SNR scalable video coder using progressive transmission of DCT coefficients

Marshall A. Robers[a], Lisimachos P. Kondi[b] and Aggelos K. Katsaggelos[b]

[a]Data Communications Technologies (DCT)
2200 Gateway Centre Blvd., Suite 201
Morrisville, North Carolina 27516

[b]Northwestern University
Department of Electrical and Computer Engineering
2145 Sheridan Road
Evanston, Illinois 60208

ABSTRACT

The importance of signal-to-noise ratio (SNR) video compression algorithms has increased in the past few years. This emergence corresponds with the vast increase of products and applications requiring the transmission of digital video streams. These new applications, including video telephony/ teleconferencing, video surveillance/ public safety, and video-on-demand, require limiting the bandwidth of the compressed bitstream to less than the capacity of the transmission channel. However, the channel capacity is frequently unknown at the time of compression, especially when the stream is to be broadcasted to many users over heterogeneous channels. SNR scalable compression allows a single compression to provide bitstreams of multiple quality. In this fashion, the transmitted bitrate can match the available channel(s) without requiring multiple encodings. In this paper, we present a novel approach to SNR scalable video compression. Our approach combines two separate methodologies for dividing the blocks of discrete cosine transform (DCT) coefficients. The flexible combination of these approaches allows each DCT block to be divided into a fixed number of scans while also controlling the size of each scan. Thus, the transmitted stream can contain any subset of scans from the overall compressed version and thereby both the transmitted bitrate and the quality or SNR are allowed to vary.

Keywords: Video compression, SNR scalable video compression, spectral selection, successive approximation

1. BACKGROUND: IMAGE COMPRESSION

The objective of image compression techniques is to remove redundancy, which typically involves the transformation of the spatial intensities (gray values). Performing this transformation involves selecting appropriate basis functions. The Karhunen-Loeve transform (KLT) statistically decorrelates the original data and therefore compacts its energy.[1] However, the computational complexity of the KLT prevents its widespread use in image and video compression. The DCT has demonstrated similar energy compaction properties as the KLT, and can be easily computed with a butterfly implementation,[2] similar to the FFT implementation of the discrete Fourier transform (DFT). Therefore, the DCT is widely accepted as the standard transformation within image compression. The DCT is a block-based approach and as such it produces many blocks of coefficients which can be scalably coded.

1.1. Spectral Selection

The energy compaction property of the DCT dictates that the majority of the signal's energy is found in the low frequency coefficients. Thus, a typical methodology for dividing a DCT block into scalable scans involves sending only the low frequency coefficients in the first scan, also known as the baselayer. This approach is called spectral selection (SS).[3] In order to rank each two-dimensional coefficient by its frequency content, a zig-zag ordering is used. In terms of this zig-zag ordering, spectral selection involves transmitting coefficients 0 to $L_1 - 1$ in the baselayer, L_1 to $L_2 - 1$ in scan two, and so on until all coefficients are included. Graphically spectral selection is represented by Figure 1.

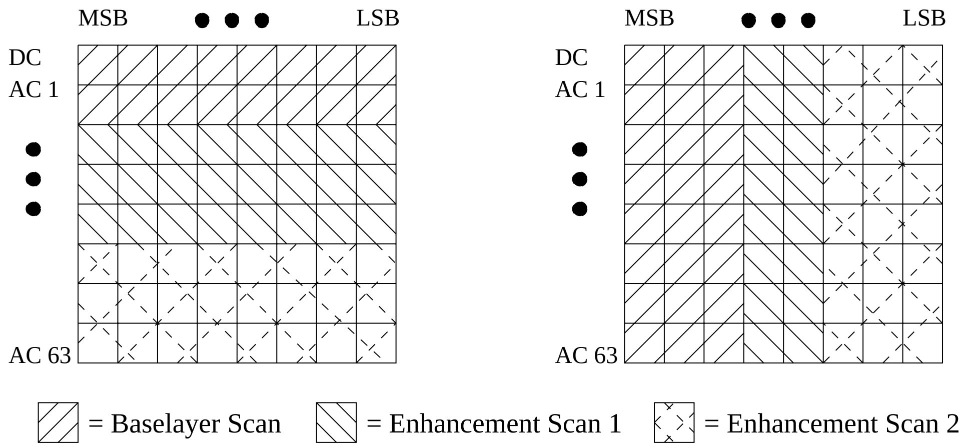

Figure 1. Typical scan definition for dividing an 8 x 8 block of DCT coefficients using spectral selection (left) and successive approximation (right)

