Transform Trellis Coding of Images at Low Bit Rates

S. FARKASH, DAVID MALAH, FELLOW, IEEE, AND WILLIAM A. PEARLMAN, SENIOR MEMBER, IEEE

Abstract—**A transform trellis coding scheme is applied to encode images at rates below 1 b/pel resulting in high SNR values and very good subjective performance. A discrete cosine transform (DCT) is used to decorrelate the data samples, and a trellis encoder is used to encode the transformed coefficients. To overcome image nonstationarity a clustering algorithm is used to segment the transformed image into distinct regions and a separate trellis diagram is then constructed for each cluster. To further improve the image quality, particularly reducing the blocking effect, a scalar quantizer is used to quantize large magnitude coefficients of the error image in the transform domain. The performance of the proposed scheme is found to be better than other recently reported schemes.**

I. INTRODUCTION

THE transform trellis coding (TTC) technique was proved by Mazor and Pearlman [1] to be asymptotically optimal for stationary Gaussian sources and the squared-error distortion measure. In [2], Mazor and Pearlman applied this technique to encode speech, and the same basic ideas were used in [3] and [18], to encode images. The main problem encountered in [3] is the blocking effect which usually accompanies block transform coding schemes at low rates. This paper expands the work in [3] and [18] and presents an improved transform trellis coding scheme, which encodes images at rates below 1 bit per pel and alleviates the blocking effect problem. The proposed scheme uses a clustering algorithm based on the LBG algorithm, which was originally used to construct code books for vector quantizers (VQ) [14], to partition the image into clusters having similar spectral characteristics. A similar classification method was used in [19]. The blocks in each cluster are then encoded by a trellis which is constructed to fit the characteristics of that cluster. Further reduction of the blocking effect is obtained by applying a scalar quantizer to encode the high magnitude coefficients of the error image in the transform domain.

The proposed TTC coding scheme performs two passes on the source image. On the first pass the image is divided into s blocks. Each block is transformed by a two-dimensional discrete cosine transform (DCT). The DCT is considered to be the best transform, among all known data-independent orthogonal transforms, from the viewpoint of data compactness and ease of implementation [4], [5]. The generated transformed coefficients are less redundant and are almost uncorrelated. The s transformed blocks are rearranged as vectors and are classified by a clustering algorithm to c classes (clusters). For each class an estimated spectrum is produced and used to construct the trellis diagram for that class. The same estimated spectrum is sent to the receiver and enables the construction of the decoder. On the second pass the encoder searches the trellis, which is populated by a random selection of Gaussian code vectors, for a code vector which minimizes the squared-error between the source vector and the code vectors. The corresponding path-map on the diagram is forwarded to the receiver.

Paper approved by the Editor for Image Communication Systems of the IEEE Communications Society. Manuscript received August 11, 1988; revised September 28, 1989. This work was supported by the Lady Davis Trust. This paper was presented in part at the 15th IEEE Convention of Electrical and Electronics Engineers in Israel, Tel-Aviv, April 1987, and EUSIPCO-88, Grenoble, France, September 1988.

S. Farkash and D. Malah are with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel.

W. A. Pearlman is with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180.

IEEE Log Number 9038344.

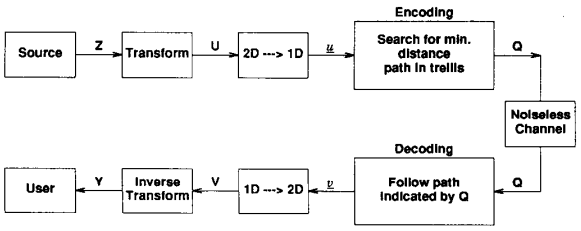


Fig. 1. TTC basic scheme.

To remove the blocking effect, which is typical of transform coders at low bit rates, and was also pronounced in [3], an error vector is constructed by subtracting the code vector, composed of codewords residing along the above path-map, from the source vector. A uniform scalar quantizer is used to quantize the error vector. When the encoding process is completed, the nonzero coefficients of the quantized error vector and their locations are coded and sent to the decoder as additional side-information.

The decoder which constructs the same trellis diagram, using the estimated spectrum side-information, uses the received path-map to uniquely determine the code vectors. The quantized error coefficients are added to the code vectors, at the proper locations, to produce the reconstructed vectors, which are then reordered into two-dimensional blocks and inverse transformed to obtain the reproduction image.

The remainder of this paper describes in more details the proposed coding scheme and the results obtained. Section II introduces the basic transform trellis coding scheme. Section III describes practical implementation considerations and the additional means used to reduce the blocking-effect. Simulation results are contained in Section IV, and finally, a summary and conclusions are presented in Section V.

II. TRANSFORM TRELLIS CODING

Fig. 1 presents the basic transform trellis coding scheme applied to image coding. The first stage of the scheme, the transformation stage, is composed in principle of an optimal two-dimensional transformation (the Karhunen–Loeve transform—KLT) which is applied to a block Z of the image to produce a block U of uncorrelated elements. The transformed block U is reordered as a vector $\boldsymbol{u}$ which serves as an input data vector to the second stage of the scheme—the coding/decoding stage. The coding stage is based on a trellis diagram shown in Fig. 2(a), with the following parameters:

K—Trellis constraint length (size of decoder shift register).

L—Depth of trellis ($K \ll L$).

q—Number of branches per node (branching factor).

n_m—Number of code letters in branch word on each branch at depth m of the trellis.

