An Operational Rate-Distortion Optimal Single-Pass SNR Scalable Video Coder

Lisimachos P. Kondi, *Member, IEEE*, and Aggelos K. Katsaggelos, *Fellow, IEEE*

***Abstract*—In this paper, we introduce a new methodology for single pass signal-to-noise ratio (SNR) video scalability based on the partitioning of the DCT coefficients. The DCT coefficients of the displaced frame difference (DFD) for inter-blocks or the intensity for intra-blocks are partitioned into a base layer and one or more enhancement layers, thus, producing an embedded bitstream. Subsets of this bitstream can be transmitted with increasing video quality as measured by the SNR. Given a bit budget for the base and enhancement layers the partitioning of the DCT coefficients is done in a way that is optimal in the operational rate-distortion sense. The optimization is performed using Lagrangian relaxation and dynamic programming (DP). Experimental results are presented and conclusions are drawn.**

***Index Terms*—Layered coding, operational rate-distortion theory, scalable video coding.**

I. Introduction

A scalable video codec is defined as a codec that is capable of producing a bitstream which can be divided into embedded subsets. These subsets can be independently decoded to provide video sequences of increasing quality. Thus, a single compression operation can produce bitstreams with different rates and reconstructed quality. A small subset of the original bitstream can be initially transmitted to provide a base layer quality with extra layers subsequently transmitted as enhancement layers.

Scalability is supported by most of the video compression standards such as MPEG-2, MPEG-4 and H.263. Version 2 of the H.263 standard (also known as H.263+) [1], [2] supports SNR, spatial and temporal scalability. In SNR scalability, the enhancement in quality translates in an increase in the SNR of the reconstructed video sequence, while in spatial and temporal scalability the spatial and temporal resolution, respectively, are increased.

An important application of scalability is in video transmission from a server to multiple users over a heterogeneous network, such as the Internet. Users are connected to the network at different speeds, thus, the server needs to transmit the video data at bit rates that correspond to these connection speeds. Scalability allows the server to compress the data only once and serve each user at an appropriate bit rate by transmitting a subset of the original bitstream.

Another important application of scalability is in error resilient video transmission. It has been shown [3] that it is advantageous to use scalability and apply stronger error protection to the base layer than to the enhancement layers (unequal error protection). Thus, the base layer will be successfully decoded with high probability even during adverse channel conditions. Had we not used scalability but instead protected the whole bitstream equally, there would be a much higher probability of catastrophic errors that would result in a poor quality reconstructed video sequence.

In this paper, we present a new method for SNR scalability, which differs from the one supported by the standards. The method is based on the optimal partitioning of the discrete cosine transform (DCT) coefficients of the displaced frame difference (DFD) or the image intensity values. We first introduced an SNR scalable codec that is based on the partitioning of the DCT coefficients in [4]. Here, we present a more general partitioning scheme, as well as, an algorithm for the rate-distortion optimal partitioning of the DCT coefficients into scalable layers.

The paper is organized as follows. In Section II previously proposed methods for SNR scalability are described. In Section III the problem formulation is presented. In Section IV a Dynamic Programming solution to the problem is discussed. In Section V details on the proposed Rate Distortion optimal SNR scalable coder are given. In Section VI the extension of the proposed algorithm to more than two scalable layers is discussed. In Section VII experimental results are presented. Finally, in Section VIII conclusions are drawn. A pseudocode for the proposed algorithm is presented in the Appendix.

II. Methods for SNR Scalability

The traditional method for SNR scalability as utilized by the video compression standards (MPEG-2 [5], H.263 [1]) consists of the following steps. The base layer is created by quantizing and encoding the DFD, as in a nonscalable encoder. Then, the difference between the reconstructed base layer and the original frame is computed. This residual error is encoded the same way the DFD is encoded in nonscalable video encoders. In order to produce more enhancement layers, the same procedure is repeated by reconstructing the enhanced frame and encoding the new residual error. This method produces sufficient results but requires a relatively high computational complexity for both the encoder and the decoder, due primarily to the additional forward and inverse DCT's required at each stage. Furthermore, since the residual error is encoded like a regular frame, it also carries a significant bit overhead.

Manuscript received May 10, 2000; revised August 28, 2001. This work was supported in part by a grant from the Motorola Center for Communications at Northwestern University. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Luis Torres.

L. P. Kondi is with the Department of Electrical Engineering, State University of New York at Buffalo, Buffalo, NY 14260 USA (e-mail: lkondi@eng.buffalo.edu).

A. K. Katsaggelos is with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208 USA (e-mail: aggk@ece.northwestern.edu).

Publisher Item Identifier S 1057-7149(01)09734-2.

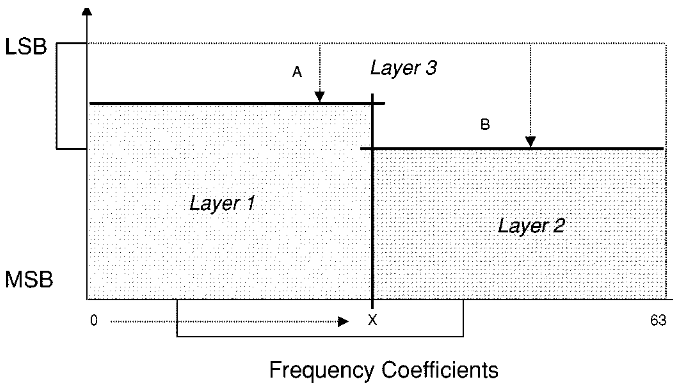


Fig. 1. Partition of the DCT coefficients in three layers according to [4] and [6].

