OPTIMUM TRANSFORM CODING OF IMAGERY

Glen P. Abousleman

Motorola, Systems Solutions Group
8201 E. McDowell Road, MD H1176
Scottsdale, AZ 85257
glen1@kuma.geg.mot.com

ABSTRACT

A system is presented for transform coding of imagery. Specifically, the system uses the 2-D discrete cosine transform (DCT) in conjunction with adaptive classification, entropy-constrained trellis-coded quantization, optimal rate allocation, and adaptive arithmetic encoding. Adaptive classification, side rate reduction, and rate allocation strategies are discussed. Entropy-constrained codebooks are designed using a modified version of the generalized Lloyd algorithm. This entropy-constrained DCT-based system is shown to achieve outstanding coding performance as compared to other DCT-based systems.

1. INTRODUCTION

Recent advances in image compression technology have yielded coders capable of outstanding compression efficiency [1]. This new generation of image coder has been based primarily on subband decomposition. Although subband decomposition enables very efficient decorrelation of the image data prior to quantization, it suffers from two major drawbacks. First, the associated computational complexity is roughly an order of magnitude greater than that of an 8 × 8 DCT (assuming reasonably short length filters). Secondly, various types of imagery which contain significant high-frequency information suffer from "subband artifacts" when encoded at low bit rates. An example of imagery that is not well suited to subband decomposition is synthetic aperture radar (SAR) imagery. It can be been shown that although subband techniques can produce compressed SAR imagery with greater signal-to-noise ratios than DCT-based systems, DCT coding is capable of producing imagery of greater perceptual quality as judged by trained image analysts, in spite of possible blocking artifacts.

The purpose of this paper is to combine DCT decomposition with several advanced technologies used by the latest generation of subband coders. The technologies chosen for this coder are among the best presented in the literature. These technologies include fully-optimized entropy-constrained trellis-coded quantization, maximum-coding-gain adaptive classification, optimal rate allocation, and adaptive arithmetic encoding. These technologies combine to produce a DCT-based coder that exceeds the performance of other DCT systems, and rivals the performance of many subband-based systems. Moreover, the computational complexity of this system is much less than those of most competing systems.

In the coder discussed herein, the image is divided into non-overlapping 8 × 8 blocks and transformed using the DCT. Each block is classified into one of J classes by maximizing the coding gain. All resulting sequences are quantized using entropy-constrained trellis-coded quantization (ECTCQ). Codebooks are optimized for different generalized Gaussian distributions. Codebook design uses a modified version of the generalized Lloyd algorithm in a training-sequence-based iterative scheme. Rate allocation is performed in an optimal fashion by an iterative technique which uses the rate-distortion performance of the various trellis-based quantizers. The quantization indices are losslessly encoded using adaptive arithmetic coding.

2. ENTROPY-CONSTRAINED TRELLIS-CODED QUANTIZATION

Trellis Coded Quantization was developed in [2]. For encoding a memoryless source at R bits per sample, a codebook of size 2^{R+1} is partitioned into four subsets, each containing 2^{R-1} codewords. These subsets are labeled D_0, D_1, D_2 and D_3, and are used as labels for the branches of a suitably chosen trellis. An example is shown in Figure 1 for R = 2.

Sequences of codewords that can be produced by the TCQ system are those that result from "walks" along the trellis from left to right. For example, if beginning in the top left state of the trellis in Figure 1, the first codeword must be chosen from either D_0 or D_2. If a codeword from D_2 is chosen, then we walk along the lower branch (shown with a heavy line) to the second state from the bottom, at which we must choose a codeword from either D_1 or D_3.

Note that at each step in the encoding, the codeword must be chosen from either $A_0 = D_0 \cup D_2$ or $A_1 = D_1 \cup D_3$. Each of these "supersets" contains 2^R codewords and hence, given an initial trellis state, the sequence of selected codewords can be transmitted using one R-bit label for each sample.

Given an input data sequence $x_1, x_2, \ldots$, the best (minimum mean-squared-error) allowable sequence of codewords is determined as follows. For the i^{th} stage in the trellis (corresponding to x_i), the best codeword in the j^{th} subset ($j = 0, 1, 2, 3$), say c_j, is chosen and the associated cost $\rho_j = (x_i - c_j)^2$ is calculated. Each branch in the i^{th} stage of the trellis that is labeled with subset D_j is assigned cost ρ_j. The Viterbi algorithm [3] is then used to find the path through the trellis with the lowest overall cost.

