A Zerotree Wavelet Video Coder

Stephen A. Martucci, *Member, IEEE*, Iraj Sodagar, *Member, IEEE*, Tihao Chiang, *Member, IEEE*, and Ya-Qin Zhang, *Senior Member, IEEE*

***Abstract*—This paper describes a hybrid motion-compensated wavelet transform coder designed for encoding video at very low bit rates. The coder and its components have been submitted to MPEG-4 to support the functionalities of compression efficiency and scalability. Novel features of this coder are the use of overlapping block motion compensation in combination with a discrete wavelet transform followed by adaptive quantization and zerotree entropy coding, plus rate control. The coder outperforms the VM of MPEG-4 for coding of I-frames and matches the performance of the VM for P-frames while providing a path to spatial scalability, object scalability, and bitstream scalability.**

***Index Terms*—Low bit rates, motion compensation, MPEG-4, video compression, zerotree.**

I. Introduction

VERY low bit-rate video coding has received considerable attention lately in academia and industry in terms of both coding algorithms and standards activities. The recently adopted ITU-T Recommendation H.263 provides a solution for very low bit-rate video telephony [1]. Currently, MPEG-4 is working toward a standard for coding video in a way that provides functionalities such as content-based access, content-based manipulation, content-based editing, combined natural and synthetic data coding, robustness, content-based scalability, as well as improved coding efficiency [2]. In this paper, we present a coder developed at the David Sarnoff Research Center intended to address the needs of MPEG-4, particularly those in the areas of improved coding efficiency at very low bit rates and content-based scalability [3]. The key element of Sarnoff's MPEG-4 coder is a new, efficient method for encoding wavelet coefficients called zerotree entropy (ZTE) coding.

The structure of our proposed coder is similar to other motion-compensated, block-based, discrete cosine transform (DCT) video coders such as MPEG-1 and H.263, but we use a discrete wavelet transform (DWT), overlapping for motion to better match the DWT, and the zerotree concept for coding the wavelet coefficients. The wavelet transform reduces the blocking artifacts seen at very low bit rates while providing a better way to address the scalability functionalities of MPEG-4 [14]. The specific components of the coder are: 1) block motion estimation to track local motion [1]; 2) overlapping block motion compensation to remove temporal redundancy [1]; 3) an adaptive discrete wavelet transform of the residual to remove spatial correlation [5]; 4) quantization of the wavelet coefficients to remove irrelevancy; 5) the use of zerotrees and an arithmetic coder to losslessly encode the quantized coefficients with a minimum number of bits [6], [7]; and 6) a rate control scheme to control the quantization factor to achieve a desired mean bit rate [4], [15].

Overlapping in the block motion compensation reduces significantly the artifacts that would otherwise arise from the mismatch between the block nature of motion compensation and the global nature of the discrete wavelet transform. At high compression ratios, ringing is a major problem of the wavelet transform; the proposed coder remedies this problem by allowing different filter lengths at each stage of the decomposition. Quantization and coding of the wavelet coefficients are done using the zerotree coding concept. The coder incorporates the well-known embedded zerotree wavelet (EZW) algorithm to deliver bitstream scalability, and a new ZTE coding algorithm specifically tailored to give the best performance for encoding wavelet coefficients of very low bit-rate video.

In addition to high compression performance, the proposed coder provides a path to spatial scalability, object scalability, and bitstream scalability. At the heart of the coder is the zerotree coding method for the wavelet coefficients combined with a means for content-based adaptation of quantization. The coder incorporates a new zerotree coding algorithm that is highly efficient for motion-compensated video residuals and that provides a direct association between the wavelet coefficients and what they represent spatially, making possible a content-based control of their encoding. Finally, a new rate control scheme specific to the needs of this coder is included.

The organization of the paper is as follows. We begin in Section II with a general overview of the coder and a brief discussion of its salient features. In Section III, we describe in detail each of the major components of the coder. We present in Section IV the results obtained from using the coder to encode the test sequences of MPEG-4. The conclusion is in Section V.

II. General Overview and Features of the Algorithm

This section gives an overview of the proposed encoder. The system block diagram is shown in Fig. 1.

Frames of video are encoded either as intra or inter, where intra is used for the first frame and inter is used for the remaining frames of the test sequences. The first intra frame can be coded using either the EZW algorithm [9] or the ZTE coding algorithm [8] newly developed for this codec. EZW is recognized as one of the best ways to encode still images at a target bit rate. The embedded feature of the algorithm makes it possible to encode the first frame at exactly the chosen rate.

Manuscript received March 15, 1996; revised July 1, 1996. This paper was recommended by Guest Editors Y.-Q. Zhang, F. Pereira, T. Sikora, and C. Reader.

The authors are with the David Sarnoff Research Center, Subsidiary of SRI International, Princeton, NJ 08543-5300 USA.

Publisher Item Identifier S 1051-8215(97)00938-5.