The method we propose in this paper builds upon and generalizes a method we have proposed earlier [4], [6]. In this earlier method a single DCT and quantization operation of the DFD is involved. The coefficients are then partitioned in a number of sets which form the scalable layers. If, for example, three layers are involved, the partitioning is depicted in Fig. 1. In it the zig-zag scanned coefficients are shown in the horizontal axis, while the bit representation of each coefficient (most significant bit -MSB- to least significant bit -LSB- is shown in the vertical axis. The base layer includes coefficients (actually their quantization levels) $0-X$, without their A least significant bits. The first enhancement layer consists of coefficients $X+1$ to 63 without their B least significant bits. All remaining bits of all coefficients are transmitted with the second enhancement layer. Motion vectors and other overhead information are transmitted with the base layer. This algorithm combines the successive approximation and spectral selection approaches for scalability supported by JPEG [7]. The parameters X, A and B are adjusted by a rate control algorithm based on heuristics. This SNR scalable algorithm has a lower computational complexity and overhead than the method supported by the standards and described in the previous paragraph. However, it implicitly makes the assumption that the DFD data are lowpass, which is not necessarily true. In addition, the three parameters X, A and B which control the rate give us few degrees of freedom.

The generalization of the method outlined above is based on the following observations. Clearly, setting the least significant bits of a coefficient to zero is equivalent to subtracting a certain value from it. The variable length code (VLC) tables used in the standards use smaller length code words for smaller coefficient magnitudes. Thus, subtracting a value from a coefficient reduces the number of bits required for its representation but clearly increases the distortion. The decoder reconstructs the quantized DCT coefficients by adding the subtracted values (if available to it) to the values it received with the base layer. These observations form the basis of the proposed partitioning technique for the DCT coefficients which is much more general than the one discussed in the previous paragraph. The base layer is constructed by subtracting a value from each DCT coefficient. These subtracted values then represent the enhancement layer (See Fig. 2). If more than two scalable layers are required, the values subtracted for the creation of the base layer are further broken into other values. For example, if we want to transmit a coefficient with magnitude of quantization level 9 using three layers, we can transmit quantization level 5 as base layer, quantization level 2 as first enhancement layer and quantization level 2 as second enhancement layer. Next, we present a formulation and optimal solution to the problem of partitioning the DCT coefficients under this scheme [8], [9]. The nonscalable mode of the H.263 standard was used as a basis for our implementation, however, any motion-compensated DCT-based video codec could be used.

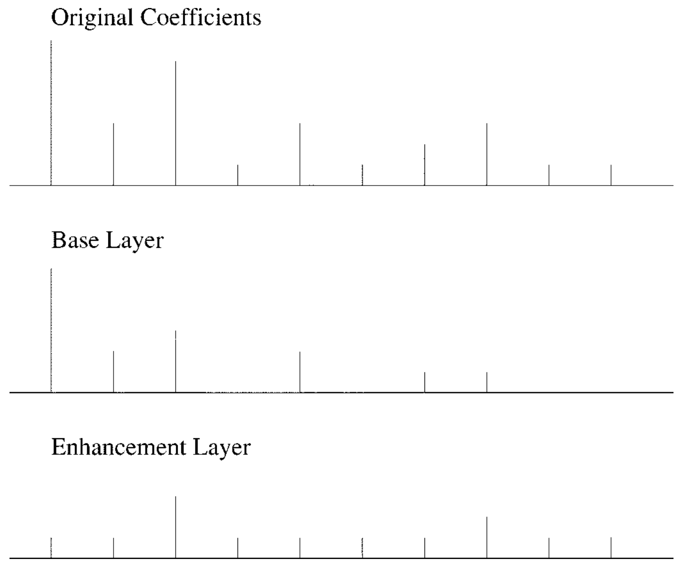


Fig. 2. Proposed partitioning of DCT coefficients for SNR scalability.

III. Problem Formulation

We assume that the DCT transform of the DFD (or the intensity for intra blocks) is taken and quantized. That is, a triplet (LEVEL, RUN, LAST) is transmitted using suitable VLC tables, where LEVEL is the quantization level of the coefficient, RUN is the number of zero-valued coefficients that precede it and LAST specifies whether the current coefficient is the last in the block. An extra bit is appended to the VLC to denote the sign of LEVEL. Therefore, in the following discussion, LEVEL will refer to the absolute value of the quantization index.

In forming an SNR scalable bitstream the following problem is formulated and solved. Let X be the set of original (unquantized) DCT coefficients in a frame and C the set of quantization levels that results from the quantization of X with quantization parameter QP. If X_i is a DCT coefficient, C_i, the corresponding quantization level (for inter blocks) is given by

$$C_i = (|X_i| - QP/2)\,/\,(2 \times QP) \tag{1}$$

where the $/$ operation denotes integer division with truncation toward zero. A similar equation is used for intra blocks. The decoder receives the set of quantization levels C and converts it into a set of "dequantized" DCT coefficients $\hat{X}$. The value of a "dequantized" coefficient $\hat{X}_i$ is given by [1]

$$\hat{X}_i = \begin{cases} 0, & \text{if } C_i = 0 \\ 2 \times QP \times C_i + QP, & \text{if } C_i \neq 0 \text{ and } QP \text{ odd} \\ 2 \times QP \times C_i + QP - 1, & \text{if } C_i \neq 0 \text{ and } QP \text{ even.} \end{cases} \tag{2}$$

Our goal is, given a set of DCT coefficients X with corresponding quantization levels C and "dequantized" values $\hat{X}$, to find a set of quantization levels $\tilde{C}$ by subtracting a certain value l_i from each coefficient quantization level C_i, so that a bit constraint is satisfied. The value l_i can be different for each coefficient quantization level C_i. The set of "dequantized" values that corresponds to $\tilde{C}$ is $\tilde{X}$. We will call $\tilde{X}$ a *trimmed* version of $\hat{X}$. The set of quantization levels $\tilde{C}$ is transmitted as the base layer (along with motion vectors and overhead information). Then, given a bit budget for the base layer, our problem is to find $\tilde{C}$ as the solution to the constrained problem

$$\min_{\tilde{C}} \left[D\left(X, \tilde{X}\right) | C \right] \text{ subject to } R\left(\tilde{C}\right) \leq R_{\text{budget}} \quad (3)$$

where $D(.,.)$ and $R(.)$ are the distortion and rate functions, respectively and R_{budget} is the available bit budget for the base layer.

The problem of (3) can be solved using Lagrangian relaxation. The problem now becomes the minimization of the Lagrangian cost

$$J(\lambda) = D\left(X, \tilde{X}\right) + \lambda R\left(\tilde{X}\right) \quad (4)$$

and the specification of the Lagrange multiplier λ so that the budget constraint is satisfied.