It was shown in [4], that the codeword labels (as described above) can be noiselessly compressed using one variable-length code for each superset. The theoretical encoding rate achievable by this process is the conditional entropy of the codebook C, given

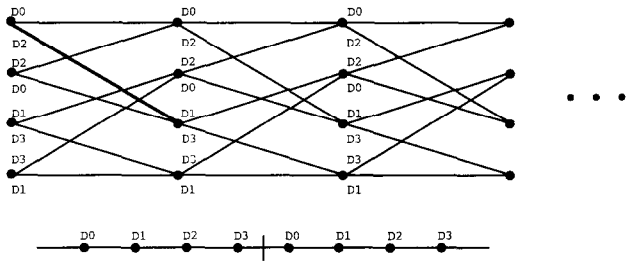

Figure 1: A 4-state trellis with subset labeling and codebook.

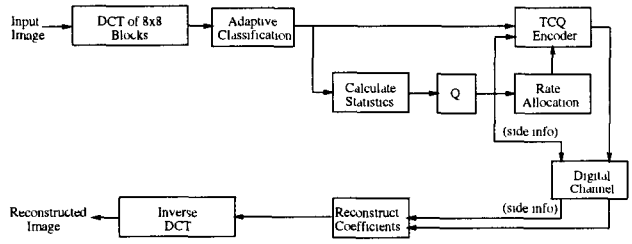

Figure 2: Adaptive transform image coder.

the superset:

$$H(C|A) = -\sum_{i=0}^{1} \sum_{c \in A_i} P(c|A_i)P(A_i)\log_2 P(c|A_i). \quad (1)$$

It can be shown that a 4-state ECTCQ system comes within 0.55 dB of the rate-distortion function for various generalized Gaussian distributions.

In the current system, adaptive arithmetic encoding [5] is used to losslessly encode the quantization indices. A separate probability table is maintained for each superset, and various codebook truncation procedures are implemented at high bit rates.

3. ADAPTIVE TRANSFORM IMAGE CODER

Coding of the image is based on the system presented in [6]. The basic configuration is shown in Figure 2. The image to be coded is divided into non-overlapping 8×8 blocks and transformed using the 2-D DCT. Prior to application of the DCT, each block is assigned to one of J classes by maximizing the coding gain. For each class, DCT coefficients corresponding to the same frequency within each block are grouped into sequences to be encoded using ECTCQ. All DCT coefficients are normalized by subtracting their mean (only the sequences corresponding to the DC transform coefficients have non-zero mean) and dividing by their respective standard deviations. Since each class contains 64 DCT coefficient sequences, the total number of sequences to be encoded is $64J$. The DCT coefficient sequences are assumed to have various generalized Gaussian statistics. Accordingly, codebooks were designed using sample sequences derived from generalized Gaussian pseudo-random number generators. Additionally, rate allocation is performed using the algorithm in [7].

The classification algorithm is similar to that presented in [8]. Consider a source X of length NL divided into N blocks of L consecutive samples, with each block assigned to one of J classes. If the samples from all blocks assigned to class i ($1 \le i \le J$) are grouped into source X_i, the total number of blocks assigned to source X_i is N_i. Let σ_i^2 be the variance of X_i and p_i be the probability that a sample belongs to X_i (i.e., $p_i = N_i/N, 1 \le i \le J$). The algorithm in [8] is a pairwise maximization of coding gain and is repeated here for convenience:

1. Initialize $N_1, N_2, \ldots, N_J$ to satisfy $\sum_{i=1}^{J} N_i = N, N_i > 0$ for $1 \le i \le J$. Let $j = 1$ and $\underline{N}_{prev} = [N_1, N_2, \ldots, N_J]'$.
2. Find N_j' and N_{j+1}' such that $N_j' + N_{j+1}' = N_j + N_{j+1}$ and $(\sigma_j^2)^{p_j'}(\sigma_{j+1}^2)^{p_j'}$ is minimized.
3. $N_j = N_j'$ and $N_{j+1} = N_{j+1}'$.
4. $j = j + 1$. If $j < J$, go to step 2.
5. Let $\underline{N} = [N_1, N_2, \ldots, N_J]'$. If $\underline{N}$ is equal to $\underline{N}_{prev}$, then **STOP**. Otherwise, $j = 1$. $\underline{N}_{prev} = \underline{N}$. Go to step 2.

Here, the average mean squared energy of a block (i.e., $E = (\sum_{i=1}^{L} x_i^2)/L$) is the criterion for classification.

The side information required to encode the image consists of the mean of the J DC sequences, and the standard deviations of all $64J$ sequences. These quantities are quantized using 16-bit uniform scalar quantizers to yield $16(64J + J)$ bits. Since 4-state trellises are used exclusively throughout the system, the initial trellis state of each sequence requires 2 bits. The total side information is then $1168J$ bits, which corresponds to $2336/(512)^2 = 0.0089$ bits per pixel (bpp) for a 512×512 image and $J = 2$, or $4672/(512)^2 = 0.0178$ bpp for $J = 4$.

