HD-VCR Codec for Studio Application Using Quadtree Structured Binary Symbols in Wavelet Transform Domain

Hyun Meen Jung, Yongkyu Kim, Seunghyeon Rhee,
Hong-Yeop Song, and Kyu Tae Park

***Abstract*— A hierarchical wavelet transform coding structure which satisfies the requirements of high definition video cassette recorder (HD-VCR) for studio application in wireless environments is proposed. All of the coefficients in a transformed image are adaptively quantized and hierarchically converted into a single stream of binary symbols using the monotone decreasing property. The resulting bit stream goes through an adaptive arithmetic coder, and two-level control of bit rate is performed so as to maintain a constant target bit rate. The proposed algorithm can be efficiently implemented with much reduced computing time for the real world application. It is verified by simulation that for some test sequence of images at least 4 dB of PSNR improvement can be achieved compared with other DCT-based scheme at a constant target bit rate, avoiding the blocking effect which would be impossible otherwise.**

I. INTRODUCTION

Advancements in communications technology have accompanied an increased demand for services that provide high quality visual data. The digital signal processing can be a solution for such services. As the development of the digital high definition television (HDTV) systems has achieved a great progress, there has been a growing need for a new recording system, usually called "high definition video cassette recorder" (HD-VCR), with much increased data rate. This is mainly a consequence of the fact that the data rate of HDTV signals such as 1125/60 or 1250/50 is approximately up to five times that of standard definition TV signals.

In order to use HDTV signals for broadcasting in wireless environments, the medium for data recording has to meet several conditions. Two important factors are data transfer rate and storage capacity. If both the picture quality and the data amount are to be considered, the tape is known as the most efficient medium among the present recording media in order to store HDTV signals of duration at least 60 minutes [1]. Even though we record HDTV signal on the tape, however, the existing system cannot accommodate the recording capacity and the transfer rate of 1.2 Gb/s usually required by most types of HDTV [1].

Therefore, the two challenges in wireless environments for the digital video data processing of recording on and playing back from the tape are 1) the transmission (transfer) rate of a channel to send huge amount of visual data and 2) the capacity of a recorder to store them. One solution for both challenges would be to *compress* the visual data before any processing and to *decompress* it for the end users by which it can fit the limited transmission rate of the channels and the storage capacity of the recording systems.

A video coding method in video cassette recorders (VCR) should be frame-based in order to enable editing on the *frame-by-frame basis*, which is one of the highly needed features of HD digital VCR for studio application. For easy implementation of relatively *high speed playback*, the coding algorithm should be closely linked with the recording system. In addition, the compression should be done so that it maintains the almost original picture quality even after several rounds of duplication. To meet these requirements, the *intraframe coding* method should be employed. For HD digital VCR for studio application, it is widely accepted that the compression ratio should be less than 4 : 1 [2].

Wavelet transform [3]–[5] is a solution to avoid the *blocking effect* that occurs in any DCT-based coding scheme widely used currently for the data compression. The wavelet-transformed image has certain *geometric similarities* among the frequency bands having the same orientation in the spatial two dimensions. This similarity can be represented by various tree structures. Recently, several coding schemes in cascade with *adaptive arithmetic coders* [6]–[8] have been developed using various tree structures. The main limiting factors of these algorithms are the requirement of multiple scans for a given transformed image and multiple arithmetic coders each using different numbers of symbols.

Multiple scans will inevitably cause either the increased computational (or hardware) complexity or the increased transmission delay, and the multiple arithmetic coders each using different number of symbols will produce the final output consisting of multiple types of symbols to be carefully and sophisticatedly formatted for a tape to be recorded. For an application to HD-VCR, especially for studio application as well as for wireless environments for mobiles, these are not very attractive features, and should be resolved so that the number of scans is as small as possible and it uses as small number of arithmetic coders as possible, all of which use the same type of smaller number of symbols. We will show in this paper that we may be able to achieve these goals to the point of requiring two scans and only one arithmetic coder using only the binary symbols. The use of only the binary symbols at the arithmetic coder will simplify even more the implementation by avoiding the multiplication operation which would otherwise be impossible [9], [10].

In order to describe the proposed coding algorithm in detail, this paper is organized as follows. We will list some requirements of HD-VCR for studio application such as high picture quality, frame-by-frame basis editing, the multispeed playback feature, uniform compression ratio throughout the whole frame, the standard method localizing the error propagation, and multiple rounds of duplication in Section II. An appropriate tape format to satisfy these requirements is proposed in Section III. We will assume that input image has the resolutions of 1024 × 1920 and 1024 × 960 for luminance and chrominance components, respectively. Some of the basics on the wavelet transform of images and "base block decomposition" of the transformed image are explained in Section IV. We will discuss these topics only enough for our exposition of the proposed algorithm in Section V step-by-step in detail. Section VI will present some of the interesting results of simulation comparing the proposed algorithm with the DCT-based algorithm applied to both still images and test sequences of images having two abrupt scene changes in terms of the peak signal-to-noise ratio (PSNR) and the average bit rate achieved for the test sequence of images. Section VII will contain some concluding remarks.

II. REQUIREMENTS OF VCR FOR STUDIO APPLICATION

A. High Quality

VCR for studio application requires far better picture quality than the general household VCR's, and there should be no visual

---

Manuscript received August 2, 1995; revised November 27, 1995. This paper was recommended by Associate Editor H. Gharavi. This work was supported in part by Technology Planning & Coordinations of Samsung Electronics Co., Ltd., Suwon, Korea. Part of this paper was presented in the *IEEE International Conference on Consumer Electronics*, held in Chicago, USA, June 7–9, 1995.

The authors are with the Department of Electronic Engineering, Yonsei University, 134 Shinchon-Dong, Seodaemun-Gu, Seoul, 120-749, Korea.

Publisher Item Identifier S 1051-8215(96)05195-6.