0090-6778/90/1100-1871$01.00 © 1990 IEEE

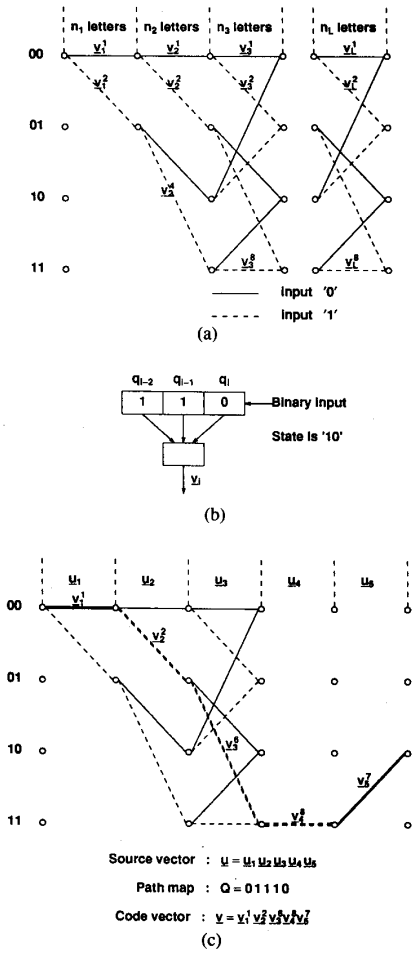


Fig. 2. (a) Trellis diagram ($K = 3$, $q = 2$). (b) Corresponding state machine. (c) Example.

The decoder can be characterized as a finite state machine [Fig. 2(b)] whose inputs are the channel symbols and whose outputs are the code words. The machine has q^{K-1} states, and for each channel symbol entering the machine, it makes a transition between states and emits the codeword corresponding to this transition. The trellis is a directed graph composed of nodes and branches. The nodes are associated with the states of the shift register, and the branches with the transition between states, when a new symbol enters the register. At depth (level) m of the trellis diagram every one of the q^K branches is populated with n_m letters, which are a copy of the decoder reproduction letters of the corresponding transition. Thus, the memory requirements for the mth level of the trellis is $n_m \cdot q^K$ letters. A simple example which demonstrates the coding process is shown in Fig. 2(c). The source vector is denoted by $\boldsymbol{u}$, the code words which form the lowest distortion code vector $\boldsymbol{v}$ are identified in the figure as $v_1^1, v_2^2, v_3^6, v_4^8, v_5^7$, and the corresponding path-map marked by bold lines is (0, 1, 1, 1,0).

The coding process is performed as follows.

For a vector vector $\boldsymbol{u}$ corresponding to a block of the image the encoder searches the trellis for the lowest distortion path, i.e., the path along the trellis whose branches contain the reproduction vector that minimizes the squared-error. The path-map Q corresponding to the chosen path is sent through the channel (assumed here to be noiseless) to the decoder. At the decoder end, the received sequence is fed to the shift register which emits the reconstructed vector $\boldsymbol{v}$. Finally, the vector $\boldsymbol{v}$ is appropriately reordered as a block V, which is inverse transformed to obtain the reconstructed block Y. The above procedure is repeated for every block of the image.

The basis of this coding scheme is established in Mazor and Pearlman works [1], [2], and their basic ideas are repeated here for the sake of clarity and completeness. In addition, we shall quote the basic results concerning the coding of a stationary Gaussian source and the squared-error distortion measure. A more detailed discussion can be found in [12 pp. 111-113]. For the sake of simplicity, the results quoted in here are for a one-dimensional signal and transform.

A. Rate Distortion Results

The solution of the rate distortion equation renders the bit allocation that further determines the branch population and the probability distributions from which the branch letters are drawn. Let $z(t)$, $t = ,\cdots, -1, 0, 1, \cdots$, be a sequence from a stationary, zero-mean, Gaussian source with a continuous and bounded power spectral density $S_z(\omega)$. Let Φ_z be the covariance matrix associated with a vector $\boldsymbol{z}$ of N observations from the source, and Γ the unitary KLT matrix which decorrelates $\boldsymbol{z}$ through $\boldsymbol{u} = \Gamma^T \boldsymbol{z}$. The covariance matrix of $\boldsymbol{u}$, is the diagonal matrix $\Lambda_u = \Gamma^T \Phi_z \Gamma = [\lambda_l \delta_{lk}]$, and $\boldsymbol{u}$ is a jointly zero-mean Gaussian vector with variances $\{\lambda_l\}_{l=1}^N$. Correspondingly, a reproduction vector $\boldsymbol{y}$ is related to its transform $\boldsymbol{v}$ by $\boldsymbol{y} = \Gamma \boldsymbol{v}$.

Since the transformation matrix Γ is invertible it preserves the average mutual information $I(\boldsymbol{u}, \boldsymbol{v})$ [i.e., $I(\boldsymbol{u}, \boldsymbol{v}) = I(\boldsymbol{z}, \boldsymbol{y})$], and since it is unitary, it also preserves the squared-error distortion

$$d(\boldsymbol{z}, \boldsymbol{y}) = d(\boldsymbol{u}, \boldsymbol{v}) = \|\boldsymbol{u} - \boldsymbol{v}\|^2. \tag{1}$$

This enables us to solve the rate-distortion function in the transform domain and do the coding in that domain.

The N-tuple rate-distortion function $R_N(D)$ of a source, determines the minimum rate required to represent a N-tuple vector of that source for a given average distortion D. A parametric solution of the rate-distortion function in this case is given by

$$D_\theta = \frac{1}{N} \sum_{l=1}^{N} d_\theta^l = \frac{1}{N} \sum_{l=1}^{N} \min(\theta, \lambda_l) \tag{2}$$

$$R_N(D_\theta) = \frac{1}{N} \sum_{l=1}^{N} r_\theta^l = \frac{1}{N} \sum_{l=1}^{N} \max\left(0, \frac{1}{2} \log \frac{\lambda_l}{\theta}\right) \tag{3}$$

where

D_θ—the average distortion between the source and the reproduction vectors, for a given parameter θ.

d_θ^l—the distortion between the lth letter in the source and reproduction vectors.

r_θ^l—the rate (in bits) of the letter u_l (lth element of $\boldsymbol{u}$).