Without lack of generality, in our implementation of the algorithm, we determine a bit budget for the base layer for a group of blocks (GOB). This is done because an outside rate control mechanism updates the quantization parameter (QP) at the beginning of each GOB and thus determines the total available bit budget for the GOB (for all scalable layers). The bit budget for the base layer is a fixed percentage of the total available bit budget for the GOB. This percentage is determined by the target bit rates for each scalable layer. In H.263 with QCIF-sized frames, one GOB consists of one line of 16×16 macroblocks (11 macroblocks). Each macroblock consists of four luminance and two chrominance 8×8 blocks. Since the encoding of the DCT coefficients is done independently for each block (except for the dc coefficient of intra blocks which is differentially encoded and transmitted with the base layer anyway), $J(\lambda)$ is expressed as the sum of individual Lagrangian costs (one for each block) and the minimization is performed individually for each block, using the same λ [10], [11]. Then, if the bit budget for the whole GOB is met for a specific λ, we are guaranteed that the minimization of the individual Lagrangian costs results in an optimal bit allocation across the whole GOB. The λ for which the bit budget is met is found iteratively. A large λ results in a point in the rate-distortion curve with low rate and high distortion. Conversely, a small λ results in a point with high rate and low distortion. Therefore, a simple method, such as bisection, can be used to find the desired λ iteratively. More sophisticated algorithms, such as, the fitting of a Bezier curve [11], can also be used.

The problem now reduces to finding the set of quantization levels $\tilde{C}$ and corresponding trimmed DCT coefficients $\tilde{X}$ for every block that would minimize the Lagrangian cost of the block

$$J_{\text{block}}(\lambda) = D_{\text{block}}\left(X, \tilde{X}\right) + \lambda R_{\text{block}}\left(\tilde{X}\right) \quad (5)$$

for a given λ. The admissible candidate set $\tilde{C}$ is constructed as follows. Each nonzero coefficient in the block with quantization level C_i is either dropped completely or a value $l_i < C_i$ is subtracted from it. Although there is a finite number of admissible sets $\tilde{C}$, the minimization of the Lagrangian cost in (5) using exhaustive search is computationally prohibitive. The problem has however a structure which can be exploited using dynamic programming (DP) for its solution, as will be described in the next section.

Algorithms have been proposed in [12] for quality enhancement in JPEG or MPEG and in [13], [14] for the optimization of the enhancement layer in SNR scalability as defined in MPEG-2. In [12] the objective is to improve the SNR of a JPEG image by using a finer quantizer and dropping the unimportant coefficients instead of using a coarser quantizer that would yield the same bit rate. In [13] the goal is to adjust the DCT coefficients of the enhancement layer (which is the differential image, as defined in the standard) in order to make its encoding more efficient. In this work, the objective is, given the quantized DCT coefficients, to define the partitioning that will yield the optimal rate-distortion performance. Thus, although similar mathematical tools are used, the application is significantly different from [12] and [13].

IV. Dynamic Programming Solution

As mentioned in the previous section, the 2-D DCT coefficients are ordered in one dimension using the zig-zag scan and encoded using Variable Length Codes (VLC's) that correspond to the triplets (LEVEL, RUN, LAST). Let us assume for a moment that the coefficients are coded using pairs (LEVEL, RUN), i.e., the same VLC is used whether the coefficient is the last nonzero coefficient in the block or not. We will explain the modifications to the algorithm for (LEVEL, RUN, LAST) later. Then, suppose that we consider the problem of minimizing the Lagrangian cost given that coefficient k is the last nonzero coefficient in the block to be coded and coefficients $k+1$ to 63 are all thresholded to zero. Assuming that we have the solution to this problem, it can be used to solve the problem when coefficient k' is the last nonzero coefficient, where $k' > k$.

In order to see this, let us consider the problem of trimming the first k DCT coefficients in a block in order to minimize the Lagrangian cost (henceforth referred to as Problem 1). The rest of the DCT coefficients in the block are thresholded to zero and they are therefore not included in the base layer. Let us assume that the solution of Problem 1 results in minimum Lagrangian cost J_k^*. Let us also consider the problem of thresholding the first k' DCT coefficients in the block ($k' > k$) in order to minimize the Lagrangian cost. Let us also assume that coefficient k' is the last nonzero coefficient in the block and coefficient k is the penultimate nonzero coefficient in the block (Problem 2). Let the minimum Lagrangian cost in Problem 2 be $J_{k'}^*$. Then, $J_{k'}^* = J_k^* + \Delta J_{k,k'}^{l_{k'}}$ where $\Delta J_{k,k'}^{l_{k'}}$ is the difference in cost caused by including coefficient k' in the block and subtracting $l_{k'}$ from it. Thus, the solution of Problem 1 and the corresponding minimum Lagrangian cost J_k^* can be used to solve Problem 2. To prove this, let us assume that the minimum Lagrangian cost of Problem 2 is $J_{k'}^* = J_k' + \Delta J_{k,k'}^{l_{k'}}$. However, this would mean that the minimum Lagrangian cost of Problem 1 should be J_k'. Thus, $J_k' = J_k^*$. Therefore, the solution to the smaller problem can be used as part of the solution to a larger problem. This is a characteristic of problems which can be solved using DP techniques. We next describe in detail the proposed algorithm.

Let us utilize the incremental Lagrangian cost $\Delta J_{j,k}^{l_k}$ as the difference in the cost incurred by including coefficient k trimmed by l_k in the base layer when the previous nonzero coefficient is j. It is defined by

$$\Delta J_{j,k}^{l_k} = -E_k^{l_k} + \lambda R_{j,k}^{l_k} \text{ for } j < k \tag{6}$$

where $E_k^{l_k}$ represents the difference in distortion incurred by including coefficient k and is defined by $E_k^{l_k} = X_k^2 - (X_k - \tilde{X}_k^{l_k})^2$, X_k is the original kth unquantized coefficient and $\tilde{X}_k^{l_k}$ is the "dequantized" coefficient which corresponds to quantization level $\tilde{C}_k^{l_k} = C_k - l_k$, with C_k the original quantization level and $R_{j,k}^{l_k}$ is the rate (in bits) that would be required to encode quantization level $\tilde{C}_k^{l_k}$ given that the previous nonzero coefficient was coefficient j.

