac{1}{\sqrt{3}step} \end{bmatrix} \mathbf{x} \tag{16}$$

where step is the quantization step size.

Step 2: Shift $\mathbf{x}'$ by $(-\frac{1}{2}, -\frac{1}{2})$, then round both unshifted $\mathbf{x}'$ and shifted $\mathbf{x}'$ to integer.

$$\mathrm{f}(\mathbf{x}) = \mathrm{round}(\mathbf{x}') \tag{17}$$

$$\mathrm{f}(\mathbf{x} - \frac{\mathbf{1}}{\mathbf{2}}) = round\left(\mathbf{x}' - (\frac{1}{2}, \frac{1}{2})\right) \tag{18}$$

Step 3: Shift $\mathbf{f(x-\frac{1}{2})}$ by $(\frac{1}{2},\frac{1}{2})$, then re-scale boyh $\mathbf{f(x)}$ and shifted $\mathbf{f(x-\frac{1}{2})}$ by multiplying the matrix M^{-1}.

$$\mathbf{y_0(x)} = \mathbf{M^{-1}f(x)} \tag{19}$$

$$\mathbf{y_1(x)} = \mathbf{M^{-1}}\left[\mathbf{f(x-\frac{1}{2})} + (\frac{1}{2},\frac{1}{2})\right] \tag{20}$$

$$M^{-1} = \begin{bmatrix} step & 0 \\ 0 & \sqrt{3}step \end{bmatrix} \tag{21}$$

Step 4: Compare $\mathbf{y_0(x)}$ and $\mathbf{y_1(x)}$ and choose one which has smaller distance $\mathbf{d_i}$ to $\mathbf{x}$ as the closest A_2 lattice point.

$$d_0 = \|\mathbf{x} - \mathbf{y_0(x)}\| \tag{22}$$

$$d_1 = \|\mathbf{x} - \mathbf{y_1(x)}\| \tag{23}$$

D_4 lattice

D_4 lattice is generated by the following matrix.

$$U = \begin{bmatrix} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{u_3} \\ \mathbf{u_4} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \tag{24}$$

D_4 lattice quantization algorithm is the following procedure.

Step 1: Scale input vector x by

$$\mathbf{x'} = \frac{\sqrt{2}}{step} \times \mathbf{x} \tag{25}$$

Step 2: Round $\mathbf{x'}$ to integer.

$$\mathbf{f(x)} = round(\mathbf{x'}) \tag{26}$$

Step 3: If the sum of vector components of $\mathbf{f(x)}$ is even, take the $\mathbf{f(x)}$ as the closest D_4 lattice point $\mathbf{y}'$.

$$\mathbf{f(x)} = (\mathrm{f_1}, \mathrm{f_2}, \mathrm{f_3}, \mathrm{f_4}) \tag{27}$$

if $\sum_{i=1}^{4} \mathrm{f_i} = even$,

$$\mathbf{y}' = \mathbf{f(x)} \tag{28}$$

Step 4: If not, re-round the component of $\mathbf{x}$, which is the furthest from an integer, to the next nearest integer, then take it as the closest D_4 lattice point $\mathbf{y}'$.

$$f_{max} = (f_i \mid \max_i(|x_i - f_i|)) \tag{29}$$

if $(x_{max} - f_{max}) \geq 0$,

$$\mathbf{y}' = (\mathrm{f_1}, ..., \mathrm{f_{max}} + 1, ..., \mathrm{f_4}) \tag{30}$$

else,

$$\mathbf{y}' = (\mathrm{f_1}, ..., \mathrm{f_{max}} - 1, ..., \mathrm{f_4}) \tag{31}$$

Step 5: Re-scale $\mathbf{y}'$ by

$$\mathbf{y} = \frac{step}{\sqrt{2}} \times \mathbf{y}' \tag{32}$$

$\mathbf{y}$ is the closest D_4 lattice point.

E_8 lattice

E_8 lattice is obtained by extended Hamming code of length 8. The generation matrix is

$$U = \begin{bmatrix} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{u_3} \\ \mathbf{u_4} \\ \mathbf{u_5} \\ \mathbf{u_6} \\ \mathbf{u_7} \\ \mathbf{u_8} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{33}$$

E_8 lattice quantization algorithm is the following.

Step 1: Scale input vector x by

$$\mathbf{x}' = \frac{1}{step} \times \mathbf{x} \tag{34}$$

Step 2: Round $\mathbf{x}'$ both to the nearest integer and to the next nearest integer.

$$\mathbf{f(x)} = \mathrm{round}(\mathbf{x}') \tag{35}$$

$$f_{max} = \left(f_i \mid \max_i(|x_i - f_i|) \right) \tag{36}$$

if $(x_{max} - f_{max}) \geq 0$,

$$\mathbf{g(x)} = (\mathrm{f_1}, ..., \mathrm{f_{max}} + 1, ..., \mathrm{f_8}) \tag{37}$$

else,

$$\mathbf{g(x)} = (\mathrm{f_1}, ..., \mathrm{f_{max}} - 1, ..., \mathrm{f_8}) \tag{38}$$

Step 3: Take $\mathbf{f(x)}$ or $\mathbf{g(x)}$, whichever has an even sum of components, as the lattice point $\mathbf{y'_0}$.

if $\sum_{i=1}^{8} \mathrm{f_i} = even$,

$$\mathbf{y'_0} = \mathbf{f(x)} \tag{39}$$

else,

$$\mathbf{y'_0} = \mathbf{g(x)} \tag{40}$$

Step 4: Shift $\mathbf{x}'$ by

$$\mathbf{x}' - \frac{\mathbf{1}}{\mathbf{2}} = \mathbf{x}' - \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right) \tag{41}$$

Step 5: Round $(\mathbf{x}' - \frac{\mathbf{1}}{\mathbf{2}})$ to integer both the correct way and the wrong way.

$$\mathbf{f}(\mathbf{x} - \frac{\mathbf{1}}{\mathbf{2}}) = \mathrm{round}(\mathbf{x}' - \frac{\mathbf{1}}{\mathbf{2}}) \tag{42}$$

$$\mathbf{f}(\mathbf{x} - \frac{\mathbf{1}}{\mathbf{2}}) = (\mathrm{f}_1, \mathrm{f}_2, \mathrm{f}_3, \mathrm{f}_4, \mathrm{f}_5, \mathrm{f}_6, \mathrm{f}_7, \mathrm{f}_8) \tag{43}$$

$$f_{max} = (f_i \mid \max_i(|x_i - f_i|)) \tag{44}$$

if $(x_{max} - f_{max}) \geq 0$,

$$\mathbf{g}(\mathbf{x} - \frac{\mathbf{1}}{\mathbf{2}}) = (\mathrm{f}_1, ..., \mathrm{f}_{\mathrm{max}} + 1, ..., \mathrm{f}_8) \tag{45}$$

else,

$$\mathbf{g}(\mathbf{x} - \frac{\mathbf{1}}{\mathbf{2}}) = (\mathrm{f}_1, ..., \mathrm{f}_{\mathrm{max}} - 1, ..., \mathrm{f}_8) \tag{46}$$

Step 6: Take $\mathbf{f}(\mathbf{x} - \frac{\mathbf{1}}{\mathbf{2}})$ or $\mathbf{g}(\mathbf{x} - \frac{\mathbf{1}}{\mathbf{2}})$, whichever has an even sum of components, as the lattice point $\mathbf{y}_1(\mathbf{x} - \frac{\mathbf{1}}{\mathbf{2}})$.

Step 7: Shift $\mathbf{y}_1(\mathbf{x} - \frac{\mathbf{1}}{\mathbf{2}})$ by

$$\mathbf{y}'_1 = \mathbf{y}_1(\mathbf{x} - \frac{\mathbf{1}}{\mathbf{2}}) + \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right) \tag{47}$$

Step 8: Choose $\mathbf{y}'_0$ or $\mathbf{y}'_1$, whichever is closer to $\mathbf{x}$, as the closest lattice point $\mathbf{y}'$.

Step 9: Re-scale $\mathbf{y}'$ by

$$\mathbf{y} = step \times \mathbf{y}' \tag{48}$$

$\mathbf{y}$ is the closest E_8 lattice point.

The orthogonal lattice Z^n is given by simply rounding the input vector to integers.

References

[1] R. Forchheimer and T. Kronander. *Image Coding - From Waveforms to Animation.* IEEE Trans. A.S.S.P, Vol.37, No.12, pp.2008-2023, Dec. '89.

[2] A. N. Netravali and B. G. Haskell. *Digital Pictures.* Plenum Press, '88.

[3] N. S. Jayant, and P. Noll. *Digital Coding of Waveforms.* Prentice-Hall, '84.

[4] W. K. Pratt. *Digital Image Processing.* Wiley-Interscience Pub., '78

[5] R. E. Crochiere, S. A. Webber, and J. L. Flanagan. *Digital Coding of Speech in Sub-bands.* Bell Syst. Tech. J., Vol.55, pp.1069-1085, Oct. 1976.

[6] P. H. Westerink, D. E. Boekee, J. Biemond, and J. W. Woods. *Subband Coding of Images Using Vector Quantization.* IEEE Trans. Comm., Vol.36, No.6, '88.

