A low-complexity bit-plane entropy coding and rate control for 3-D DWT based video coding

Evgeny Belyaev, IEEE Member, Karen Egiazarian, IEEE Senior Member and Moncef Gabbouj, IEEE Fellow

Abstract—This paper is dedicated to fast video coding based on three-dimensional discrete wavelet transform. First, we propose a novel low-complexity bit-plane entropy coding of wavelet subbands based on Levenstein zero-run coder for low entropy contexts and adaptive binary range coder for other contexts. Second, we propose a rate-distortion efficient criterion for skipping 2-D wavelet transforms and entropy encoding based on parent-child subband tree. Finally, we propose one pass rate control which uses virtual buffer concept for adaptive Lagrange multiplier selection.

Simulations results show that the proposed video codec has a much lower computational complexity (from 2 to 6 times) for the same quality level compared to the H.264/AVC standard in the low complexity mode.

Index Terms—3-D DWT, entropy coding, rate control

I. INTRODUCTION

Video coding and transmission are core technologies used in numerous applications such as streaming, surveillance, conferencing, broadcasting and so on. Taking into account high bit error ratios, packet losses and time-varying bandwidth, the scalable video coding (SVC) is the most preferable compression method for video transmission [1], [2].

A scalable extension of the H.264/SVC standard [3], which is currently the most popular video coding approach, includes temporal, spatial and quality scalability and provides high compression efficiency due to motion compensation and inter-layer prediction exploiting the video source temporal redundancy and redundancy between different layers.

However, high computational complexity is a significant barrier for usage of the H.264/SVC on mobile devices, personal computers and other systems such as high definition video surveillance, wireless home TV, mobile IPTV broadcast etc. [4], [5], [6], which compress one or more high resolution video sources in real-time. This becomes an even greater a challenge in future real-time scalable multiview video coding systems which must be able to handle several views simultaneously.

The high computational complexity of H.264/SVC, as well as other hybrid video coding algorithms, are due to the following main reasons:

1) Motion compensation and estimation require a lot of computations even if fast motion vector searching algorithms are used. Additional problem occur in case of high resolution video sequences which requires high motion search radius for efficient coding in rate-distortion sense. This contradicts with low complexity coding case which requires as less as possible motion search radius. Therefore, in real-time applications the motion estimation can be not so efficient, especially taking into account that in some cases the compressed video sources does not contain significant motion.

2) To avoid resynchronization with the decoder side the motion compensation, as well as other encoder parts like intra-prediction, require backward loop in the encoding scheme. This loop includes inverse transform and quantization, debllocking filter and other extra calculations.

3) Macro-block coding mode selection as well as rate control also requires additional computations.

In real-life applications the H.264/SVC encoder should be significantly simplified, resulting in a decrease of its rate-distortion performance [7]. However, even after simplifications, listed above encoding parts have relatively high computational complexity. Taking into account that the future video coding standard HEVC [8] (High Efficiency Video Coding) is also based on hybrid video coding, its scalable extension is likely to have a high computational complexity too.

As an alternative to H.264/SVC encoders, scalable video encoder based on three-dimensional discrete wavelet transform (3-D DWT) can be used. During the last decade, several 3-D DWT approaches were proposed such as 3-D Set partitioning in hierarchical trees (3-D SPIHT) [9], 3-D Embedded subband coding with optimal truncation (3-D ESCOT) [10], Embedded video coding using zeroblocks of motion compensated 3-D subband/wavelet coefficients (MC-EZBC) [11] and so on. Due to intensive investigation of Motion Compensated Temporal Filtering (MCTF), currently 3-D DWT schemes have a comparable rate-distortion performance with the H.264/SVC [12]. However, as in the previous case, these schemes require also high computational resources. Therefore, development of an efficient low-complexity scalable video coders is an important practical problem.

In [5], a real-time 3-D run-length wavelet video encoder was proposed, which, however, does not include a rate control algorithm which is a key component of any video codec. In [6] and [13], new rate control algorithms based on an extension of the Embedded Block Coding with Optimized Truncation (EBCOT) [14] were proposed. These rate controllers minimize distortion for a given bit rate budget, but require a lot of computations due to the use of lossless compression of wavelet subbands requiring Lagrange multiplier selection. Other works propose rate control algorithms based on heuristics and additional video source characteristics (see, for example, [15]) and also require extra computations.
The main contribution of this work is a scalable low-complexity video codec based on 3-D DWT efficiently optimizing the rate-distortion-complexity trade-off and comprising the following tools:

1) The proposed codec uses two entropy encoding cores: Levenshtein zero-run coder for low entropy binary contexts and adaptive binary range coder (ABRC) [16] for the remaining binary contexts. Comparing to the traditional bit-plane encoders, e.g. [17], it allows to decrease the computation complexity without any significant rate-distortion performance degradation.