If coefficient k is dropped completely, the contribution toward the total mean squared error is X_k^2. If the quantized and trimmed coefficient is transmitted instead, the contribution toward the total mean squared error is $(X_k - \tilde{X}_k^{l_k})^2$. Therefore, $E_k^{l_k}$ represents the difference in mean squared error between dropping coefficient k from the base layer and transmitting the quantized coefficient trimmed by l_k. Since the error is initialized as if all coefficients are dropped, i.e., for the kth frequency index the error is equal to X_k^2, the inclusion of the trimmed by l_k kth coefficient will increase the error by E_k^2, that is the resulting error is equal to $X_k^2 - E_k^{l_k} = (X_k - \tilde{X}_k^{l_k})^2$.

$\Delta J_{j,k}^{l_k}$ therefore represents the incremental Lagrangian cost of going from coefficient j to coefficient k (dropping the coefficients between them) and subtracting l_k from quantization level C_k. The algorithm keeps track of the minimum Lagrangian cost for each coefficient k assuming that it is the last coefficient to be coded in the block. We will denote this cost as J_k^*.

If we drop all ac coefficients of an intra block, the rate will be zero and the distortion will be equal to

$$J_0^* = E_{\text{intra}} = \sum_{i=1}^{63} X_i^2 \tag{7}$$

since the DCT transform is unitary and we can therefore calculate the mean squared error in either the spatial or the frequency domain. For inter blocks, we allow for the possibility of dropping all coefficients, including the dc. Then, we define

$$J_{-1}^* = E_{\text{inter}} = \sum_{i=0}^{63} X_i^2. \tag{8}$$

As mentioned earlier, we need to take into account the fact that different VLC's are used depending on whether the coefficient to be encoded is the last one in the block or not. Therefore, we define a second incremental cost $\Delta J_{j,k,\text{last}}^{l'_k} = -E_j^{l'_k} + \lambda R_{j,k,\text{last}}^{l'_k}$, where $R_{j,k,\text{last}}^{l_k}$ is the number of bits that are required to encode quantization level $\tilde{C}_k^{l_k}$ given that j was the previous nonzero coefficient and coefficient k is the last one to be encoded in the block. We also keep the minimum Lagrangian costs $J_{k,\text{last}}^*$ for each coefficient k given that it is the last coefficient to be coded in the block.

V. Proposed Rate Distortion Optimal SNR Scalable Coder

We are now ready to give the details of the algorithm. The algorithm is recursive and stores the minimum Lagrangian costs for the block when coefficient k is the last nonzero coefficient in the block, where $k = 0, \ldots, 63$ for inter blocks and $k = 1, \ldots, 63$ for intra blocks. In the following discussion, we consider the case for inter blocks. For intra blocks, the only difference is that the recursion starts at $k = 1$ instead of $k = 0$.

The recursion begins with "coefficient" -1 which means that the imaginary coefficient -1 is the last coefficient in the block to be coded and all coefficients between 0 and 63 are dropped from the base layer. The cost of dropping all coefficients is stored as J_{-1}^* and is given by (8). Then, we proceed to find the minimum cost path that ends in coefficient 0 $(k = 0)$. Clearly, this means that coefficient 0 will be kept and all others will be dropped but we need to find what value l_0 will be subtracted from its quantization level. That l_0 is the one which minimizes the expression $J_{-1}^* + \Delta J_{-1,0}^{l_0}$. The resulting cost is the minimum cost when coefficient 0 is the last one to be encoded in the block and is equal to J_0^*. We also need to perform the same procedure using $\Delta J_{-1,0,\text{last}}^{l'_0}$ instead of $\Delta J_{-1,0}^{l_0}$. Thus, we compute $J_{0,\text{last}}^* = \min_{l'_0} \{J_{-1}^* + \Delta J_{-1,0,\text{last}}^{l'_0}\}$ where l'_0 is not necessarily the same as l_0.

For $k = 1$, we can either keep coefficients 0 and 1 or just coefficient 1. Again, we need to determine the value to be subtracted from C_1. Now, the minimum cost will be $J_1^* = \min_{i,l_1} J_i^* + \Delta J_{i,1}^{l_1}$, for $i = -1, 0$. We also need to calculate $J_{1,\text{last}}^*$ in a similar manner.

For a general k, the minimum costs are found as

$$J_k^* = \min_{i,l_k} \left\{ J_i^* + \Delta J_{i,k}^{l_k} \right\}, \text{ for } i = -1, \ldots, k-1 \tag{9}$$

and

$$J_{k,\text{last}}^* = \min_{i,l'_k} \left\{ J_i^* + \Delta J_{i,k,\text{last}}^{l'_k} \right\}, \text{ for } i = -1, \ldots, k-1. \tag{10}$$

The algorithm calculates J_k^* and $J_{k,\text{last}}^*$ for all $k = 0, \ldots, 63$ and also stores the last nonzero coefficients (predecessors) i and the subtracted values l_k and l'_k which minimize (9) and (10), respectively. The k which results in the minimum $J_{k,\text{last}}^*$ will be denoted as k^*. Clearly, $J_{k^*,\text{last}}^*$ is equal to the minimum Lagrangian cost J_{block} for the whole block. Therefore, we know that coefficient k^* will be included in the base layer and we look up the value to be subtracted from it. Then, we look up the optimal predecessor i which resulted in $J_{k^*,\text{last}}^*$. Let us denote this coefficient as k_1. Then, k_1 will be included in the base layer and the value to be subtracted from it is looked up. Then we look up

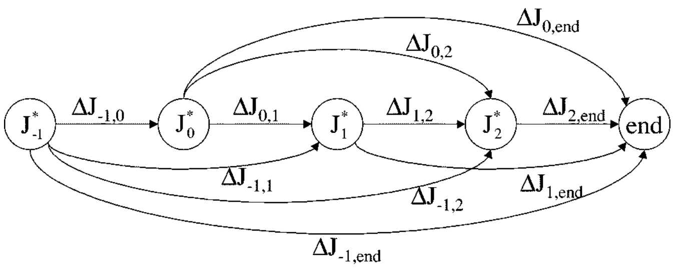


Fig. 3. Directed acyclic graph representation of the optimal DCT coefficient partitioning problem.

the predecessor that resulted in $J^*_{k_1}$ and continue recursively in the same fashion until we arrive at the imaginary coefficient -1.