In addition, the classification map must also be transmitted. The map is arithmetically encoded using "on-the-fly" multi-order modeling. The map requires a maximum of 8192 bits for a 512×512 image (6648 bits for the Lenna image). The Kurtosis values for all $64J$ sequences are also arithmetically encoded (after being rounded to one of five possible values).

3.1. Codebook Design

The probability distribution of each sequence to be encoded is modeled by the so-called *Generalized Gaussian Distribution* (GGD), whose probability density function (pdf) is given by

$$f_X(x) = \left[\frac{\alpha\eta(\alpha,\sigma)}{2\Gamma(1/\alpha)}\right] \exp\{-[\eta(\alpha,\sigma)|x|]^{\alpha}\} \quad (2)$$

where

$$\eta(\alpha,\sigma) \equiv \sigma^{-1}\left[\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}\right]^{1/2}. \quad (3)$$

The shape parameter α describes the exponential rate of decay and σ is the standard deviation of the associated random variable [9]. The gamma function $\Gamma(\cdot)$ is defined as

$$\Gamma(n) = \int_0^{\infty} e^{-x}x^{n-1}dx. \quad (4)$$

Distributions corresponding to $\alpha = 1.0$ and 2.0 are Laplacian and Gaussian, respectively. Figure 3 shows generalized Gaussian pdfs corresponding to $\alpha = 0.5, 1.0, 1.5, 2.0$, and 2.5.

It can be shown that

$$E[X^4] = \int_{-\infty}^{\infty} x^4 f_X(x)dx = K\sigma^4 \quad (5)$$

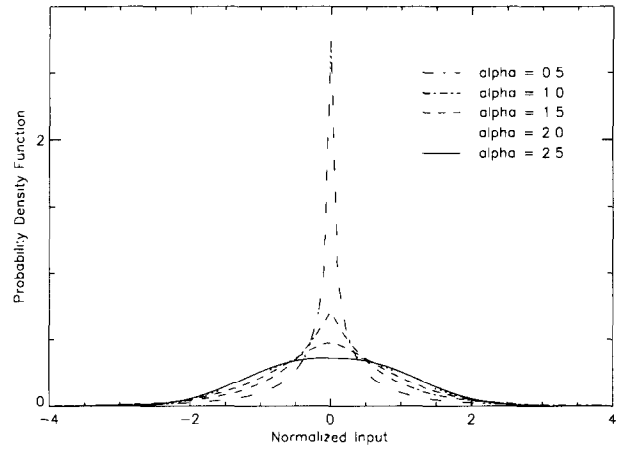

Figure 3: Probability density function for generalized Gaussian distributions with alpha values of 0.5, 1.0, 1.5, 2.0, and 2.5.

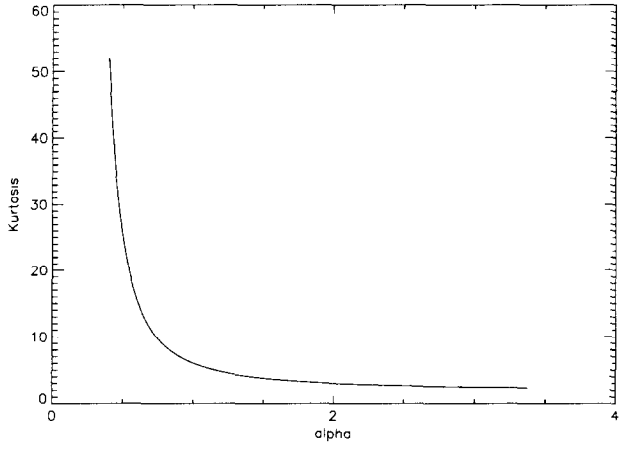

Figure 4: Kurtosis vs. alpha.

or

$$K = \frac{E[X^4]}{\sigma^4} = \frac{\Gamma(5/\alpha)\Gamma(1/\alpha)}{\Gamma(3/\alpha)^2} \tag{6}$$

where K is the fourth central moment, or Kurtosis. Recall that the Kurtosis is a measure of the peakedness of a given distribution. If a pdf is symmetric about its mean and is very flat in the vicinity of its mean, the coefficient of Kurtosis is relatively small. Similarly, a pdf that is peaked about its mean has a large Kurtosis value.