$\log a$—base 2 logarithm of a.

The conditional probability density function $P(\boldsymbol{v} \mid \boldsymbol{u})$, which solves the rate-distortion function, and the source density function $P(\boldsymbol{u})$, are used to define the density function $P(\boldsymbol{v})$ of the transformed reproduction vectors. This density function will be used to construct the code vectors and is given by

$$P(\boldsymbol{v}) = \prod_{l=1}^{N} P_l(v_l) \tag{4}$$

where

$$P_l(v_l) = \begin{cases} \dfrac{1}{\sqrt{2\pi(\lambda_l - \theta)}} \exp\left\{-\dfrac{v_l^2}{2(\lambda_l - \theta)}\right\}, & \lambda_l > \theta \\ \delta(v_l), & \lambda_l \le \theta \end{cases} \tag{5}$$

where $\delta(v_l)$ the Dirac delta function, denotes the distribution of coefficients with zero variance.

B. Code Construction

The trellis structure (Fig. 2) is completely determined by the choice of the branching factor q and the decoder constraint length K. As mentioned above, at depth m of the trellis, each branch is populated by a vector v_m of n_m code letters selected randomly from a source characterized by $P(v)$ as in (4). The code letters within a vector and in different vectors are statistically independent, and are distributed according to $P_l(v_l)$ given in (5). The rate, in bits per pel, of each code letter at level m of the trellis is

$$\rho_m = \frac{\log(q)}{n_m}, \qquad m = 1, 2, \cdots, L. \tag{6}$$

The average trellis code rate, in bits per pel, is therefore given by

$$R = \frac{1}{N}\sum_{m=1}^{L} n_m \rho_m = \frac{L}{N}\log(q). \tag{7}$$

To complete this part we give the formal algorithm used to construct the trellis code.

i) For a given desired rate R determine the parameter θ such that

$$R = \frac{1}{N}\sum_{l=1}^{N} \max\left\{0, \frac{1}{2}\log\left(\frac{\lambda_l}{\theta}\right)\right\} = \frac{1}{N}\sum_{l=1}^{N} r_\theta^l \tag{8}$$

where

R—the desired code rate,
λ_l—the lth eigenvalue of the covariance matrix of u,
N—the vector size, and
r_θ^l—the optimal rate of the lth coefficient of u

ii) Choose q arbitrarily, but large enough to satisfy

$$\log(q) \geq \max_l \{r_\theta^l\}. \tag{9}$$

iii) Choose an integer value for K (taking into consideration memory and computation requirements proportional to q^K).

iv) Determine the set of possible branch population numbers $\{n_m\}$ by

1) set $n_{\min}$ to be the largest positive integer such that

$$\frac{\log(q)}{n_{\min}} > \max_l \{r_\theta^l\} \tag{10}$$

2) set $n_{\max}$ to be the smallest integer to satisfy

$$\frac{\log(q)}{n_{\max}+1} < \min_l \{r_\theta^l\}. \tag{11}$$

The set of available population numbers $\{n_m\}$ are the set of integers from $n_{\min}$ to $n_{\max}$. Correspondingly, the set of possible level rates ρ_m is given in (6).

v) Set the actual rate of a source letter u_l with optimal rate r_θ^l to ρ_m if $\rho_m > r_\theta^l > \rho_{m+1}$. This is done by associating n_m code letters to the mth level in the trellis, which will be used to code the source letters $\{u_i\}_{i=l}^{l+n_m-1}$. When less than n_m source letters have optimal rate in the above range, letters with rate less than ρ_{m+1} are appended. When more than n_m source letters have rate in the above range, successive levels in the trellis are assigned letters with the same rate, and hence they have the same population numbers. Fig. 3 describes schematically this procedure.

iv) Populate each branch at level m of the trellis with a random vector v_m composed of n_m letters. Each letter is chosen independently from a Gaussian source with a zero mean and a variance of $\lambda_l - \theta$ as given in (5).

According to the theorem proved in [1] the ensemble average of trellis codes, constructed as described above, is optimal for a stationary Gaussian source, in the sense that as K goes to infinity, it achieves an average MSE distortion D_θ with a code rate R arbitrarily close to $R_N(D_\theta)$. Therefore, there exists at least one trellis code from that ensemble, whose MSE performance approaches that of the rate-distortion function with increasing K.

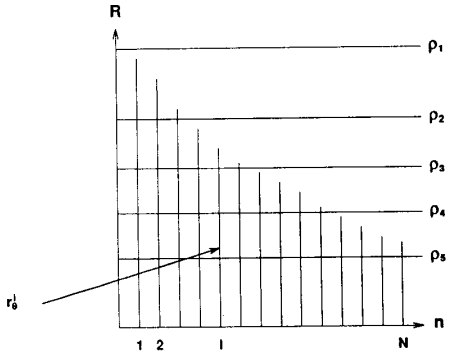


Fig. 3. Bit allocation among trellis levels. (In this figure the rates r_θ^l are monotonically decreasing, corresponding to a rearrangement of the eigenvalues in decreasing order of magnitude, as explained in Section III-C.)

III. Implementation Considerations and System Description

The implementation of the transform trellis coder presented in the previous section suffers from several shortcomings. First, the computation of the optimal KLT is a too complicated and expensive task, as it requires computation of the covariance function and its eigenfunctions and eigenvalues. Another weakness of the scheme is that it does not take into account the nonstationary nature of the images we usually have to code. This nonstationarity prevents the transform trellis coder from obtaining good performance for every block in the image. The blocking effect, which is usually encountered in transform coding at low bit rates, is also a problem in the above scheme. This section presents our solutions to these difficulties and presents the improved coding scheme.