[7] P. J. Burt, and E. H. Adelson. *The Laplacian Pyramid as Compact Image Code.* IEEE Trans. Comm., Vol.COM-31, No.4, '83.

[8] G. Karlsson and M. Vetterli. *Subband Coding of Video for Packet Networks.* Optical Engineering, Vol.27, No.7, pp.574-586, July '88.

[9] J. C. Darragh. *Subband and Transform Coding of Images.* Dissertation UCLA '89.

[10] R. Gray. *Vector Quantization.* IEEE ASSP Magazine pp.4-29, '84.

[11] J. Makhoul, S. Roucos, and H. Gist. *Vector Quantization in Speech Coding.* Proc. IEEE, Vol.73, No.11, '85.

[12] N. M. Nasrabadi, and R. A. King. *Image Coding Using Vector Quantization: A Review.* IEEE Trans. Comm., Vol.36, No.8, '88.

[13] A. Gersho. *On the Structure of Vector Quantization.* IEEE Trans. on Inf. Theory, Vol. IT-28, No. 2, pp.157-166, '82.

[14] B. Ramamurthi and A. Gersho. *Classified Vector Quantization of Images.* IEEE Transactions on Communications, vol.COM.-34, no.11, pp.1105 - 1115, Nov. '86.

[15] Y. Linde, A. Buzo, R. M. Gray. *An Algorithm for Vector Quantizer Design.* IEEE Trans. on Comm., Vol. COM-28, No. 1, pp.84-94, Jan. '80.

[16] P. Romano Jr. *Vector Quantization for Spatiotemporal Sub-band Coding.* MS Thesis MIT, '89.

[17] W. J. Butera. *Multiscale Coding of Images.* MS Thesis MIT, '88.

[18] J. L. Bentley. *Multidimensional Binary Search Trees Used for Associative Searching.* Comm. of the ACM, Vol.18, N0.9, pp.509-517, Sept. '75.

[19] J. H. Friedman, J. L. Bentley, and R. AS. Finkel. *An Algorithm for Finding Best Matches in Logarithmic Expected Time.* ACM Trans. om Mathematical Software, Vol.3, N0.3, pp.209-226, Sept. '77.

[20] A. Gersho. *Asymptotically Optimum Block Quantization.* IEEE Trans. on Inf. Theory, Vol. IT-25, No. 4, pp.373-380, July '82.

[21] J. H. Conway and N. J. A. Sloane *Voronoi Regions of Lattices, Second Moments of Polytopes, and Quantization.* IEEE Trans. on Information Theory, vol.IT-28, No.2, pp.211 - 226, March '82.

[22] N. J. A. Sloane. *Tables of Sphere Packings and Spherical Codes.* IEEE Trans. on Information Theory, vol.IT-27, No.3, pp.327 - 338, May '81.

[23] J. H. Conway and N.J.A. Sloane. *A Fast Encoding Method for Lattice Codes and Quantizers.* IEEE Transactions on Information Theory, vol.IT-29, no.6, pp.820 - 824, Nov. '83.

[24] J. H. Conway and J. A. Sloane. *Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes* IEEE Trans. on Information Theory, vol.IT-28, pp.227 - 232, March '82.

[25] P. A. Chou, T. Lookabaugh, and R. M. Gray. *Entropy-Constrained Vector Quantization.* IEEE Trans. ASSP, Vol. 37, No. 1, '89.

[26] P. A. Chou *Application of Entropy-Constrained Vector Quantization to Waveform Coding of Images.* SPIE Vol. 1199 Visual Comm. and Image Proc. '89.

[27] D. Esteban and C. Galand. *Applications of Quadrature Mirror Filters to Split Band Voice Coding Systems.* Proc. ICASSP '77, pp.191-195.

[28] J. D. Johnston. *A Filter Family Designed for Use in Quadrature Mirror Filter Banks.* IEEE proc. ICASSP '80, pp.291-294.

[29] E. P. Simoncelli. *Orthogonal Sub-band Image Transforms.* MS Thesis MIT, '88.

[30] R. J. Clarke. *Transform Coding of Images.* Academic Press, '85.

Table 1: Example of 9-tap quadrature mirror filter coefficients

coefficient	LPF	HPF
C_0	0.5645751	0.5645751
C_{-1}, C_1	0.2927051	-0.2927051
C_{-2}, C_2	-0.05224239	-0.05224239
C_{-3}, C_3	-0.04270508	0.04270508
C_{-4}, C_4	0.01995484	0.01995484

List of Figure Captions

Fig.1: Subband decomposition and reconstruction.

Fig.2: Frequency response of 9-tap quadrature mirror filter bank.

Fig.3: 4-level Subband pyramid decomposition of image Lenna, each of the subbands has been normalized.

Fig.4: Data distribution of the low frequency subband on the fourth pyramid level.

Fig.5: Data distribution of the horizontally high subband on the fourth pyramid level.

Fig.6: Illustration of (a) 2-D vector space with distribution density contour map, (b) quantized by full-search LBG VQ, (c) quantized by entropy-constrained VQ, (d) quantized by k-D tree VQ, (e) quantized by Hadamard k-D tree VQ, (f) quantized by ternary k-D tree VQ, (g) quantized by hexagonal lattice VQ, (h) quantized by orthogonal lattice VQ with reproduction vectors.

Fig.7: $R(D)$ curves comparing uniform memoryless quantizer (SCALAR), LBG full search fixed length VQ (LBG8), entropy-constrained VQ (ECVQ8), k-D tree VQ (KD8), Hadamard k-D tree VQ (HKD8), E_8-lattice VQ (E8-LTC), and Shannon lower bound of $R(D)$ curve (SLB) for a first-order Gauss-Markov source with correlation coefficient $r = 0.95$.

Fig.8: $R(D)$ curves comparing uniform memoryless quantizer (SCALAR), LBG full search fixed length VQ (LBG8), entropy-constrained VQ (ECVQ8), k-D tree VQ (KD8), Hadamard k-D tree VQ (HKD8), Tri-kD tree VQ (Tri-KD8), E_8-lattice VQ (E8-LTC), and Shannon lower bound of $R(D)$ curve (SLB) for Laplacian distribution without memory.

Fig.9: Training pictures.

Fig.10: $R(D)$ curves comparing uniform memoryless quantizer (SCALAR), LBG full search fixed length VQ (LBG8), entropy-constrained VQ (ECVQ8), k-D tree VQ (KD8), Hadamard k-D tree VQ (HKD8), and E_8-lattice VQ (E8-LTC) for 512 × 512 Lenna.

Fig.11: $R(D)$ curves comparing uniform memoryless quantizer (SCALAR), LBG full search VQ (LBG8), entropy-constrained VQ (ECVQ8), k-D tree VQ (KD8), Hadamard k-D tree VQ (HKD8), tri-kD tree VQ (Tri-KD8), Z^8-lattice VQ (UNIF8), and E_8-lattice VQ (E8-LTC) for the horizontally high frequency subband on the first pyramid level (H0).

Fig.12: $R(D)$ curves comparing uniform memoryless quantizer (SCALAR), LBG full search VQ (LBG4), entropy-constrained VQ (ECVQ4), k-D tree VQ (KD4), Tri-kD tree VQ (Tri-KD4), Z^4-lattice VQ (UNIF4), and D_4-lattice VQ (D4-LTC) for the vertically high frequency subband on the second pyramid level (V1).

Fig.13: Magnified $R(D)$ curves comparing uniform memoryless quantizer (SCALAR), LBG full search VQ (LBG2), entropy-constrained VQ (ECVQ2), k-D tree VQ (KD2), Hadamard k-D tree VQ (HKD2), A_2-lattice VQ (A2-LTC), and Z^4-lattice VQ (UNIF4) for the low frequency subband on the fourth pyramid level (L3).

Fig.14: Lenna original 512 × 512 pixels, 8 bits/pixel.

Fig.15: Lenna coded by subband lattice quantizer at 0.136bpp, peak SNR = 30.909dB.

Fig.16: Lenna coded directly by entropy constrained vector quantizer without subband pyramid de-

composition at 0.137bpp, peak SNR = 19.194dB.

Fig.17: Bit allocation among subbands over distortion contribution of each subband, bit rate is normalized by the whole image size.

Fig.18: Spatio-temporal subband pyramid scheme for moving image, high frequency components are optimally pruned.

Fig.19: Original first frame of Alley 480 × 640 pixels, 8 bits/pixel.

Fig.20: First frame of Alley encoded by spatio-temporal subband pyramid and orthogonal lattice quantizers at 1.08Mbps, peak SNR = 39.4dB.

Fig.21: Peak SNR fluctuation of encoded Alley.3 sequences according to the frame number.

Biography

Takanori Senoo received the B.Sc. degree and M.Sc. degree in Electrical Engineering from Osaka University, Japan, in 1973 and 1975, respectively. Since 1975, he has been with Matsushita Electric Industrial Co. Ltd., where he was involved in the development of digital audio equipment. He has been working on error correction and digital signal processing. Since 1988, he has been a research affiliate of The Media Laboratory at Massachusetts Institute of Technology, his research interests include image coding, audio coding, information theory, and coding theory. Mr. Senoo is a member of SMPTE, SPIE, AES, ITE of Japan, IEICE of Japan, and ASJ in Japan.