2) We propose an efficient rate-distortion criterion for skipping 2-D wavelet transforms and entropy encoding based on parent-child subbands tree. Using this criterion, the simulations results show that the computational complexity of the proposed codec tends to the complexity of the 1-D temporal transform since as the compression ratio increases, more and more 2-D subbands are skipped and 2-D transforms and entropy coding are only applied to at most 5% of the 2-D subbands.

3) We propose a one-pass rate control which uses the virtual buffer concept in order to estimate the Lagrange multiplier without resorting to lossless compressing or any additional computation.

The rest of the paper is organized as follows.

Section II presents a briefly review and categorization of existing wavelet-based video codecs along with a discussion of their computational complexity requirements. Section III describes a basic simplified JPEG2000 entropy encoding algorithm of wavelet subbands. Section IV introduces a low-complexity bit-plane entropy coding and parent-child skip criterion. Section V proposes the one-pass rate control based on adaptive Lagrange multiplier selection using the virtual buffer concept. The rate-distortion-complexity comparisons for the proposed codec, HEVC reference software, 3-D SPIHT and existing real-time and non real-time software implementations of H.264/AVC standard are presented in Section VI. Finally, conclusions are drawn in Section VII.

II. OVERVIEW OF 3-D DWT VIDEO CODING SCHEMES

From a video data decorrelation point of view, wavelet-based video coding schemes in the literature can be grouped in four major classes as follows:

1) The class of volumetric coding techniques, which treat the video sequence as a volume and perform a 3-D DWT followed by entropy coding of the resulting subbands [18], [19];

2) The class of in-band motion-compensated temporal filtering (MCTF) schemes performing first spatial discrete wavelet transforms followed by in-band MCTF and encoding of the resulting spatio-temporal subbands [20], [21];

3) The class of spatial-domain MCTF schemes which apply MCTF on the original image data and then transform and encode the residuals using a critically-sampled wavelet transform [11];

4) An extension of 2) and 3), a generic spatio-temporal wavelet decomposition structure alternating spatial transforms, MCTF and subsequent spatial transforms [22].

As computational complexity plays a key role, we shall next briefly evaluate the four classes in this regard. The main disadvantage of the second class is the necessity to calculate the spatial wavelet transform for each frame, even for video sequences which have group of frames (GOF) with low motion or without any motion. For such type of sequences, it is obvious that it is better to apply a temporal wavelet transform first, and then, depending on the motion level, one may skip spatial transform for several high-frequency frames in the GOF.

The third class and its extension (the fourth class) require motion compensation and motion estimation, which, as was mentioned in the introduction, improve rate-distortion performance but require a high computational complexity.

Following the reasoning above, for low complexity video coding, we consider in this paper the first class of wavelet-based video coding schemes and we use a simplified 3-D extension of the JPEG 2000 standard [23], [24] as the basic video coder. The adopted scheme is shown in Figure 1.

First, a group of frames (GOF) of length \( N \) are accumulated in the input frame buffer. Then, one-dimensional multilevel DWT of length \( N \) in the temporal direction is applied. All frames in the GOF are processed starting from low-frequency to high-frequency frames. For each frame, the spatial subbands are also processed from low-frequency to high-frequency spatial subbands.

Depending on a required bit rate and motion level in the current GOF, the rate controller chooses one of the following options for each spatial subband:

1) It permits calculation of 2-D spatial transform, selects the Lagrange multiplier \( \lambda \) (which defines the rate-distortion trade-off for the subband) and then permits entropy encoding for the subband;

2) It prohibits calculation of the 2-D spatial DWT and entropy coding for the subband. At the decoder side, the corresponding transform coefficients in the subband are considered to be zero.

III. BASIC SUBBAND ENTROPY ENCODER

Each wavelet subband is independently compressed using bit-plane entropy coding with a simplified JPEG2000 context modeling to split bit-planes into a set of binary sources.
After splitting, each binary source is compressed by an adaptive binary arithmetic coder from H.264/AVC standard [3] (M-coder) which is faster and more efficient than the MQ-coder used in JPEG2000 standard [25].