A. Discrete Cosine Transform – DCT

The optimal trellis code is achieved only when the KLT is used. Because of its signal dependence and its complexity, the KLT is replaced by the suboptimal DCT as defined, e.g., in [4], [8]. The suboptimality of the DCT stems from the fact that it doesn't completely decorrelate the covariance matrix of the data block. However, it has been shown [5], [6] that the DCT is asymptotically optimal and converges to the KLT (as the block size increases) when operated on Markovian signals. Furthermore, for finite size vectors from a Markovian source, the DCT performance is the closest to the performance of the KLT among all other suboptimal transforms [5]. The DCT overcomes the main disadvantages of the KLT, i.e., it is signal independent, and there exist fast algorithms for its computation. The algorithm applied for computing the DCT is the one presented in [4], which uses the FFT algorithm on a reordered version of the input block.

B. Clustering

In the discussion of the optimal coding system, we have assumed the source to be stationary and Gaussian. This assumption does not hold for most natural images. The justification for the assumption that the transformed coefficients are Gaussian comes from the Central Limit Theorem and the fact that DCT is a weighted sum of random variables [8, p. 275]. However, later work [15] showed that the Laplace distribution is more suitable to model the DCT coefficients. We decided to proceed with the Gaussian assumption and leave to later studies the subject of trellis coding using the Laplace distribution. A more serious problem we face is the nonstationary nature of the images. To solve this problem we assign the image blocks to different clusters based on the spectral characteristics of the blocks (without the dc term). Then, we estimate a spectrum for each cluster and use it to construct a trellis diagram for that cluster.

Assuming stationarity among blocks belonging to the same cluster, the trellis code is considered optimal.

To explain the clustering problem we introduce the set W of s vectors $W = \{w_i \mid i = 1, \cdots, s\}$ and define the term *partition* as a collection of sets of nonoverlapping vectors $C = \{C^1, \cdots, C^c\}$ such that

$$\bigcup_{k=1}^{c} C^k = W. \tag{12}$$

The clustering problem deals with the partitioning of the set W into c clusters C^k, $k = 1, \cdots, c$, each represented by its mean value m^k

$$m^k \triangleq \frac{1}{\gamma_k} \sum_{\substack{i \\ w_i \in C^k}} w_i \tag{13}$$

where γ_k is the cardinality of the kth cluster.

In our scheme the classified vectors w_i are taken to be the transformed source vectors, squared term by term, i.e.,

$$w_i = \{u_1^2 \; u_2^2 \; \cdots \; u_N^2\}^T.$$

Given a distance measure $d(w_i, m^k)$ between a vector w_i and a cluster representative m^k (the Euclidian distance in our case), the problem is to find among all possible partitions of the set W into c clusters, the partition which minimizes the clustering criterion $J(C)$, which is given by

$$J(C) \triangleq \sum_{k=1}^{c} \sum_{\substack{i \\ w_i \in C^k}} d(w_i, m^k) = \sum_{k=1}^{c} J^k(C). \tag{14}$$

The problem may be stated as: among all possible partitions find the one which minimizes the sum of all distances of the vectors in C^k from their representative vector m^k for all the clusters ($k = 1, \cdots, c$).

There are two basic approaches to perform clustering [7].

1) The dynamic clustering method which employs an iterative algorithm to optimize the clustering criterion function $J(C)$.

2) The hierarchical clustering method which uses all s vectors as initial clusters. Then, by repeatedly merging the two most similar clusters, the number of clusters is reduced, until the desired number of clusters is reached.

The hierarchical clustering method is suboptimal in the sense that it does not necessarily reach a minimum of the function $J(C)$. Furthermore, when we applied this method to the problem at hand, we found that it is also more complicated than the dynamic clustering method. The reason for this complication is the relatively small number of clusters needed, as compared to the total number of vectors (e.g., $s = 1024$, $c = 16$ to 64), a fact which results in many iterations. The dynamic clustering method, on the other hand, reaches a minimum of the criterion function $J(C)$, although it may be a local one. Using this method, we found that the algorithm converges to a solution in a relatively small number of iterations. Because of these reasons we have chosen to apply the dynamic clustering method in our scheme.

The dynamic clustering algorithm, known also as the LBG algorithm [14], can be stated as follows.

1) Choose an arbitrary partition of W into c clusters C^k, $k = 1, \cdots, c$; compute their mean vectors m^k, and evaluate $J(C)$ for this partition [denoted by $J^0(C)$].

2) Assign each vector w_i in the set W to the cluster whose mean vector m^k is closest to w_i, thus defining a new partition of W.

3) Update the mean vectors m^k, $k = 1, \cdots, c$, and compute the new value of $J(C)$ [denoted by $J^1(C)$].

4) If $(J^1(C) - J^0(C))/J^1(C) > \epsilon$ set $J^0(C) = J^1(C)$ and go to step 2); else stop.

The partition obtained in the last iteration is the desired partition.

The algorithm we eventually used circumvents the need for setting an initial partition by using the splitting method proposed in [14]. This method was originally developed to construct codebooks for vector quantizers (VQ).

The clustering algorithm with the splitting method works as follows.

1) Assign all the vectors in W to one cluster and compute its mean vector m using (13).

2) Split the mean vector into two vectors $m \pm \Delta$, where Δ is a fixed perturbation vector.

3) Use the dynamic clustering algorithm that is introduced above to find the optimal partition and the mean vectors of that partition.