"Pruning" of nonoptimal predecessors i in (9) and (10) can be performed, that is based on the observation that in the standard H.263 VLC table (and in practice, most custom made VLC tables), the number of bits $R^{l_j}_{i,j}$ required for the encoding of coefficient j given that the previous nonzero coefficient was coefficient i is monotonically increasing with the zero run length $j-i-1$. Therefore, for $i < j$, $\Delta J^{l_k}_{i,k} \geq \Delta J^{l_k}_{j,k}$ and $\Delta J^{l'_k}_{i,k,\mathrm{last}} \geq \Delta J^{l'_k}_{j,k,\mathrm{last}}$. Thus, if $J^*_i > J^*_j$, then $J^*_i + \Delta J^{l_k}_{i,k} > J^*_j + \Delta J^{l_k}_{j,k}$ and $J^*_i + \Delta J^{l'_k}_{i,k,\mathrm{last}} > J^*_j + \Delta J^{l'_k}_{j,k,\mathrm{last}}$, for every l_k and l'_k. Thus, i cannot be an optimal predecessor to k. We will denote the set of coefficients to be considered as optimal predecessors of coefficient k as S_k.

It is interesting to point out that the proposed algorithm is equivalent to finding the shortest path in an directed acyclic graph (DAG). Fig. 3 shows a DAG for the case of just three DCT coefficients (instead of 64). The vertices of the DAG correspond to the Lagrangian costs J^*_k while the edges correspond to the differential costs $\Delta J_{i,k}$. For simplicity, in this graph we assume that coefficients can either be included or dropped from the base layer, i.e., no "trimming" is involved. The first vertex takes the value $J^*_{-1} = E$, where E is equal to E_{intra} or E_{inter}, depending on the type of the macroblock. The last vertex designated as "end" is needed to show that the last coefficient for the block has been encoded. Clearly, $\Delta J_{i,\mathrm{end}} = 0$ for all i. The solution of finding the shortest path of the DAG is exactly the algorithm we described.

The computational complexity of the proposed algorithm heavily depends on how successful the pruning is and how many iterations are needed to find the appropriate λ for each block for a particular video sequence. Our implementation of the algorithm shows that, although our encoder is slower than the H.263+ encoder, its computational complexity is reasonable (the current nonoptimized implementation of the proposed encoder is about five times slower than the UBC implementation of H.263+ encoder, which does not use rate control for the enhancement layer). Our goal in designing this codec was to have a low complexity decoder (which only requires a single inverse DCT step) at the expense of higher encoder complexity. If both an encoder and decoder with lower computational complexity than H.263+ are required, an algorithm employing a heuristic selection of DCT coefficients as in [4] and [6] can be used at the expense of lower video quality.

VI. Extension of the Algorithm to a Multiple Number of Layers

We have thus far presented an optimal algorithm for partitioning a set of quantized DCT coefficients into two layers. A bit budget is set for the base layer and the outside rate control is responsible for maintaining the total target bit rate for all layers. We now extend the algorithm to more than two layers. As described earlier, in order to partition the DCT coefficients into more than two layers, we first perform the partitioning into two layers as described in the previous section. We then partition the enhancement layer into two layers making the total number of layers equal to three. Now, the problem is formulated as follows. Given the quantized DCT coefficients and the partitioning into two layers, partition the coefficients of the enhancement layer (of the original partitioning) into two layers such that the distortion between the original unquantized coefficients and the coefficients reconstructed using the new base layer and the new first enhancement layer is minimized. The minimization is subject to a bit budget constraint for the first enhancement layer. Let us assume that we have already partitioned the DCT coefficients into two layers and the set of coefficient quantization levels for the second layer is C_{enh}. We now want to partition C_{enh} into two sets of coefficients, namely C_2 and C_3. The coefficients of the base layer C_1 have already been selected during the partitioning of the coefficients into two layers. Let X_{1+2} be the "dequantized" DCT coefficients when the first two of the three layers are utilized. Then, our problem is to choose the coefficient quantization levels C_2 that will make up scalable layer 2 (first enhancement layer) such that

$$\min_{C_2} \left[D\left(X, X_{1+2}\right) | C_{\mathrm{enh}}\right] \text{ subject to } R\left(C_2\right) \leq R_{\mathrm{budget},2} \tag{11}$$

where $R_{\mathrm{budget},2}$ is the bit budget for scalable layer 2 (first enhancement layer).

The above problem can be solved using the algorithm already described. The only difference is in the definition of E^l_k and J^*_{-1}. The value of J^*_{-1} should be equal to the distortion that occurs when no coefficients are selected for layer 2 (the first enhancement layer). Therefore, the distortion will be the distortion incurred when including only the base layer, that is,

$$J^*_{-1} = E = \sum_{i=0}^{63} \left(X_i - \tilde{X}_i\right)^2 \tag{12}$$

where $\tilde{X}$ is the coefficients selected for the base layer as determined by the partitioning of the DCT coefficients into two layers. It should be pointed out that since in the case of intra blocks the dc coefficient goes to the base layer, for this part of the algorithm there is no distinction between intra and inter blocks. Thus, the first Lagrangian cost is J^*_{-1} and the recursion starts at $k = 0$ and not $k = 1$.

The second difference is in the definition of $E^{l_k}_k$. Now we have

$$E^{l_k}_k = \left(X_k - \tilde{X}_k\right)^2 - \left(X_k - X^{l_k}_{1+2,k}\right)^2 \tag{13}$$

where $X^{l_k}_{1+2,k}$ is the "dequantized" coefficient k when including layers 1 and 2 (the base layer and the first enhancement layer).