Let us define \( x[i,j] \) as a value of the coefficient with coordinates \( (i,j) \) in a wavelet subband of size \( S_w \times S_h \), \( s[i,j] \) as a significance flag, \( f[i,j] \) as a flag which shows that the current bit is (or is not) first after the significant bit. The proposed bit-plane entropy encoding procedure is given in Algorithm 1.

First, the number of significant bit-planes in subband \( n_{max} \) is calculated and encoded, subband distortion \( D \) is calculated and flags \( s[i,j] \) and \( f[i,j] \) are set to zero. Then each coefficient is processed from the highest to the lowest bit plane in a raster scan. For each coefficient \( x[i,j] \) the current bit value \( bit \) in bit-plane \( n \) is determined\(^1\).

If \( x[i,j] \) is not significant \( (s[i,j] = 0) \), then the number of significant neighbors \( H, V, D \) are calculated, context number \( c \) is determined as it is shown in Table I and \( bit \) value is compressed by an adaptive binary arithmetic encoder corresponding to the context number \( c \). If \( bit = 1 \) then \( x[i,j] \) becomes significant. In this case the significance flag \( s[i,j] \) is modified and the coefficient sign is compressed.

If \( x[i,j] \) is significant \( (s[i,j] = 1) \), then the current bit is compressed depending on the position after a significant bit. Finally, the subband distortion \( D \) is recalculated.

After encoding of each bit plane the bit stream size \( R \) of the subband is determined. If the Lagrange sum \( D + \lambda R \) for the bit plane \( n \) is greater than the same sum for the bit plane \( n+1 \), then the bit stream is truncated at the bit plane \( n+1 \). Otherwise, encoding is continued.

### Algorithm 1 Subband encoding procedure

**Input:** Subband \( \{x[i,j]\} \), \( \lambda \)

1: \( n_{max} \leftarrow \left\lfloor \log_2(\max_{(i,j)} x[i,j]) \right\rfloor \), \( D \leftarrow \sum_{(i,j)} x^2[i,j] \)

2: \( \forall (i,j) : s[i,j] \leftarrow 0, f[i,j] \leftarrow 0 \)

3: arithmetic_encode\( (n_{max}) \)

4: \( \psi \leftarrow D \)

5: for \( n = n_{max}...0 \) do

6: for \( i = 0...S_h - 1 \) do

7: for \( j = 0...S_w - 1 \) do

8: if \( |x[i,j]| \& 2^n \neq 0 \) then

9: \( \text{bit} \leftarrow 1 \)

10: else

11: \( \text{bit} \leftarrow 0 \)

12: end if

13: if \( s[i,j] = 0 \) then

14: \( H \leftarrow s[i-1,j] + s[i+1,j] \)

15: \( V \leftarrow s[i, j-1] + s[i,j+1] \)

16: \( D \leftarrow s[i-1,j-1] + s[i+1,j+1] \)

17: \( c \leftarrow \text{context_model}(H, V, D) \)

18: arithmetic_encode\( (\text{bit}, c) \)

19: if \( \text{bit} = 1 \) then

20: \( s[i,j] \leftarrow 1 \)

21: arithmetic_encode\( (\text{sign}(x[i,j]), 11) \)

22: end if

23: else

24: arithmetic_encode\( (\text{bit}, 9+f[i,j]) \)

25: \( f[i,j] \leftarrow 1 \)

26: end if

27: \( D \leftarrow D - ((|x[i,j]| & (2^{n+1}-1))^2 \)

28: \( D \leftarrow D + ((|x[i,j]| & (2^n-1))^2 \)

29: end for

30: end for

31: \( R \leftarrow \text{arithmetic_get_bit_stream_size()} \)

32: if \( \psi < D + \lambda R \) then

33: stop encoding

34: else

35: \( \psi \leftarrow D + \lambda R \)

36: end if

37: end for

---

\(^1\)Here we use "−" as the assignment operation, and \& as arithmetic 'and' operation

---

### IV. Proposed Low-Complexity Entropy Coder

#### A. Combined entropy coding

In comparison with the M-coder from H.264/AVC standard, the adaptive binary range coder (ABRC) proposed in [16] with the probability estimation based on a virtual sliding window has less computational complexity due to use of bytes as an output bit stream element and byte renormalization; and better compression efficiency due to the assignment of a specific window length selected according to the statistical properties of the corresponding binary source. Therefore, the computational complexity of the bit-plane entropy coder, described in section II, can be reduced due to the replacement of the M-coder by the ABRC. As shown in Figure 5, using the ABRC increases the encoding speed of 3-D DWT coder from 20% to 30% on average with an increase in compression performance. The encoding speed in this paper is defined as the number of frames which can be encoded in one second on the hardware platform with processor Intel Core 2 DUO CPU 3.0GHz. The computational complexity is considered as inverse value of the encoding speed.