The sample Kurtosis of any sequence can be calculated easily and used as a measure by which the distribution of the sequence can be determined. Figure 4 shows the relationship between the shape parameter α and K. This graph is used to determine the appropriate α for a particular sequence.

Codebooks were designed for generalized Gaussian distributions with α values of 0.5, 0.75, 1.0, 1.5, and 2.0, using the algorithm in [4]. It was shown in [4] that for the Gaussian distribution, optimum codebooks do not yield significant MSE improvement over uniform codebooks at rates greater than 2.5 bits/sample. Experimentation revealed that this is also true for $\alpha = 1.5$ and $\alpha = 1.0$. However, for $\alpha = 0.75$, optimum codebooks are superior up to 3.0 bits/sample, while for $\alpha = 0.5$, optimum codebooks should be used up to 3.5 bits/sample. Accordingly, for α values of 2.0, 1.5, and 1.0, optimum codebooks were designed in one-tenth bit increments up to 2.5 bits/sample, while for $\alpha = 0.75$ and $\alpha = 0.5$, optimum codebooks were designed in one-tenth bit increments up to 3.0 and 3.5 bits/sample, respectively. Thereafter, uniform codebooks were designed in one-tenth bit increments up to 12 bits/sample. Training sequences consisted of 100,000 samples derived from generalized Gaussian pseudo-random number generators, each tuned to the appropriate α value.

3.2. Rate Allocation

Rate allocation is performed by using the algorithm presented in [7]. The overall MSE incurred by encoding the coefficient sequences using ECTCQ at an average rate of R_s bits/coefficient is represented by

$$E_s = \sum_{i=1}^{K} \alpha_i \sigma_i^2 E_{ij}(r_i) \tag{7}$$

where σ_i^2 is the variance of sequence i, $E_{ij}(r_i)$ denotes the rate-distortion performance of the j^{th} quantizer (e.g., the quantizer corresponding to the Kurtosis of sequence i) at r_i bits/sample, K is the number of data sequences, and α_i is a weighting coefficient to account for the variability in sequence length. For 8×8 blocks and J classes, $K = 64J$.

The rate allocation vector $B = (r_1, r_2, \ldots, r_K)$ is chosen such that E_s is minimized, subject to an average rate constraint:

$$\sum_{i=1}^{K} \alpha_i r_i \leq R_s \text{ bits/coefficient.} \tag{8}$$

It is shown in [7] that the solution $B^*(r_1^*, r_2^*, \ldots, r_K^*)$ to the unconstrained problem

$$\min_{B} \left\{ \sum_{i=1}^{K} (\alpha_i \sigma_i^2 E_{ij}(r_i) + \lambda \alpha_i r_i) \right\} \tag{9}$$

minimizes E_s subject to $\sum_{i=1}^{K} \alpha_i r_i \leq \sum_{i=1}^{K} \alpha_i r_i^*$. Thus, to find a solution to the constrained problem of equations (7) and (8), it suffices to find λ such that the solution to equation (9) yields $\sum_{i=1}^{K} \alpha_i r_i^* \leq R_s$. Procedures for finding the appropriate λ are given in [7].

For a given λ, the solution to the unconstrained problem is obtained by minimizing each term of the sum in (9) separately. If S_j is the set of allowable rates for the j^{th} quantizer and r_i^* is the i^{th} component of the solution vector B^*, then r_i^* solves

$$\min_{r_i \in S_j} \{\alpha_i \sigma_i^2 E_{ij}(r_i) + \lambda \alpha_i r_i\}. \tag{10}$$

4. RESULTS AND CONCLUSIONS

Coding simulations were performed on the 512×512 8-bit Lenna image. The number of classes was chosen to vary from zero to four. The compression rates were calculated from actual bit streams produced by the coder, and include all side information.

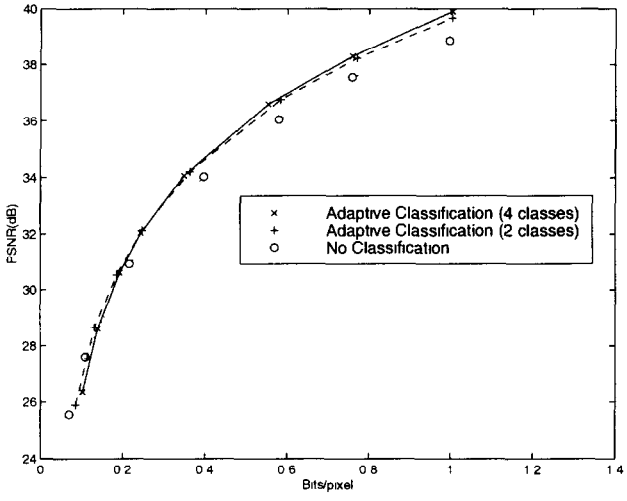

Figure 5: Comparison of different number of classes for the Lenna image.