Biographical Sketch

Bernd Girod
MIT Media Laboratory
E15-392, Cambridge, MA 02139

Last update: April 24, 1990

Bernd Girod received an M.S. degree in Electrical Engineering from Georgia Institute of Technology in 1980, and the Doctoral degree "with highest honours" from University of Hannover, West Germany, in 1987.

He has worked for Siemens Central Research Laboratories in Munich in 1981 on speech synthesis and speech recognition. Until 1988 he was member of the research staff of the Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung of the University of Hannover working on moving image coding, human visual perception, and information theory. Since 1988, Dr. Girod has been with M.I.T., first as a Visiting Scientist with the Research Laboratory of Electronics, and now as an Assistant Professor of Media Technology at the Media Laboratory. His research interests include multidimensional signal processing, information theory, video signal coding, human and machine vision, sensing and modeling of three-dimensional scenes. For several years, he has been serving as a consultant to companies and government agencies.

Dr. Girod is the principal author of over 30 papers in the field. He is a member of the IEEE and of the German Informationstechnische Gesellschaft (ITG) des VDE that honoured him with their 1988 Best Paper Award. He was the banquet speaker of last year's IEEE International Conference on Computer Design. He has chaired the recent SPIE conference on "Sensing and Reconstruction of Three-Dimensional Objects and Scenes," and he is a guest editor for the upcoming special issue "64 kbit/s Coding of Moving Video" of the EURASIP journal "Image Communication."

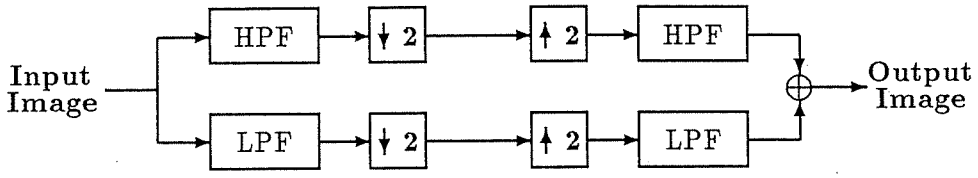

Figure 1: Subband decomposition and reconstruction.

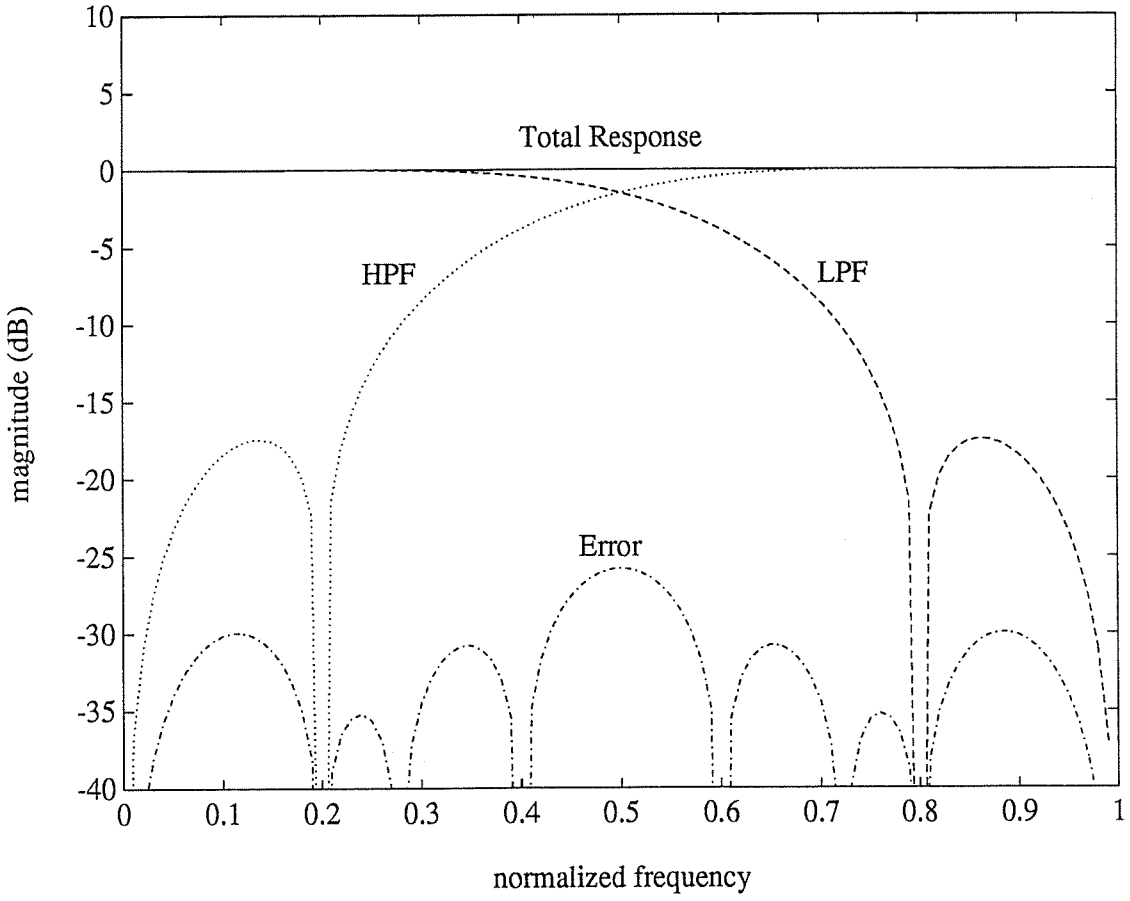

Figure 2: Frequency response of 9-tap quadrature mirror filter bank.

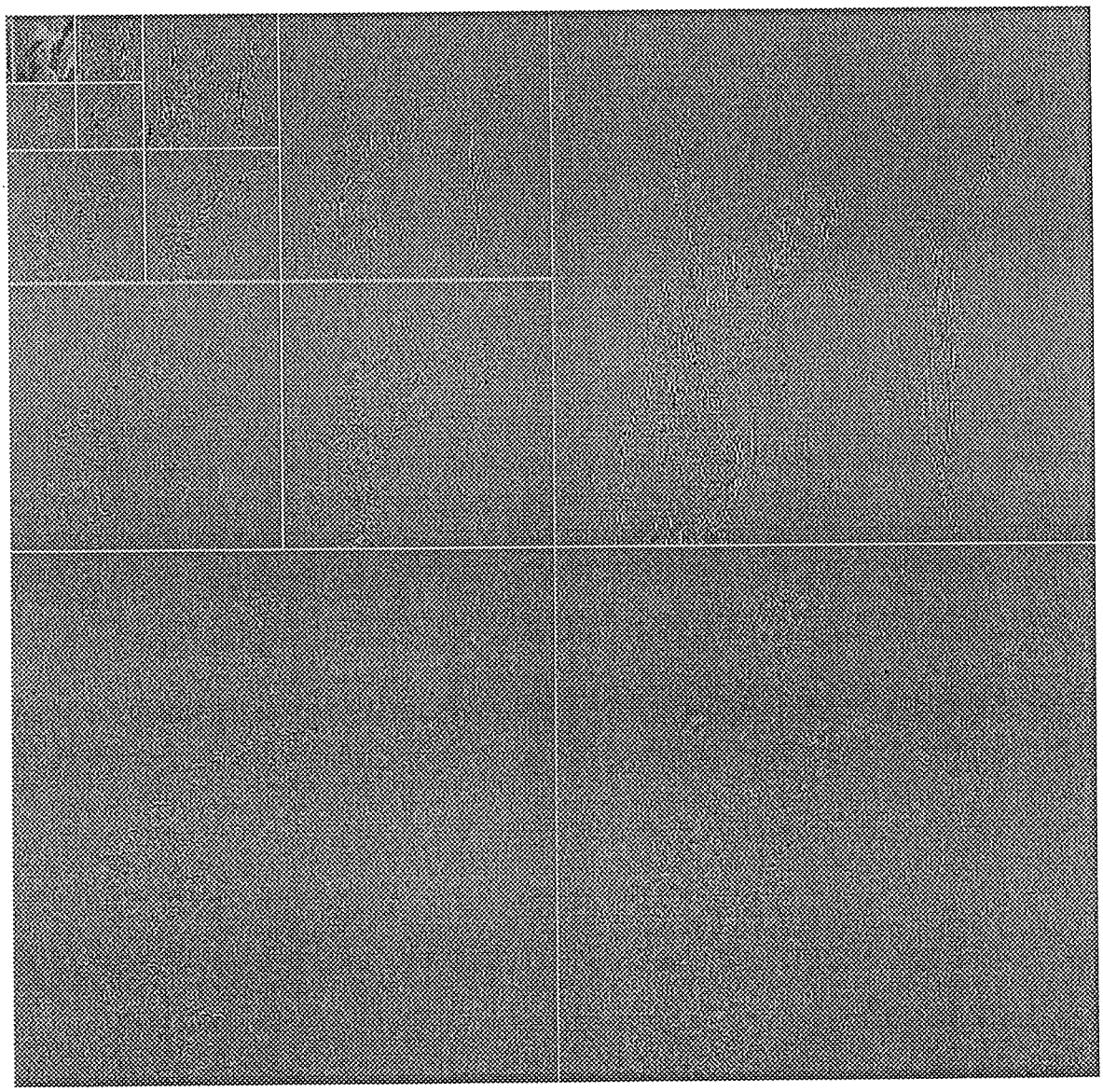

Figure 3: Subband pyramid decomposition of image Lenna, each of the subbands has been normalized.