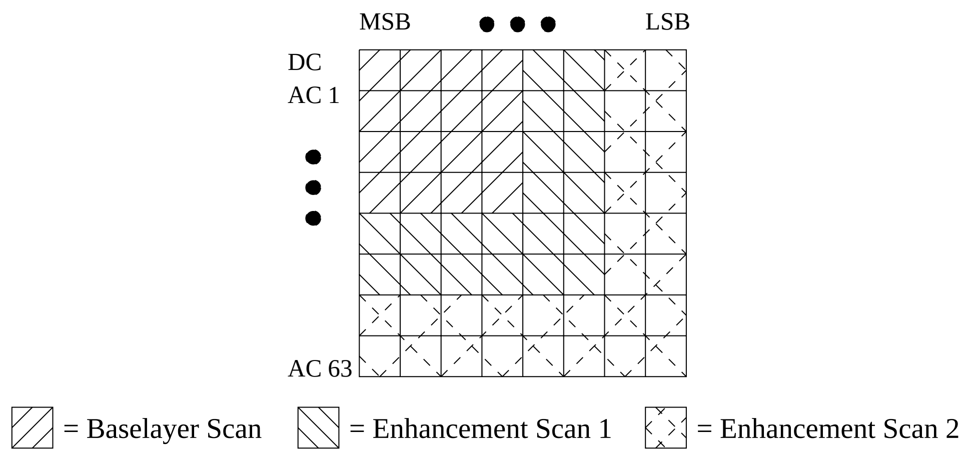

Figure 2. Typical scan definition for dividing an 8 x 8 block of DCT coefficients using both spectral selection and successive approximation

1.2. Successive Approximation

In contrast to SS, successive approximation (SA) involves including all coefficients in each scan, but increasing the resolution of each coefficient in subsequent scans. This technique corresponds to bit-plane coding techniques; here we effectively reduce the quantization coefficients by a factor of two between each scan. Graphically, successive approximation is represented by Figure 1.

1.3. Combination of SS and SA

Within a block of DCT coefficients, the low frequency coefficients represent trends, or regions with relatively constant intensity. These coefficients represent the majority of the information content of most image blocks. In contrast to the trends, the high frequency coefficients represent areas of highly varying intensity or edges. While edges are not present in all blocks, the information which they convey is significant to the overall content/meaning of the image. Thus, in order to have some tradeoff between edges and trends, a combination of spectral selection and successive approximation can be used to divide a DCT block. An example of a combination of SS and SA is given in Figure 2.

2. BACKGROUND: VIDEO COMPRESSION

In the previous section, we reviewed two methods for dividing blocks of DCT coefficients into several scans and noted that both techniques could be combined for greater flexibility. We saw that the most flexible approach was a combination of both spectral selection and successive approximation. In this section, we extend the concepts of scalable image compression to scalable video compression. Within video compression, source sequences possess both spatial and temporal redundancy. The temporal redundancy or correlation between subsequent frames can lead to significant increases in compression ratio.

Temporal redundancy is typically exploited by predicting the current frame from the previously decoded frame. Block matching techniques are used to determine the best match block from a region of predetermined size around the current block. The resulting displacement motion vector indicating the selected block in the previously decoded frame is then entropy coded with a variable length code (VLC).

2.1. H.263 video compression

H.263 constitutes an international standard for video compression of color sequences at low bitrates. This standard specifies the approaches and the exact syntax for the video compression algorithm.[4] H.263 is a block-based compression approach that allows for both non-predictive (intra or I) and predictive (inter or P) blocks. Obviously, the first frame must contain only I-blocks. In addition, blocks in subsequent frames containing new information or with a complex motion pattern are typically intracoded. The rest of the blocks are predicted from the previous blocks through use of motion vectors.

2.2. H.263+ Video Compression

The H.263 standard for video compression at low bitrates has been expanded upon in the video coding standard called H.263+.[5] The H.263 standard discussed in the previous section does not include any measures for SNR scalability. The successor to H.263, H.263+, has measures for three forms of video scalability; namely, H.263+ allows true temporal scalability in the form of B-frames, spatial scalability, and SNR scalability.

Within H.263+, SNR scalable coding involves coding a single frame multiple times. The first scan for each frame is determined using standard motion compensation as described in H.263. Then, to form the next scan for a frame, the encoder calculates the difference between the actual frame and the representation of the frame given by the first scan. This difference or "error frame" is then coded in the same way as the first scan; we can use motion estimation to predict this enhancement layer from the baselayer (in the case of an EI layer) or from both the baselayer and the enhancement layer from the previous frame (in the case of an EP layer). Finally the DCT is taken for the pixels representing the difference between the predicted enhancement layer and the actual enhancement layer. These DCT coefficients are quantized and coded in the same way as the baselayer coefficients. Figure 3 depicted the typical scenario for H.263+ SNR scalability in which enhancement I and P blocks are used to code the error from I and P baselayer blocks respectively. The obvious difference between an I and a P enhancement block is that the latter block also uses prediction from the previous frame's enhancement block. The standard specifies that predicting from only the current frame's previous layer uses no motion vectors. Motion vectors are only used in the enhancement layers when we are predicting from the previous layer (i.e., EP block).