In order to further decrease the computational complexity, let us consider the following properties of different binary contexts. Figure 2 shows the typical fraction of the context \( c = 0 \) in all input binary data depending on Peak signal-to-
Fig. 2: Fraction of context 0 in all input binary data

Fig. 3: Binary entropy estimation for different contexts

Fig. 4: Proposed combined entropy coder

Fig. 5: Diagram of the proposed combined entropy coder

A bit-plane entropy encoding with two buffers was proposed first by the authors in [26], but in this work contexts with a high entropy are not compressed at all.
corresponding coefficient in the parent subband. This property is widely used in image and video compression algorithms based on inter-subband correlation (see, e.g. [9]). Using the same reasoning, we first process subbands from low-frequency to high-frequency temporal subbands and from low-frequency to high-frequency spatial subbands. Second, we make the following coding assumption: if for any subband, inequality (2) holds, then for all temporal-spatial child-subbands, the same inequality also holds. In this case, all child-subbands will not be processed and hence spatial transform calculation and entropy coding steps are skipped. At the decoder side, the corresponding transform coefficients are considered to be zero.

Figure 6 illustrates the proposed approach for a GOF size \( N = 4 \) with two-level temporal and two-level spatial wavelet decomposition. Frames after temporal wavelet decomposition are denoted as \( L3, H2, H1, H0 \), where \( L3 \) is the low-frequency frame. If the spatial subband \( HH1 \) in the frame \( H2 \) is skipped then the corresponding child-subbands \( HH0 \) in the frame \( H2, HH1 \) and \( HH0 \) in the frame \( H2, HH1 \) and \( HH0 \) in the frame \( H0 \) are skipped without any processing.

Figure 7 shows the fraction of skipped wavelet subbands depending on the Y-PSNR for the different test video sequences. One can easily notice that this fraction is highly dependent on the motion level in the video sequence (compare for instance the results for Football which contains high motion with Vtc1nw which contains low motion). On the other hand, in all cases, as Y-PSNR decreases (or the compression ratio increases), the fraction of skipped subbands increases and reaches 95% in the limit. Therefore, as the compression ratio increases, the computational complexity of the proposed algorithm tends to the complexity of the 1-D temporal transform.

As shown in Figure 5, the proposed combined-coder with a parent-child tree skipping allows to increase the encoding speed of 3-D DWT coder from 2 to 7 times in comparison with one using the M-coder without any significant degradation of the rate-distortion performance.

Fig. 5: Rate-distortion-complexity comparison of different bit-plane entropy encoding approaches

Fig. 6: Example of parent-child skipping tree

Fig. 7: Efficiency of the proposed parent-child tree based subband skip criterion
V. PROPOSED LOW-COMPLEXITY RATE CONTROL

A. Optimal rate allocation based on Lagrangian relaxation

Consider \( n = \{n_i\} \) to be the truncation vector, where \( n_i \) signifies that \( i \)th subband in a GOF is truncated after encoding of bit-plane \( n_i \). Denote the overall GOF distortion after the truncation as

\[
D(n) = \sum_{i=0}^{i_{\text{max}}-1} w_i \cdot d_i^{n_i},
\]

where quantity \( d_i^{n_i} \) corresponds to the distortion value of \( i \)-th subband truncated in the bit-plane number \( n_i \), \( w_i \) is the weighting distortion coefficient of the \( i \)th subband. Similarly, denote the resulting GOF bit stream size as

\[
R(n) = \sum_{i=0}^{i_{\text{max}}-1} r_i^{n_i}.
\]

Then for each GOF we should determine the truncation vector \( n^* \), so that

\[
\begin{cases} 
    n^* = \arg \min_{n \in \mathbb{N}} D(n) \\
    R(n^*) \leq R_{\text{max}}
\end{cases}
\]

where \( R_{\text{max}} \) is the bit budget for each GOF which is defined as

\[
R_{\text{max}} = \frac{N \cdot C}{f},
\]

where \( N \) is the number of frames in the GOF, \( C \) is the required bit rate, and \( f \) is the video source frame rate.