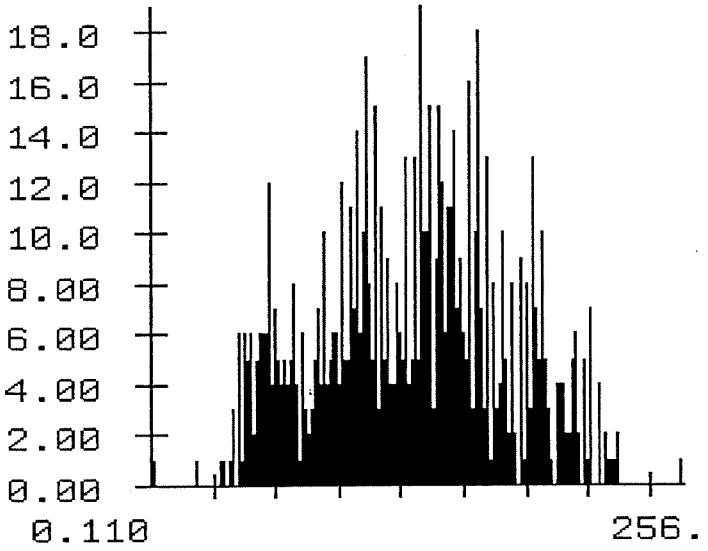

Figure 4: Data distribution of the low frequency subband on the fourth pyramid level.

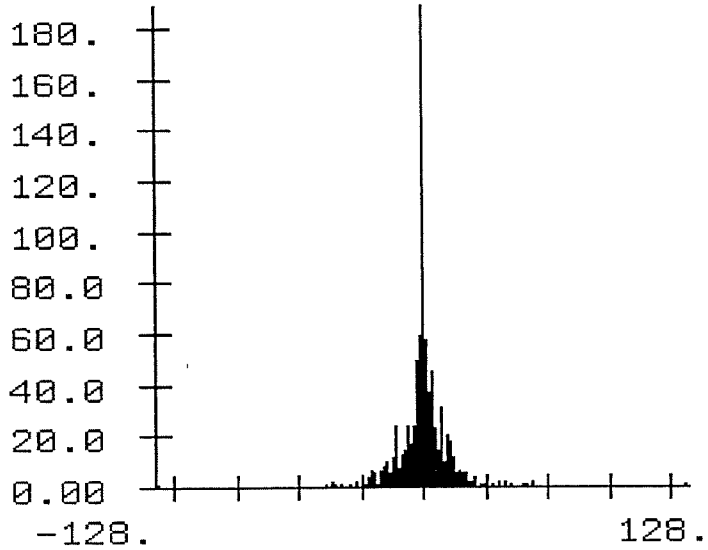

Figure 5: Data distribution of the horizontally high subband on the fourth pyramid level.

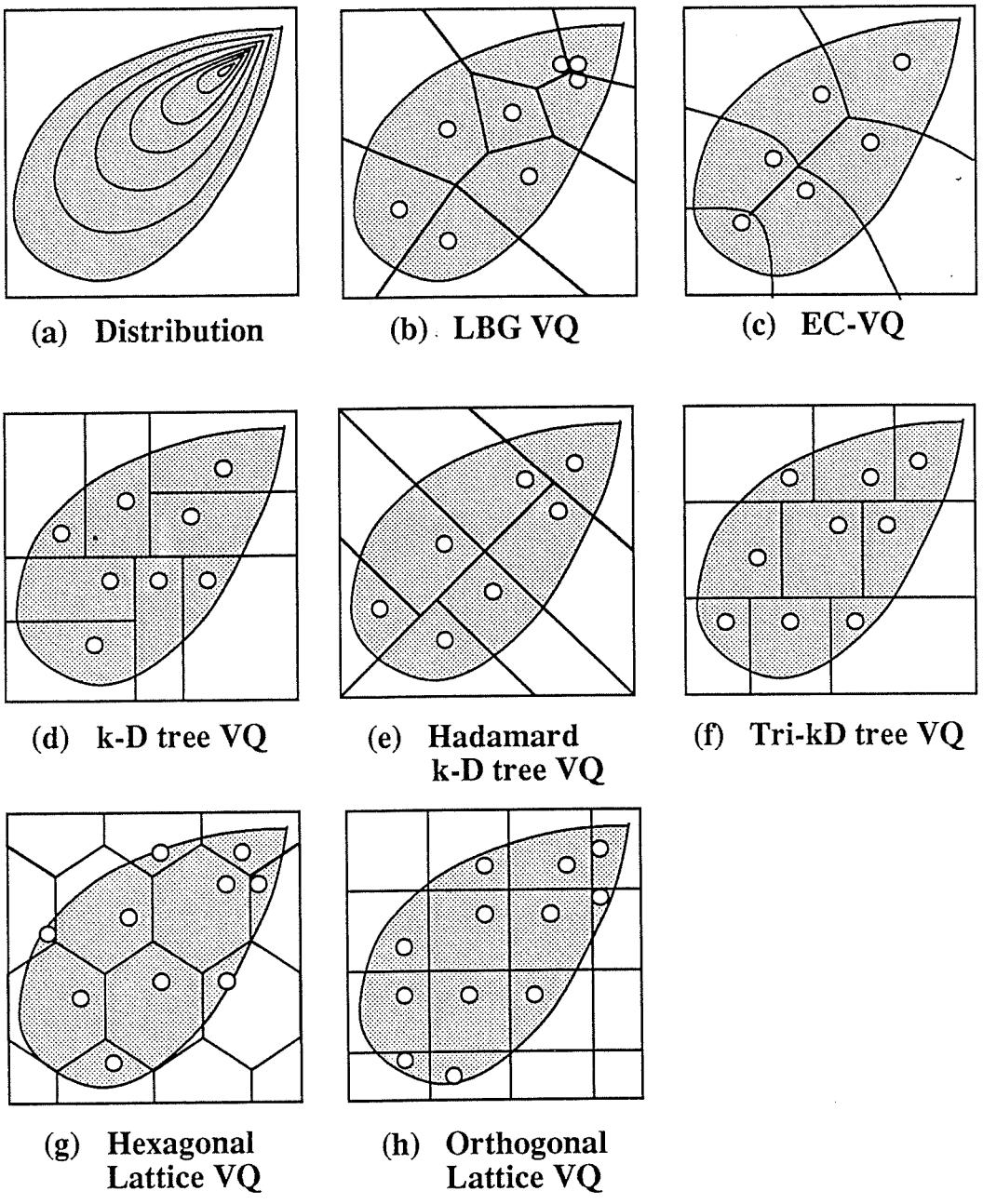

Figure 6: Illustration of (a) 2-D vector space with distribution density contour map, (b) quantized by full-search LBG VQ, (c) quantized by entropy-constrained VQ, (d) quantized by k-D tree VQ, (e) quantized by Hadamard k-D tree VQ, (f) quantized by ternary k-D tree VQ, (g) quantized by hexagonal lattice VQ, (h) quantized by orthogonal lattice VQ with reproduction vectors.