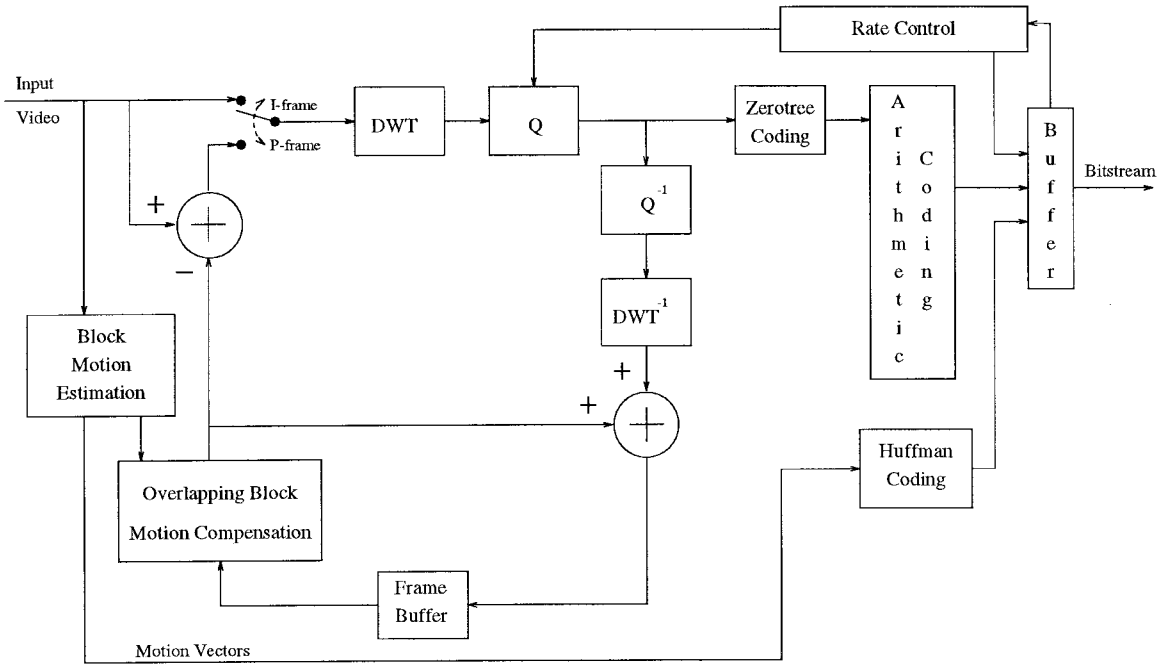


Fig. 1. Block diagram of proposed encoder.

ZTE is an improvement over EZW that does not produce an embedded bitstream but offers better performance and greater flexibility for video coding.

All succeeding frames of the video sequence are encoded from the preceding frame as forward-predicted "P" frames. Bidirectionally-predicted "B" frames are possible but have not yet been implemented. A block-based motion estimation scheme is used to detect local motion. The prediction is done using the block motion estimation scheme of H.263 where estimation is done to half-pel accuracy on blocks of 16×16 or 8×8.

Next, each block is predicted using the overlapping block motion compensation scheme of H.263. After all blocks have been predicted, the residuals are pieced together to form a complete residual frame for subsequent processing by the wavelet transform. The overlapping in the motion compensation ensures that a coherent residual is presented to the wavelet transform, without any artificial block discontinuities. For those blocks where prediction fails, the intra mode is selected and the original image block is coded. To turn this block into a residual similar to the other predicted blocks, the mean is subtracted from the intra blocks and sent as overhead. In this way, all blocks now contain a prediction error, constituting either the difference between the current block and its overlapping motion-compensated prediction or the difference of the block pixels from their mean.

The wavelet transform is flexible, allowing the use of different filters at each level of the decomposition. Shorter filters are used at the later levels of the decomposition to reduce their effective length and thereby reduce the extent of ringing after quantization of the coefficients. At the start of the decomposition, longer filters are needed to avoid any artificial blockiness that short filters would cause.

The wavelet coefficients are organized into *wavelet trees*, each of which is rooted in the low-low band of the decomposition then extends into the higher frequency bands at the same spatial location. Each wavelet tree provides a correspondence between the wavelet coefficients and the spatial region they represent in the frame. The coefficients of each wavelet tree can be rearranged to form a *wavelet block*. Each wavelet block is a fixed size block of the frame and comprises those wavelet coefficients at all scales and orientations that correspond to that block.

The wavelet trees are scanned and the wavelet coefficients are quantized. Because each tree relates to a distinct block of the frame, quantization can be varied according to what content of the image is covered by each block. It is also possible to vary the quantization within the block in a frequency-dependent fashion. The coder permits the use of any desired quantizer. We use a midriser uniform quantizer with dead zone around zero with adjustable step size.

The extreme quantization that is required to achieve a very low bit rate produces many zero coefficients. This is exploited by using zerotree coding to significantly compress those quantized coefficients. The coder can either use the EZW algorithm or the new, improved zerotree algorithm called ZTE coding. The EZW algorithm would be used where bitstream scalability is required or where exact bit rate use per frame must be met. The new ZTE coding scheme uses the zerotrees of EZW, but differs from EZW in ways that enable it to deliver much better performance for video coding. One feature of ZTE coding is that quantization is done explicitly and therefore can be optimized and dynamically adapted to scene content. A second feature is that the scanning and encoding of the wavelet coefficients is done in an order that exploits the close connection between the coefficients and what they represent

in the frame, thereby allowing allocation of bits on an object-by-object basis. Third, in ZTE coding, the use of zerotrees has been enhanced by defining a new set of symbols designed specifically for very low bit-rate coding of video. In place of the embedding property of EZW, ZTE provides greater flexibility, adaptability, and improved coding efficiency.

Finally, the symbols generated by the zerotree scanning process and the quantized coefficients are losslessly encoded using an adaptive arithmetic coder. The arithmetic coder uses adaptive models to track the statistics of its inputs and code them close to their entropy.

A rate control scheme is included that permits a target bit rate to be met. The scheme varies the quantization factor used for each wavelet block within each frame using a novel quadratic modeling of the rate distortion function. The rate control algorithm performs both interframe rate control and intraframe rate control.

III. Technical Details of the Algorithm

A. Motion Estimation and Compensation

1) Block Motion Estimation: The proposed coder uses the block motion estimation technique of H.263 [1]. Motion estimation is performed on the luminance 16 $\times$ 16 and 8 $\times$ 8 blocks. The distortion measure is the sum of absolute difference (SAD). The full pel motion estimation is done using the previous *original* frame. A full search is used and the search area is up to 15 pixels in all four directions from the center of the macroblock. The SAD for the zero translation vector for the 16 $\times$ 16 block is reduced by a bias, set to 100 by default, to favor the zero motion vectors. The SAD is calculated for each 16 $\times$ 16 macroblock and its four 8 $\times$ 8 blocks.

The motion estimation algorithm also has intra/inter mode decision. For each macroblock, its mean value is also calculated. The SAD between the macroblock and its mean value is also calculated (call it MSAD). If this value (MSAD) is smaller than SAD calculated by motion estimation by a set margin, 500 by default, the intra mode is chosen and no motion vectors are sent. In this case, the block is predicted by its mean and the mean is sent as overhead. Otherwise, the inter mode is chosen and the macroblock is estimated using motion vectors as described in the previous paragraph.