The rate control task (5) can be solved using the concept of Lagrangian relaxation [27]. It can be proven, that for each \( \lambda \geq 0 \), the set of truncation vectors \( n^*_i \) minimizing

\[
D(n) + \lambda \cdot R(n)
\]

is the solution of the rate control task (5) for \( R_{\text{max}} = R(n^*_i) \).

From this statement it follows that to solve (5) we should find \( \lambda \), so that \( R(n^*_\lambda) = R_{\text{max}} \). It can be proven, see [27], that \( R(n^*_{\lambda}) \) is a non-increasing function of \( \lambda \), i.e. \( \lambda \) can be found by bisection method.

For the considered codec, the rate-distortion function for the subband \( i \) is independent of the truncation of other subbands; therefore,

\[
\min_{\mathbb{N}} \{D(n) + \lambda \cdot R(n)\} = \sum_{i} \min_{n_i} \{w_i \cdot d_i^{n_i} + \lambda \cdot r_i^{n_i}\},
\]

It means that a solution of the rate control task (5) can be written as \( n^* = \{n_0^*, n_1^*, ..., \} \), where

\[
n_i^* = \arg \min_{n_i} \{w_i \cdot d_i^{n_i} + \lambda \cdot r_i^{n_i}\}.
\]

B. One-pass rate control based on adaptive Lagrangian multiplier selection

To find the necessary \( \lambda \) value with the bisection method, we need to losslessly compress each subband in order to determine all sets of truncation points, which requires significant computational resources. As an alternative to the bisection method the rate control based on progressive construction of the convex hull of the rate-distortion function can be used [28]. In this approach encoder progressively adds bit-planes of all subbands in the GOF (starting from the highest bit-planes) until the required bit budget \( R_{\text{max}} \) is reached. But in this approach, the calculation of 2-D wavelet transforms for all subbands is needed in order to determine the number of significant bit-planes in each subband. It contradicts the 2-D wavelet transform calculation skipping proposed in Section IV-B.

In this paper we propose the one-pass rate control algorithm which does not require calculation of 2-D wavelet transform for all subbands in the GOF and uses the virtual buffer [29] concept to estimate \( \lambda \) without lossless coding. Let us define \( b_{\text{virt}} \) as the number of bits in a virtual buffer and \( B_{\text{max}} \) as the virtual buffer size determined as

\[
B_{\text{max}} = L \cdot C,
\]

where \( L \) is the required buffering latency [30] at the decoder side for continuous video playback. Then the rate control can be proposed as shown in Algorithm 2.

Algorithm 2 One-pass rate control procedure

1: \( b_{\text{virt}} \leftarrow b_{\text{virt}}(0) \)
2: for \( j = 0, ... , j_{\text{max}} - 1 \) do
3: \( \lambda \leftarrow \lambda_{\text{max}} \left( \frac{b_{\text{virt}}}{B_{\text{max}}} \right)^\gamma \)
4: for \( i = 0, ... , i_{\text{max}} - 1 \) do
5: For subband \( i \) in GOF \( j \) find
6: \( n_i^* = \arg \min_{n_i} \{w_i \cdot d_i^{n_i} + \lambda \cdot r_i^{n_i}\} \)
7: \( b_{\text{virt}} \leftarrow b_{\text{virt}} + r_i^{n_i^*} \)
8: end for
9: \( b_{\text{virt}} \leftarrow b_{\text{virt}} - R_{\text{max}} \)
10: end for

At step 3, Algorithm 2 selects the Lagrange multiplier \( \lambda \) value in the proportion to the virtual buffer fullness, where \( \gamma \) defines the proportion degree. It allows to increase or decrease the compression ratio for the current GOF depending on the buffer fullness. Then for subband \( i \) in GOF \( j \) the truncation point \( n_i \) is selected according to (9). Minimization of \( \psi(n_i) = w_i \cdot d_i^{n_i} + \lambda \cdot r_i^{n_i} \) is provided by steps 32–36 of Algorithm 1 with the assumption that \( \psi(n_i) \) is a convex function.