1051–8215/96$05.00 © 1996 IEEE

deterioration after image compression and decompression. Under the subjective evaluation of image quality, one should not be able to distinguish the original images from those that underwent compression. When an objective evaluation method is applied, high quality means high PSNR. In order to maintain high quality, visual data must be compressed at a compression ratio of no higher than 4 : 1 in general [2].

B. Frame-by-Frame Editing

When a VCR is used as broadcasting equipment, the editing should be done on the frame-by-frame basis. In other words, one must be able to perform frame addition, frame deletion, frame retouch, etc., without affecting the other frames, which will only be possible if "intraframe coding scheme" is used instead of interframe coding schemes. On the other hand, Moving Pictures Expert Group (MPEG) is one example of an interframe coding scheme and, thus, cannot be used if the frame-by-frame editing is required.

C. Multispeed Playback

Since a VCR that uses the tape as a recording media can perform only the sequential playback, the multispeed playback is the most tricky problem for digital VCR. Slow or still playback is relatively easy to implement, but the fast forward feature is now being regarded as the hottest research topic in this area [1]. To achieve the fast forward functionality when neither the dynamic tracking head [11] nor the depth recording method is used, the input images should be decomposed into some smaller parts and then each part is compressed and recorded on a specific area of the tape. Therefore, there should be a close relation between the tape format and the coding algorithm to be employed. To satisfy these requirements, the coding algorithm must be designed such that each smaller segment of the image can be independently encoded and then decoded of other segments. Furthermore, if the compression can be done into a uniform size (at the price of low coding efficiency), it will be straightforward to realize the multispeed playback.

D. Uniform Compression by Frame

It is of utmost importance to compress each frame of the image into uniform size in order to 1) be able to effectively edit on the frame-by-frame basis and 2) realize multispeed playback with ease. We note that it is quite difficult to achieve this with such standard image compression methods as MPEG and JPEG (Joint Photographic Experts Group) because they use an adaptive quantizer, Huffman coding, and/or interframe coding.

In general, there are backward and forward controls [12] in the bit allocating methods to control the bit rate. The forward control works better for VCR for studio application than the backward control when the uniform bit control per frame of a very *narrow span* is implemented. To do the forward control [12], it is necessary to determine the value of such variables as quantizer step size using some statistics after images are placed on an input buffer.

E. Reduction of Error Propagation

Most errors in digital VCR systems usually take place between the tape and the magnetic head during the process of recording or playback. To minimize the error propagation, the use of error correcting code (ECC) is the most commonly adopted method in which the images are compressed and an appropriate overhead information is appended. Since the error correction capability of any ECC is quite limited, it is preferred and sometimes essential that the image compression technique itself be able to minimize the error propagation [13]. To localize the propagation of errors that have occurred beyond the ECC capability, it is preferred that the input image frame is decomposed into as small segments as possible.

F. Multiple Duplication

In case of digital VCR for studio application, degradation of picture quality is an unavoidable consequence of multiple duplication due to the electric error occurred at the rear part of ECC no matter how strong ECC may be employed. In practical terms, this phenomenon is usually far more crucial an element than the image compression with high definition quality intact. This requirement has some similarity with the previously mentioned requirement of minimizing the error propagation. In this case, the errors that have already occurred should remain as local as possible if they cannot be corrected.

III. TAPE FORMAT

HD-VCR coding scheme for studio application must achieve the uniform compression ratio per frame as well as per each of the smaller segments of a frame in order to realize the multispeed playback. This will definitely lower the coding efficiency since the characteristics of each segment (or frame) cannot be fully utilized. To increase the coding efficiency, therefore, it is necessary to predict the precise amount of output bits by thorough analysis of the input image. At the decoder, if errors beyond the error correction capability of the ECC take place during the process of recording or playback, it is necessary to signal such occurrences to the decoder so that an error concealment process [13] can be performed. Such a scheme will localize the error propagation only if each of the smaller segments of a frame (instead of a frame by itself) is independently encoded and can be decoded later at the decoder. To further increase the coding efficiency, we may maintain the uniform compression ratio per frame (instead of per its smaller segment).

We propose a decomposition of a frame into several smaller segments, each of them called a "base block." It refers to the *minimum unit* of the wavelet-transformed image for an independent encoding. As will be clearly described in Section IV, a base block consists of a "dc component" and all the higher frequency components having the same orientation for a given dc component. In this paper, we will consider the input image consisting of Y component (for luminance) of size 1024×1920, Cr and Cb components (for chrominance) each of size 1024×960. If we use the wavelet transform with $n = 4$ levels of approximation, then the wavelet-transformed data will contain "dc components" of size 64×120 for luminance and 64×60 for chrominance. Therefore, each of these N_b coefficients (dc components), where $N_b = 64 \times 120 + 2 \times (64 \times 60)$, will be a root node of N_b base blocks, and each base block will contain $16 \times 16 = 256$ transformed coefficients.

While the basic unit of encoding is a base block, the basic unit of recording on and playback from the VCR tape will be called a "SYNC block." A SYNC block consists of four base blocks which are two base blocks from Y frame and one base block from each of Cb and Cr frames. We choose and fix the size of a SYNC block as small as possible to efficiently satisfy two of the constraints of digital VCR described in Section II: 1) to ameliorate (localize) the error propagation and 2) to easily implement the multispeed playback.