In order to achieve better estimates, the half-pel motion estimation is done using the previous *reconstructed* frame. The search is performed only on the luminance component. The range of search is one-half pixel in all four directions. Bilinear interpolation is used to obtain half pixels. For the chrominance components, motion vectors are divided by two and also quarter-pel interpolation is done to obtain the predictions of the chroma blocks.

If the 16 $\times$ 16 macroblock match results in smaller SAD, the corresponding vector is chosen and sent; otherwise four motion estimated vectors corresponding to four 8 $\times$ 8 blocks are sent as motion information. If the motion vector with smallest SAD is the zero vector, no vector is sent; this condition is indicated by the prediction mode. Therefore, there are four possible prediction modes for each 16 $\times$ 16 motion-estimated macroblock: 1) estimated by one 16 $\times$ 16 macroblock requiring no motion vector (motion vector is the zero vector); 2) estimated by one 16 $\times$ 16 macroblock requiring one motion vector; 3) estimated by four 8 $\times$ 8 blocks requiring four motion vectors; 4) intra macroblock requiring no motion vector. The motion vector field is differentially coded by predicting from a spatial neighborhood of three motion vectors that were already transmitted. The motion vector prediction errors are Huffman encoded following the tables of H.263. For details, refer to the H.263 recommendation [1].

2) Overlapping Block Motion Compensation: Overlapping block motion compensation (OBMC) is an advanced scheme for block motion compensation that overlaps, windows, and sums prediction blocks prior to subtraction from the current block being predicted in order to reduce the effect of block boundary discontinuities [11], [12]. Each block of the current frame to be encoded is predicted using the OBMC method of H.263 [1]. In this method, each 8 $\times$ 8 block is overlapped with two major neighboring blocks: 1) the upper left 8 $\times$ 8 block of each 16 $\times$ 16 macroblock is overlapped with the adjacent blocks located at the above and left sides; 2) the upper right 8 $\times$ 8 block of each 16 $\times$ 16 macroblock is overlapped with the adjacent blocks located at the above and right sides; 3) the lower left 8 $\times$ 8 block of each 16 $\times$ 16 macroblock is overlapped with the adjacent blocks located at the below and left sides; 4) the lower right 8 $\times$ 8 block of each 16 $\times$ 16 macroblock is overlapped with the adjacent blocks located at the below and right sides. Therefore, each pixel of motion compensated frame is a weighted sum of three prediction values from the previous reconstructed frame: one value predicted using the current block motion vector and two other values predicted using the neighboring motion vectors. This overlapping provides a coherent motion-compensated frame and therefore, a coherent motion residual frame free of artificial block discontinuities that a nonoverlapping motion compensation scheme would produce. Note that intra blocks are not overlapped with their neighboring blocks. In order to smooth the motion-compensated residual frame at the borders of intra blocks, their means are subtracted and sent as side information.

B. Discrete Wavelet Transform

A two-dimensional DWT is at the core of the proposed coder. The wavelet transform performs decomposition of video frames or motion-compensated residuals into a multiresolution subband representation. The DWT has been made extremely flexible by allowing explicit specification of parameters such as the number of decomposition levels, the filter coefficients to use, what filters to use at each level of the decomposition, and the filter-bank/wavelet-packet structure for the decomposition.

Multidimensional discrete wavelet transforms are usually implemented in the form of hierarchical tree structures of filter banks. The implementation of separable filter banks is efficient due to the fact that the decomposition is applied in each dimension separately. Therefore, using a simple iterative routine, the residual frame can be decomposed into four

subimages in each iteration. Our algorithm uses symmetric extension at the frame boundaries for the implementation of the filters of the wavelet transform in order to effect a nonexpansive decomposition [10].

One important set of parameters involves the choice of filters in the DWT. Linear-phase filters are preferred; otherwise, phase distortion around edges would be very visible. Therefore, we restrict the use of linear filters to linear phase filters only. Since the decorrelation of the subband signal is a desired property in a compression system, we use orthogonal or near-orthogonal filter banks. A special feature of our implementation is the ability to use different filter banks at each level of the decomposition. This is important because longer filters provide good frequency localization but can cause ringing artifacts along the edges of objects, while the use of shorter filter banks such as Harr results in less ringing artifacts but more blockiness in the reconstructed frame. Therefore, a combination of long filters for first levels and shorter filters for later levels provides a good tradeoff between ringing and blockiness artifacts.

C. Quantization and Zerotree Coding

1) Embedded Zerotree Wavelet Coding: EZW coding is a proven technique for coding wavelet transform coefficients. Besides superior compression performance, the advantages of EZW coding include simplicity, an embedded bitstream, scalability, and precise bit-rate control. These features enable EZW coding to address the MPEG-4 functionalities of improved coding efficiency, scalability, and error robustness, as well as providing other useful functionalities.

EZW coding is based on three key ideas: 1) exploiting the self-similarity inherent in the wavelet transform to predict the location of significant information across scales; 2) successive approximation quantization of the wavelet coefficients; and 3) universal lossless data compression using adaptive arithmetic coding. We give here a brief description of the EZW coding algorithm. Reference [9] describes the algorithm in further detail.

EZW coding is applied to coefficients resulting from a DWT. In our implementation we use a DWT identical to a hierarchical subband decomposition. The DWT decomposes the input frame into a set of subbands of varying resolutions. The coarsest subband is a lowpass approximation of the original frame, and the other subbands are finer-scale refinements. In the hierarchical subband system such as that of the wavelet transform, with the exception of the highest frequency subbands, every coefficient at a given scale can be related to a set of coefficients of similar orientation at the next finer scale. The coefficient at the coarse scale is called the parent, and all coefficients at the same spatial location and of similar orientation at the next finer scale are called children. As an example, Fig. 2 shows a wavelet tree descending from a coefficient in the subband HH_3. For the lowest frequency subband, LL_3 in the example, the parent-child relationship is defined such that each parent node has three children, one in each subband at the same scale and spatial location but different orientation.

