High-Performance Wavelet-Based Video Coding Using Variable Block-Size Motion Compensation and Adaptive Arithmetic Coding

Detlev Marpe
Image Processing Department
Heinrich-Hertz-Institute (HHI)
Einsteinufer 37
10587 Berlin, Germany
email: marpe@hhi.de

Hans L. Cycon
University of Applied Sciences
FHTW Berlin
Allee der Kosmonauten 20–22
10315 Berlin, Germany
email: hcycon@fhtw-berlin.de

ABSTRACT
In this paper, we present a wavelet-based video coding algorithm within a conventional hybrid framework of temporal motion-compensated prediction and transform coding. Our proposed algorithm involves the incorporation of multi-frame motion compensation as an effective means of improving the quality of the temporal prediction. For an efficient coding control, we follow the rate-distortion optimizing strategy of using a Lagrangian cost function to discriminate between different decisions in the video encoding process. As a key element for fast adaptation and high coding efficiency, our approach uses an entropy coding scheme based on context-based adaptive arithmetic coding. In addition, the combination of overlapped block motion compensation and frame-based transform coding guarantees blocking artifact free and hence subjectively more pleasing video. In comparison with a highly optimized MPEG-4 Advanced Simple Profile coder, our proposed scheme provides significant performance gains in objective quality of 2–2.5 dB PSNR.

KEY WORDS
Video compression, wavelet-based video coding, very low bit-rate coding, H.26L, MPEG-4

1 Introduction

Multi-frame prediction [1] and variable block-size motion compensation in a rate-distortion optimized motion estimation and mode selection process [2, 3] are powerful tools to improve the coding efficiency of today's video coding standards like MPEG-4 Visual [4] and ITU-T H.263 [5]. In this paper, we present the novel *design of a video coder* (*DVC*) that demonstrates how these state-of-the-art coding tools as implemented in the current test model TML-8 [6] of the ITU-T H.26L video compression standardization project can be successfully integrated in a blocking artifact free video coding environment. In addition, we propose a solution for an efficient macroblock-based intra coding mode within a frame-based residual coder, which is extremely beneficial for improving the subjective quality. We further demonstrate how to improve the efficiency of a wavelet-based residual coder by using appropriately designed entropy coding tools as introduced in our previous work [7, 8].

This paper is organized as follows. Section 2 gives an overview of our proposed video coding system. The instruments of motion compensated prediction along with the macroblock-based intra coding mode are described in Section 3. The specific wavelet transform used in our approach is introduced in Section 4, and the associated pre-coding scheme for transform coefficients is reviewed in Section 5. In Section 6, more details of the adaptive entropy coding scheme are given. Experimental results comparing our proposed wavelet-based DVC coder against an optimized MPEG-4 and H.26L TML-8 coder [3, 6] are presented in Section 7. Section 8 finally contains concluding remarks and suggestions for future work.

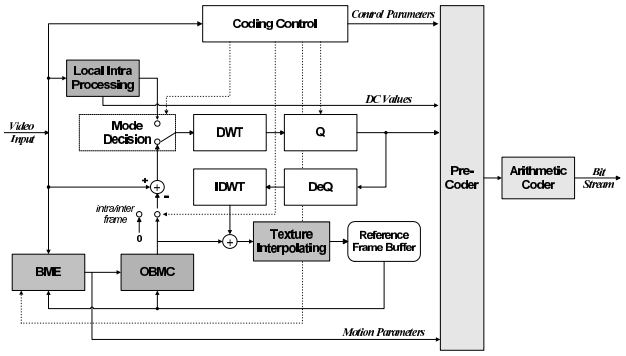


Figure 1. Block diagram of the proposed coding scheme

2 Overview of the DVC Scheme

Fig. 1 shows a block diagram of our proposed coder. As a hybrid system, it consists of a temporal predictive feedback loop along with a spatial transform coder. Temporal prediction is performed by using *block motion estimation* (BME) and *overlapped block motion compensation* (OBMC), where the reference of each predicted block can be obtained from a long-term *reference frame memory*.