4) Repeat steps 2) and 3), doubling the number of clusters in each iteration, until the desired number of clusters is reached.

The result of the clustering process is a segmented image where each block belongs to a cluster of blocks having similar spectra. Fig. 6(c) demonstrates a segmented image for $c = 16$ clusters.

C. Code Construction and Adaptation

The clustering process segments the image into c classes with different spectral characteristics. The next step is the code construction and adaptation. To determine the trellis structure we first have to estimate the eigenvalues (spectrum) corresponding to each cluster. These eigenvalues are used to compute the rate for each cluster. The estimated spectra and the cluster rates are then used to determine the structures of the trellis diagrams as described in Section II. A smoothing operation is used in the spectral estimation algorithm to improve the spectrum estimation and to reduce the side information. Because each block in a cluster usually has different average power, an additional adaptation process is needed. This adaptation is provided by modifying, for each block in the cluster, the population of the basic trellis for that cluster, as explained in the sequel. The algorithm which estimates the eigenvalues (spectrum) of each cluster is as follows.

1) Compute the set of normalized variances of the vectors (blocks) of the kth cluster C^k, according to the following equation:

$$\tilde{S}^k(l) = \frac{1}{\gamma_k} \sum_{\substack{i \\ u_i \in C^k}} \frac{(u_i(l))^2}{P_i^k}, \quad l = 1, 2, \cdots, N \tag{15}$$

where

$\tilde{S}^k(l)$—the normalized variance of the lth coefficient in C^k

N—the vector size

γ_k—the cardinality of C^k

$u_i(l)$—the lth coefficient of the ith vector in C_k

P_i^k—the power of the ith vector in C^k

$$P_i^k = \sum_{l=1}^{N} (u_i(l))^2, \quad u_i \in C^k. \tag{16}$$

2) Average the normalized variances $\tilde{S}^k(l)$ over successive M_s neighbors to get $N_s = N/M_s$ terms in $\bar{S}^k(l)$, according to

$$\bar{S}^k(l) = \frac{1}{M_s} \sum_{j=l}^{l+M_s-1} \tilde{S}^k(j), \qquad l = 1, 1 + M_S, 1 + 2M_S, \cdots, N - M_S + 1. \tag{17}$$

3) The remaining terms in $\bar{S}^k(l)$ are obtained by straight-line interpolation between the already computed N_s terms.

4) Compute the gain factor for the ith block in cluster k

$$G_i^k = \frac{P_i^k}{\sum_{l=1}^{N} \bar{S}^k(l)}. \tag{18}$$

5) The values of the estimated spectrum for the kth cluster which is used to determine the cluster's rate and to construct the trellis

diagram, is then given by

$$\hat{S}^k = \frac{\bar{S}^k}{\gamma_k} \sum_{\substack{i \\ u_i \in C^k}} G_i^k = \bar{S}^k G^k \tag{19}$$

where G^k is the average gain factor of cluster k, i.e.,

$$G^k = \frac{1}{\gamma_k} \sum_{\substack{i \\ u_i \in C^k}} G_i^k. \tag{20}$$

The above procedure is repeated for each cluster in the image, $(k = 1, \cdots, c)$.

The next step is the rate allocation among clusters. If the same rate is allocated to all the clusters, the scheme would not be optimal (in the MMSE sense) because clusters containing high variance blocks need higher rate than clusters with low variance blocks. Therefore, rate allocation among the clusters should be performed, such that for a given total rate the best performance (i.e., the smallest MSE) is obtained. The algorithm which is used here to allocate rates among the clusters is motivated by the rate-distortion function solution for Gaussian sources given in (2) and (3). The algorithm is formulated as follows.

1) For a given desired total average rate R (in bit/pel) determine the parameter θ such that

$$R = \frac{1}{P} \sum_{k=1}^{c} \sum_{l=1}^{N} \gamma_k \max \left\{ 0, \frac{1}{2} \log \left(\frac{\hat{S}^k(l)}{\theta} \right) \right\} \tag{21}$$

where

R—the desired total code rate,
P—the number of pels in the image ($P = N \sum_k \gamma_k$),
N—the number of pels in a block,
$\hat{S}^k(l)$—the lth coefficient of the average spectrum of cluster k,
γ_k—The cardinality of the kth cluster.

2) The average rate (in bits/pel) of all the blocks in cluster k, $k = 1, \cdots, c$ is given by

$$R^k = \frac{1}{N} \sum_{l=1}^{N} \max \left\{ 0, \frac{1}{2} \log \left(\frac{\hat{S}^k(l)}{\theta} \right) \right\} \tag{22}$$

where θ is the parameter found in step 1).

The cluster rate R^k and a reordered version of the estimated spectrum $\hat{S}^k$ are used to construct the trellis diagram of the kth cluster, as was described in the previous section. We use a reordered version of the estimated spectrum in decreasing order of its coefficient energy, since such a reordering improves the performance. This is because code coefficients with similar rates share the same level of the trellis, i.e., they will be coded at the same rate. The same reordering can be done at the receiver since the estimated spectrum $\hat{S}^k$ is transmitted as side information. In addition, since the first $K - 1$ stages of the trellis are not fully developed, i.e., they have the form of a tree, the performance of this part of the trellis is inferior, to later levels of the trellis. Therefore, we arranged the trellis levels in inverted order, so that the first levels correspond to low energy coefficients.

The last step in the code construction is the population of the trellis diagram. As was said before the population is performed separately for each block in the cluster, to improve the adaptivity of the scheme. The values of the estimated variances of the ith block in the kth cluster that are used to populate the diagram are given by

$$V_i^k = \frac{\hat{S}^k}{G^k} G_i^k \tag{23}$$

where

V_i^k—the estimated variances of the ith block of the kth cluster,
$\hat{S}^k$—the estimated spectrum of the kth cluster,
G^k—the average gain factor of the kth cluster [defined in (20)],
G_i^k—the gain factor of the ith block in the kth cluster [defined in (18)].