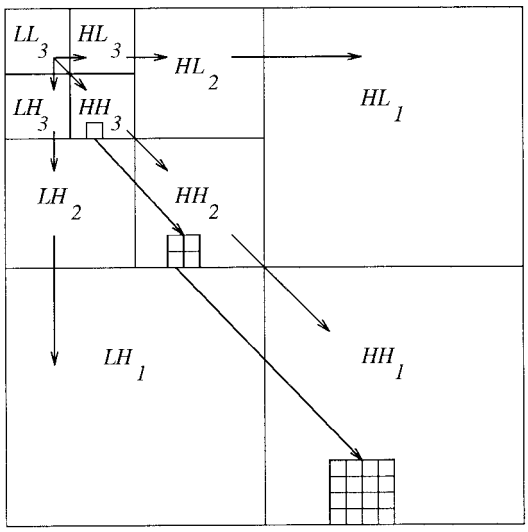


Fig. 2. Parent-child dependencies for creating wavelet trees.

EZW introduced a data structure called a *zerotree*, built on the parent-child relationship. The zerotree structure takes advantage of the principle that if a wavelet coefficient at a coarse scale is *insignificant* (quantized to zero) with respect to a given threshold T, then all wavelet coefficients of the same orientation at the same spatial location at finer wavelet scales are also likely to be insignificant with respect to that T. The zerotree structure is similar to the zigzag scanning and end-of-block symbol commonly used in coding DCT coefficients.

EZW scans wavelet coefficients subband by subband. Parents are scanned before any of their children, but only after all neighboring parents have been scanned. Each coefficient is compared against the current threshold T. A coefficient is significant if its amplitude is greater than T; such a coefficient is then encoded using one of the symbols *negative significant* or *positive significant*. The *zerotree root* symbol is used to signify a coefficient below T, with all its children in the zerotree data structure also below T. The *isolated zero* symbol signifies a coefficient below T, but with at least one child not below T. For significant coefficients, EZW further encodes coefficient values using a successive approximation quantization (SAQ) scheme. Coding is done bit-plane-by-bit-plane. The successive approximation approach to quantization of the wavelet coefficients leads to the embedded nature of an EZW coded bitstream.

2) Zerotree Entropy Coding: ZTE coding is a new efficient technique for coding wavelet transform coefficients of motion-compensated video residuals or of video frames. The technique is based on, but differs significantly from, the EZW algorithm. Like EZW, this new ZTE algorithm exploits the self-similarity inherent in the wavelet transform of images and video residuals to predict the location of information across wavelet scales. ZTE coding organizes quantized wavelet coefficients into wavelet trees and then uses zerotrees to reduce the number of bits required to represent those trees. ZTE differs from EZW in four major ways: 1) quantization is explicit instead of implicit and can be performed distinct from the zerotree growing process or can be incorporated into the process,

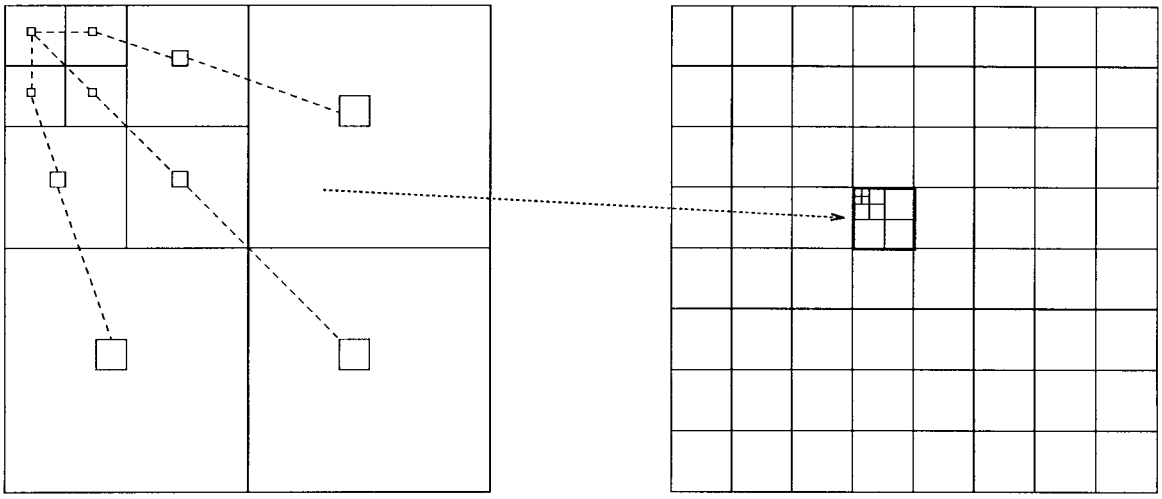


Fig. 3. Reorganization of a wavelet tree into a wavelet block.

thereby making it possible to adjust the quantization according to where the transform coefficient lies and what it represents in the frame; 2) coefficient scanning, tree growing, and coding are done in one pass instead of bit-plane-by-bit-plane; 3) coefficient scanning is changed from subband-by-subband to a depth-first traversal of each tree; and 4) the alphabet of symbols for classifying the tree nodes is changed to one that performs significantly better for very low bit-rate encoding of video. The ZTE algorithm does not produce an embedded bitstream as EZW does, but by sacrificing the embedding property, this scheme gains flexibility and other advantages over EZW coding, including substantial improvement in coding efficiency.

In ZTE coding, the coefficients of each wavelet tree are reorganized to form a wavelet block as shown in Fig. 3. Each wavelet block comprises those coefficients at all scales and orientations that correspond to the frame at the spatial location of that block. The concept of the wavelet block provides an association between wavelet coefficients and what they represent spatially in the frame.