Coding of inter frames (*P-frames*) as well as of intra frames (*I-frame)* is performed by first applying a *discrete wavelet transform* (DWT) to an entire frame. Uniform scalar *quantization* (Q) with a central dead-zone around zero similar to that designed for H.263 is then used to map the dynamic range of the wavelet coefficients to a reduced alphabet of decision levels. Prior to the final *arithmetic coding* stage, the *pre-coder* further exploits redundancies of the quantized wavelet coefficients in a 3-stage process of partitioning, aggregation and conditional coding.

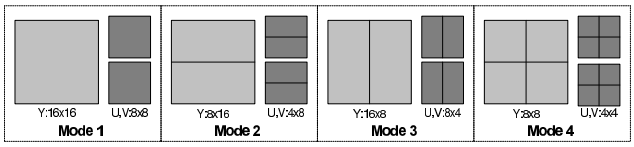


Figure 2. Illustration of the four macroblock partition modes used for temporal prediction

3 Motion-Compensated Prediction

3.1 Motion Model

The motion model used in our approach is very similar to that of the H.26L TML8 design [6]. Basically, it relies on a simple model of block displacements with variable block sizes. Given a partition of a frame into macroblocks *(MB)* of size 16×16 pels, each macroblock can be further subdivided into smaller blocks, where each sub-block has its own displacement vector. Our model supports 4 different partition modes, as shown in Figure 2.

Each macroblock may use a different reference picture out of a long-term frame memory. In addition to the predictive modes represented by the 4 different MB partition modes in Fig. 2, we allow for an additional macroblock-based intra coding mode in predicted inter frames. This local intra mode is realized by computing DC components for each 8×8 sub-block and each spectral component (Y,U,V) in a macroblock and by embedding the DC-corrected sub-blocks into the residual error frame in a way, which is further described in Section 3.4 below.

3.2 Motion Estimation and Mode Decision

Block motion estimation involves an exhaustive search over all integer pel positions within a pre-defined search window around the motion vector predictor, which is obtained from previously estimated sub-blocks in the same way as in TML-8 [6]. In a number of subsequent steps, the best integer pel motion vector is refined to the final $\frac{1}{4}$-pel accuracy by searching in a 3×3 sub-pel window around the refined candidate vector. All search positions are evaluated by using a Lagrangian cost function $J = D + \lambda R$, which involves a rate term R and a distortion term D coupled by

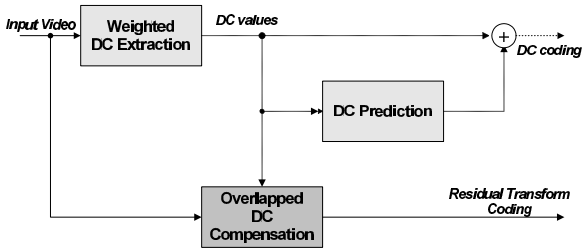


Figure 3. Block diagram of local intra coding mode

a Lagrangian multiplier λ. For all fractional-pel displacements, distortion D is estimated in the transform domain by using the Walsh-Hadamard transform, while the rate of the motion vector candidates is estimated by using a fixed, pre-calculated table. This search process involves each of the 4 different macroblock partition modes and each reference frame. Finally, the cost of the overall best predictive mode decision is compared against the cost of the local intra mode decision and the macroblock mode with minimum Lagrangian cost is chosen.

3.3 Overlapped Block Motion Compensation

The prediction error signal of the luminance signal is formed by using the weighted sum of the differences between all 16×16 overlapping blocks from the current frame and their related overlapping blocks with displaced locations in the reference frame, which have been estimated in the BME stage for the corresponding core blocks of size 8×8. As a weighting function w, we used the 'raised cosine' window function given by

$$w(n,m) = w_n \cdot w_m, \quad w_n = \frac{1}{2}\left[1 - \cos\frac{2\pi n}{N}\right] \qquad (1)$$

for all $(n,m) \in [0,N] \times [0,N]$. In our presented approach, we have chosen $N = 16$ ($N = 8$) for the luminance (chrominance, resp.) in Eq. (1) resulting in a block of 16×16 (8×8) pixel support centered over a "core" block of size 8×8 (4×4) pels for the luminance (chrominance, resp.).