We will describe the differences between this H.263+ scalability and our approach for scalability later in the paper. It is important to note that the H.263+ approach typically involves a re-quantization of coefficients and transmission of additional motion vectors for each new layer of SNR scalability. Our approach will involve only a single set of motion vectors per frame and a single quantization.

3. SCALABLE VIDEO CODER

The proposed approach to scalable video coding[6] applies concepts of the progressive JPEG image coding technique to a sequence of images (i.e., a video sequence). As such, this approach partitions the quantized DCT coefficients for both inter and intra blocks to allow for several scans of increasing quality. We have developed scan-dependent variable length codes (VLCs) which take advantage of the characteristics and properties of each scan. We have also implemented a rate-control mechanism that modifies the scan definitions to meet prespecified bitrate constraints.

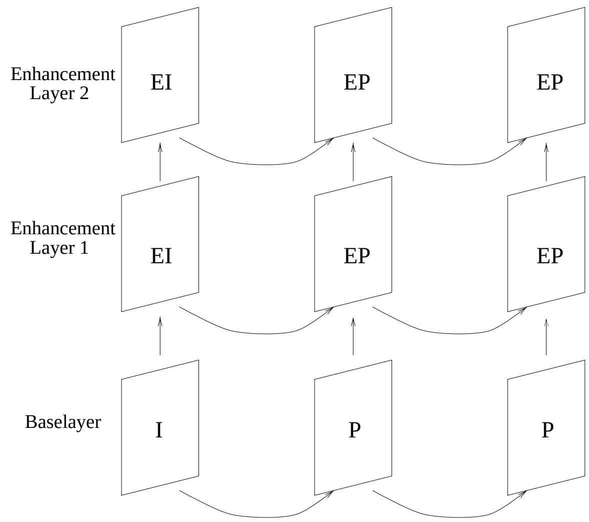

Figure 3. H.263 scalability using three layers (typical)

3.1. Progressive Partitioning of DCT Coefficients

The proposed scalable methodology embeds a fixed number of scans of increasing quality within the bitstream. Since the number of scans is prespecified, this approach constitutes discrete scalability. We saw previously that H.263+[5] incorporated SNR scalability by recoding the error or difference between the baselayer and the actual frame. The typical implementation of the H.263+ approach required requantization as well as additional prediction and motion vectors in order to code an enhancement layer. We wish to deviate from this scheme since a second quantization seems both inappropriate and imprecise. We wish to produce all scans of the DCT block from a single quantization and division of the original DCT block. Thus, the baselayer provides a "reasonable" quality version of the image and subsequent scans further refine this initial estimate by including higher frequency coefficients or bits of lower significance than those in the baselayer. An additional drawback of the typical implementation of the H.263+ approach is the cost of determining and transmitting an additional set of motion vectors for the enhancement layer; this technique decreases the speed of the algorithm considerably. Ideally, we wish to keep the proposed SNR scalable algorithm as close to real-time as possible. Speed was a primary motivation behind initially choosing the DCT; therefore, we do not wish to produce an algorithm which cannot be implemented easily and quickly.

Figure 4 provides a block diagram of the proposed SNR scalable encoder. It only requires a single quantization and a single set of motion vectors. The proposed algorithm uses a block-based motion compensated scheme identical to H.263.[4] Then after the DCT of each block is taken, the DCT coefficients are quantized a single time using a fairly small quantizer stepsize. We will discuss this quantizer stepsize in more detail later. After quantization, we partition the block of DCT coefficients using a combination of spectral selection and successive approximation. In this way, we form a number of scans. Each scan constitutes a subset of the original quantized block of DCT coefficients. Thus, using all scans, we have the complete block of DCT coefficients which were quantized with a small quantizer stepsize. An important thing to notice from the block diagram is that motion compensation uses only the baselayer from the previous reconstructed frame. This sacrifice is necessary to assure that the decoder can reproduce the encoder's motion compensation without having the enhancement layers. This will also be discussed in additional detail later.