To use ZTE, a symbol is assigned to each node in a wavelet tree describing the wavelet coefficient corresponding to that node. Quantization of the wavelet transform coefficients can be done prior to the construction of the wavelet tree, as a separate task, or quantization can be incorporated into the wavelet tree construction. In the second case, as a wavelet tree is traversed for coding, the wavelet coefficients can be quantized in an adaptive fashion, according to spatial location and/or frequency content.

The extreme quantization required to achieve a very low bit rate produces many zero coefficients. Zerotrees exist at any tree node where the coefficient is zero and all the node's children are zerotrees. The wavelet trees are efficiently represented and coded by scanning each tree depth-first from the root in the low-low band through the children, and assigning one of three symbols to each node encountered: *zerotree root*, *valued zerotree root*, or *value*. A zerotree root denotes a coefficient that is the root of a zerotree. Zerotrees do not need to be scanned further because it is known that all coefficients in such a tree have amplitude zero. A valued zerotree root is a node where the coefficient has a nonzero amplitude and all four children are zerotree roots. The scan of this tree can stop at this symbol. A value symbol identifies a coefficient with amplitude either zero or nonzero, but also with some nonzero descendant. The symbols and quantized coefficients are then losslessly encoded using an adaptive arithmetic coder.

The new ZTE improves upon EZW for very low bit-rate video coding in several significant ways. In EZW, quantization of the wavelet coefficients is done implicitly using successive approximation. When using ZTE, the quantization is explicit and can be made adaptive to scene content. Quantization can be done entirely before ZTE, or it can be integrated into ZTE and performed as the wavelet trees are traversed and the coefficients encoded. The quantizer we use is a midriser uniform quantizer with dead zone around zero, but other quantizers could be used if desired.

If coefficient quantization is performed as the trees are built, then it is possible to dynamically specify a global quantizer step size for each wavelet block (a *quant* for each block), as well as an individual quantizer step size for each coefficient of a block (a *quant matrix*). These quantizers can then be adjusted according to what the coefficients of a particular block represent (scene content), or according to what frequency band the coefficient represents, or both. The advantages of incorporating quantization into ZTE are: 1) the status of the encoding process and bit usage are available to the quantizer for adaptation purposes, and 2) by quantizing coefficients as the wavelet trees are traversed, information such as spatial location and frequency band is available to the quantizer for it to adapt accordingly and thus provide content-based coding.

Another advantage of ZTE coding over EZW comes from how the coefficients are scanned. EZW scans subband by subband. In ZTE, all coefficients that represent a given spatial block are scanned, in ascending frequency order from parent, to child, to grandchild, and so on, before the coefficients of the next adjacent spatial location are scanned. This is extremely valuable for rate control that adapts quantization block-by-block.

(a)

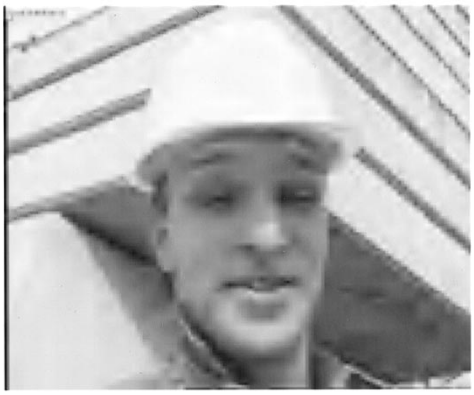
(b)

(c)

Fig. 4. I-frame coding of first frame of Akiyo at 14 kb by: (a) VM, (b) EZW, and (c) ZTE.

(a)

(b)



(c)

Fig. 5. I-frame coding of first frame of Foreman at 14 kb by: (a) VM, (b) EZW, and (c) ZTE.

Finally, the greatest advantage of ZTE over EZW coding is that, for ZTE coding, the alphabet of symbols used for classifying the wavelet tree nodes has been designed specifically for very low bit-rate coding of video. Consequently, ZTE yields significantly better performance than EZW for encoding motion-compensated video residuals and the same or better performance than EZW for I-frame coding at the very low bit rates.

3) Adaptive Arithmetic Coding: Symbols and quantized coefficient values generated by the zerotree stage are all encoded using an adaptive arithmetic coder, such as presented in [13]. The arithmetic coder is run over several data sets simultaneously. A separate model with associated alphabet is used for each. The arithmetic coder uses adaptive models to track the statistics of each set of input data, then encodes each set close to its entropy. The symbols encoded differ based upon whether EZW or ZTE coding is used.

For EZW, a four-symbol alphabet is used for the significance map, and a different two-symbol alphabet is used for the SAQ information. The arithmetic coder is restarted every time a new significance map is encoded or a new bit plane is encoded by SAQ.

(a)

(b)

(c)

Fig. 6. I-frame coding of first frame of News at 27 kb by: (a) VM, (b) EZW, and (c) ZTE.

For ZTE, symbols describing node type (zerotree root, valued zerotree root, or value) are encoded using a three-symbol alphabet. The list of nonzero quantized coefficients that correspond one-to-one with the valued zerotree root symbols are encoded using an alphabet that does not include zero. The remaining coefficients, which correspond one-to-one to the value symbols, are encoded using an alphabet that does include zero. For any node reached in a scan that is a leaf with no children, neither root symbol can apply. Therefore, bits are saved by not encoding any symbol for this node and encoding the coefficient along with those corresponding to the value symbol using the alphabet that includes zero.

ZTE coding quantizes wavelet coefficients and generates the zerotrees and quantized values in a specific order. These maps and values are saved in three different tables. Then, each table is losslessly coded using an arithmetic coding scheme. In order to be able to allocate bits among different wavelet blocks, the quantization, generation of zerotrees, and arithmetic coding of the wavelet coefficients must be performed in one single loop. In this case, we can calculate the number of bits used for coding of each wavelet coefficient right after arithmetic coding of its value and therefore control the bit rate to the block level.

In the arithmetic coder, three different tables (*type*, *valz*, *valnz*) must be coded at the same time. The statistics of each table are different and therefore the arithmetic coder must track at least three different probability models, one for each table. For each wavelet coefficient in any wavelet block, first the coefficient is quantized, then its type and value are calculated, and last, these values are arithmetic coded. The probability model of the arithmetic coder is switched appropriately for each table. At the end of each wavelet block, the number of bits used is calculated. This value is fed back to the rate-control algorithm in order to adjust the quantization bins for the next wavelet blocks.