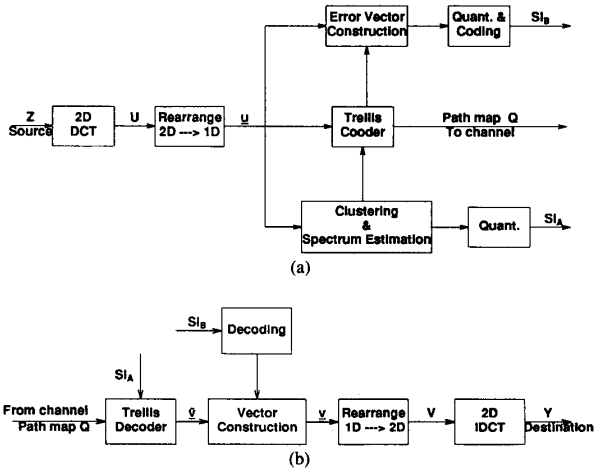


Fig. 4. (a) Transmitter. (b) Receiver.

D. Search Algorithm

A trellis code requires a search of the trellis for the path-map. This path is populated with a sequence of branch code-vectors, which provide an adequate match to the input sequence of source vectors. An exhaustive search which covers all candidate sequences will determine the best possible match. However, this search is not instrumentable for a trellis with a large number of possible paths. A class of algorithms which overcomes this difficulty is the class of parallel search algorithms based on the Viterbi algorithm [13]. In our simulations, we used a suboptimal version of the Viterbi algorithm known as the M algorithm [13]. The M algorithm finds at each level of the trellis the best M states, and extends them to the next level. The process is repeated until the end of the trellis is reached and the best of the resulting M path-maps is transmitted to the decoder.

E. Blocking Effect Suppression

One of the most disturbing effects in transform coding at low bit rates is the blocking effect. Several methods were reported to address this problem, namely the overlap method [20], [21] which operates on the source image, and the filtering method [20] which operates on the reconstructed image. In the proposed scheme we use the fact that both the source and the reconstructed images are available at the encoder, and apply a method which operates on the error image, i.e., the difference between the source and the reconstructed images. The error vector is constructed by subtracting the code reproduction vector, comprised of concatenated branch words residing along the chosen path-map on the trellis diagram, from the source vector. A uniform scalar quantizer is then used to quantize the terms in this error vector having large magnitude. The magnitudes and locations of the nonzero coefficients of the quantized error vector are coded efficiently by Huffman codes, and sent as side information denoted by (SI_B) to the decoder. The rate SI_B is determined by the scalar quantizer step size. In essence, the final coder is a two stage coder composed of a trellis coder followed by a scalar quantizer.

The encoding of the large magnitude terms in the error vector significantly reduces the blocking-effect. The additional complexity is very small, since the decoding process needed to create the reconstructed image, which is used in the creation of the error image, is done anyway as part of the search process in the trellis. The complexity accompanying the Huffman coder is negligible.

F. Summary of System Description

The implemented encoder [Fig. 4(a)] performs two passes on the input image.

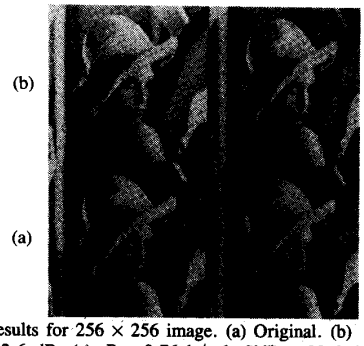


Fig. 5. Results for 256 × 256 image. (a) Original. (b) R = 0.61 b/pel, SNR = 32.6 dB. (c) R = 0.76 b/pel, SNR = 33.6 dB. (d) R = 0.92 b/pel, SNR = 35.2 dB.

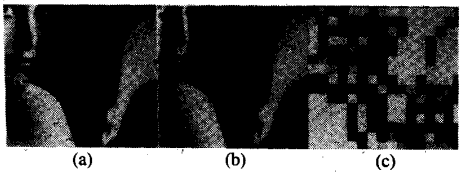


Fig. 6. Results for 512 × 512 image. (a) Original. (b) R = 0.41 b/pel, SNR = 37.4 dB. (c) Clustering image 16 classes.

In the first pass the structure of the encoder is determined. The image of size $[P' \times P']$ (total of P pels) is divided into s blocks of size $[N' \times N']$ (total of N pels). Each block undergoes a two-dimensional discrete cosine transform (DCT) and produces an equal size block of transform coefficients. The s transformed blocks are rearranged as vectors and are classified into c clusters according to the clustering algorithm which was introduced in Section III-B above. For each of the clusters that were created a cluster spectrum $\hat{S}^k$ is estimated. The estimated spectrum, along with the dc coefficient of each transformed block and its gain factor, each of them represented by an 8 bit codeword, are sent as side information (SI_A) to the decoder. The eigenvalues $\lambda^k(l)$ of each cluster k = 1 to c, which are used in the code construction, are taken to be the elements $\hat{S}^k(l)$ of the estimated spectrum (19) in decreasing order of magnitude. The variances used to populate the trellis for each block are computed from the estimated spectrum and the corresponding gain factor [as given in (23)].