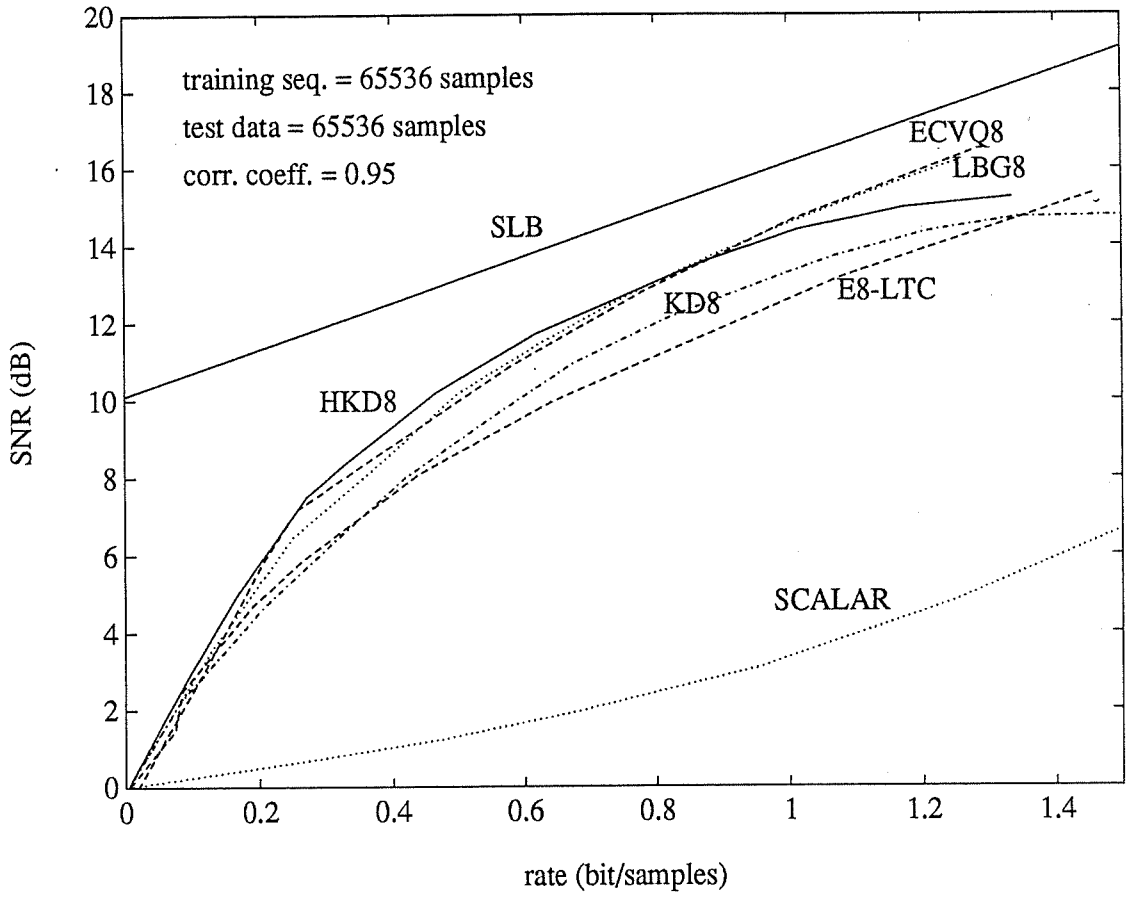

Figure 7: $R(D)$ curves comparing uniform memoryless quantizer (SCALAR), LBG full search fixed length coding (LBG8), entropy-constrained VQ (ECVQ8), k-D tree (KD8), Hadamard k-D tree (HKD8), E_8-lattice (E8-LTC), and Shannon's lower bound of $R(D)$ curve (SLB) for a first-order Gauss-Markov source with correlation coefficient $r = 0.95$.

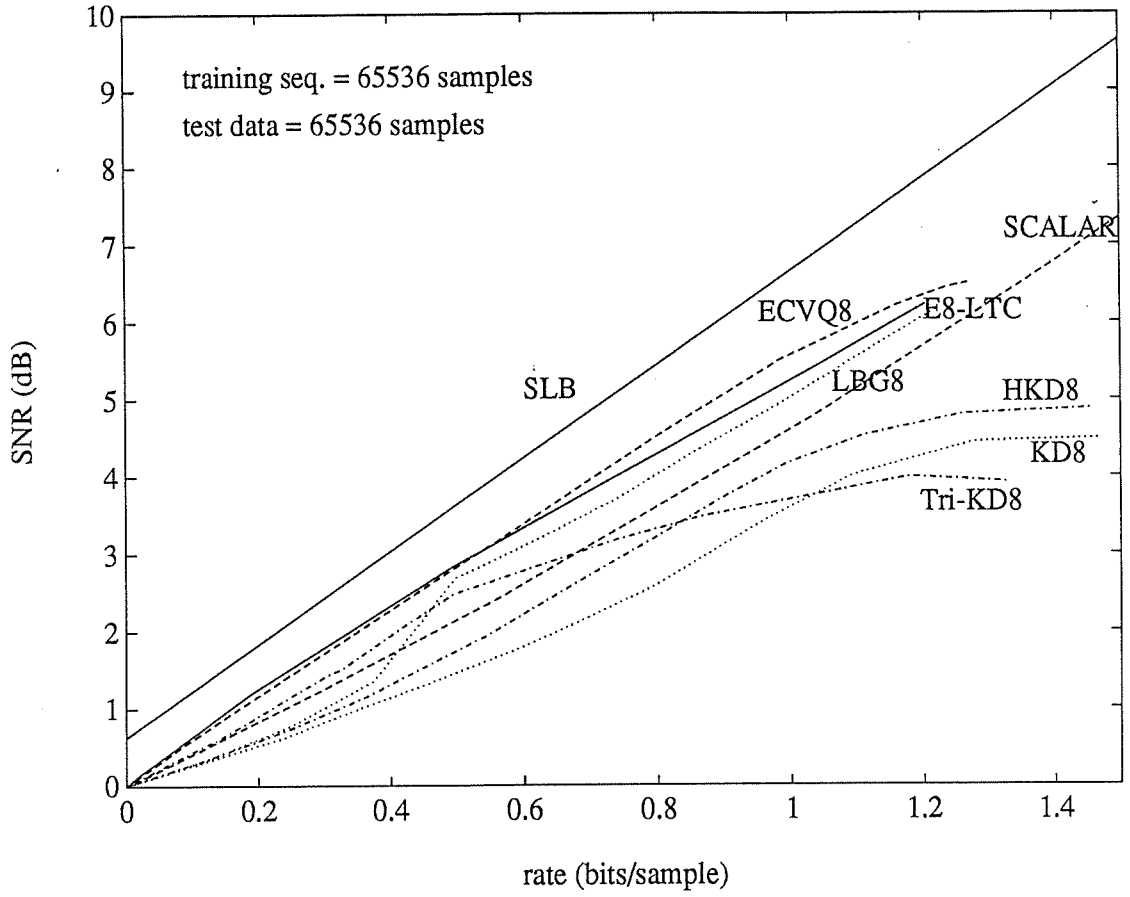

Figure 8: $R(D)$ curves comparing uniform memoryless quantizer (SCALAR), LBG full search fixed length (LBG8), entropy-constrained VQ (ECVQ8), k-D tree (KD8), Hadamard k-D tree (HKD8), Tri-kD tree (Tri-KD8), E_8-lattice (E8-LTC), and Shannon's lower bound of $R(D)$ curve (SLB) for Laplacian distribution without memory.

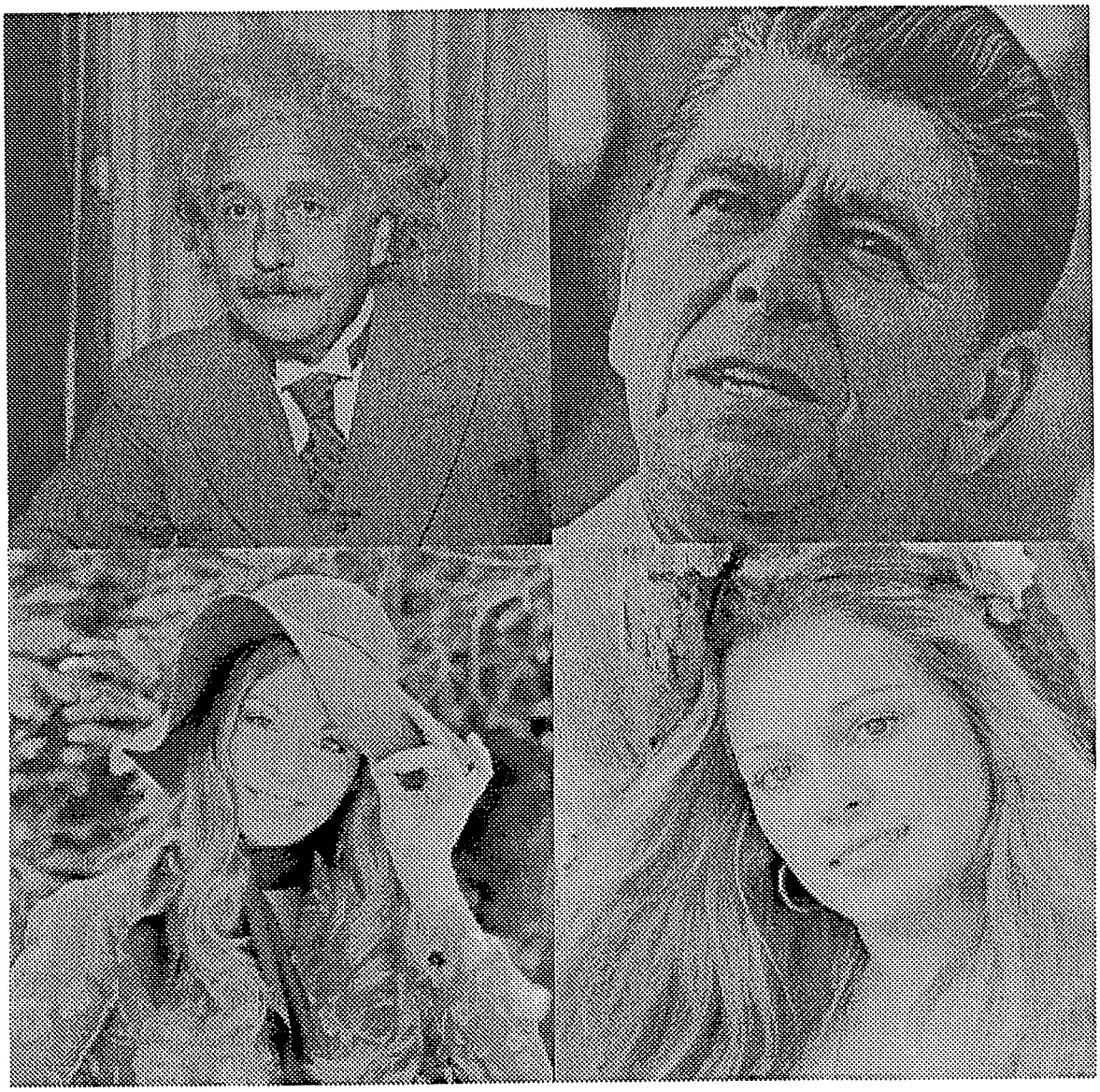

Figure 9: Training pictures.