The tape format proposed in this paper is explained in Table I. If the compression ratio is 4 : 1 [2] for the input frame of images described above, a SYNC block will contain, *on the average*, 256 bytes. If we use the values defined in Table I, each SYNC block will include two types of code bits in up to three fields as shown in Fig. 1. This consists of *independently decodable code* (IDC) and *dependently decodable code* (DDC) [12]. DDC is the code bits from the least significant bits (LSB) of Cr and Cb components in the

TABLE I
TAPE FORMAT

Item	Data
Compression Ratio	4:1
Image Size	1024 × 1920 (Y)
	1024 × 960 (Cb)
	1024 × 960 (Cr)
Rotations	5400 rpm
1 Frame	24 track/frame
	4 channel/frame
	3840 SYNC/frame
1 Channel	6 track/channel
1 Track	160 SYNC/track
1 SYNC Block	4 base block/SYNC
	256 byte/SYNC
1 Base Block	16 × 16 pixel/base block

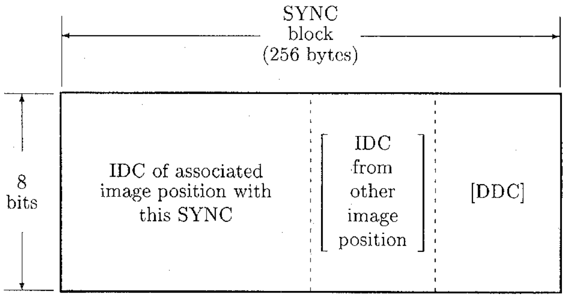

Fig. 1. Format of a SYNC block : IDC, IDC of other SYNC blocks, and DDC. Last two parts are optional and may be omitted if IDC part is too long.

highest frequency band, while IDC is the code bits from all the other coefficients of a frame. Again, two main reasons of distinguishing IDC from DDC are the implementation of the multispeed playback at the decoder and the control of a constant target bit rate at the encoder. The reconstructed frame without any bit of DDC still can maintain excellent image quality since most of the information of a given frame resides in IDC. During the high speed playback, therefore, the frame can be reconstructed from only IDC.

Each SYNC block has its own IDC that can be independently decoded regardless of any other SYNC blocks. DDC of any one SYNC block can be stored in any other SYNC blocks as long as these SYNC blocks have some available empty spots. A single SYNC block may include four base blocks and accommodates the compression results corresponding to the resolution of 16 × 32 in luminance (Y component) and two 16 × 16 in chrominance (Cr and Cb components).

Fig. 2 illustrates the packing of 160 SYNC blocks (after the entropy coding and bit rate control) onto one "track" of a VCR tape. Since each SYNC block may contain, *on the average*, 256 bytes, some SYNC blocks (blocks with the number 1, 2, 3, 4, and 7 in Fig. 2) may have less than 256 bytes and others (blocks with number 5 and 6) may have more. The amount of a SYNC block data in excess of 256 bytes is assigned to the position for some of the neighboring (short) SYNC blocks in order to 1) control the output bit rate, 2) increase the efficiency of compression, and 3) maintain the picture quality. In Fig. 2, it is illustrated that the excess amount of data in the SYNC block with number 5 is taken over to those with numbers 1 and 2, and the excess in the SYNC block with number 6 goes to those with numbers 2, 3, 4, and 7.

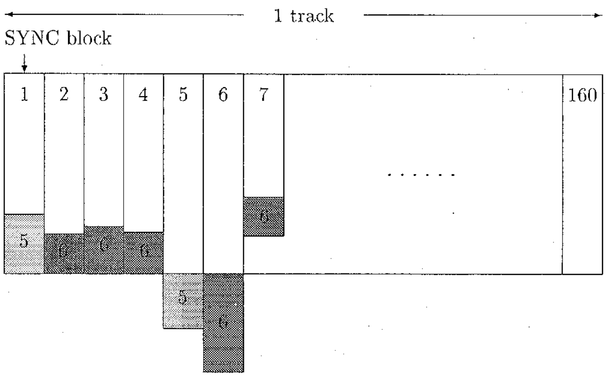

Fig. 2. Packing 160 SYNC blocks onto a track. The surplus bits of a SYNC block is assigned to some neighbors if possible.

IV. WAVELET TRANSFORM OF IMAGES AND QUADTREES

The wavelet transform does the following on an input image [3]–[5]. The input image is divided into four subbands (2 × 2 in both horizontal and vertical directions) that are equally spaced in logarithmic scale, and this can be considered as an *octave-band decomposition* in both directions. The lower frequency subband is further divided into the smaller four subbands and this process is continued until the desired level of decomposition (approximation) is achieved.

In general, when the original image has $N \times M$ pixels (nodes) and if the n levels of approximation are used in the wavelet transform, then the block corresponding to the lowest frequency band (dc components in short) becomes of size $N/2^n \times M/2^n$, and all of these $NM/2^{2n}$ coefficients will reflect the dc components of the original image.

For each of these $NM/2^{2n}$ dc coefficients, there corresponds to three coefficients in the directions of horizontal, vertical, and diagonal. Each of these three coefficients is associated with 2 × 2 coefficients for the same location in the higher frequency band and this relation continues until the highest frequency band is reached. Therefore, if we associate one of $NM/2^{2n}$ coefficients in the lowest frequency with a root, and its three coefficients with three different directions, these four and the rest of the coefficients for the same orientation in each successively higher frequency band form a "quadtree structure."

The above structure of the transformed image naturally gives a decomposition into $NM/2^{2n}$ quadtrees each of which is merely the disjoint set of coefficients maintaining the spatial and directional consistency in all the successive subbands. We call each of these quadtrees a "base block." The set of all the coefficients of one base block is shown in Fig. 3. Note that every base block contains $2^n \times 2^n$ coefficients provided that n levels of approximation are used in the wavelet transform. Therefore, from now on, we will use

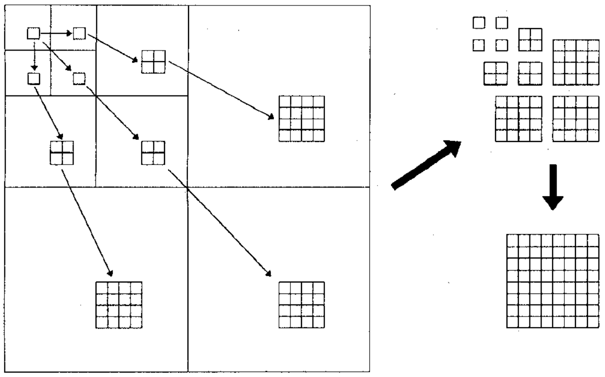

Fig. 3. Relation between a wavelet-transformed image and one of its base blocks. This example illustrates $n = 3$ levels of decomposition by the transform. The root and its three children are at the first level shown in the upper left-hand corner. Second level contains three 2 × 2 blocks, and third does 12 2 × 2 blocks.

the terms "coefficient" and "node" interchangeably due to the above correspondence of "base block" and "quadtree."