TABLE I
COMPARISON OF THE AVERAGE PSNR OF THE PROPOSED ALGORITHM AND THE H.263 STANDARD SNR SCALABILITY ALGORITHM AT 14–18 KBPS

Bit rate (kbps)	14	18
Proposed Algorithm PSNR (Akiyo)	36.41	36.62
UBC Codec PSNR (Akiyo)	35.41	35.59
Proposed Algorithm PSNR (Foreman)	28.59	29.32
UBC Codec PSNR (Foreman)	28.49	28.97

TABLE II
COMPARISON OF THE AVERAGE PSNR OF THE PROPOSED ALGORITHM AND THE H.263 STANDARD SNR SCALABILITY ALGORITHM AT 28.8–56 KBPS

Bit rate (kbps)	28.8	56
Proposed Algorithm PSNR (Akiyo)	39.57	40.64
UBC Codec PSNR (Akiyo)	38.68	40.33
Proposed Algorithm PSNR (Foreman)	30.29	31.98
UBC Codec PSNR (Foreman)	30.06	32.24

Layer 2 has been constructed by subtracting value l_k from the quantization index of $C_{\rm enh}$.

The partitioning of the DCT coefficients into more layers is done by repeatedly partitioning the enhancement layer into two layers. If we want to partition the coefficients into n layers assuming that they are already partitioned into $n-1$ layers, we need to solve the problem of finding the set of quantization levels for layer $n-1$, C_{n-1} such that

$$\min_{C_{n-1}} \left[D\left(X, X_{1+2+\cdots+n-1}\right) | C_{\rm enh}\right] \quad \text{subject to } R\left(C_{n-1}\right) \le R_{{\rm budget},n-1} \tag{14}$$

where now $C_{\rm enh}$ is the (last) enhancement layer of the partitioning into $n-1$ layers and $X_{1+2+\cdots+n-1}$ is the set of "dequantized" DCT coefficients when the first $n-1$ layers are utilized.

Similarly to the case of three layers, we define J^*_{-1} and $E_k^{l_k}$ as

$$J^*_{-1} = E = \sum_{i=0}^{63} \left(X_i - X_{1+2+\cdots+n-1,i}\right)^2 \tag{15}$$

and

$$E_k^{l_k} = \left(X_k - X_{1+2+\cdots+n-2,k}\right)^2 - \left(X_k - X^{l_k}_{1+2+\cdots+n-1,k}\right)^2 \tag{16}$$

where $X_{1+2+\cdots+n-2,k}$ is the "dequantized" coefficient k when layers up to $n-1$ are used.

VII. EXPERIMENTAL RESULTS

We tested the above algorithm with the "Akiyo" and "Foreman" sequences and compared it with results obtained using the H.263+ public domain codec from the University of British Columbia (UBC) [2]. The results are shown in Table I for a base layer bit rate of 14 kbps and an enhancement layer of a total bit rate of 18 kbps and in Table II for a base layer bit rate of 28.8 kbps and an enhancement layer of 56 kbps. The original frame rate of these sequences is 30 frames per second and the original length is 300 frames (10 s). The resulting encoded frame rate is close to 8 frames per second in all cases. The peak signal-to-noise ratio (PSNR) reported was calculated as the average PSNR of all components (Y, C_b, C_r) and all encoded frames. As can be seen from the Appendix, in order to reduce the computational complexity of the proposed algorithm, the maximum subtracted value l_j for any coefficient j is L where, in our simulation, $L = 5$. As can be seen in step 5 of the Appendix, this reduces the number of comparisons required to find J^*_k and $J^*_{k,{\rm last}}$. We can see that the proposed algorithm outperforms H.263 in the case of the "Akiyo" sequence while for the "Foreman" sequence, the results are comparable.

Fig. 4. Frame 122 of the "Foreman" sequence encoded using the optimal single-pass codec at 28.8–56 kbps (28.8 kbps layer).

Fig. 5. Frame 122 of the "Foreman" sequence encoded using the optimal single-pass codec at 28.8–56 kbps (56 kbps layer).

Figs. 4 and 5 show representative frames of the "Foreman" sequence encoded using the optimal single-pass codec at 28.8–56 kbps. Fig. 6 shows a plot of the PSNR versus frame number for both the proposed codec and the UBC codec. It can be seen that, although the average PSNR is similar for the two codecs (see Table II), the PSNR of the proposed codec is higher than the PSNR of the UBC codec for most frames for both layers but it is lower around frames 160–230. Fig. 7 shows a similar plot for the "Akiyo" sequence. The public domain version of the UBC codec provides rate control for the base layer but not for the enhancement layer. Thus, an appropriate constant QP was used for the enhancement layer so that the average target bit rate was met. In the case of the "Akiyo" sequence, the constant QP used was significantly smaller than the QP used for frame 0 using the proposed method. For this reason, the UBC codec exhibits a higher PSNR for the enhancement layer of the first several frames. As can be seen in Table II, the proposed algorithm has a higher average PSNR.

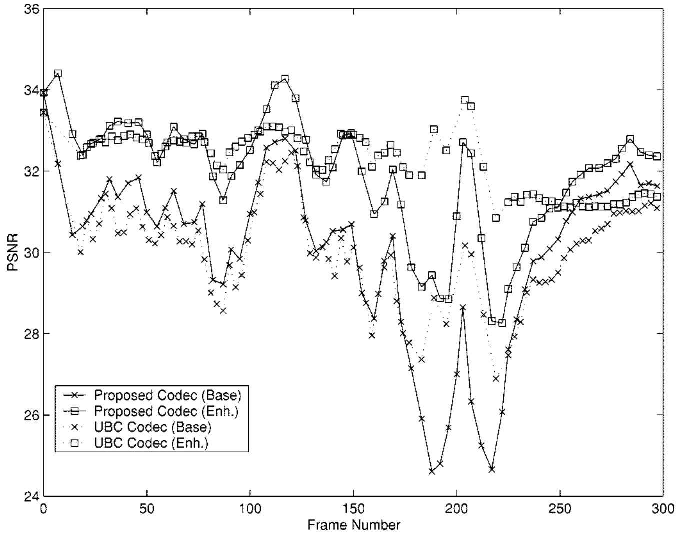


Fig. 6. PSNR versus frame number plot for the "Foreman" sequence encoded at 28.8–56 kbps.