In the second pass, the encoding pass, the DCT coefficients are reordered to match the order of the corresponding code-words on the trellis, and this vector enters the encoder. The encoder then searches the trellis for a code-vector that minimizes the squared-error between this vector and the input vector, and forwards the corresponding path-map Q (q-ary letters) to the receiver. Along with the trellis encoding process, an error vector is constructed and those of its coefficients having large magnitude are quantized, Huffman-coded, and sent to the receiver. The receiver [Fig. 4(b)] uses SI_A to replicate the encoder trellis, and uses the received path-map to uniquely determine the code vector $\tilde{v}$. The error vector coefficients that were transmitted as side information (SI_B) are added to the code vector $\tilde{v}$ to produce a yet unordered reconstructed vector. SI_A is then used to reorder the elements of this vector into its proper order, to obtain the vector v, which is then rearranged as a block V. Finally the block V is inverse transformed to obtain the spatial domain reproduction block Y.

IV. Simulation Results

To evaluate the performance of the proposed coding scheme we simulated it on a VAX/750 computer with a Gould IP8500 image display system. We use the woman's head and shoulder image ("LENNA") of size 256 × 256 pels shown in Fig. 5(a), as test image #1, and a high resolution version (512 × 512 image size) of the same image shown in Fig. 6(a) as test image #2. The performance is evaluated by the following.

TABLE I
Performance Comparison (256 × 256 image)

Reference		Rate [bits/pel]	SNR [dB]
[9]		0.67	30.9
		1	32.5
[11]		0.74	32.4
		1	32.8
[10]		0.5	27.5
		1	30.9
proposed coder	(Fig 5b)	0.61	32.6
	(Fig 5c)	0.76	33.6
	(Fig 5d)	0.92	35.2

1) The subjective quality of the reconstructed image.

2) The signal-to-noise ratio (SNR), expressed in dB, of the reconstructed image which is defined as $10 \log_{10}$ of the ratio of the peak to peak signal (255) to the root mean square (RMS) error between the original image and the reconstructed one [8].

The optimality of MSE as a measure of distortion is an open question especially with regard to the human observer. To take into account known characteristics of the human visual system, one could weight the spectral coefficients by the relative contrast sensitivity in the derivation of bit allocation formulas and design of the scalar quantizer of the large magnitude error terms. However, we prefer to present the scheme optimized for squared-error only, as almost all lossy coding schemes are derived and tested for their squared-error performance. The test of the worth of our scheme should be in squared-error and visual comparisons against other reported squared-error-based schemes. In fact, fair comparison to other schemes can be made only on this basis and are available only on this basis. Adjustments for visual system characteristics can than be made once the scheme has shown sufficient merit in these performance comparisons.

Fig. 5(b)-(d) shows reconstructed images of test image #1 at three different rates. Fig. 6 shows a reconstructed image of test image #2, along with its rate and SNR value achieved at R = 0.41 b/pel. The subjective quality of the reconstructed images in Fig. 5 are rated according to our judgment, from very good quality—for the image in Fig. 5(d), to good quality—for the image in Fig. 5(c), to better than fair quality—for the image in Fig. 5(b). The SNR achieved in our simulations for test image #1 are presented in Table I, along with the performance of other coding systems on the same test image which were recently published. The table shows that the performance of our coder is better than the performance obtained in [9], [10], [11]. The subjective quality of the reconstructed images resulting from the more redundant image [Fig. 6(b)] is very high, and one can not see any difference between the original and the reconstructed images. The SNR value achieved for test image #2 is very high compared to the performance achieved for test image #1, as is expected due to the higher correlation between pels in the higher resolution image. The results achieved for test image #2 are much better than the results reported in the literature for this image. Table II presents the results that are achieved by the proposed scheme for test image #2 along with the performance of other coding schemes for the same test image. The table shows that the performance of our coder is better than the performance obtained in [16], [17]. In all the above simulations the constraint length is K = 3 and the branching factor is q = 256. This branching factor enables allocation of 8 bits for the largest coefficient of the block. The block size used in the simulations for Fig. 5 is 8 × 8, whereas the block size used in Fig. 6 is 16 × 16. Typical side information rates used in the scheme are summarized in Table

TABLE II
PERFORMANCE COMPARISON (512 × 512 IMAGE)

Reference		Rate [bits/pel]	SNR [dB]
[16]		0.557	32.19
[17]		0.65	31.3
proposed coder	(Fig 6b)	0.41	37.44

TABLE III
TYPICAL SIDE INFORMATION RATES

	R	SI_A [bits/pel]	SI_B [bits/pel]
Fig. 5d	0.92	0.185	0.117
Fig. 6b	0.41	0.062	0.091

III. The overall rate is denoted by R, the side information needed to send the estimated spectra and the gain factors is denoted by SI_A, while the side information corresponding to the scalar quantizer is denoted by SI_B. The overall rate R can be determined in advance, however, the allocation of this rate between the trellis (i.e., the path-map and SI_A) and the scalar quantizer (SI_B) is image dependent. The optimal allocation could be determined iteratively, but as a rule of thumb one can allocate 85% of the desired rate to the trellis and 15% to the scalar quantizer. We found by experience that the deviations in the scheme performance, in terms of SNR, is less than 1 dB, when small modifications are done on the above suggested allocation.

V. SUMMARY AND CONCLUSION

The transform trellis coding scheme described in this work encodes cosine transform coefficients of an image on a trellis diagram. The image is segmented into several clusters using a dynamic clustering algorithm. Each cluster is coded by a different trellis diagram, constructed by using an estimated spectrum of that cluster. Each block in a cluster is encoded by the corresponding trellis diagram using code letters which are scaled to match the power of that block. To reduce the blocking effect encountered in transform coding schemes, a scalar quantizer is used to encode large magnitude coefficients of the error image in the transform domain.