Note that if during the encoding process of a video sequence the number of bits in the virtual buffer requires

\[
0 \leq b_{\text{virt}} \leq B_{\text{max}},
\]

then the average bit stream size for each GOF \( \bar{R}(n) \) is bounded by

\[
R_{\text{max}} \leq \bar{R}(n) \leq R_{\text{max}} + \frac{B_{\text{max}}}{j_{\text{max}}},
\]

where \( j_{\text{max}} \) is the number of GOF’s in a video sequence. From (12) it follows that if (11) holds during the encoding process, then the average bit stream size for each GOF \( \bar{R}(n) \) tends to \( R_{\text{max}} \) with an increasing \( j_{\text{max}} \).

To further explain the proposed rate control Algorithm 2, let us suppose that the video encoder compresses the series of identical GOF’s. Let us define \( \lambda^* \) as the Lagrange multiplier value computed using the bisection method described in
Section V-A and consider the virtual buffer fullness when the initial buffer fullness is

\[ b_{virt}(0) = b_{virt}^\ast(0) = B_{\text{max}} \left( \frac{\lambda^\ast}{\lambda_{\text{max}}} \right)^{\frac{1}{\gamma}}. \]  

(13)

Taking into account that for the bisection method \( R(\lambda^\ast) = R_{\text{max}} \), after processing each GOF, the buffer fullness will be constant and equal to \( b_{virt} = b_{virt}^\ast(0) \).

Now let us analyze the buffer fullness in the case when the initial buffer fullness \( b_{virt}(0) \neq b_{virt}^\ast(0) \). Let us assume that after compression of the previous GOF the current buffer fullness is \( b_{virt} < b_{virt}^\ast(0) \). It means that \( \lambda \) value calculated by line 3 of Algorithm 2 for the current GOF will be less than the optimal \( \lambda^\ast \). Taking into account that \( R(\lambda) \) is a non-increasing function of \( \lambda \), after compression of this GOF the difference \( R(\lambda) - R_{\text{max}} > 0 \). Therefore, a new buffer fullness calculated by line 9 will be higher than previous one. It means that after compression of the next GOF, the current value of \( \lambda \) will be closer to the optimal \( \lambda^\ast \) and so on, until the current value becomes equal or exceeds \( \lambda^\ast \).

Analogically, if the current \( b_{virt} > b_{virt}^\ast(0) \), then after compression of the next GOF the current value of \( \lambda \) will also be closer to the optimal \( \lambda^\ast \) and so on, until the current \( \lambda \) value becomes equal or smaller than \( \lambda^\ast \).

Taking into account the reasoning described above, the value of \( \lambda \) calculated by the proposed algorithm will oscillate around the optimal value \( \lambda^\ast \).

As an example, Figure 8 shows the Lagrange multiplier selected by the bisection method and the proposed rate control for different initial virtual buffer fullness. One can see that, in practice, after some adaptation period, the value of the \( \lambda \) approaches the optimal one, computed by the bisection method.

It is important to notice, that in comparison with other heuristic rate control techniques, the proposed algorithm does not require any additional computations and provides close to the optimal Lagrange multiplier as in the bisection method.

VI. RATE-DISTORTION-COMPLEXITY COMPARISONS

Simulation results were obtained for the test video sequences [32], [33] with different frame resolutions: 640×480 (“Vassar_0”, “Ballroom_0”, “Vtc1nw” and “Football”), 704×576 (“Crew” and “Harbour”) and 1920×1080 (“Pedestrian Area”, “Rush Hour” and “Tractor”).

For our experiments, the proposed video coding algorithm is compared with the x.264 codec [31] which, as shown in [7], provides close to optimum rate-distortion performance for the H.264/AVC standard when computational complexity is significantly restricted. Therefore, this codec can be used as an upper bound of rate-distortion performance which can be achieved by H.264/AVC standard in a low complexity case.

The proposed codec was run with GOF size \( N = 16 \) using the Haar wavelet transform in the temporal direction and the 5/3 spatial lifting wavelet transform at three-levels of the decomposition\(^3\). x.264 codec was run in very low complexity mode which corresponds to the Baseline profile of H.264/AVC with intra-frame period 16\(^4\).

Additionally, for the first four test video sequences with low frame resolution we have measured the rate-distortion-complexity performance for the reference software of the HEVC standard (HM version 9.1 in low delay configuration), 3-D SPIHT codec, x.264 codec in high complexity mode\(^5\) (which corresponds to the High profile of the H.264/AVC standard) and x.264 codec in very low complexity mode with Intra-frame coding only\(^6\). In all cases, the codecs were simulated using a constant bit rate mode and the encoding speed was measured without any use of assemblers, threads, or other program optimization techniques.