Figure 4 indicates that either the previous reconstructed baselayer frame or the previous reconstructed frame from all scans can be used for motion estimation. The process of motion estimation does not need to be duplicated at the decoder and therefore allows flexibility to select which version of the reconstructed frame to use. Our results indicate a slight improvement in overall quality when the previous baselayer reconstructed frame is used for motion estimation. Thus, our demonstrated results in section 4 use the previous baselayer frame for motion estimation. This favoritism toward using the previous baselayer frame can most likely be attributed to its ability to more faithfully

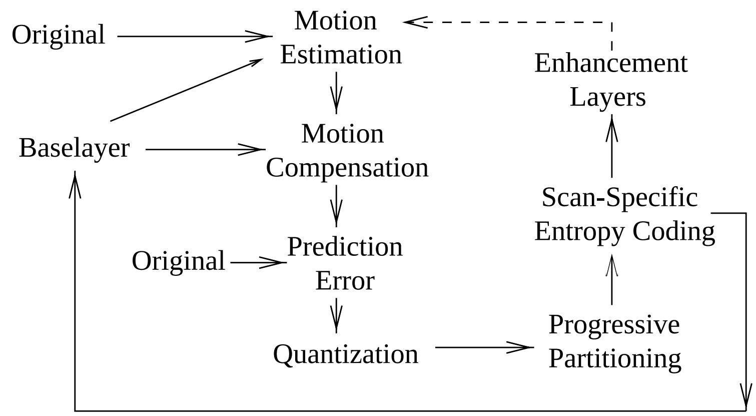

Figure 4. Block diagram of SNR scalable encoder

represent the baselayer of the current frame since the quality of the baselayer is emphasized by its use in motion compensation.

To see the difference between the H.263+ approach to scalability and the proposed approach, consider the following. Using H.263+, if we wish to obtain a baselayer of a certain size, we must select the baselayer quantizer so that transmission of the complete baselayer blocks of DCT coefficients will meet this bitrate constraint. This quantizer size is usually quite large. Thus, all DCT coefficients are transmitted within the baselayer but the precision with which each coefficient is represented suffers; this approach implies or restricts the division of DCT coefficients to use a modification of successive approximation since each scan contains all coefficients. In other words, the baselayer with its coarse quantizer gives a minimal representation of all coefficients, and subsequent scans use a lower (finer) quantizer to add precision to the estimate from the baselayer. Our approach allows the baselayer to use a combination of successive approximation and spectral selection. With this flexibility, the baselayer can contain more significant bits of the low frequency coefficients and less (or no) information about the high frequency coefficients. With this scheme, we use a single quantizer for the whole block of DCT coefficients and then only transmit a subset of these coefficients in each scan.

There are many possible valid divisions of the block of DCT coefficients using a combination of spectral selection and successive approximation. We have experimented with different configurations in order to obtain a reasonable partitioning of the DCT block. As we will see in Section 3.2, we have developed a rate control methodology to allow the scan definitions to change during the coding process so that multiple bitrate constraints can be met. Even when we are allowing the scan definitions to vary throughout the coding, we need both an initial setup for the scan definitions and a range of permissible scan definitions. This will also be discussed in the rate control section.

It should be noted that we can use a different scan definition for intra and inter blocks. The primary difference between the scan definitions for intra and inter blocks is that the base-layer for intra blocks contains complete information about the lower frequency DCT coefficients whereas the inter block base-layer definition omits the least significant bit. This discrepancy in scan definition helps to assure that the baselayer maintains a "reasonable" representation of the current frame since this information will have to be used during motion compensation for the next frame.

The baselayer is the only scan that is guaranteed to be included in any compressed bitstream. In other words, all applications using this scalable coding are required to transmit at least the baselayer. As such, the baselayer contains the only essential information that the decoder needs in order to reproduce the encoder's motion compensation. If for some reason (i.e., packet loss or delay) the baselayer is not received by the decoder, the encoder and decoder will end up with different versions of the reconstructed frame. This scenario presents a major difficulty for motion-compensated video coders since errors will propagate from frame to frame. The most likely solution to such difficulties is to require the encoder to provide conditional replenishment. In other words, the encoder would code a different part of each image as intra blocks (non-predictive) so that any skew between the encoder's and decoder's version of the reconstructed frame can be eliminated over some specific number of frames.

Since the encoder's motion compensation is based solely on the baselayer, we must pay particular attention to the definition of the baselayer. We must ascertain that the baselayer's quality remains reasonably good in order to take advantage of the temporal redundancy inherent to most video sequences. To demonstrate this concept, we can consider two successive frames with a high degree of correlation (i.e., high temporal redundancy). Now, for the purpose of generic video coding (i.e., non-scalable) we could use very few bits to represent the second frame since we could predict this frame reasonably well from the previous frame. However, for scalable coding where prediction is based solely on the previous frame's baselayer, the number of bits needed to represent the second frame is dependent on the baselayer of the previous frame. Thus if the baselayer is not an adequate representation of the first frame, the prediction of the second frame will not be adequate, and we must use many bits to represent the second frame despite the high correlation with the first frame. In fact, using prediction based on the previous frame's baselayer is the primary disadvantage of scalable video coding compared with standard video coding.