The trellis coder is used in our work due to its optimality for signals having a Gauss–Markov model, considered to properly model real images, and because its implementation is simpler than tree and block coders (for the same performance). The deviation of the image from the assumed model is compensated by the clustering process and the scalar quantizer used. The performance obtained with the proposed scheme is superior to those reported in the literature with other coders (9, 10, 11, 16, 17). The complexity of the scheme is indeed very high. However, most of the complexity is in the encoder part of the scheme. This fact enables employment of the proposed scheme in image storage and retrieval systems, where the decoder simplicity is a vital property. With the continued development of VLSI technology, it is believed that this scheme could in the future be implemented also in real-time systems.

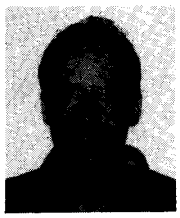
ACKNOWLEDGMENT

The authors would like to thank the reviewers for their careful reading and valuable comments.

REFERENCES

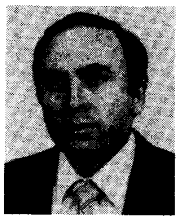
[1] B. Mazor and W. A. Pearlman, "A trellis code construction and coding theorem for stationary Gaussian sources," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 924–930, May 1983.
[2] ——, "An optimal transform trellis code with application to speech," *IEEE Trans. Commun.*, vol. COM-33, pp. 1109–1116, Oct. 1985.
[3] S. Farkash, "Transform trellis coding of images," in *Proc. IEEE 15th Convention Elec. Electron. Eng. Israel*, Tel-Aviv, Israel, Apr. 1987, pp. 2.5.4/1–2.5.4/4.
[4] J. Makhoul, "A fast cosine transform in one and two dimensions," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 27–34, Feb. 1980.
[5] M. Hamidi and J. Pearl, "Comparison of the cosine and Fourier transforms of Markov-1 signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 428–429, Oct. 1976.
[6] Y. Yemini and J. Pearl, "Asymptotic properties of discrete unitary transforms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 366–371, Oct. 1978.
[7] P. A. Devijuer and J. Kittler, *Pattern Recognition A Statistical Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[8] W. K. Pratt, *Digital Image Processing*. New York: Wiley, 1978.
[9] J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1278–1288, Nov. 1986.
[10] H. Hang and J. W. Woods, "Predictive vector quantization of images," *IEEE Trans. Commun.*, vol. COM-33, pp. 1208–1219, Nov. 1985.
[11] W. A. Pearlman, M. M. Leung, and P. Jakatdar, "Adaptive transform tree coding of images," *ICASSP-85*, Tampa, FL, 1985, pp. 145–148.
[12] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.
[13] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
[14] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantization," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.
[15] R. C. Reininger and J. D. Gibson, "Distribution of the two-dimensional DCT coefficients for images," *IEEE Trans. Commun.*, vol. COM-31, pp. 835–839, June 1983.
[16] A. Tran and K. Liu, "An efficient pyramid image coding system," in *Proc. ICASSP-87*, 1987, pp. 733–736.
[17] V. J. Mathews, R. W. Waite, and T. D. Tran, "Image compression using vector quantization of linear (one-step) prediction error," in *Proc. ICASSP-87*, 1987, pp. 733–736.
[18] S. Farkash, W. A. Pearlman, and D. Malah, "Transform trellis coding of images at low rates with blocking effect removal," in *Proc. EUSIPCO-88*, Grenoble, France, Sept. 1988.
[19] Y. Kato *et al.* "A motion picture coding algorithm using adaptive DCT encoding based on coefficient power distribution classification," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 1090–1099, Aug. 1987.
[20] H. C. Reeve and J. S. Lim, "Reduction of the blocking effect in image coding," in *Proc. ICASSP-83*, 1983, pp. 1212–1215.
[21] H. S. Malvar and D. H. Staelin, "Reduction of the blocking effect in image coding with a lapped orthogonal transform," in *Proc. ICASSP-88*, 1988, pp. 781–784.

★



Shmuel Farkash was born in Israel in 1956. He received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion—Israel Institute of Technology, Haifa, Israel, in 1979 and 1987, respectively. He is currently working towards the D.Sc. degree in electrical engineering at the Technion.

From 1979 to 1984 he was with the Israeli Defense Forces as an electronics engineer. From 1984 to 1985 he was with EL-OP Electro-Optics industries where he participated in image processing projects. Since 1985 he is a teaching assistant at the Technion. His research interests are in image processing and time-frequency representation of signals.



David Malah (S'67-M'71-SM'84-F'87) was born in Poland on March 31, 1943. He received the B.Sc. and M.Sc. degrees in 1964 and 1967, respectively, from the Technion—Israel Institute of Technology, Haifa, Israel, and the Ph.D. degree in 1971 from the University of Minnesota, Minneapolis, all in electrical engineering.

During 1971-1972 he was an Assistant Professor at the Electrical Engineering Department of the University of New Brunswick, Fredericton, N.B., Canada. In 1972 he joined the Electrical Engineering Department of the Technion, where he is presently an Associate Professor. From 1979 to 1981 he was on sabbatical and leave at the Acoustic Research Department of AT&T Bell Laboratories, Murray Hill, NJ, and a consultant at Bell Labs., during the summers of 1983, 1986, and 1988. During 1988-1989 he was on sabbatical leave at the Signal Processing Research Department of AT&T Bell Laboratories, Murray Hill. Since 1975 (except during 1979-1981 and 1988-1989) he is in charge of the Signal Processing Laboratory, at the EE Department, which is active in speech and image communication research and real-time hardware development. His main research interests are in image and speech coding, image and speech enhancement, and digital signal processing techniques.

★

William H. Pearlman (S'61-M'64-SM'84) for a photograph and biography, see p. 703 of the May 1990 issue of this TRANSACTIONS.