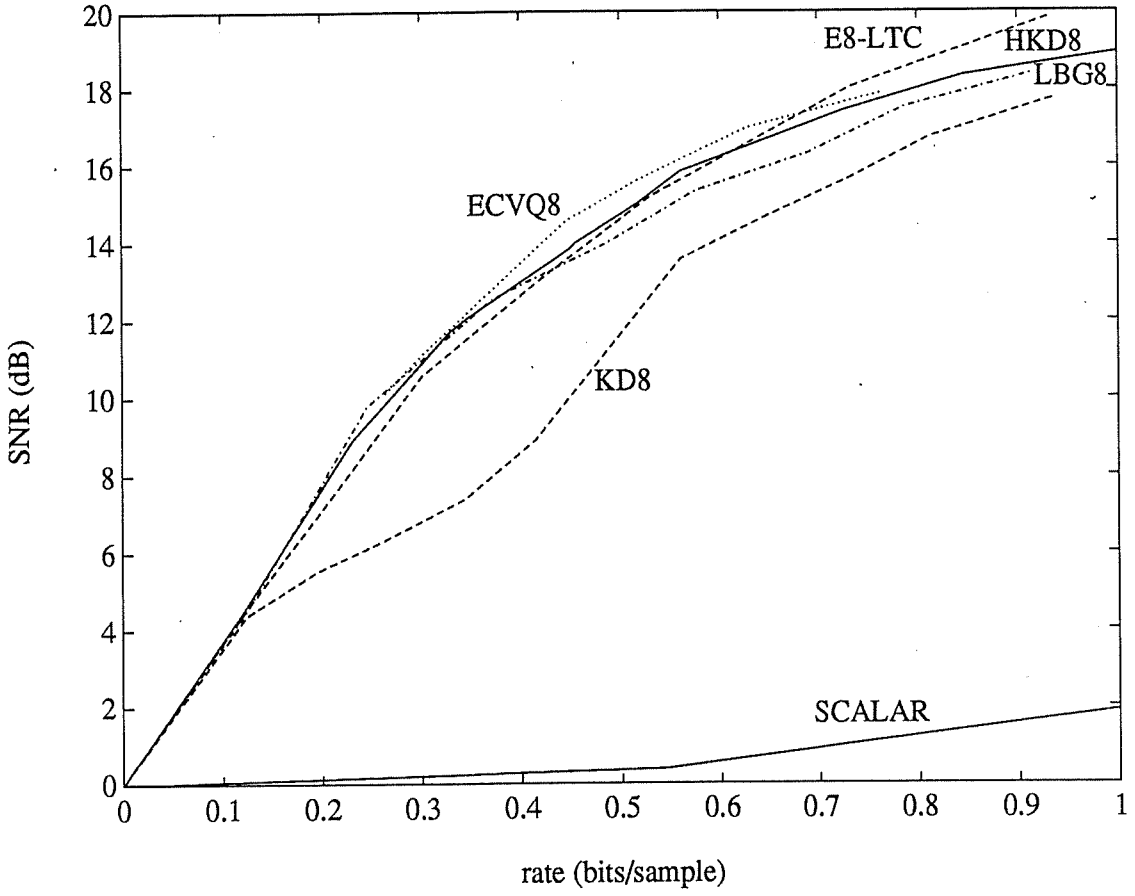

Figure 10: $R(D)$ curves comparing uniform memoryless quantizer (SCALAR), LBG full search fixed length (LBG8), entropy-constrained VQ (ECVQ8), k-D tree (KD8), Hadamard k-D tree (HKD8), and E_8-lattice (E8-LTC) for 512×512 Lenna.

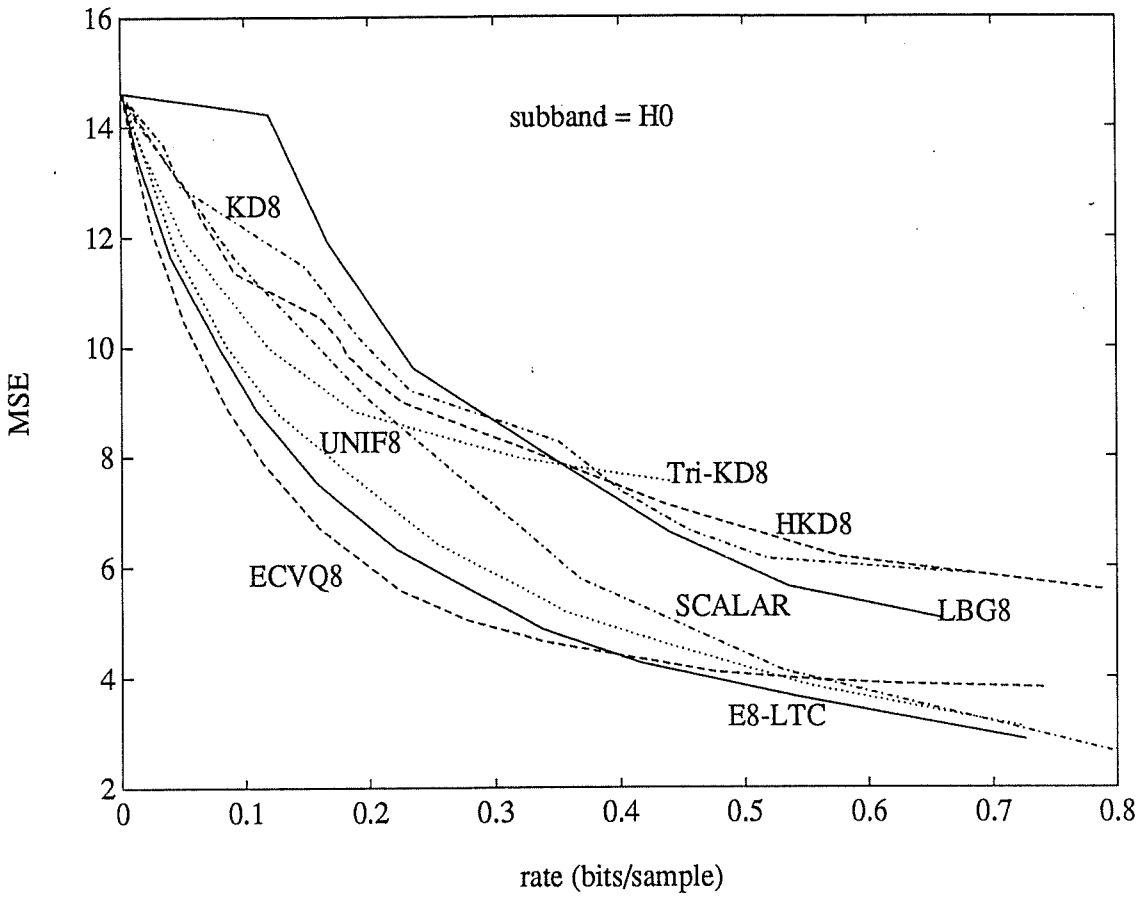

Figure 11: $R(D)$ curves comparing uniform memoryless quantizer (SCALAR), LBG full search (LBG8), entropy-constrained VQ (ECVQ8), k-D tree (KD8), Hadamard k-D tree (HKD8), tri-kD tree (Tri-KD8), Z^8-lattice (UNIF8), and E_8-lattice (E8-LTC) for the horizontally high frequency subband on the first pyramid level (H0).