Since the size of the baselayer is dictated by the application(s), we can only control the content of the baselayer (i.e., how the available bits are allocated). Obviously, we wish to spend the majority of the allotted baselayer bitrate on the low frequency DCT coefficients. In addition, we have to spend many bits giving the location of significant high frequency coefficients even if these high frequency coefficients are run-length coded. Thus, a typical baselayer for this scalable coder would include most of the bits for the low frequency DCT coefficients and very little (if any) information about the high frequency coefficients. This emphasis on the low frequency coefficients produces a low-pass filter effect on the compressed sequence. This low-pass effect will be discussed in the results section. Here it is sufficient to note that this low pass effect is often more visually pleasing than the H.263+ alternative of having a minimal representation of all DCT coefficients in a block.

We should point out here that the baselayer also contains the motion vectors and the high level parameters (i.e., the headers for frames, group of blocks, and blocks). Thus the content of the enhancement layers is limited to refinements of the DCT block coefficients. Therefore, the enhancement layers increase the quality of a particular frame. However, their effect is not cumulative; i.e., including enhancement layers for one frame does not increase the SNR for any layer of the subsequent frame. As mentioned earlier, this is in sharp contrast with any non-scalable coder where the whole previous frame aids in the prediction of the current frame.

3.2. Rate Control

When designing a video transmission scheme for real-time communication channels, practical limits are set on the allowable bandwidth of the encoded video subsets. Thus, our progressive partitioning of the DFD using both spectral selection and successive approximation must be adaptive so the bitrate constraints can be met. We have devised a scheme to adjust the quantization stepsize, the coded framerate, and the scan definitions to obtain the desired bitrates.

The quantization parameter and the coded framerate are adjusted based on the desired bitrate for all scans combined. The approach for selecting and modifying both the quantization parameter and the coded framerate are taken from the TMN6 video codec test model.[7]

We have developed a dynamic partitioning scheme to divide the total incoming bits into subsets of specified sizes. The basic idea of the scheme is to change the boundaries of the scans based on the target bitrates for each of the scans. This approach assumes that maximum bitrates have been specified for each scan. In other words, we can assume separate transmission channels, and therefore unused bits in one scan can not increase the bits available to subsequent scans. It should be noted that for transmission over a single channel, unused bits from a previous layer could be used by a higher layer with a sophisticated multiplexing algorithm. Such a multiplexing strategy has not been implemented. For convenience, we have chosen to modify the scan parameters at the beginning of each macroblock line since this coincides with the modification of the quantization parameter.

In order to dynamically modify the scan parameters, we must first explicitly specify the scan parameters. We have parameterized the boundaries between each scan. Our scheme can be adapted for use for DCT block divisions using an arbitrary number of scans; here we will present an example based on a video sequence with three subsets. Typically, three subsets will be sufficient to meet the needs of the intended applications. Despite the fact that

Scan Number	AC Start	AC End	Which Bits
1	0	X	except A LSBs
2	X+1	63	except B LSBs
3	0	X	A LSBs
3	X+1	63	B LSBs

Table 1. Division of DCT block into three subsets

our scheme could allow an arbitrary number of scans, increasing the number of scans increases the overhead and typically causes the efficiency to suffer. Table 1 shows the proposed scan definition. Note that scan three contains the uncoded LSBs from all DCT coefficients. This division into three subsets yields three parameters (A,B, and X) which our scheme can dynamically adjust. We also enable the scans for intra and inter blocks to differ. This is done by expressing the parameters for intra blocks in terms of the parameters for inter blocks. Typically, the X used for intra blocks was greater by twenty than the X for inter blocks, and A was less by one for intra than inter blocks. Obviously, these conversions are limited by the allowable dynamic ranges for X, A, and B.

Our partitioning scheme changes the scan parameters based on the number of bits spent on each scan during the last frame. In other words, we maintain buffers for each scan which hold the bits used for representing one frame up to the macroblock line under consideration. Then as each macroblock line in the new frame is coded, these bits are added to the appropriate buffers and the bits spent on this macroblock line in the previous frame are removed. The number of bits in these scan buffers at the end of each macroblock line is used to calculate a Target Bit Error (TBE) for each scan, where

$$TBE_i = bits\_in\_buffer_i - target\_bits\_per\_frame_i \tag{1}$$

and i denotes the scan number. Of course, the target number of bits per frame for each scan depends on the coded framerate.