For the texture interpolation of sub-pel positions, we used the same filters as specified in TML-8 [6].

3.4 Macroblock-Based Intra Coding Mode

Macroblock-based intra coding involves three steps as depicted in Fig. 3: computation of weighted DCs, removal of the DC components in an overlapping framework and predictive coding of the DC values. More precisely, for each macroblock, we first compute the weighted DC value dc_i for a 16×16 overlapping block related to the correspond-

ing 8×8 core block:

$$dc_i = \frac{\sum_{m,n=1}^{16} w(n,m) s_i(n,m)}{\sum_{m,n=1}^{16} w(n,m)}, \qquad (2)$$

where $s_i(n,m)$ denotes a sample of the i-th overlapping block (Y: $i = 1,\ldots,4$; U: $i = 5$; V: $i = 6$). In a second step, the weighted sum of the differences between the overlapping blocks of the corresponding core intra blocks and its related DC values is formed. Finally, before entering the pre-coding stage, each DC component is predicted by using spatially neighboring DC components.

4 Wavelet Transform

In wavelet-based image compression, the Daubechies 9/7-tap biorthogonal wavelet with compact support [9] is the most popular choice. Our proposed coding scheme, however, utilizes a class of biorthogonal wavelet bases associated with infinite impulse response (IIR) filters, which was recently constructed by Petukhov [10]. His approach relies on the construction of a dual pair of rational solutions of the matrix equation

$$\mathbf{M}(z)\tilde{\mathbf{M}}^T(z^{-1}) = 2\mathbf{I}, \qquad (3)$$

where $\mathbf{I}$ is the identity matrix, and

$$\mathbf{M}(z) = \begin{pmatrix} h(z) & h(-z) \\ g(z) & g(-z) \end{pmatrix}, \tilde{\mathbf{M}}(z) = \begin{pmatrix} \tilde{h}(z) & \tilde{h}(-z) \\ \tilde{g}(z) & \tilde{g}(-z) \end{pmatrix}$$

are so-called *modulation matrices*.

Petukhov constructed a one-parametric family of filters h_a, g_a, $\tilde{h}_a$ and $\tilde{g}_a$[1] satisfying Eq. (3):

$$h_a(z) = \frac{1}{\sqrt{2}}(1+z), \qquad (4)$$

$$\tilde{h}_a(z) = \frac{c(z^{-1}+3+3z+z^2)(z^{-1}+b+z)}{(z^{-2}+a+z^2)}, \qquad (5)$$

$$g_a(z) = c(z^{-1}-3+3z-z^2)(-z^{-1}+b-z), \qquad (6)$$

$$\tilde{g}_a(z) = \frac{1}{\sqrt{2}}\frac{1-z^{-1}}{z^{-2}+a+z^2}, \qquad (7)$$

where $c = \frac{2+a}{4\sqrt{2}(2+b)}, b = \frac{4a-8}{6-a}, \quad |a| > 2, \quad a \neq 6.$

In order to adapt the choice of the wavelet basis to the nature and statistics of the different frame types related to intra and inter coding mode, we performed a numerical simulation on this one-parametric family of IIR filter banks yielding the optimal parameter choice of $a = 8$ for intra frame mode and $a = 25$ for inter frame mode in Eqs. (4)–(7). Note that the corresponding wavelet transforms can be efficiently realized with a composition of recursive filters [10].

[1] h_a and g_a denote low-pass and high-pass filters of the decomposition algorithm, respectively, while $\tilde{h}_a$ and $\tilde{g}_a$ denote the corresponding filters for reconstruction.