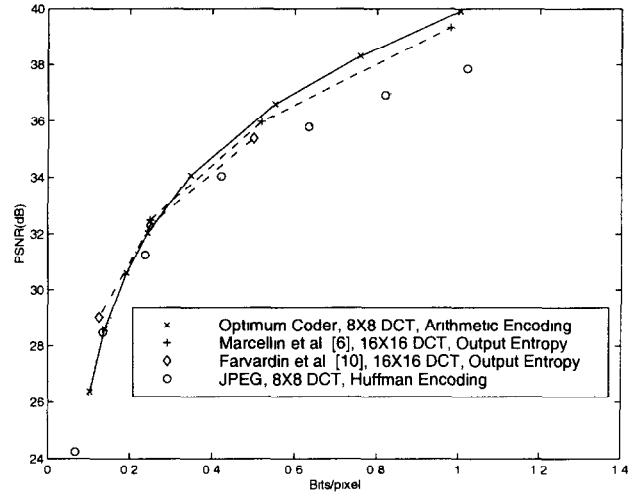

Figure 6: Comparison of encoding performance for the Lenna image.

Figure 5 shows the PSNR performance of the coder at various bit rates. Note that for higher encoding rates, greater performance is obtained as the number of classes increases. However, no performance is gained when the number of classes in increased beyond four. In this case, the increased side information overwhelms the increase in coding gain. The performance of the four-class system is greater than that of the one-class system at rates above 0.15 bits/pixel. Below that rate, the additional side information required by the four-class system becomes a very large percentage of the overall bit rate, thus leaving less rate for the quantizers. Finally, the two-class system performs almost as well as the four-class system at high rates, and outperforms the four-class system at rates below 0.25 bits/pixel. Note that the two-class system outperforms the system with no classification above 0.1 bits/pixel.

Figure 6 shows the performance of the four-class system as compared to other DCT-based coders in the literature. The competing coders are the DCT/ECTCQ coder presented in [6], the perceptually-weighted adaptive DCT/ECTCQ coder presented in [10], and JPEG. Our coder outperforms the coders in [6] and [10] at rates above 0.3 bits/pixel. Below that rate, the reported performance of the two competing coders is greater than our system. However, the systems in [6] and [10] utilize 16×16 DCTs, which provides a vast performance increase over the 8×8 DCT used in our system. Additionally, the bit rates reported for the two competing systems are output entropies, and are not calculated from actual bit streams.

We have presented an image coder which combines the 2-D DCT, maximum-coding-gain adaptive classification, fully optimized entropy-constrained trellis-coded quantization, optimal rate allocation, and adaptive arithmetic encoding. To our knowledge, this coder outperforms any system that uses an 8×8 DCT, and that produces an actual bit stream. In addition, our coder is competitive with many systems using subband decomposition, while being vastly less computationally complex.

5. REFERENCES

[1] R. L. Joshi, H. Jafarkhani, H. Kasner, T. R. Fischer, N. Farvardin, and M. W. Marcellin, "Comparison of different methods of classification in subband coding of images," *IEEE Trans. Image Proc.*, vol. IP-6, pp. 1473–1486, Nov. 1997.

[2] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. Commun.*, vol. COM-38, pp. 82–93, Jan. 1990.

[3] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.

[4] M. W. Marcellin, "On entropy-constrained trellis coded quantization," *IEEE Trans. Commun.*, vol. 42, pp. 14–16, Jan. 1994.

[5] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, pp. 520–540, June 1987.

[6] M. W. Marcellin, P. Sriram, and K. Tong, "Transform coding of monochrome and color images using trellis coded quantization," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 3, pp. 270–276, Aug. 1993.

[7] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. 36, pp. 1445–1453, Sept. 1988.

[8] R. L. Joshi, T. R. Fischer, and R. H. Bamberger, "Optimum classification in subband coding of images," *Proc. International Conference on Image Processing*, pp. 883–887, Nov. 1994.

[9] N. Farvardin and J. W. Modestino, "Optimum quantizer performance for a class of non-gaussian memoryless sources," *IEEE Trans. Inform. Th.*, vol. 30, pp. 485–497, May 1984.

[10] N. Farvardin, X. Ran, and C.-C. Lee, "Adaptive DCT coding of images using entropy-constrained trellis coded quantization," in *Conf. Proceedings*, 1993 Int. Conf. on Acoust., Speech, and Signal Proc., Minneapolis, MN, Apr. 1993.