Next we normalize each of the $TBEs$ based on the assumption that exceeding the target bitrate by a fixed number of bits requires more significant and immediate action for a scan with a smaller target bitrate. This normalization produces a normalized Target Bit Error ($NTBE$) for each scan, where

$$NTBE_i = TBE_i / target\_bits\_per\_frame_i \tag{2}$$

Finally we compare the NTBEs to determine if the scan parameters need adjusting. We calculate three scan differences ($\Delta_{i,j}$) by comparing the $NTBEs$ for each scan; that is,

$$\Delta_{1,2} = NTBE_1 - NTBE_2 \tag{3}$$

$$\Delta_{1,3} = NTBE_1 - NTBE_3 \tag{4}$$

$$\Delta_{2,3} = NTBE_2 - NTBE_3. \tag{5}$$

These $\Delta_{i,j}$s are compared to pre-established thresholds ($T_{i,j}$) which depend on the maximum allowable deviation from the desired scan bitrates. If the threshold is exceeded, the appropriate scan parameter is adjusted. Table 2 provides a description of which parameter should be adjusted when one of the thresholds is exceeded. Obviously each scan adjustment must result in a feasible solution; i.e., X is limited to [0,63]. In addition, we impose the constraint that A and B are limited to [0,3]. The amount by which A,B, and X are incremented/decremented is given by the the following equation:

$$\Delta_{param} = \lfloor \frac{\Delta_{i,j}}{T_{i,j}} \rfloor \tag{6}$$

where $\lfloor x \rfloor$ denotes the largest integer not greater than x. An upper bound on the magnitude of the scan adjustments is also used to avoid sending the parameters into rapid oscillation. Typically, we limit Δ_X to five coefficients and Δ_A and Δ_B to one bit. These limitations prevent the scan parameters from oscillating rapidly, but at the same time do not pose difficulty for meeting the imposed bitrate constraints. Rapid oscillations of the scan parameters is

Condition	Action Required
$\Delta_{1,2} > T_{1,2}$	decrease X
$\Delta_{1,2} < -T_{1,2}$	increase X
$\Delta_{1,3} > T_{1,3}$	increase A
$\Delta_{1,3} < -T_{1,3}$	decrease A
$\Delta_{2,3} > T_{2,3}$	increase B
$\Delta_{2,3} < -T_{2,3}$	decrease B

Table 2. Dynamic Adjustment of Scan Parameters

undesirable since it will cause the baselayer quality to deteriorate when the baselayer is small. Due to the scarcity of intra frames in low bitrate video, when the quality of the baselayer becomes poor, it can affect the quality of the prediction for many subsequent frames.

Obviously we must inform the decoder of any adjustments to the scan parameters by coding these changes in the header for each macroblock line (i.e., GOB). The H.263 syntax already allows the change in the quantization parameter in the GOB header. We have modified this syntax to also allow changes in the scan parameters to be coded. The number of bits required is minimal since the magnitude of the scan adjustments has been limited, as mentioned previously.

3.3. Entropy Coding (VLCs)

This section describes the selection of VLCs in order to minimize the number of bits necessary to represent the blocks of DCT coefficients. As we saw in previous sections, the content of each scan (i.e., the scan definitions) can vary widely; therefore, the most appropriate scheme will not only allow these variations in the scan limits but will also take advantage of the particular characteristics and probability distribution of the symbols for each scan.

In H.263,[4] each non-zero coefficient is run-length entropy coded. In fact a specialized 3D VLC is used which combines three variables in a single VLC for each significant coefficient. The VLC gives the magnitude of the significant coefficient, the number of preceding insignificant coefficients, and whether this coefficient is the last significant coefficient in the DCT block. We follow a similar VLC structure. It is clear that by coding multiple events (i.e., run, magnitude, and last) as a single symbol, we gain compression efficiency compared with coding the events as separate symbol. In fact, increasing the number of events coded as a single entity can only improve the compression performance.[8] In other words, when designing a Huffman code, we can get closer to the entropy of the source by increasing the number of symbols coded as one entity.

We also group the three events (run, level, last) into a single symbol. We have yet to specify the process of obtaining the VLCs that will be used to code these symbols. In general, there are two ways to obtain these VLCs; either the VLCs can be prespecified based on their expected probability of occurrence, or they can be derived for the specific source image or video. Image compression techniques typically allow the VLCs to vary based on the source statistics. However, for Huffman coding, such an approach implies a two pass approach in which the source statistics are obtained in the first pass; the VLCs are then generated based on these statistics and used during the second pass. This approach forces the encoder to spend some bits to indicate which VLCs are being used. The primary drawback of this approach, however, is the time it takes the source coder to make two passes for an image. Compression schemes for real-time video applications cannot utilize this two pass approach and therefore must pre-determine an acceptable set of VLCs.

