AN OPTIMAL SINGLE PASS SNR SCALABLE VIDEO CODER

Lisimachos P. Kondi and Aggelos K. Katsaggelos

Northwestern University
Dept. of Electrical and Computer Engineering
2145 Sheridan Road
Evanston, IL 60208
E-Mail: {lkon,aggk}@ece.nwu.edu

ABSTRACT

In this paper, we introduce a new methodology for SNR video scalability which is based on the partitioning of the DCT coefficients. The partitioning is done in a way that is optimal in the Rate-Distortion sense. The optimization is performed using Lagrangian relaxation and Dymamic Programming (DP). Experimental results are presented and conclusions are drawn.

1. INTRODUCTION

A scalable video codec is defined as a codec that is capable of producing a bit stream 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 bit streams with different rates and reconstructed quality. A small subset of the original bit stream can be initially transmitted to provide a base layer quality with extra layers subsequently transmitted as enhancement layers.

There are three types of scalability supported in the H.263 video compression standard: SNR, spatial and temporal. 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.

A major 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 bit stream.

Another important application of scalability is in error resilience. It has been shown [1] that it is advantageous to use scalability and apply stronger error protection for the base layer than for the enhancement layers (Unequal Error Protection). Thus, we can almost guarantee a base quality even during adverse channel conditions. Had we not used scalability and protected the whole bit stream 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. This method is based on the optimal partitioning of the DCT coefficients of the Displaced Frame Difference (DFD). We also discuss other methodologies for SNR scalability and explain how our method differs from them.

2. METHODS FOR SNR SCALABILITY

The traditional method for SNR scalability as utilized by the video compression standards (MPEG-2, H.263) is as follows. The base layer is created by quantizing and encoding the DFD, as in a non-scalable encoder. Then, the base layer is reconstructed and the difference between the reconstructed and original frame is computed. This residual error is then transmitted in the same way as the DFD is transmitted in non-scalable video encoders. In order to produce more enhancement layers, the same procedure is repeated by reconstructing the enhanced frame and transmitting the new residual error. This method produces good results but requires an extra forward and inverse Discrete Cosine Transform (DCT) for each enhancement layer. Furthermore, since the residual error is transmitted like a regular frame, it also carries a significant bit overhead.

Another method we have proposed earlier [2, 3] involves only a single DCT and quantization operation.

This work was supported in part by a grant from the Motorola Center for Communications at Northwestern University.

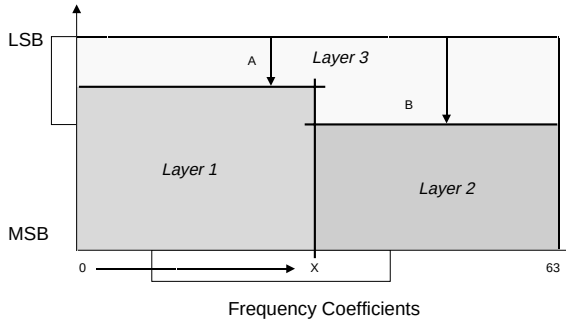

Figure 1: Partition of the DCT coefficients in three layers according to [2,3].

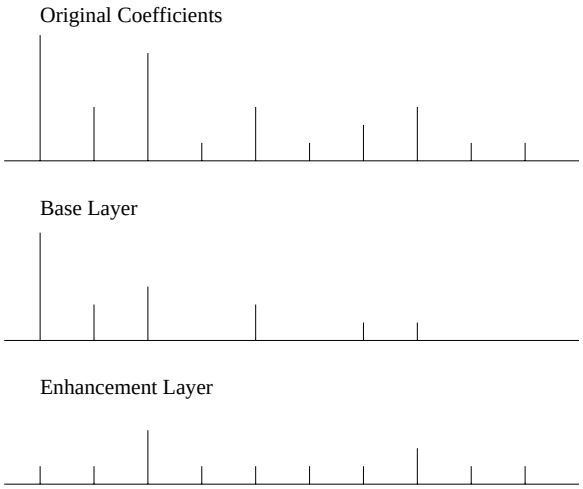

Figure 2: Proposed partitioning of DCT coefficients for SNR scalability.