V. PROPOSED ALGORITHM

In this section, we will describe a coding algorithm that will generate a stream of binary symbols from the wavelet-transformed image and a given quantizer step size D. The proposed algorithm is designed such that 1) the output consists of a single stream of binary symbols for simple implementation, 2) the resulting bit stream is as efficiently encoded as possible by the entropy coder employed, 3) bit rate control is easily achieved, and 4) the final bit stream after entropy coding satisfies various requirements described in Section II so that it is appropriate for the HD-VCR tape for studio application. The entire block diagram of the encoder is shown in Fig. 4.

Recall the correspondence between the base block and its quadtree structure described in Section IV. We will use X (or Y) to denote any node in this quadtree, and S_X for the children of X as well as its all other descendants. For each node X, let $C(X)$ denote the quantized coefficient associated with X, and define $L'(X)$ as

$$L'(X) \triangleq \begin{cases} 0, & \text{if } C(X) = 0 \\ \lfloor 1 + \log_2 |C(X)| \rfloor, & \text{otherwise} \end{cases} \tag{1}$$

where $\lfloor \lambda \rfloor$ is the largest integer not exceeding λ.

Steps of the proposed algorithm can be described as follows.

A) *Quantization:* Quantize every coefficient into $C(X)$ corresponding to each node in the wavelet-transformed image using the quantizer step size D, and identify $L_{\max}$ which is the maximum of $L'(X)$ over all X.

B) *Bottom-Up Scanning of the Base Blocks (Quadtrees):* Decompose the resulting image into the base blocks (quadtrees) so that each base block can be independently encoded. For each base block, bottom-up scanning is performed in order to generate "binary length tree" such that the resulting tree satisfies "monotone decreasing property" for the binary lengths.

C) *Binary Bit Stream Generation:* For each base block, convert every $C(X)$ of the base block into a stream of binary symbols using the corresponding "binary length tree" constructed above.

D) *Entropy Coding:* Feed the resulting bit stream into the arithmetic coder and produce the output stream to be stored in a buffer for the bit rate control.

E) *Bit Rate Control:* Identify whether the bit stream in the buffer belongs to IDC or DDC. The output of the buffer at the constant bit rate will be fed into the SYNC block formatter for the tape. Some parameters of the coding results will be stored for the determination of the quantizer step size D for the next frame.

A. Quantization

The quantizer step size for the current frame should depend on the bit rate control of the previous frame so as to maintain a constant target bit rate. For the given current (wavelet-transformed) image, the same value, say D, is used throughout the entire coefficients of a frame. A uniform quantizer of midtread type with a dead zone is used. The resulting value $C(X)$ is given by

$$C(X) \triangleq \begin{cases} -k & \text{if } -(k+1)D < x \le -kD \\ 0 & \text{if } -D \le x < D \\ k & \text{if } kD \le x < (k+1)D \end{cases} \tag{2}$$

where x is the unquantized value (raw coefficient of the wavelet-transformed image) of node X. The maximum length, $L'(X)$, of $|C(X)|$ over the whole frame is identified during the quantization process, and we let $L_{\max}$ be $L'(X)$ of this node X.

B. Bottom-Up Scanning of the Base Block (Quadtree)

As described earlier, the wavelet-transformed image can be decomposed into base blocks, each of which can be independently encoded of other base blocks. For each base block, we first generate "binary length tree" (or, binary length block) by performing a bottom-up scanning. It will also decide L_b which is the maximum (over all X in the base block) of the lengths required to represent each $C(X)$. In most of the cases, for a given node X we have the property that $|C(X)| \ge |C(Y)|$ for any $Y \in S_X$, but this is not at all guaranteed. One reason for performing the bottom-up scanning is to create a "binary length tree" for this base block in such a way that $L'(X)$ is replaced with the maximum of $L'(Y)$ for $Y \in S_X$ if $L'(X) < L'(Y)$. We define $L(X)$ to be equal to either $L'(X)$ or the resulting value of such replacement. That is, letting Y to be one of (immediate) children of X

$$L(X) \triangleq \begin{cases} L'(Y) & \text{if } L'(Y) > L'(X) \text{ for some } Y \in S_X \\ L'(X) & \text{if } L'(Y) \le L'(X) \text{ for all } Y \in S_X. \end{cases} \tag{3}$$

This will ensure that for a given node X the resulting "binary length tree" has the property that $L(X) \ge L(Y)$ for all $Y \in S_X$. We call this a monotone decreasing property in binary lengths (MDP for short). For every node X (except for leaves at the bottom level), we attach a bit, called MDP_Bit, to indicate whether $L(X)$ is equal to $L'(X)$ or not

$$\text{MDP\_Bit}(X) \triangleq \begin{cases} 0 & \text{if } L(X) = L'(X) \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

After the binary length tree is constructed, all the values $C(X)$ in some specific order are converted into a single stream of binary symbols. We assume that $L_{\max}$ of the current frame is known to the decoder by some other method. The order of visits to the coefficients [14] is given as (1,1), (1,2), (2,1), (2,2) for a 2 × 2 block, and from the lowest frequency band to the higher frequencies, where (i, j) represents i in the horizontal direction from left to right and j in the vertical direction from top to bottom. The same pattern is used to visit four 2 × 2 blocks at the higher frequency band.