This approach of pre-specifying the VLCs for transmission of DCT coefficients is the technique used within H.263 compression. However, the task of VLC development is much more difficult for a scalable coder for which the content of the DCT blocks can vary dramatically with variations in the scan definitions. For this reason, we had to develop scan specific VLC tables instead of merely using the VLC table provided by H.263. To see the importance of scan specific VLC tables, consider the case when the last scan typically contains only the LSB for the DCT coefficients. The H.263 VLCs would not provide an effective representation for this scan since the probability of having a coefficient of magnitude one is greatly increased for this final scan. Likewise, the probability of having a coefficient of magnitude greater than one is close to (or equal to) zero since this scan typically provides only the LSB. There are similar examples requiring specialized VLCs for scans utilizing spectral selection since the permissible run lengths are reduced. Thus, it is concluded that the development of scan specific VLC can greatly increase the compression efficiency within a scalable coder.

When developing the VLC tables, we wish to have a VLC for all symbols with some non-zero probability of occurrence. Otherwise, we have to resort to ESCAPE coding; ESCAPE coding is used to code the symbols which have not been pre-assigned a VLC. ESCAPE coding uses many bits, and we wish to avoid it whenever possible. However, it is not practical or feasible to have a VLC for each and every combination of run, level, and last which could occur. Complete enumeration of all possible symbols would require an enormous number of VLCs; in addition, having a large number of VLCs makes it difficult to avoid start code emulation. A start code constitutes a sequence of bits used in the bitstream to indicate the start of a frame. When a transmission error occurs, the decoder can resynchronize by looking for the next start code. It is therefore important that the start code is not duplicated within the VLCs.

We determined a small, but nonzero threshold, and all symbols with a probability of occurrence above the threshold were assigned a specific VLC. All other symbols will be ESCAPE coded; thus, the probability of having an ESCAPE code is nonzero. However, with a reasonably small threshold (about 10^{-5}) and accurate symbol source probabilities, the probability of an ESCAPE code remains close to zero.

In order to pre-specify the VLC tables, we need to know the typical or average source probabilities for the different symbols. To obtain these probabilities, we ran the encoder on a number of different sequences using the rate control mechanisms specified in Section 3.2. It is important to note that when estimating symbol probabilities while using a rate-control mechanism, we must make multiple passes. In other words, the rate control changes the scan definitions and the quantizers based on the number of bits so far which clearly depends on the current VLCs. This selection of scan definitions and quantizer stepsize affects the size of the blocks of DCT coefficients. Therefore, the VLCs used when determining the source probabilities are important. We needed to assign a reasonable set of initial VLCs, and subsequently encode a number of test sequences. The symbol probabilities obtained when encoding these sequences can be used within a standard Huffman coding algorithm[9] to obtain a new set of VLCs. This process can repeat until the changes in the VLC tables are small. Typically, only a few passes are necessary.

Scalable coding using either spectral selection or successive approximation (or both) requires an additional symbol that is not necessary for non-scalable coding. We have called this symbol ZERO_LAST. This symbol occurs when a scan should be skip coded, but the DCT block as a whole contains some significant coefficients. Skip coding refers to the case when a block is predicted from the previous frame, and no DCT information is provided. The macroblock's parameters indicate the presence of a skip block so that the decoder knows that this block will have no DCT information. The difficulties occur when we perform scalable coding with a single set of macroblock parameters. These parameters will indicate whether or not the block as a whole is skip coded. Thus, when a block as a whole contains some significant coefficients, but a specific scan for this block does not contain any nonzero coefficients, we use this ZERO_LAST symbol. This symbol can occur quite frequently when using successive approximation. Typically the MSBs are all zero, but the LSBs in subsequent passes are significant.

4. RESULTS

In this section, the results produced by the proposed algorithm are reviewed. We compare the proposed scheme to standard (non-scalable) H.263. Such a comparison will show some bias against the proposed scheme since H.263 uses motion compensation based on the complete previous frame. As mentioned earlier, the bits used in the enhancements scans in our scalable algorithm only contribute to the quality of a single frame.

Some extracted coded frames (intensity only) produced by the proposed scalable coder are shown first. The frames shown are inter frames from the "Coastguard" sequence. The subset bitrates were 14, 18 and 22 Kb/sec. Figure 5 depicts the baselayer representation of the frame (left) and the baselayer plus enhancement layer 1 representation (right). Figure 6 shows the complete (all scans) representation of the frame.