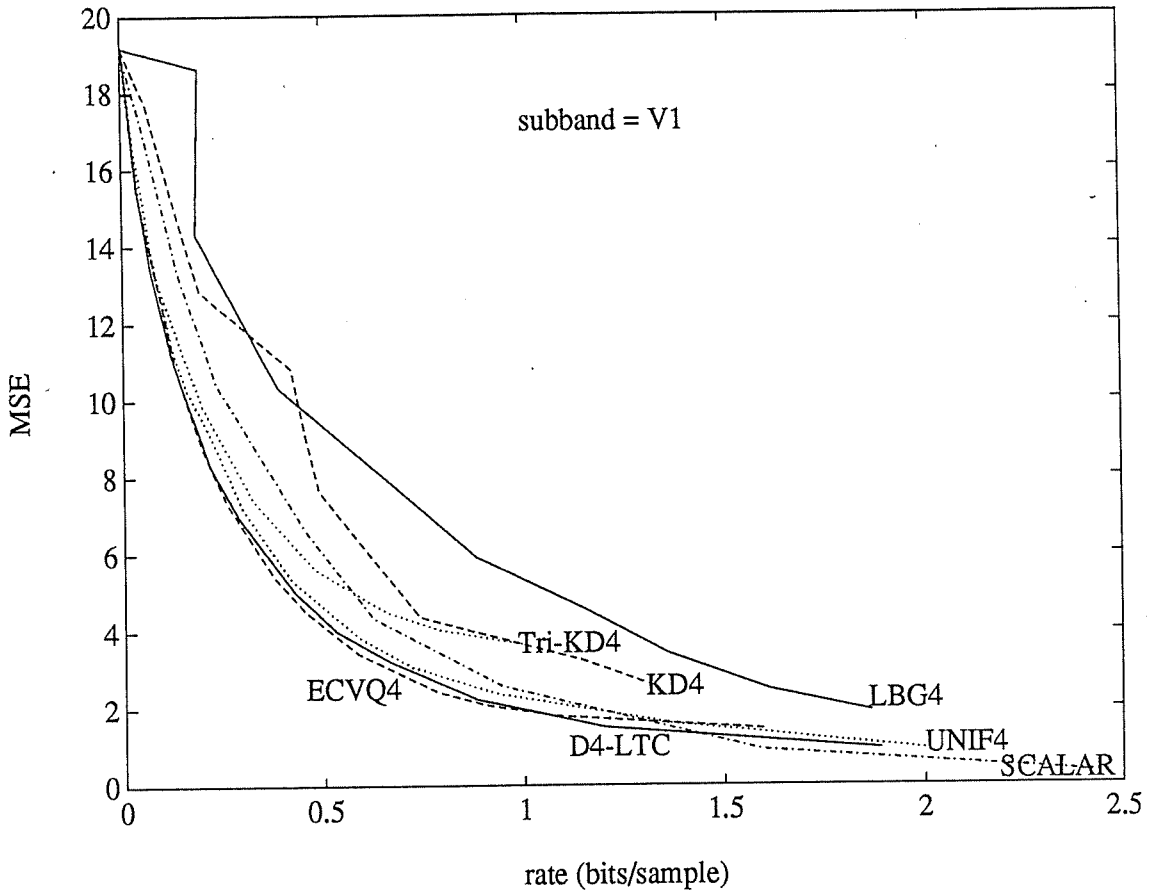

Figure 12: $R(D)$ curves comparing uniform memoryless quantizer (SCALAR), LBG full search (LBG4), entropy-constrained VQ (ECVQ4), k-D tree (KD4), Tri-kD tree (Tri-KD4), Z^4-lattice (UNIF4), and D_4-lattice (D4-LTC) for the vertically high frequency subband on the second pyramid level (V1).

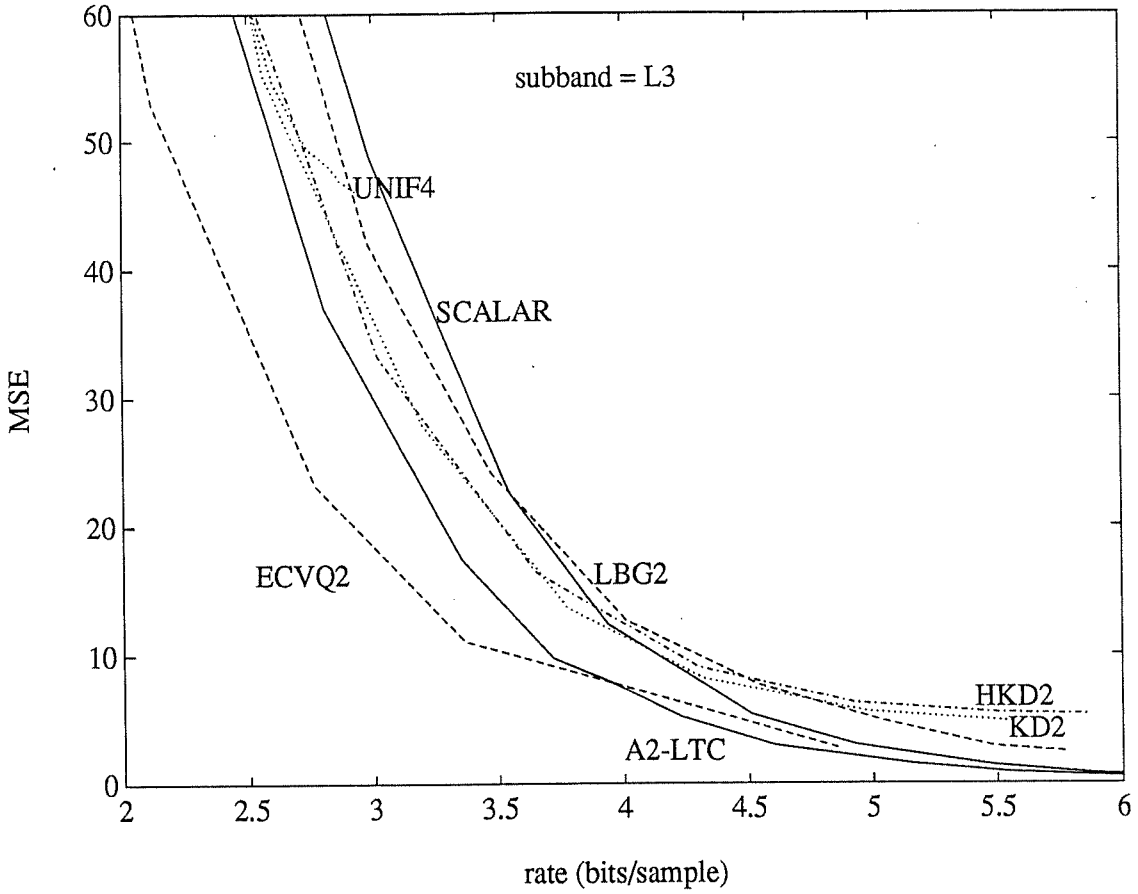

Figure 13: Magnified $R(D)$ curves comparing uniform memoryless quantizer (SCALAR), LBG full search (LBG2), entropy-constrained VQ (ECVQ2), k-D tree (KD2), Hadamard k-D tree (HKD2), A_2-lattice (A2-LTC), and Z^4-lattice (UNIF4) for the low frequency subband on the fourth pyramid level (L3).

Figure 14: Lenna original 512 × 512, 8 bits/pixel.

Figure 15: Lenna coded by subband lattice quantizer at 0.136bpp, peak SNR = 30.909dB.

Figure 16: Lenna coded directly without subband pyramid decomposition by entropy constrained vector quantizer at 0.137bpp, peak SNR = 19.194dB.

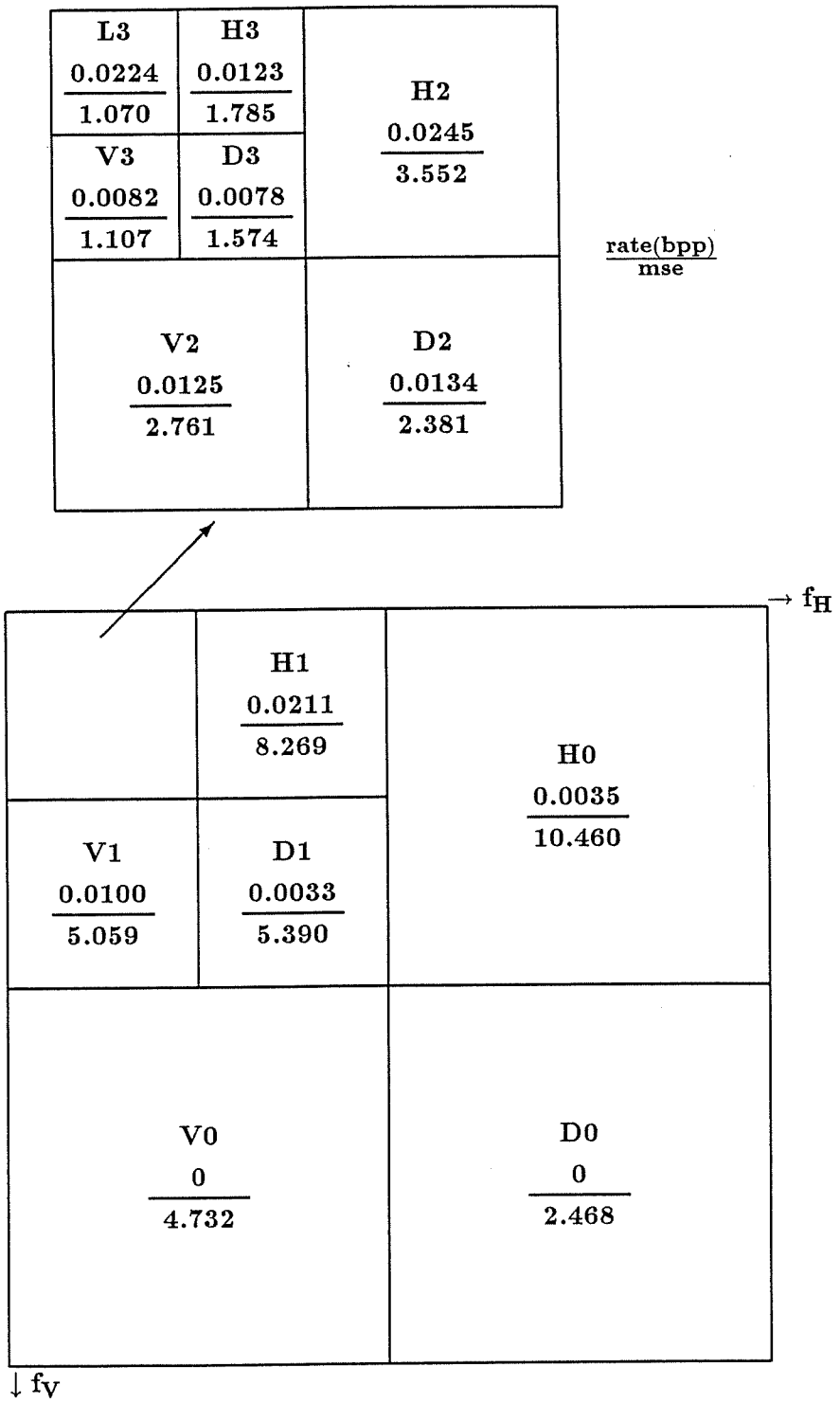

Figure 17: Bit allocation among subbands over distortion contribution, bit rate is normalized by the whole image size.