The output of the ZTE coder is one single bitstream for each luminance and color component. Therefore, three different bitstreams are generated for each motion-compensated residual frame. The three bitstreams are concatenated and appropriate header is added to fit in the main output bitstream of the coder. In the cases in which all of the luminance or chrominance residual components are quantized to zero, a skip code is sent to minimize the coding cost of that residual component.

D. Rate Control

The proposed coder includes an optional advanced rate control scheme. The rate control scheme is implemented in both the picture and wavelet tree level where a second-order rate-distortion model is used for bit allocation. Based on a linear regression (LR) analysis, a new formula is derived to yield a smoother bit rate for each individual frame.

The rate control mechanism is performed in two stages: determination of the number of bits to be spent for each frame followed by the specific allocation of those bits within the frame. The target bit rate for each frame is calculated before the encoding of each frame. Assuming that each frame of the same prediction type has strong correlation in coding complexity, a target bit rate for the current frame is set as an average of the bits used in the previous frame and the available bits per frame. The weighting factor is the coefficient of the first-order autoregressive (AR) model.

In our coder, the rate controller can choose to perform intraframe rate control or no intraframe rate control. If the intraframe rate control is not activated, the rate controller needs to compute a quantization parameter (QP) for the whole frame. QP is obtained by solving a second-order model which is estimated by reviewing the buffer activities in the past. Such a technique is generalized to include I, P, and B picture encoding as described in [4]. It is desirable to have a single

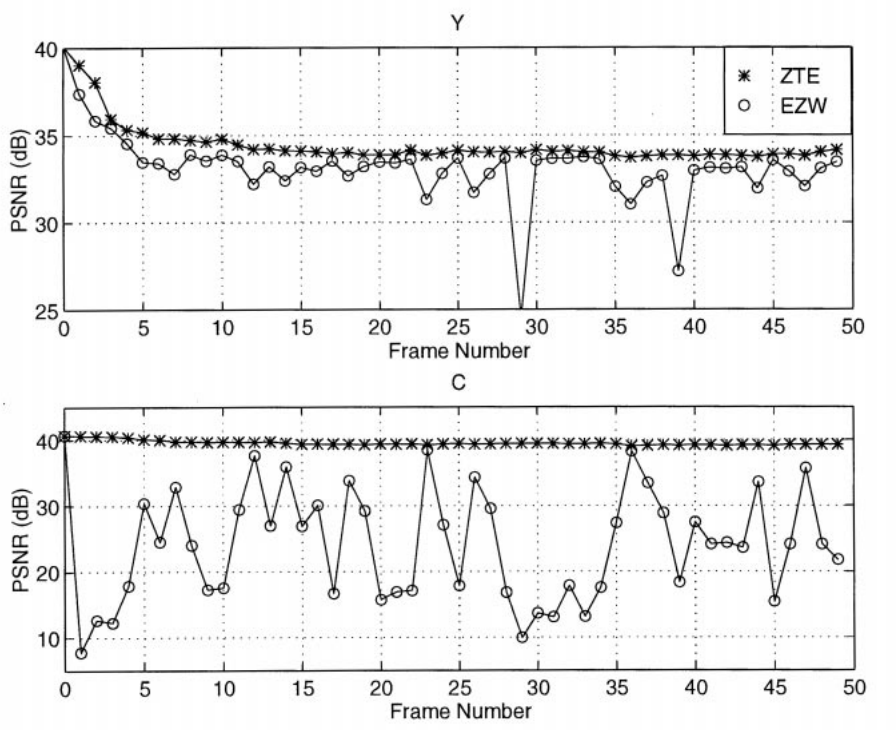


Fig. 7. P-frame coding PSNR results for "Akiyo" at 10 kb/s and 5 f/s. Results for ZTE are from a closed-loop coder. Results for EZW are valid for each residual only.

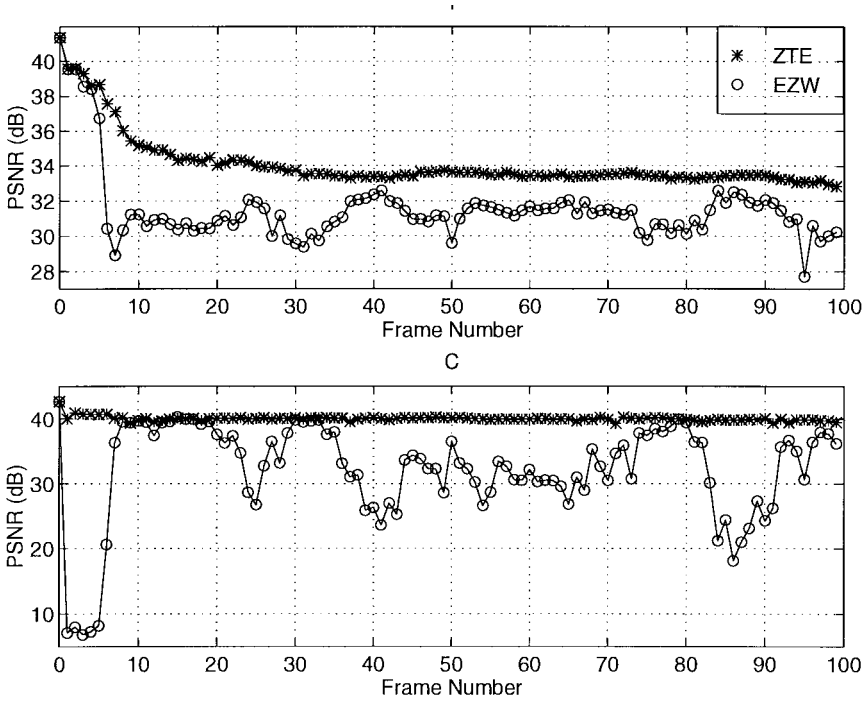


Fig. 8. P-frame coding PSNR results for Hall Monitor at 24 kb/s and 10 f/s. Results for ZTE are from a closed-loop coder. Results for EZW are valid for each residual only.

pass approach with low complexity. This approach has been adopted by MPEG-4 as the VM5.0 rate control scheme [15].