The bit stream begins by converting the value L_b of this base block into $L_{\max}$ binary symbols in which we insert one 1 at the L_b-th position from the right. Recall that $L_{\max}$ is the maximum length of $|C(X)|$ over the whole frame, and L_b is the maximum length of $|C(X)|$ over the current base block which is equal to the maximum of $L(X)$ over the current base block.

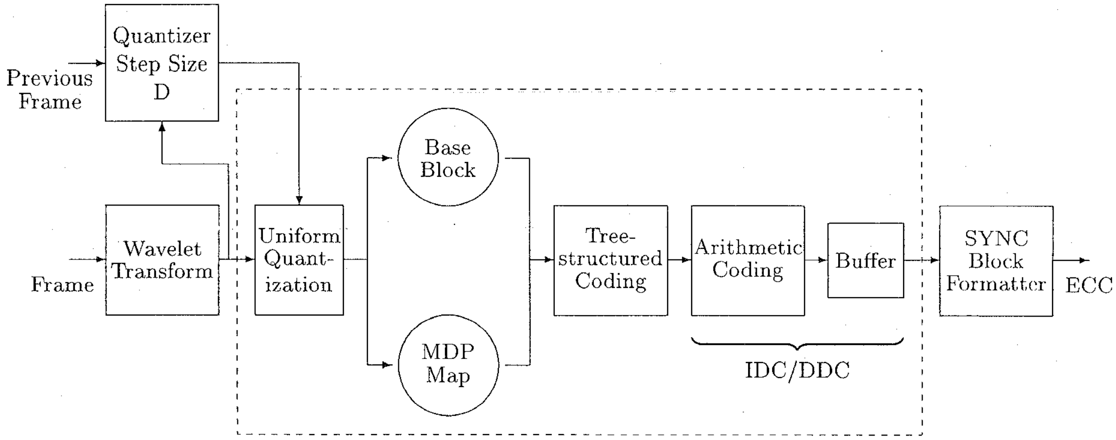

Fig. 4. Block diagram of the encoder in the proposed algorithm.

TABLE II
THREE FIELDS TO REPRESENT $C(X)$ IN BINARY SYMBOLS

Field	MDP_Bit(X)	$\lvert C(X)\rvert$ in the binary representation	SIGN_Bit(X)
No. of digits	1 or 0	$l(X)$	1 or 0

TABLE III
COMPARISON OF JPEG AND THE PROPOSED SCHEME

	Lena (512 × 512)		Baboon (512 × 512)	
	JPEG (dB)	Proposed (dB)	JPEG (dB)	Proposed (dB)
0.25 bpp	30.78	32.87	21.45	22.06
0.5 bpp	34.72	36.05	23.80	24.64
0.75 bpp	36.55	37.92	25.28	26.40
1.0 bpp	37.93	39.22	26.52	27.89

A node X of a base block becomes a leaf node if and only if any one of the following is satisfied: I) $L(X) = 0$ whenever X is *not* at the leaf level of the quadtree; II) X is at the leaf level of the quadtree regardless of the value $L(X)$. The bit stream will end whenever all the leaf nodes (whether or not at the leaf level) are converted. Note that the condition I) implies both $C(X) = 0$ and every coefficient of S_X is zero.

C. Binary Bit Stream Generation

Each of $C(X)$ in the order described earlier is converted into binary symbols in which there are three different *fields* in the order shown in Table II with the number of digits required for each field. MDP_Bit(X) will *not* be included if and only if X is at leaf level. Otherwise, MDP_Bit(X) defined in (4) of one symbol length comes first. SIGN_Bit(X) will *not* be included if $C(X) = 0$. Otherwise, SIGN_Bit(X) represents whether $C(X)$ is positive or negative and defined as

$$\text{SIGN\_Bit}(X) \triangleq \begin{cases} 1 & \text{if } C(X) > 0 \\ 0 & \text{if } C(X) < 0. \end{cases} \tag{5}$$

Let Y be the parent node (at exactly one level up) of X, and define $l(X)$ as

$$l(X) \triangleq \begin{cases} L_b & \text{if } X \text{ is a root} \\ L(Y) - 1 & \text{if MDP\_Bit}(X) \text{ is } 1 \\ L(Y) & \text{if MDP\_Bit}(X) \text{ is } 0 \text{ or if} \\ & \quad \text{there is no MDP\_Bit.} \end{cases} \tag{6}$$

Then, $|C(X)|$ is converted into the ordinary binary representation with some number of leading 0's stuffed to make its total length equal to $l(X)$.

We note that the above conversion process not only encodes $C(X)$ but also the "delimiter" of each $C(X)$ such that the resulting bit stream is *uniquely decodable* before the entropy coding, and furthermore, *comma-free* [15]. We also note that it is this difference between the proposed algorithm and any of the DCT-based algorithm in which the delimiter is inserted *during* the run-length Huffman coding instead of before the entropy coding. Therefore, it will be not easy to separately assess the contribution only of the proposed scan by using the Huffman coding after the above conversion process (delimiter will be inserted twice), and vice versa (no delimiter at all is encoded so that the result becomes not uniquely decodable), either.

D. Entropy Coding

The adaptive arithmetic coding [16] is an alternative method of the entropy coding to the widely used Huffman coding. Computational load of an arithmetic coder critically depends on the number of symbols used. The proposed algorithm uses only two symbols, and hence, it will be simple and relatively efficient to implement in a hardware. We note that using only the binary symbols makes it possible to dispense with the multiplication operation in the implementation.