According to it, the coefficients are partitioned in a number of sets which form the scalable layers. As shown in Fig. 1 for the case of three layers, the base layer involves the transmission of coefficients (actually their quantization levels) $0-X$ in a zig-zag scan, 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. The parameters X, A and B are adjusted by a rate control algorithm based on heuristics. This SNR scalability algorithm has a much lower computational complexity and overhead than the method supported by the standards and described in the previous paragraph. However, it makes the assumption that the DFD data are lowpass, which is not always true. Also, the three parameters X, A and B which control the rate give us few degrees of freedom. Subsequently, results obtained with this algorithm exhibit lower PSNR for the same bit rates than results obtained by the SNR scalability method described in H.263.

In this paper we propose a generalization of the method in [2, 3] outlined above. 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 fewer bits for smaller coefficient magnitudes. Thus, subtracting a value from a coefficient quantization level reduces the number of bits required for its transmission but clearly increases the distortion. The decoder reconstructs the quantized DCT coefficients by adding the subtracted values (if available) to the values it received with the base layer. These observations show us that we can propose a partitioning technique for the DCT coefficients that is much more general than the one proposed in the previous paragraph. The base layer is constructed by subtracting a value from each DCT coefficient. The subtracted values are then sent as enhancement (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 level 5 as base layer, 2 as first enhancement layer and 2 as second enhancement layer. Next, we present a formulation and optimal solution to the problem of partitioning the DCT coeficients of this scheme.

3. THE ALGORITHM

It is assumed that the DCT transform of the DFD (or the intensity for intra blocks) is taken and quantized according to the H.263 standard. Then, the proposed algorithm takes over to create the scalable layers. The quantized coefficients are coded as follows in H.263. A triplet (LEVEL,RUN,LAST) is transmitted using suitable VLC tables, where LEVEL is the quantization level, RUN is the number of zero-valued coefficients that precede it and LAST specifies whether the current coefficient is the last in the block.

The problem is formulated as follows. Given a set of DCT coefficients X and their quantized version $\hat{X}$, find a set $\tilde{X}$ by subtracting a certain value l from each coefficient quantization LEVEL, so that a bit constraint is satisfied. It should be pointed out that $\tilde{X}$ as well as $\hat{X}$ are the "dequantized" values and not the quantization levels. We will call $\tilde{X}$ a *trimmed* version of $\hat{X}$. Let us denote by C the set of the quantization levels that correspond to the quantized coefficients $\hat{X}$ and $\tilde{C}$ the set of quantization levels that correspond to trimmed coefficients $\tilde{X}$. 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 $\tilde{X}$ as the solution to the constrained problem

$$\min[D(X,\tilde{X})|\hat{X}] \text{ subject to } R(\tilde{X}) \leq R_{budget}, \quad (1)$$

where $D(.,.)$ and $R(.)$ are the distortion and rate functions, respectively.

The problem of Eq. 1 can be solved using Lagrangian relaxation. The problem now becomes the minimization of the Lagrangian cost

$$J(\lambda) = D(X,\tilde{X}) + \lambda R(\tilde{X}). \quad (2)$$

In our implementation of the algorithm, we determine a bit budget for the base layer for a whole Group of Blocks (GOB). 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 at each block (except for the dc coefficient of intra blocks which is transmitted with the base layer anyway), it is well-known that it is possible to express $J(\lambda)$ as a sum of individual Lagrangian costs (one for each block) and perform the minimization individually for each block. The same λ should be used for all blocks. 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 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

$$J_{block} = D_{block} + \lambda R_{block} \quad (3)$$

for a given λ. We assume that each admissible candidate set $\tilde{C}$ is constructed as follows. Each non-zero coefficient in the block with quantization level u is either dropped completely or a value l where $l = 0, \ldots, L$ is subtracted from it. Clearly, $l < u$ and L is an appropriate constant. Therefore, there is a finite number of admissible sets $\tilde{C}$ and the minimization of the Lagrangian cost in Eq. 3 could be done using exhaustive search. However, the computational cost would be prohibitive. Luckily, the problem has a structure can be exploited using a Dynamic Programming (DP) solution. The Dynamic Programming algorithm is described in the next section. A similar algorithm has been proposed in [4] for the quality enhancement in JPEG or MPEG and in [5] for the optimization of the enhancement layer in SNR scalability as defined in MPEG-2. In this work, however, the same principles are applied to a completely different context.