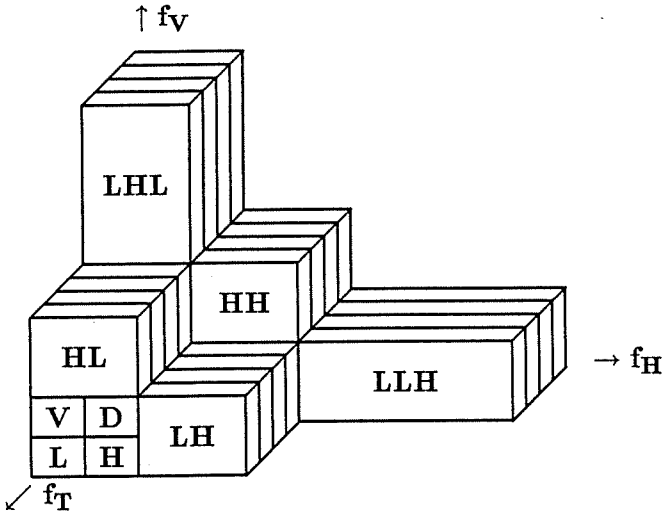

Figure 18: Spatio-temporal subband scheme for moving image

Figure 19: Original first frame of Alley 480 × 640 pixels, 8 bits/pixel.

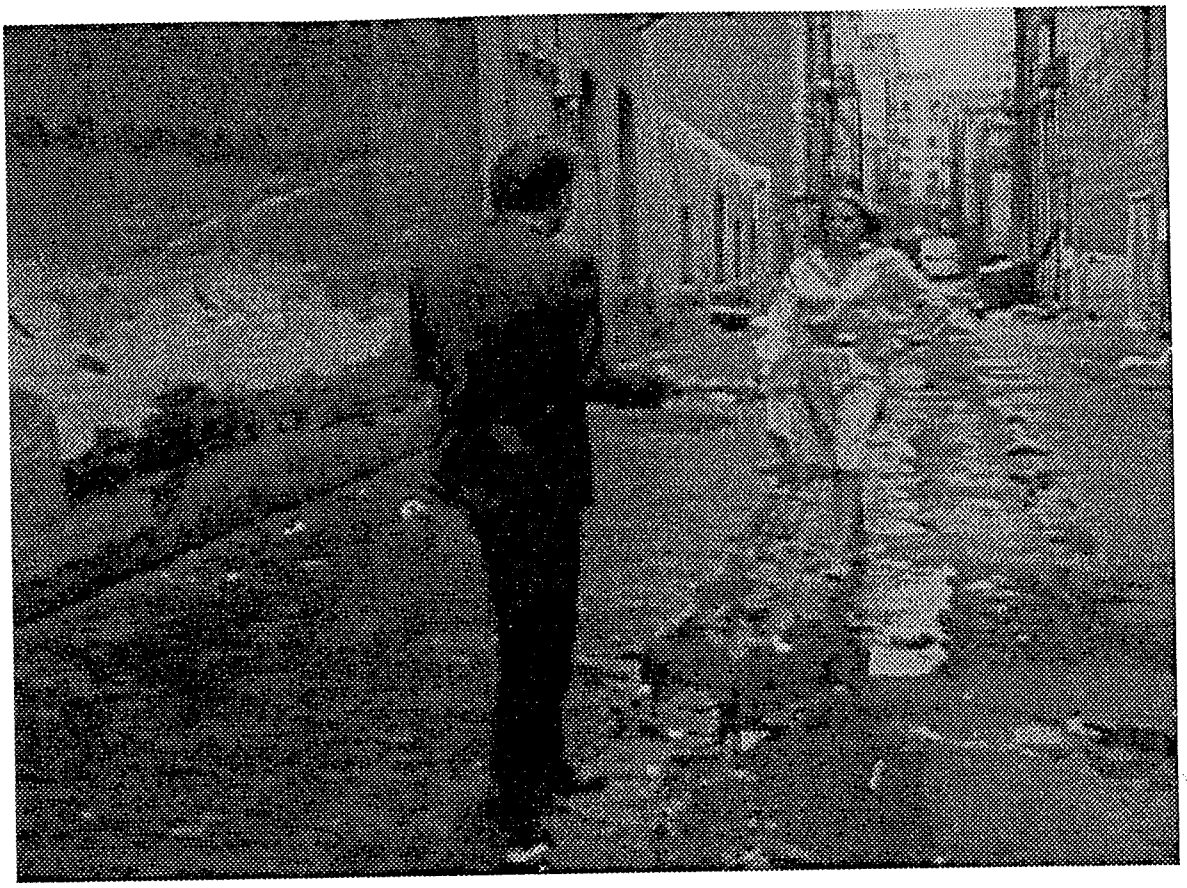

Figure 20: First frame of Alley encoded by subband orthogonal lattice quantizer at 1.08Mbps, peak SNR = 39.4dB.

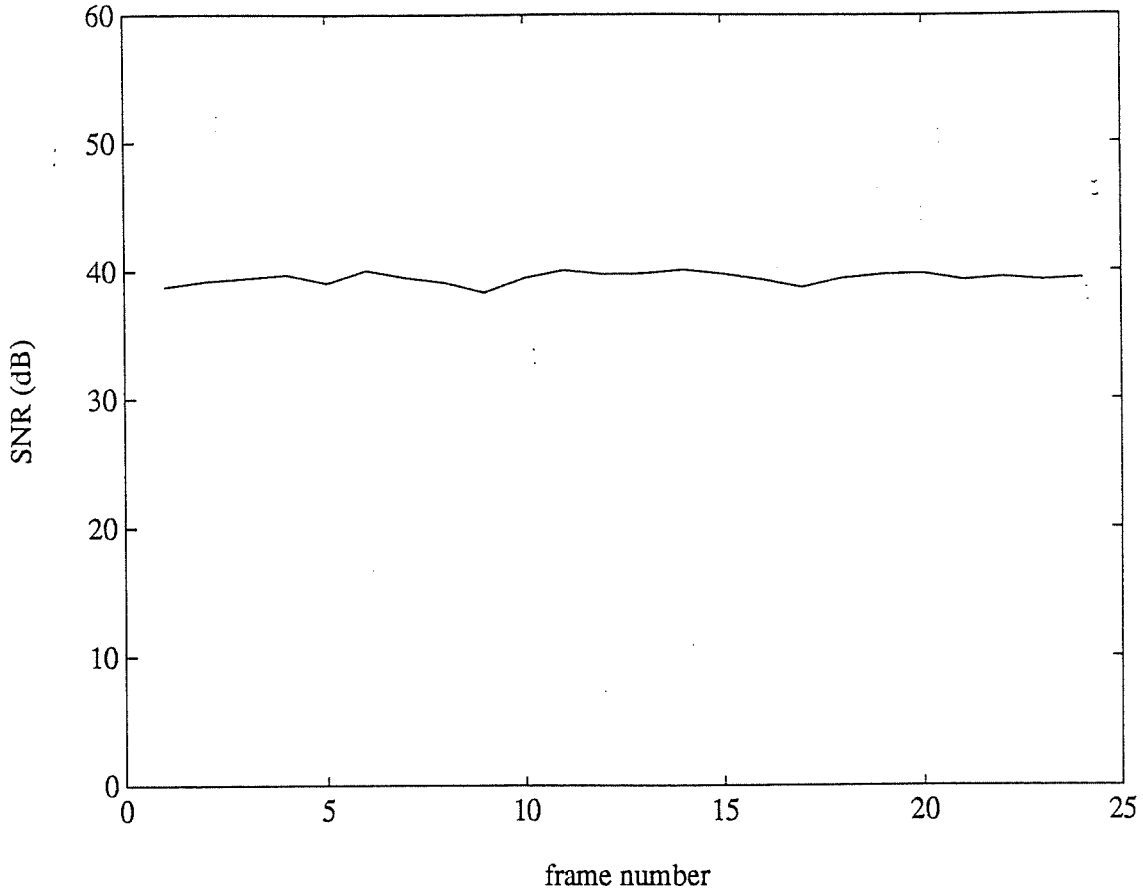

Figure 21: Peak SNR fluctuation of encoded Alley.3 sequences according to the frame number.