The proposed algorithm produces a "highly unbalanced" bit stream of 0's and 1's in that it contains many more 0's than 1's. This is because 1) almost all of the nodes at the leaf level are 0's, 2) stuffed bits in the leading positions are all 0's whenever $l(X)$ is larger than the necessary number of binary digits for $|C(X)|$, and 3) most of MDP_Bit's are 0's. This will definitely lower the entropy of the input bit stream and contribute to the overall performance in addition to others.

E. Constant Bit Rate

Two steps of bit rate control are performed in the proposed algorithm: (a) adaptively change the quantizer step size D for each

98	34	-30	9	20	-2	0	0
-52	-13	5	0	12	0	0	0
0	0	-6	0	-20	0	0	0
0	-8	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)

7	6	5	4	0	0	0	0
6	4	5*	0	0	0	0	0
0	0	3	0	0	0	0	0
0	3	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b)

max length	$\overset{7}{001000000}$			
level 1	0 $\overset{98}{1100010}$ $\overset{+}{1}$	0 $\overset{34}{0100010}$ $\overset{-}{0}$	0 $\overset{52}{0110100}$ $\overset{-}{0}$	0 $\overset{13}{0001101}$ $\overset{-}{0}$
level 2	0 $\overset{30}{011110}$ $\overset{-}{0}$ 0 $\overset{0}{000000}$ 0 $\overset{6}{0110}$ $\overset{-}{0}$	0 $\overset{9}{001001}$ $\overset{+}{1}$ 0 $\overset{0}{000000}$ 0 $\overset{0}{0000}$	1 $\overset{5}{00101}$ $\overset{+}{1}$ 0 $\overset{0}{000000}$ 0 $\overset{0}{0000}$	0 $\overset{0}{000000}$ 0 $\overset{8}{001000}$ $\overset{-}{0}$ 0 $\overset{0}{0000}$
level 3	$\overset{20}{10100}$ $\overset{+}{1}$ $\overset{0}{0000}$ $\overset{20}{10100}$ $\overset{-}{0}$ $\overset{0}{0000}$ $\overset{0}{000}$	$\overset{2}{00010}$ $\overset{-}{0}$ $\overset{0}{0000}$ $\overset{0}{00000}$ $\overset{0}{0000}$ $\overset{0}{000}$	$\overset{12}{01100}$ $\overset{+}{1}$ $\overset{0}{0000}$ $\overset{0}{00000}$ $\overset{0}{0000}$ $\overset{0}{000}$	$\overset{0}{00000}$ $\overset{0}{0000}$ $\overset{0}{00000}$ $\overset{0}{0000}$ $\overset{0}{000}$

(c)

Fig. 5. An example of the proposed algorithm. (a) Base block. (b) Its binary length block(tree). (c) The coded bit stream. We assume that $L_{\max}$ is nine, and $\mathbf{5}^*$ in (b) indicates the reflection of sub-nodes and hence MDP_Bit is one.

input image, and (b) appropriately control the amount of DDC to be inserted in SYNC blocks according to the amount of IDC.

1) Determining the Quantizer Step Size D: The coefficient of a wavelet-transformed and quantized frame is regarded as "significant" if its absolute value exceeds some predefined threshold. If D increases, then the bit rate (in terms of bit per pixel) will decrease, and if D decreases, then the bit rate will increase. We note that the latter will be likely to cause the severe damage to the reconstructed image since some of the significant coefficients have to be discarded in order to maintain the constant bit rate.

Usually, not many differences exist between two consecutive frames except for the scene change and so on. This will also be true in the number of the significant coefficients between two consecutive frames. Thus, the value of D for the current frame is determined based upon the coding result of the previous frame as well as the histogram of the transform coefficients of the current frame [17]. The same value of D is used for both luminance and chrominance components to provide almost the same image quality.

2) Bit Rate Control: It is practically impossible to predict the precise amount of output bits from the arithmetic coder no matter how elaborate a forward control method may be used to decide the value of D. To maintain the constant bit rate per frame with the arithmetic coder, and to catch up with the target bit rate as fast as possible even when there is an abrupt scene change between the consecutive frames, the proposed algorithm uses the two-level control that depends on IDC and DDC [12]. The reconstructed frame without DDC can supposedly maintain the excellent image quality by assigning most of the "essential information" of the frame into IDC. The idea is that the frame is reconstructed from only IDC during the multispeed playback. The quantizer step size D should be determined such that the bit rate of only IDC is slightly smaller than the target and that the bit rate of the sum of IDC and DDC is slightly bigger than the target. This will leave a possibility of fine-tuning of the final bit rate by deciding the amount of DDC to be included in the final output, which, in turn, is to be fed into the SYNC block formatter for the tape.

F. Example

A complete example of the steps in V-B and V-C using a single base block of size 8 × 8 is shown in Fig. 5. The wavelet-transformed and quantized base block has three levels of nodes in addition to the root node. The value L_b of this base block is seven, and $L_{\max}$ (of the whole frame) is assumed to be nine. Note that we have $L_b \leq L_{\max}$ always.

Note that in any row of Fig. 5(c), the number of digits to represent $|C(X)|$ is $l(X)$ given in (6) in which some leading 0's are stuffed if longer than essentially required. Note also that the number of digits for MDP_Bits and SIGN_Bits in this example.

Finally, we observe the following two points: 1) approximately three quarters of the coefficients in this base block are not at all included in the output bit stream, and 2) the number of 0's is much more than the number of 1's in the output bit stream. The

TABLE IV
COMPARISON OF CODING PERFORMANCE (20 FRAMES EACH)

	Lena (512 x 512)		Baboon (512 x 512)	
	JPEG (dB)	Proposed (dB)	JPEG (dB)	Proposed (dB)
0.25 bpp	30.78	32.87	21.45	22.06
0.5 bpp	34.72	36.05	23.80	24.64
0.75 bpp	36.55	37.92	25.28	26.40
1.0 bpp	37.93	39.22	26.52	27.89

first point essentially provides the efficiency of the tree coding of a wavelet-transformed image, and the second is closely related to the performance of the arithmetic coder for the binary symbols.