Simulation results, presented in Figures 9–11, show the rate-distortion-complexity comparison for the considered codecs. Additional computation complexity comparisons based on CPU cycles measurements in Tables II–V are presented.

From Figures 9–10, it can be seen that the highly complex video codecs (HEVC, 3-D SPIHT and x.264 in high complexity mode) provide significantly better rate-distortion performance than low complexity video codecs and, as mentioned in the introduction, the reduction in complexity causes a significant degradation in the rate distortion performance of H.264/AVC, especially for sequences with high level of motion (for example, more than 4.5 dB for Football).

Furthermore, the proposed 3-D DWT video codec provides from 2 to 6 times lower computational complexity for the same Y-PSNR level compared to the H.264/AVC in the low complexity mode, when the number of frames which can be encoded in one second is used as the complexity metric (see the encoding speed in Figures 9–11), and from 2 to 11

\(^3\)The proposed codec can be found at http://www.cs.tut.fi/~belyaev/3d_dwt.htm

\(^4\)Command line example: x264.exe –output vassar_0.264 vassar_0.avs – preset ultrafast –keyint 16 –bitrate 1000 –no-asm –threads 1

\(^5\)Command line example: x264.exe –output vassar_0.264 vassar_0.avs – preset veryslow –keyint 16 –bitrate 1000 –no-asm –threads 1

\(^6\)Command line example: x264.exe –output vassar_0.264 vassar_0.avs – preset ultrafast –keyint 1 –bitrate 1000 –no-asm –threads 1
times lower computational complexity, when CPU cycles are used as the complexity metric (see Tables II–V). The second complexity metric does not take into account the time needed for memory access. Therefore the maximum value of the complexity gain measured by this metric is higher than the one measured using the first metric.

For visual assessment we depicted two frames from different video sequences: frame 45 of video sequence “Ballroom_0”, when the required bit rate is 500 kbps (see Figures 12–17), and frame 45 of video sequence “Football”, when the required bit rate is 1000 kbps (see Figures 18–23). One can see, that the visual quality for the proposed codec is comparable with x.264 in the low-complexity mode (see Figures 14 and 16 or Figures 20 and 22) and significantly better than x.264 in the INTRA mode (see Figures 17 and 23).

VII. CONCLUSION

In this paper a real-time scalable video codec based on 3-D DWT with a low-complexity bit-plane entropy coding and rate control was presented. Simulations showed that the proposed codec exhibits a lower computational complexity compared to the most efficient software implementation of H.264/AVC with comparable rate-distortion performance.

|TABLE II: CPU cycles for “Vassar_0”|

<table>
<thead>
<tr>
<th>Required bit rate, kbps</th>
<th>CPU cycles for x.264 ultrafast</th>
<th>CPU cycles for the proposed 3-D DWT</th>
<th>Computation complexity gain, times</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000</td>
<td>34726795256</td>
<td>13973509441</td>
<td>2.49</td>
</tr>
<tr>
<td>1000</td>
<td>24600141863</td>
<td>6374088375</td>
<td>3.86</td>
</tr>
<tr>
<td>500</td>
<td>19930517294</td>
<td>4441827624</td>
<td>4.49</td>
</tr>
<tr>
<td>250</td>
<td>17903180874</td>
<td>378875063</td>
<td>4.73</td>
</tr>
</tbody>
</table>

|TABLE III: CPU cycles for “Ballroom_0”|

<table>
<thead>
<tr>
<th>Required bit rate, kbps</th>
<th>CPU cycles for x.264 ultrafast</th>
<th>CPU cycles for the proposed 3-D DWT</th>
<th>Computation complexity gain, times</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000</td>
<td>3526087063</td>
<td>13605051412</td>
<td>2.30</td>
</tr>
<tr>
<td>1000</td>
<td>2535869723</td>
<td>713677272</td>
<td>3.55</td>
</tr>
<tr>
<td>500</td>
<td>23043646383</td>
<td>5199852570</td>
<td>4.43</td>
</tr>
<tr>
<td>250</td>
<td>20842157619</td>
<td>3645741812</td>
<td>5.72</td>
</tr>
</tbody>
</table>