The rate controller can perform a more elaborate fine control if more implementation complexity is permissible. Bit allocation within a frame is accomplished by varying the quantization bin size for each wavelet tree. The encoder will perform multiple pass dummy encoding of the residuals to construct the rate distortion curve with a second-order function using LR analysis. In order to obtain enough data for modeling, the number of passes must be at least three. Each individual wavelet tree has a rate distortion model available for the frame to be encoded.

TABLE I
I-FRAME CODING PSNR RESULTS

Sequence	Bits	Y/C	VM	EZW	ZTE
Akiyo	14k	Y	33.06	33.77	34.62
		C	36.31	35.71	36.19
Akiyo	28k	Y	38.42	40.18	40.18
		C	40.81	39.50	40.81
News	14k	Y	28.60	29.38	29.38
		C	33.82	31.78	33.47
News	27k	Y	33.38	34.68	34.49
		C	37.39	35.12	36.84
Foreman	14k	Y	30.11	30.85	30.86
		C	38.27	37.57	38.69
Foreman	27k	Y	35.05	35.49	35.27
		C	40.71	41.23	40.76

TABLE II
ENTIRE SEQUENCE PSNR RESULTS

Sequence	Bit Rate	Y/C	VM	ZTE
Akiyo	10kbps	Y	34.61	34.61
@ 5 frames/sec		C	39.48	39.86
Hall Monitor	10kbps	Y	30.38	30.25
@ 5 frames/sec		C	37.78	38.05
Akiyo	24kbps	Y	37.46	36.64
@ 10 frames/sec		C	42.15	44.02
Hall Monitor	24kbps	Y	34.46	34.11
@ 10 frames/sec		C	39.38	39.63
Coastguard	48kbps	Y	29.74	29.20
@ 7.5 frames/sec		C	40.78	40.88
News	48kbps	Y	35.10	35.17
@ 7.5 frames/sec		C	39.11	40.46
Coastguard	112kbps	Y	31.50	31.01
@ 15 frames/sec		C	41.66	41.90
News	112kbps	Y	37.68	37.59
@ 15 frames/sec		C	41.20	42.55

IV. EXPERIMENTAL RESULTS

The proposed coder has been run to encode I and P frames using block motion estimation of sizes 16×16 or 8×8, overlapping block motion compensation, the discrete wavelet transform implementing the Daubechies' 9-3 tap filter bank at the first level of the decomposition of luminance followed by the 2-tap Haar filter for the remaining three levels of the decomposition of luminance, the discrete wavelet transform implementing the 2-tap Haar filter for all three levels of the decomposition of chrominance, rate control to set the quant for each frame, either EZW or ZTE coding for the first frame coded intra, and ZTE coding for all P predicted frames.

We ran several experiments using the three components (Y, Cb, Cr) of the MPEG-4 test sequences "Akiyo," "Hall Monitor," "Coastguard," "Foreman," and "News" at QCIF resolution. We compared ZTE coding to EZW coding and to the MPEG-4 Verification Model Version 1.0 (VM V1.0). The VM is a slightly modified DCT-based H.263 coder used by MPEG-4 as a reference for evaluation of new tools and algorithms. The VM and ZTE coders were run either with a fixed quant for all frames or with rate control used to set a quant for each frame. EZW was run to produce a bitstream at a specified bit rate.

In our first experiment we compared I-frame coding using ZTE to the VM and to EZW. We coded the first frame of Akiyo, Foreman, and News at 14 000 b and either 27 000 or 28 000 b by each of the three methods. In Table I we show the peak signal-to-noise ratio (PSNR) results for the luminance component in the row labeled "Y" and the average for the two chrominance components Cb and Cr in the row labeled "C." We see that EZW outperforms the VM but ZTE outperforms both the VM and EZW. The difference in quality can be clearly seen in Figs. 4–6.

In our second experiment we compared ZTE and EZW for P-frames. We first encoded the entire test sequence at a given frame rate and bit rate using ZTE coding and a fixed quant for all frames. Next, we re-encoded just the ZTE-derived residual for each frame using EZW to encode at the same number of bits that ZTE had used for that frame's residual. We then measured the error caused by each coding method. It is important to note that the results express the quality of ZTE versus EZW coding of identical residuals, but the higher quality ZTE residuals feed the encoder for succeeding frames. This was done in order to focus the comparison on encoding of residuals only.

Results from the second experiment for the sequences "Akiyo" and "Hall Monitor" are shown in Figs. 7 and 8, respectively. The PSNR reported for ZTE reflects the use of ZTE in a closed-loop coder as shown in Fig. 1. The PSNR result for each frame using EZW is valid only for that particular frame's residual. A closed-loop coder using feedback of EZW-coded frames would perform worse.

As the plots show, for "Akiyo" encoded using ZTE at the rate of 10 kb/s and 5 f/s, the average improvement in PSNR of ZTE over EZW for encoding luminance P-frames is 1.4 dB and for chrominance P-frames it is a staggering 15.7 dB. The average improvement in PSNR of ZTE over EZW for "Hall Monitor" encoded using ZTE at 24 kb/s and 10 f/s is 2.6 dB and 8.0 dB for luminance and chrominance P-frames, respectively.

The final experiment compared ZTE encoding of entire sequences (an initial I-frame followed by all P-frames) to VM encoding. Results for "Akiyo," "Hall Monitor," "Coastguard," and "News" at bit rates varying from 10 kb/s to 112 kb/s are shown in Table II. First the VM was run, then the ZTE coder was run with rate control set to match the bit rate achieved by the VM. Our rate control scheme, under these test conditions, was able to meet the target bit rate with an average error of only 0.09%. Average PSNR calculated over the entire coded sequences shows that our new ZTE coding algorithm achieves comparable performance to the MPEG-4 VM. However, the subjective quality resulting from ZTE coding is slightly better; in particular, blocking artifacts are reduced and objects are rendered better. The major advantage of our coder over the MPEG4 VM is its ability to address the scalability functionalities of MPEG-4 far more easily than VM 1.0.