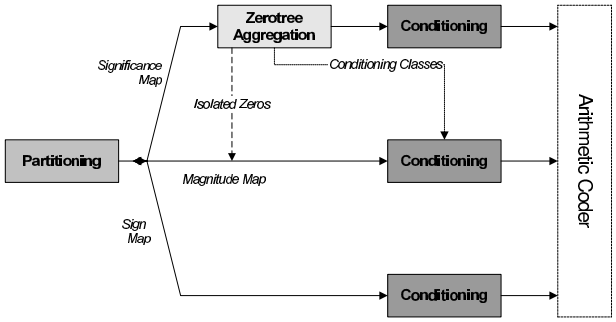


Figure 4. Schematic representation of the pre-coder used for encoding the quantized wavelet coefficients

5 Pre-Coding of Wavelet Coefficients

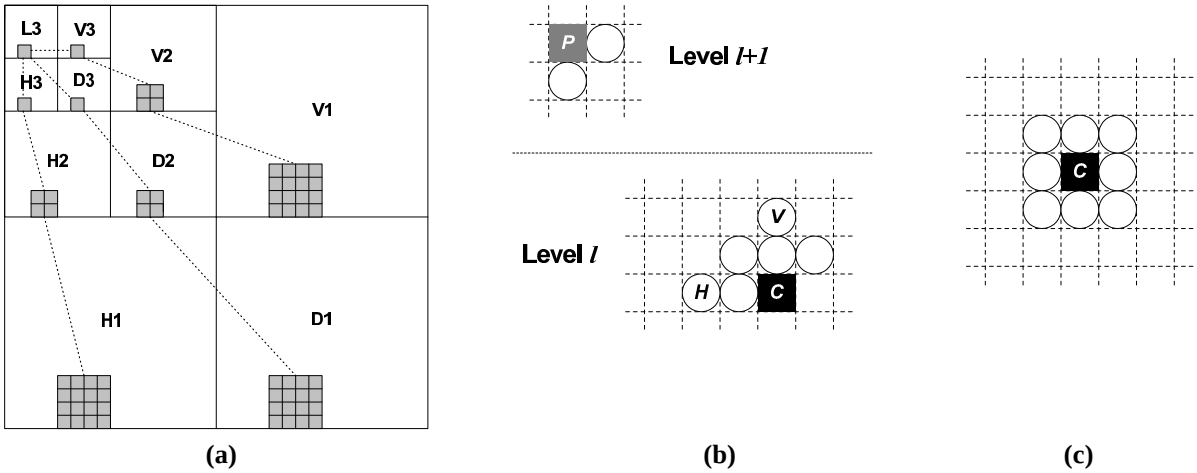
The strategy for encoding the quantized wavelet coefficients follows our conceptual ideas initially presented in [7] and later refined in [8]. Next, we give a brief review of the corresponding techniques of partitioning, aggregation and conditional coding (PACC). For more details, the readers are referred to [7, 8].

5.1 Partitioning

As shown in the block diagram of Fig. 4, an initial 'partitioning' stage divides each frame of quantized coefficients into three sub-sources: a significance map, indicating the position of significant coefficients, a magnitude map holding the absolute values of significant coefficients, and a sign map with the phase information of the wavelet coefficients. Note that all three sub-sources inherit the subband structure from the quantized wavelet decomposition as shown in Fig. 5 (a), so that there is another partition of each subsource according to the given subband structure.

5.2 Zerotree Aggregation

In a second stage, the pre-coder performs an 'aggregation' of insignificant coefficients using a quad-tree related data structure. These so-called *zerotrees* [7, 11] connect insignificant coefficients sharing the same spatial location along a fixed orientation (vertical, horizontal or diagonal) of the multiresolution decomposition as shown in Fig. 5 (a). However, in our presented approach, we do not consider zero-tree roots in bands below the maximum decomposition level, $e.g.$ level 3 in Fig. 5 (a). In inter-frame mode, coding efficiency is further improved by connecting the zerotree root symbols of all three lowest high-frequency bands (V3, H3 and D3 in Fig. 5 (a)). In that case, we build a so-called 'integrated' zerotree root residing in the lowpass band ($e.g.$ L3 in Fig. 5 (a)), if the corresponding coefficient in the lowpass band is also insignificant.