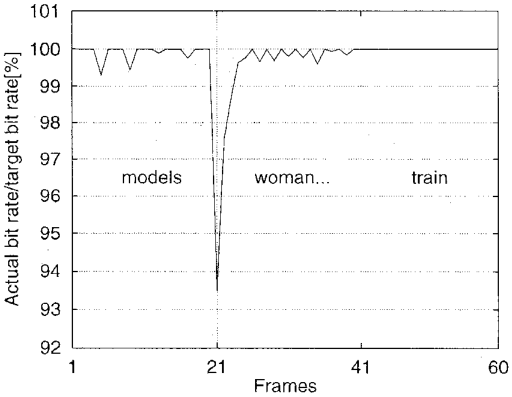

Fig. 6. Actual bit rate sequence of the 60 consecutive test frames with two changes of scenes at every 20 frames.

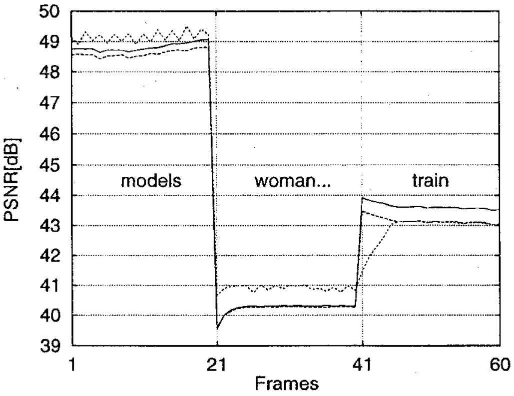

Fig. 7. PSNR of the luminance (solid line) and two chrominance (two broken lines) components in the 60 consecutive test frames of three types each with 20 frames.

VI. EXPERIMENTS AND RESULTS

The performance of the proposed algorithm is evaluated from the actual bit streams by simulations. The wavelet transform is implemented using Antonini's [3] 9-7 tap biorthogonal filters that can preserve the linear phase with relatively short taps. The coefficients of each filter are scaled so that the squared sum of the coefficients is one. The wavelet transform is performed to depth four with the symmetric signal extension method [18], [19] for the boundary of the image. Each coefficient is quantized uniformly with the given step size D. An arithmetic coder with adaptive models [16] is employed as the entropy encoder. To acquire universality of the coding scheme, three-byte header is used to indicate the quantizer step size D and $L_{\max}$ of the current frame. To evaluate the image quality, the PSNR between the original image $f(i,j)$ and the reconstructed image $f'(i,j)$ is calculated by, for frames of size $N \times M$,

$$\mathrm{PSNR} = 10\log_{10}\left\{\frac{1}{NM}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\frac{255^2}{[f(i,j)-f'(i,j)]^2}\right\}. \quad (7)$$

To compare the efficiency of the proposed coding scheme with that of JPEG, PSNR's obtained from still images (Lena and Baboon) processed by both algorithms are shown in Table III. This shows that the proposed algorithm is about 1 to 2 dB better than JPEG at the various compression ratio. These gains should have come from the tree structured scanning with MDP and the use of only the binary symbols. The tree structured coding reduces (by the adaptive quantization) the amount of coefficients to be encoded and transmitted, and the use of only the binary symbols with the unbalanced number of 1's and 0's in the encoding using MDP further reduces the amount of the output bits.

The validity of the bit rate control was evaluated by calculating the percentage of the average of actually encoded bit rates per target bit rate. In order to evaluate the performance of bit rate control and encoding as the scene changes, three kinds of video sequences of Models, Woman in Flowers, and Train are concatenated as a test sequence, so that there are two scene changes in the test sequence. Fig. 6 shows the result of the bit rate control for the above test sequence of 60 frames in terms of actual bit rate per target bit rate (%) for each frame. Fig. 7 shows the resulting PSNR of Y, Cr, and Cb components of the test frame sequences for each frame. Reconstructed test images yield average PSNR over 40 dB, and they were indistinguishable from the original images through subjective assessment. The percentage of the average of actually encoded bit rates to the target bit rate is 99.897% for the whole test image sequence including two scene changes. This clearly shows that the proposed algorithm is robust to the scene changes and yields high quality, and the proposed rate control method is fast enough to catch up with the target bit rate in a few frames (three to four frames as shown in Fig. 6). The same quantizer step size D is applied to all the color components, in order to have the same (or as close as possible) PSNR's among all the color components (as shown in Fig. 7) without any bit allocation algorithm. In addition, it is possible to allocate more bits to the luminance components than to the chrominance components since encoding of each color component can be done at relatively even PSNR level.

The results of simulation shows about 4 dB of PSNR improvement of the proposed scheme over the conventional DCT-based coding scheme [17], and furthermore, the reconstructed images from the proposed one avoid the blocking effect. Average PSNR of the 60 test frames compared to the DCT-based coding [17] for Models and Woman in Flowers are shown in Table IV at various compression ratio. The method of bit rate control in the proposed algorithm turned out to be also superior to the one in the existing method. Overall, we may conclude from these results that the proposed scheme is suitable for HD-VCR for studio application in wireless environments because 1) the constant bit rate per frame is achieved by two level coder [12], [20], 2) each SYNC block can be independently decoded of other blocks using only IDC during the high speed playback, and 3) each SYNC block is synchronized with the arithmetic coder for the robustness to the transmission errors.

VII. CONCLUSIONS

We propose a hierarchical coding (compression) scheme of moving pictures and an algorithm to control the output bit rate of the coding which altogether satisfy various requirements of HD-VCR for studio application in wireless environments.

By using the properties of the wavelet transform, the so called blocking effect is completely avoided. We use the natural correspondence between the quadtrees and the base blocks of the wavelet-transformed image in the decomposition of a frame into the base blocks to be processed independently of each other. The proposed algorithm produces a single stream of bits using only the binary symbols with highly unbalanced distribution on the number of 1's and 0's, which, in turn, results in better compression ratio at the entropy coder employed.