|TABLE IV: CPU cycles for “Vtc1nw”|

<table>
<thead>
<tr>
<th>Required bit rate, kbps</th>
<th>CPU cycles for x.264 ultrafast</th>
<th>CPU cycles for the proposed 3-D DWT</th>
<th>Computation complexity gain, times</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>34040990912</td>
<td>11357117783</td>
<td>3.00</td>
</tr>
<tr>
<td>500</td>
<td>24937234723</td>
<td>5135051986</td>
<td>4.86</td>
</tr>
<tr>
<td>100</td>
<td>23773185119</td>
<td>3149471200</td>
<td>7.55</td>
</tr>
<tr>
<td>50</td>
<td>23717868942</td>
<td>2033751456</td>
<td>11.66</td>
</tr>
</tbody>
</table>

The proposed scheme has a much lower computational complexity than the low complexity profile of H.264/AVC due to the following reasons:

1) It does not require backward loop in encoding scheme, because motion compensation, intra-prediction, deblocking filter and so on are not used and distortion caused by quantization can be estimated in wavelet domain without this loop.

2) The proposed entropy encoder uses very simple zero-run coding based on Levenstein codes for low entropy contexts. It helps avoid using any coding tables as in CA VLC used in H.264/AVC. The remaining contexts are compressed by ABRC [16]. Taking into account that ABRC is used approximately only for 12% of the input binary data, its complexity does not affects significantly the total complexity while achieving a high compression performance.

3) The proposed parent-child tree based subband skip criterion allows skipping the 2-D wavelet transform calculation and encoding without any additional computations. The use of this criterion does not lead to a significant degradation of the rate-distortion performance and, with increasing of the compression ratio, more and more spatial subbands are skipped. Therefore, the computational complexity of the proposed codec tends to the complexity of the 1-D temporal transform. It is important to notice that the same idea cannot be effectively implemented in the H.264/AVC which requires extra analysis for motion estimation, coding mode selection, and so on, even after simplifications.

4) The proposed one-pass rate control does not require any additional computations for video signal analysis and provides a near optimal Lagrange multiplier if the statistical properties of the frames in neighboring GOFs are similar.

In this paper the proposed codec is not compared with a scalable extension of the H.264/AVC standard, because the authors have not found any open source real-time software implementation of it. But, taking into account that H.264/SVC has lower rate-distortion performance [34] and higher computational complexity (due to additional inter-layer prediction, input frame downsampling etc.) than H.264/AVC single-layer coding, our scalable video compression scheme can be more preferable than H.264/SVC in many applications where computational complexity and scalability plays a critical role.

The proposed entropy encoding and rate control can be also easily adopted for spatial-domain motion-compensated temporal filtering based video coding and, combined with a complexity scalable motion compensation [35], can achieve better rate-distortion-complexity performance for high complexity encoding modes.

REFERENCES


x.264 video codec, http://x264.nl/


High Efficiency Video Coding, http://www.h265.net/


Fig. 9: Rate-distortion-complexity comparison of fast implementation of H.264/AVC standard and proposed 3-D DWT codec
Fig. 10: Rate-distortion-complexity comparison of fast implementation of H.264/AVC standard and proposed 3-D DWT codec
Fig. 11: Rate-distortion-complexity comparison of fast implementation of H.264/AVC standard and proposed 3-D DWT codec
Fig. 12: HEVC, PSNR=35.1 dB, Speed=0.02 fps

Fig. 13: 3-D SPIHT, PSNR=32.2 dB, Speed=9.2 fps

Fig. 14: x.264 ultrafast, PSNR=29.8 dB, Speed=79.6 fps

Fig. 15: x.264 veryslow, PSNR=32.8 dB, Speed=1.8 fps

Fig. 16: Proposed 3-D DWT, PSNR=31.2 dB, Speed=238 fps

Fig. 17: x.264 INTRA, PSNR=24.4 dB, Speed=74.1 fps
Fig. 18: HEVC, PSNR=34.2 dB, Speed=0.02 fps

Fig. 19: 3-D SPIHT, PSNR=30.4 dB, Speed = 8.6 fps

Fig. 20: x.264 ultrafast, PSNR=29.2 dB, Speed=61.4 fps

Fig. 21: x.264 veryslow, PSNR=32.5 dB, Speed=1.1 fps

Fig. 22: Proposed 3-D DWT, PSNR=27.8 dB, Speed=217 fps

Fig. 23: x.264 INTRA, PSNR=25.2 dB, Speed=72.1 fps