3.1. Dynamic Programming Algorithm

As noted previously, the 2-D DCT coefficients are ordered in one dimension using the zig-zag scan and encoded using Variable Length Codes (VLCs) 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 non-zero coefficient in the block or not. We will explain the modifications to the algorithm for (LEVEL,RUN,LAST) later. Then, let us suppose that we consider the problem of minimizing the Lagrangian cost given that coefficient k is the last non-zero coefficient in the block to be coded and coefficients $k+1$ to 63 are all dropped from the base layer. Assuming that we have the solution to this problem where k is the last non-zero coefficient, this solution can be used to help us solve the problem of coefficient k' being the last one, where $k' > k$. Clearly, in the problem where k' is the last coefficient, if we determine that coefficient k should be included, then all coefficients between 0 and k would be trimmed as in the problem where k was the last coefficient. 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 Dynamic Programming techniques. We next describe the basic characteristics of the algorithm.

The proposed algorithm uses incremental Lagrangian costs $\Delta J_{j,k}$ which are defined as

$$\Delta J^l_{j,k} = -E^l_k + \lambda R^l_{j,k} \text{ for } j < k \text{ and } l = 0, \ldots, L. \quad (4)$$

In the above equation,

$$E^l_k = X^2_k - (X_k - \tilde{X}^l_k)^2 \quad (5)$$

where X_k is the original kth unquantized coefficient and $\tilde{X}^l_k$ is the quantized coefficient which corresponds to quantization level $\tilde{C}^l_k = C_k - l$ where C_k is the original quantization level. $R^l_{j,k}$ is the rate (in bits) that would be required to encode quantization level $\tilde{C}^l_k$ given that the previous non-zero coefficient was coefficient j. $R^l_{j,k}$ can be found from the VLC table and is simply the length of the corresponding code.

If coefficient k is dropped completely, the increase in mean squared error is X^2_k where X_k the unquantized coefficient. If the quantized and trimmed coefficient is transmitted instead, the increase in mean squared error is $(X_k - \tilde{X}^l_k)^2$. Therefore, E^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.

$\Delta J^l_{j,k}$ represents the incremental Lagrangian cost of going from coefficient j to coefficient k (dropping

the coefficients between them) and subtracting l 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^*. For intra blocks we assume that the dc coefficient is transmitted intact with the base layer while for inter blocks, it is treated like every other coefficient. Therefore, for intra blocks, J_0^* is the Lagrangian cost of encoding only the dc coefficient and dropping all other coefficients. Since the dc coefficient is not encoded using the same VLCs as the rest of the coefficients, we will not count the bits used for the encoding of the base layer in the minimization. Therefore, if we drop all ac coefficients of a block, the rate will be zero and the distortion will be equal to

$$J_0^* = E_{intra} = \sum_{i=1}^{63} X_i^2, \tag{6}$$

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

$$J_{-1}^* = E_{inter} = \sum_{i=0}^{63} X_i^2. \tag{7}$$

As mentioned earlier, we need to take into account the fact that different VLCs 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 $\Delta J_{j,k,last}^l = -E_j^l + \lambda R_{j,k,last}^l$, where $R_{j,k,last}^l$ is the number of bits that are required to encode quantization level $\tilde{C}_k^l$ given that j was the previous non-zero coefficient and coefficient k is the last one to be encoded in the block. We also keep the minimum Lagrangian costs $J_{k,last}^*$ for each coefficient k given that it is last coefficient to be encoded in the block.

A more detailed description of the proposed algorithm can be found in [6] and [7].