By using an adaptive quantization of updating the step size appropriately and using two-level forward control, the constant target bit rate is successfully achieved. In addition, the use of IDC and DDC makes it possible to recover the image only from IDC during the high speed playback, and hence, it becomes possible to realize one of the important features of HD-VCR, the multispeed playback, for studio application.

We still have several steps left in our algorithm which will require more rigorous analysis so that the contribution of each part to such an improvement over the other scheme can be accurately assessed. Part of our efforts to further improve the proposed algorithm with more careful analysis will soon be published hopefully.

REFERENCES

[1] S. W. Wu and A. Gersho, "Rate-constrained optimal block-adaptive coding for digital tape recording of HDTV," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, no. 1, pp. 100–112, 1991.

[2] P. Kauff, S. Rauthenberg, and R. Ritter, "An improved coding scheme for studio recording of interlaced and progressive HDTV signals," presented at *Int. Workshop on HDTV '92*, Nov. 1992.

[3] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Proc.*, vol. 1, no. 2, pp. 205–220, April 1992.

[4] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 674–693, 1989.

[5] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Mag.*, vol. 8, no. 4, pp. 14–38, Oct. 1991.

[6] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[7] P. Strobach, "Quadtree-structured recursive plane decomposition coding of images," *IEEE Trans. Signal Processing*, vol. 39, no. 6, pp. 1380–1395, June 1991.

[8] Y. K. Kim, H. G. Kim, and K. T. Park, "New tree structure with conditional height difference for wavelet transform image coding," *IEE Electron. Lett.*, vol. 31, no. 2, pp. 90–92, Jan. 1995.

[9] G. G. Langdon and J. Rissanen, "Compression of black-white images with arithmetic coding," *IEEE Trans. Commun.*, vol. COM-29, no. 6, pp. 858–867, June 1981.

[10] ——, "A simple general binary source," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 5, pp. 800–803, Sept. 1982.

[11] D. L. McLaren and A. Gersho, "Multiresolution high-definition television compression algorithm for digital video tape recording," *SPIE Opt. Eng.*, vol. 32, no. 7, pp. 1549–1558, July 1993.

[12] M. S. Hong, T. E. Kim, Y. K. Kim, G. M. Park, and J. K. Kim, "Adaptive bit allocation coder for DVCR trick play," in *Proc. IEEE Conf. on Consumer Electronics*, 1992, pp. 110–111.

[13] J. Watkinson, *The Art of Digital Video*. London & Boston: Focal Press, 1990, pp. 381–455.

[14] H. Samet, *The Design and Analysis of Spatial Data Structures*. Reading, MA: Addison-Wesley, 1990.

[15] S. W. Golomb, R. E. Peile, and R. A. Scholtz, *Basic Concepts in Information Theory and Coding: The Adventures of Secret Agent 00111*. New York and London: Plenum, 1994.

[16] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 30–44, June 1987.

[17] J. H. Kim, G. M. Park, and Y. J. Kim, "Two-layered DCT based coding scheme for a digital HD-VCR," in *Proc. IEEE Conf. on Consumer Electronics*, 1994, pp. 24–25.

[18] S. A. Martucci, "Signal extension and noncausal filtering for subband coding of images," in *Visual Commun. Image Processing '91, SPIE*, Sept. 1991, vol. 1605, pp. 137–148.

[19] H. J. Barnard, J. H. Webber, and J. Biemond, "Efficient signal extension for subband/wavelet decomposition of arbitrary length signals," in *SPIE*, 1993, vol. 2094, pp. 966–975.

[20] Y. K. Kim, H. M. Jung, G. M. Park, K. T. Park, and H. G. Kim, "A tree structured wavelet transform coding scheme for digital HD-VCR," in *Proc. IEEE Conf. on Consumer Electronics*, 1995, pp. 230–231.

Efficient Block Motion Estimation Using Integral Projections

Ken Sauer and Brian Schwartz

***Abstract*—Several efficient techniques have recently been proposed to reduce the computational burden of block matching for motion estimation in video coding. The goal is efficient motion estimation with minimal error in the motion-compensated predicted image. We present a block motion estimation scheme which is based on matching of integral projections of motion blocks with those of the search area in the previous frame. Like many other techniques, ours operates in a sequence of decreasing search radii, but it performs exhaustive search at each level of the hierarchy. The projection method is much less computationally costly than block matching and has prediction accuracy of competitive quality with both full block matching and other efficient techniques. Our algorithm also takes advantage of the similarity of motion vectors in adjacent blocks in typical imagery by subsampling the motion vector field. It has the added advantage of allowing parallel computation of vertical and horizontal displacements.**

I. INTRODUCTION

Full search block matching (FBM) is perhaps the most widely studied and applied technique for block motion estimation in video coding [1]. If performance in terms of prediction error is the only criterion for a motion estimation algorithm, FBM is generally superior, since it exhaustively searches for the displacement minimizing the cost function. But it is also computationally costly, and many less expensive alternatives have been presented with a variety of abbreviations in computational operations [2]–[6]. Several of these algorithms economize by reducing the number of locations at which block matching is performed. For example, Koga's three-step search (TSS) [2] operates in a hierarchy of logarithmically decreasing search distances, at each level matching most closely to one of nine locations,

Manuscript received June 16, 1994; revised February 11, 1996. This paper was recommended by Associate Editor J. Juhola.

K. Sauer is with the Laboratory for Image and Signal Analysis, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA.

B. Schwartz is with Delco Electronics Corporation, Kokomo, IN 46904 USA.

Publisher Item Identifier S 1051-8215(96)07097-8.

1051–8215/96$05.00 © 1996 IEEE