V. Conclusion

This paper has presented a coder that represents a promising solution to several requirements of the MPEG-4 standardization effort. The major components of the coder are block motion estimation, overlapping block motion compensation, an adaptive discrete wavelet transform, the use of zerotrees and an adaptive arithmetic coder for encoding quantized wavelet coefficients, plus rate control.

The coder performs well, particularly on I-frames, and is directly extensible to provide the scalability functionalities sought by MPEG-4. Its components were submitted as tools to MPEG-4 in November 1995. The complete coder was submitted as an algorithm in January 1996 and was very well received. The components were incorporated into the core experiments as part of the MPEG-4 testing and standardization process.

Acknowledgment

The coder presented in this paper is the result of a strong team effort at the David Sarnoff Research Center. The authors would like to gratefully acknowledgment the invaluable time and effort contributed to the project by J. Lee, Z. Xiong, K. Williams, A. Marks, C. Wine, R. Jonsson, and H. Peterson.

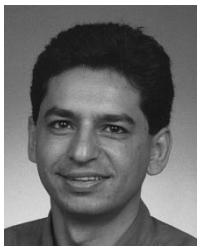
References

[1] Draft ITU-T Recommendation H.263, "Video coding for low bitrate communication," Dec. 1995.
[2] Document ISO/IEC JTC1/SC29/WG11, "Proposal package description (PPD)—Revision 2.0," Mar. 1995.
[3] Document ISO/IEC JTC1/SC29/WG11 MPEG96/M0637, "Very low bit rate video codec," Munich MPEG meeting, Jan. 1996.
[4] Document ISO/IEC JTC1/SC29/WG11 MPEG95/N0436, "A rate control scheme using a new rate-distortion model," Dallas MPEG meeting, Nov. 1995.
[5] Document ISO/IEC JTC1/SC29/WG11 MPEG95/N0437, "A flexible wavelet transform package for image and video representation," Dallas MPEG meeting, Nov. 1995.
[6] Document ISO/IEC JTC1/SC29/WG11 MPEG95/N0439, "Embedded zero-tree wavelet coding for video compression," Dallas MPEG meeting, Nov. 1995.
[7] Document ISO/IEC JTC1/SC29/WG11 MPEG95/N0441, "A zero-tree entropy coding tool for wavelet compression of video," Dallas MPEG meeting, Nov. 1995.
[8] S. A. Martucci and I. Sodagar, "Zerotree entropy coding of wavelet coefficients for very low bit rate video," presented at *Proc. 1996 IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, Sept. 1996.
[9] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.
[10] S. A. Martucci and R. M. Mersereau, "The symmetric convolution approach to the nonexpansive implementation of FIR filter banks for images," in *Proc. 1993 IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Minneapolis, MN, Apr. 1993, pp. V.65–V.68.
[11] H. Watanabe and S. Singhal, "Windowed motion compensation," in *SPIE Vol. 1605 Visual Communications and Image Processing '91: Visual Communication*, Boston, MA, Nov. 1991, pp. 582–589.
[12] C. Auyeung, J. Kosmach, M. Orchard, and T. Kalafatis, "Overlapped block motion compensation," in *SPIE Vol. 1818 Visual Communications and Image Processing '92*, Boston, MA, Nov. 1992, pp. 561–572.
[13] I. Witten, R. Neal, and J. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, June 1987.
[14] Y.-Q. Zhang and S. Zafar, "Motion-compensated wavelet transform coding for color video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 285–296, Sept. 1992.
[15] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion modeling," this issue, pp. 246–250.

Stephen A. Martucci (S'80–M'83) was born in Mineola, NY, in 1960. He received the B.E.E. degree, with highest honor, in 1982, the M.S.E.E. degree in 1987, and the Ph.D. degree in electrical engineering in 1993, all from the Georgia Institute of Technology, Atlanta. He won a DAAD (German Academic Exchange Service) scholarship and spent the period from October 1988 to December 1989 studying under Prof. H. G. Musmann at the Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung of the Technische Universität Hannover in Germany.

He is currently a Member of Technical Staff at the David Sarnoff Research Center, Princeton, NJ. His research interests include digital video, discrete transforms, multirate filter banks and wavelets, and image and video coding.

Dr. Martucci is a member of Sigma Xi, Phi Kappa Phi, Eta Kappa Nu, and Tau Beta Pi. He received the Young Investigator Award for best paper at SPIE VCIP'94.



Iraj Sodagar (S'87–M'95) was born in Hamedan, Iran, in 1964. He received the B.S. degree in electrical engineering from Tehran University, Iran, in 1987 and the M.S. and Ph.D. degrees in electrical engineering from Georgia Institute of Technology, Atlanta, in 1993 and 1994, respectively.

He was a Research Assistant at the School of Electrical Engineering at Georgia Tech from 1991 to 1994. Since 1995, he has been a Member of Technical Staff at the David Sarnoff Research Center, Princeton, NJ. His current research interests include multirate signal processing, filter banks and wavelets, and image and video coding.

Tihao Chiang (S'90–M'95) was born in Cha-Yi, Taiwan, R.O.C., 1965. He received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1987. He received the M.S. and Ph.D. degrees in electrical engineering from Columbia University, New York, NY, in 1991 and 1995, respectively.

In 1995, he joined David Sarnoff Research Center, Princeton, NJ, as a Member of Technical Staff. Since 1992 he has actively participated in ISO's Moving Picture Experts Group (MPEG) digital video coding standardization process with particular focus on the scalability/compatibility issue. He is also very active in the multiview profile ad hoc group in MPEG. His main research interests are compatible/scalable video compression, stereoscopic video coding, and ATM-based digital video application.

Ya-Qin Zhang (S'87–M'90–SM'93), for a photograph and biography, see this issue, p. 3.