4. EXPERIMENTAL RESULTS

We tested the above algorithm with the "Akiyo" and "Foreman" sequences for a base layer bit rate of 14 kbps and an enhancement layer of a total bit rate of 18 kbps and compared it with the H.263 standard algorithm results in [3]. The results are shown in Table 1. We also made the same comparison for a base layer bit rate of 28.8 kbps and an enhancement layer of 56 kbps. The results are shown in Table 2.

We can see that the proposed algorithm clearly outperforms H.263 in the case of the "Akiyo" sequence

Bit rate (kbps)	14	18
Proposed Algorithm PSNR (Akiyo)	36.10	36.40
H.263 Algorithm PSNR (Akiyo)	35.50	35.61
Proposed Algorithm PSNR (Foreman)	28.10	29.19
H.263 Algorithm PSNR (Foreman)	29.00	29.26

Table 1: Comparison of the average PSNR of the proposed algorithm and the H.263 standard SNR scalability algorithm at 14-18kbps.

Bit rate (kbps)	28.8	56
Proposed Algorithm PSNR (Akiyo)	38.70	40.66
H.263 Algorithm PSNR (Akiyo)	38.17	39.06
Proposed Algorithm PSNR (Foreman)	29.92	32.79
H.263 Algorithm PSNR (Foreman)	30.79	32.53

Table 2: Comparison of the average PSNR of the proposed algorithm and the H.263 standard SNR scalability algorithm at 28.8-56 kbps.

while for the "Foreman" sequence, the results are comparable.

Figs. 3 and 4, show representative frames of the "Foreman" sequence encoded using the H.263 SNR scalability method at 28.8-56 kbps.

Figs. 5 and 6, show representative frames of the "Foreman" sequence encoded using the Optimal Single-Pass Codec at 28.8-56 kbps.

It can be observed that, in agreement with the numerical results, the quality of the frames from the two scalable algorithms are comparable. The H.263 scalable codec yields a slightly better result for the base layer frame, while for the enhancement layer, the proposed algorithm gives a higher quality frame.

5. 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. The proposed algorithm requires only a single DCT and quantization operation and a smaller bit overhead. Experimental results show that the proposed algorithm performs at least comparably or better than the H.263 scalable codec depending on the type of the video sequence and the target bit rates.

Figure 3: Frame 82 of the "Foreman" sequence encoded using the H.263 scalable codec at 28.8-56 kbps (28.8 kbps layer).

Figure 4: Frame 82 of the "Foreman" sequence encoded using the H.263 scalable codec at 28.8-56 kbps (56 kbps layer).

Figure 5: Frame 81 of the "Foreman" sequence encoded using the Optimal Single-Pass Codec at 28.8-56 kbps (28.8 kbps layer).

Figure 6: Frame 81 of the "Foreman" sequence encoded using the Optimal Single-Pass Codec at 28.8-56 kbps (56 kbps layer).

6. REFERENCES

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

[2] M. A. Robers, L. P. Kondi, and A. K. Katsaggelos, "SNR scalable video coder using progressive transmission of DCT coefficients," in *Proc. SPIE Conference on Visual Communications and Image Processing*, pp. 201–212, 1998.

[3] L. P. Kondi, F. Ishtiaq, and A. Katsaggelos, "On video SNR scalability," in *Proc. International Conference on Image Processing*, (Chicago, IL.), pp. 934–938, 1998.

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

[5] D. Wilson and M. Ghanbari, "Optimisation of two-layer SNR scalability for MPEG-2 video," in *Proc. International Conference on Accoustics, Speech and Signal Processing*, pp. 2637–2640, 1997.

[6] L. P. Kondi and A. K. Katsaggelos, "Rate-distortion optimal SNR scalable video coding," *IEEE Trans. on Circuits and Systems for Video Technology*, 1999. Submitted for publication.

[7] L. P. Kondi, *Low Bit Rate SNR Scalable Video Coding and Transmission*. PhD thesis, Northwestern University, Dept. of ECE, June 1999.