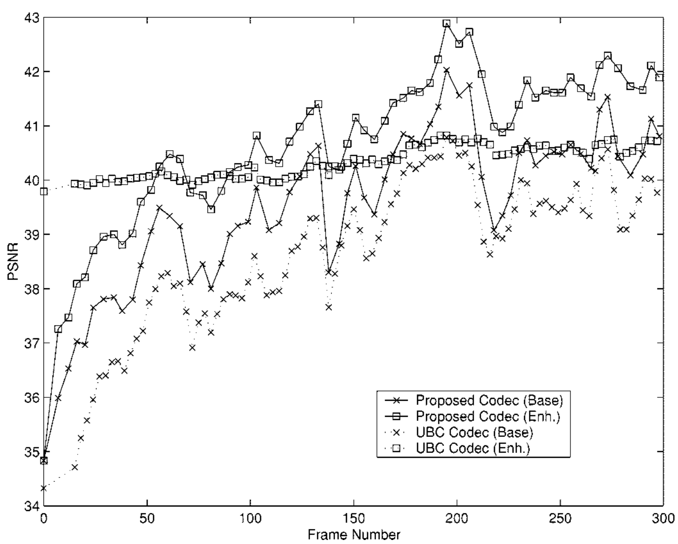


Fig. 7. PSNR versus frame number plot for the "Akiyo" sequence encoded at 28.8–56 kbps.

We also implemented the algorithm for the case of three scalable layers and compared us with results of the codec that utilizes the partitioning of DCT coefficients as in Fig. 1 [6]. A heuristic algorithm was used for the selection of the three control parameters A, B and X. The bit rates of 14, 18, and 22 kbps were used. The results are shown in Table III. It can be seen that the results of the optimal algorithm proposed here are better than those of the heuristic partitioning of Fig. 1, at the expense of increased computational complexity at the encoder.

The public domain implementation of H.263+ from UBC [2] only supports two layers while our implementation of the codec in [6] only supports three layers. This is the reason why the results of the two-layer version of our algorithm were compared with the former while our three-layer results were compared with the latter.

TABLE III
COMPARISON OF THE AVERAGE PSNR OF THE PROPOSED ALGORITHM AND THE CODEC WITH DCT COEFFICIENT PARTITIONING AS IN FIG. 1 AT 14–18–22 KBPS

Bit rate (kbps)	14	18	22
Proposed Algorithm PSNR (Akiyo)	35.93	36.10	36.40
Partitioning as in Fig. 1 PSNR (Akiyo)	34.39	34.66	35.08
Proposed Algorithm PSNR (Foreman)	28.06	28.79	29.57
Partitioning as in Fig. 1 PSNR (Foreman)	27.56	27.93	29.11

VIII. CONCLUSIONS

In this paper, we proposed an algorithm for SNR video scalability which is based on the partitioning of the DCT coefficients into layers. The partitioning is done in an optimal manner. An important advantage of the proposed algorithm is that it requires only a single DCT and quantization operation and a smaller bit overhead. Although the optimization algorithm increases the computational complexity of the encoder in comparison with the algorithm in [4], the complexity of the decoder is still the same as that of a nonscalable decoder. Experimental results show that the proposed algorithm performs at least as good as the H.263 scalable codec depending on the type of the video sequence and the target bit rates. We presented results using the "Foreman" sequence, a typical high-motion sequence and the "Akiyo" sequence, a typical low-motion sequence. The algorithm gives results that are similar with H.263 for high-motion sequences while it clearly outperforms it for lower motion sequences. The results also depend on the ratios between the target bit rates for the scalable layers. This is the case because the QP is determined by an outside rate control algorithm based on the total target bit rate (the sum of all layers). Thus, in the results we presented for three layers (28.8–56–128 kbps), the 28.8 layer has a lower PSNR than in the case of two layers (28.8–56 kbps) although the bit rate is the same. This is because in the three layer case, the quantizer was adjusted in order to provide a total bit rate of 128 kbps while in the two layer case the total target bit was 56 kbps. The quality of the base layer depends on the ratio of the target bit rate for the base layer over the total target bit rate. In our example, the 28.8/56 ratio gives better base layer quality than the 28.8/128 ratio. However, in both cases, the results are acceptable.

The methodology proposed is general and can be applied to other scalability problems.

APPENDIX

The pseudocode for the optimization algorithm is given below. For a given λ, the algorithm for minimizing J_{block} is as follows.

1) For current λ, compute $\Delta J_{i,j}^{l_j} = -E_j^{l_j} + \lambda R_{i,j}^{l_j}$ and $\Delta J_{i,j,\text{last}}^{l_j} = -E_j^{l_j} + \lambda R_{i,j,\text{last}}^{l_j}$ for all pairs (i,j) of nonzero coefficients with $j > i$ and $l_j = 0, \ldots, L$.
2) If current macroblock is intra: $k^* = 0$, $k = 0$, $S_0 = \{0\}$, $J_0^* = E_{\text{intra}}$, predecessor(0) = NULL, lastpredecessor(0) = NULL.
   If current macroblock is inter: $k^* = -1$, $k = -1$, $S_{-1} = \{-1\}$, $J_{-1}^* = E_{\text{inter}}$, predecessor$(-1)$ = NULL, lastpredecessor(-1) = NULL.
3) $k = k + 1$. If $k = 64$, go to step 7.

4) if $E_k^0 = 0$, set $S_k = S_{k-1}$ and go to step 3.
5) $J_k^* = \min_{i \in S_{k-1}, l_k=0,\ldots,L} [J_i^* + \Delta J_{i,k}^{l_k}]$; sub$(k) = l_k$ (the l_k that minimized the previous expression).
$J_{k,\mathrm{last}}^* = \min_{i \in S_{k-1}, l_k'=0,\ldots,L} [J_i^* + \Delta J_{i,k,\mathrm{last}}^{l_k'}]$; sublast$(k) = l_k'$ (the l_k' that minimized the previous expression). If $J_{k,\mathrm{last}}^* \leq J_{k^*,\mathrm{last}}^*$, then $k^* = k$.
6) $S_k = \{k\} \cup \{i | i \in S_{k-1}$ and $J_i^* \leq J_k^*\}$. Go to step 3.
7) subvalues$(0..63) = 0$, keep$(0..63) = 0$.
8) $k = k^*$, subvalues$(k^*) =$ sublast(k^*), keep$(k^*) = 1$.
9) $k =$ lastpredecessor(k^*), subvalues$(k) =$ sub(k), keep$(k) = 1$.
10) while $(k =$ predecessor$(k)) \neq$ NULL, subvalues$(k) =$ sub(k), keep$(k) = 1$.