Since the proposed algorithm was designed and intended for very low bitrate compression, the results presented in this section will be limited to these bitrates (i.e., less than 128 Kb/sec). The proposed technique with the dynamic scan boundaries was tested and compared to H.263. As mentioned a few times before, the main discrepancy

Figure 5. Left figure: Coastguard, Frame 43, Luminance, Scalable, Baselayer, 14 Kb/sec Right figure: Coastguard, Frame 43, Luminance, Scalable, Baselayer plus enhancement layer 1, 18 Kb/sec

Figure 6. Coastguard, Frame 43, Luminance, Scalable, all scans, 22 Kb/sec

Technique	Bitrate (Kb/sec)	Mean PSNR (dB)
Scalable (Baselayer Only)	14	25.61
Scalable (Baselayer + Enh. Scan 1)	18	26.01
Scalable (All 3 layers)	22	27.05
Non-Scalable H.263	14	27.31
Non-Scalable H.263	18	27.59
Non-Scalable H.263	22	27.81

Table 3. Comparison of Proposed Technique and Standard H.263 for foreman at very low bitrates

Technique	Bitrate (Kb/sec)	Mean PSNR (dB)
Scalable (Baselayer Only)	14	26.53
Scalable (Baselayer + Enh. Scan 1)	18	26.97
Scalable (All 3 layers)	22	27.75
Non-Scalable H.263	14	27.84
Non-Scalable H.263	18	28.36
Non-Scalable H.263	22	28.93

Table 4. Comparison of Proposed Technique and Standard H.263 for coastguard at very low bitrates

between the results can be attributed to the motion compensation utilizing only the previous frame's baselayer. The quantitative measures presented in the following tables use the mean PSNR of the luminance channel only (Y channel) for all coded frames. Two different tests were conducted on two different sequences. The source sequence foreman contains more motion and is more difficult to compress than the coastguard source sequence. For the first test, seen in Tables 3 and 4, the SNR scalability was set to attain subset bitrates of 14, 18 and 22 Kb/sec. That is, the baselayer was 14 Kb/sec and both enhancement layers were 4 Kb/sec. For the second test, seen in Tables 5 and 6, the SNR scalability was set to attain subset bitrates of 28.8, 56 and 128 Kb/sec. That is, the baselayer and enhancement scan 1 were each approximately 28 Kb/sec and the final enhancement scan was 56 Kb/sec. It should be mentioned here that the three scalable results required only a single compression whereas the standard H.263 results required a separate compression for each bitrate.

5. CONCLUSIONS

In this paper, we presented a novel approach for performing SNR scalable video compression. Our approach combined two schemes, spectral selection and successive approximation, for dividing the blocks of quantized DCT coefficients. In order to attain desired bitrates for each subset, the boundaries between each scan were parameterized and dynamically adjusted. The significant coefficients in each scan were entropy coded with scan-dependent VLCs to take advantage of the highly scan-specific distributions. While the scalable results were somewhat below those produced by H.263

Technique	Bitrate (Kb/sec)	Mean PSNR (dB)
Scalable (Baselayer Only)	28.8	26.52
Scalable (Baselayer + Enh. Scan 1)	56	27.48
Scalable (All 3 layers)	128	31.46
Non-Scalable H.263	28.8	28.26
Non-Scalable H.263	56	29.89
Non-Scalable H.263	128	32.50

Table 5. Comparison of Proposed Technique and Standard H.263 for foreman at low bitrates

Technique	Bitrate (Kb/sec)	Mean PSNR (dB)
Scalable (Baselayer Only)	28.8	27.97
Scalable (Baselayer + Enh. Scan 1)	56	29.05
Scalable (All 3 layers)	128	32.70
Non-Scalable H.263	28.8	29.59
Non-Scalable H.263	56	31.59
Non-Scalable H.263	128	34.02

Table 6. Comparison of Proposed Technique and Standard H.263 for coastguard at low bitrates

in terms of PSNR, the added functionality enables additional applications which were not possible without SNR scalability.

6. ACKNOWLEDGEMENTS

This work was supported in part by Motorola

REFERENCES

1. J. Lim, *Two-dimensional signal and image processing*, Prentice Hall, Englewood Cliffs, 1990.
2. W. Chen, C.H.Smith, and S.C.Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Transactions on Communications* **COM-25**, pp. 1004–1009, 1977.
3. "Information technology- digital compression and coding of continuous-tone still images (JPEG); Recommendation T.84 ." International Organization for Standardization, November 1994.
4. "Video coding for low bitrate communication; DRAFT ITU-T Recommendation H.263." International Telecommunication Union, May 1996.
5. "Video coding for low bitrate communication; ITU-T Recommendation H.263+." International Telecommunication Union, March 1997.
6. M. A. Robers, "Snr scalable video coder using progressive transmission of dct coefficients," Master's thesis, Northwestern University, May 1997. Department of Electrical and Computer Engineering.
7. "Video codec test model: TMN6." ITU Telecommunications Standardization Sector: H.263+ Ad Hoc Group, April 1996.
8. R. G. Gallager, *Information Theory and Reliable Communication*, John Wiley & Sons, 1968.
9. D. Huffman, "A method for construction of minimum redundancy codes," in *IRE*, vol. 40, pp. 1098–1101, 1962.