The output of the algorithm is arrays keep$(.)$ and subvalues$(.)$, where keep(i) is equal to 0 if coefficient i is not included in the base layer and 1 if it is. In that case, subvalues(i) is equal to the value l that is subtracted from its quantization index.

The pruning of the suboptimal predecessors is performed in step 6. Step 6 of the appendix determines the predecessor candidates for the next iteration (predecessors of coefficient $k-1$). The only allowable predecessors are those whose cost J_i^* is smaller than the cost J_k^*.

REFERENCES

[1] "Recommendation H.263: Video coding for low bitrate communications," Int. Telecom. Union, 1998.

[2] "TMN Version 3.2. H.263+ Public domain implementation," Univ. British Columbia.

[3] R. Aravind, M. R. Civanlar, and A. R. Reibman, "Packet loss resilience of MPEG-2 scalable video coding algorithms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 426–435, Oct. 1996.

[4] M. A. Robers, L. P. Kondi, and A. K. Katsaggelos, "SNR scalable video coder using progressive transmission of DCT coefficients," *Proc. SPIE*, pp. 201–212, 1998.

[5] *Generic coding of moving pictures and associated audio*, Int. Std. ISO/IEC IS-13 818, Nov. 1994.

[6] L. P. Kondi, F. Ishtiaq, and A. K. Katsaggelos, "On video SNR scalability," in *Int. Conf. Image Processing*, Chicago, IL, 1998, pp. 934–938.

[7] "Information technology-digital compression and coding of continuous-tone still images (JPEG)," International organization for standardization, Recommend. T.84, 1994.

[8] L. P. Kondi and A. K. Katsaggelos, "An optimal single pass SNR scalable video coder," in *Int. Conf. Image Processing*, Kobe, Japan, 1999.

[9] L. P. Kondi, "Low bit rate SNR scalable video coding and transmission," Ph.D., Northwestern Univ., Evanston, IL, 1999.

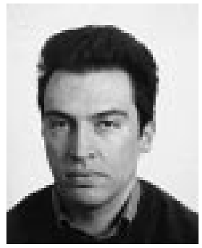
[10] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Processing*, vol. 3, pp. 533–545, Sept. 1994.

[11] G. M. Schuster and A. K. Katsaggelos, *Rate-Distortion Based Video Compression, Optimal Video Frame Compression and Object Boundary Encoding*. Norwell, MA: Kluwer, 1997.

[12] K. Ramchandran and M. Vetterli, "Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility," *IEEE Trans. Image Processing*, vol. 3, pp. 700–704, Sept. 1994.

[13] D. Wilson and M. Ghanbari, "Optimization of two-layer SNR scalability for MPEG-2 video," in *Int. Conf. Acoustics, Speech, Signal Processing*, 1997, pp. 2637–2640.

[14] ——, "Optimization of MPEG-2 SNR scalable codecs," *IEEE Trans. Image Processing*, vol. 8, pp. 1434–1438, Oct. 1999.



Lisimachos P. Kondi (S'92–M'99) received the Diploma degree in electrical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, in 1994 and the M.S. and Ph.D. degrees, both in electrical and computer engineering, from Northwestern University, Evanston, IL, in 1996 and 1999, respectively.

In August 2000, he joined the Department of Electrical Engineering, State University of New York at Buffalo, Buffalo, NY, as Assistant Professor. During the summer of 2001, he was a U.S. Navy-ASEE Summer Faculty Fellow at the Naval Research Laboratory, Washington, DC. His current research interests include video compression, wireless communications, joint source/channel coding, multimedia signal processing and communications, image restoration, resolution enhancement, and boundary encoding.

Aggelos K. Katsaggelos (S'80–M'85–SM'92–F'98) received the Diploma degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, in 1979, and the M.S. and Ph.D. degrees, both in electrical engineering, from the Georgia Institute of Technology, Atlanta, in 1981 and 1985, respectively.

In 1985, he joined the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, where he is currently Professor, holding the Ameritech Chair of Information Technology. He is also the Director of the Motorola Center for Communications. During the 1986-1987 academic year, he was an Assistant Professor with the Department of Electrical Engineering and Computer Science, Polytechnic University, Brooklyn, NY. His current research interests include image and video recovery, video compression, motion estimation, boundary encoding, computational vision and multimedia signal processing and communications. He is the editor of *Digital Image Restoration* (Berlin, Germany: Springer-Verlag, 1991), co-author of *Rate-Distortion Based Video Compression* (Norwell, MA: Kluwer, 1997), and co-editor of *Recovery Techniques for Image and Video Compression and Transmission* (Norwell, MA: Kluwer, 1998). He is the coinventor of seven international patents.

Dr. Katsaggelos is an Ameritech Fellow, a member of the Associate Staff, Department of Medicine, Evanston Hospital, and a member of SPIE. He is a member of the Board of Governors and the Publication Board of the IEEE Signal Processing Society, the IEEE TAB Magazine Committee, the IEEE Technical Committees on Visual Signal Processsing and Communications and Multimedia Signal Processing, and is Editor-in-Chief of the IEEE SIGNAL PROCESSING MAGAZINE. He has served as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING (1990–1992), area editor for the journal *Graphical Models and Image Processing* (1992–1995), a member of the Steering Committees of the IEEE TRANSACTIONS ON IMAGE PROCESSING (1992–1997) and the IEEE TRANSACTIONS ON MEDICAL IMAGING (1990–1999), and a member of the IEEE Technical Committee on Image and Multidimensional Signal Processing (1992–1998). He served as the General Chairman of the 1994 Visual Communications and Image Processing Conference, Chicago, IL, and as technical program co-chair of the 1998 IEEE International Conference on Image Processing, Chicago, IL. He is the recipient of the IEEE Third Millennium Medal (2000).