(a) **(b)** **(c)**

Figure 5. (a) Three-level wavelet decomposition with illustration of the zerotree structure; (b) Orientation dependent design of the two-level template (*white circles*) used for conditional coding of a binary event C of the significance map; V, H: additional elements used for vertical and horizontal oriented bands, respectively; P: parent of C on the next upper level $l + 1$; (c) 8-neighborhood of significance used for conditioning of a given magnitude C.

5.3 Conditional Coding

The final 'conditioning' part of the pre-coding stage supplies the elements of each source with a 'context', $i.e.$ an appropriate model for the actual coding process in the arithmetic coder. Fig. 5 (b) shows the prototype template used for conditioning of elements of the significance map. It consists of two parts. The first part involves a causal neighborhood of the actual coding event C depending on the scale and orientation of a given band. Except for the lowest frequency bands, the second part of the template uses additional information of the next upper level (lower resolution) represented by the neighbors of the parent P of C, thus allowing a 'prediction' of the non-causal neighborhood of C.

Coding of the lowest frequency band depends on the intra/inter decision. In intra mode, the lowpass band contains mostly non-zero coefficients, so that there is no need for coding a significance map. For P-frames, however, the significance of a coefficient in the lowpass band is coded by using the four-element kernel of our prototype template (Fig. 5 (b)), which is extended by the related entry of the significance map belonging to the previous P-frame.

The processing of subbands is performed band-wise in the order from lowest to highest frequency bands and the partitioned data of each band is processed such that the significance information is coded (and decoded) first. This allows the construction of special conditioning categories for the coding of magnitudes by using the local significance information. Thus, the actual conditioning of magnitudes is performed by classifying magnitudes of significant coefficients according to the local variance estimated by the significance of their 8-neighborhood (see Fig. 5 (c)). For the conditional coding of sign maps, we are using a context built of two preceding signs with respect to the orientation of a given band [8].

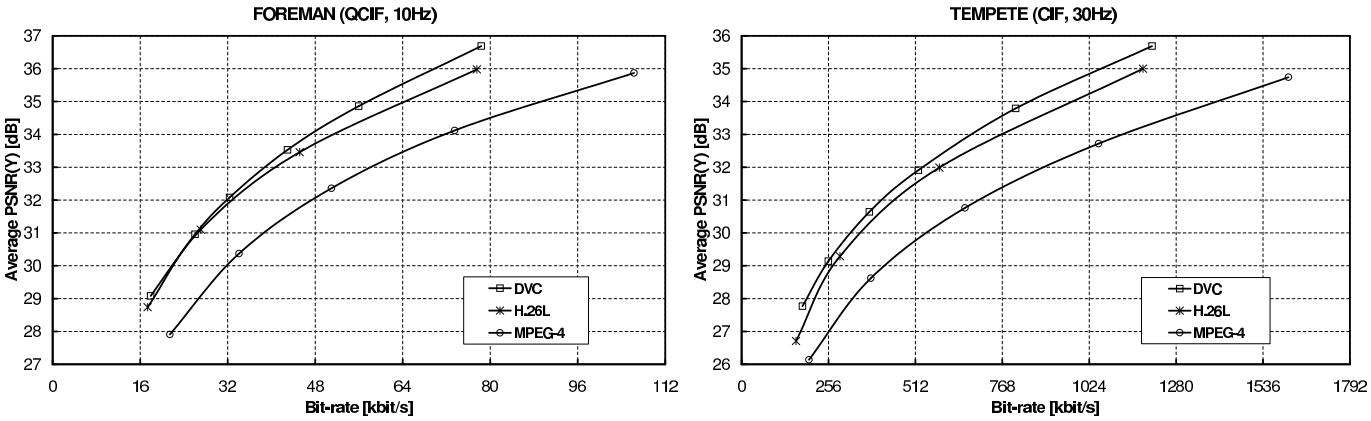
For coding the lowpass band of an intra frame, our proposed scheme uses a DPCM-like approach with a spatially adaptive predictor and a backward driven classification of the prediction error using a six-state model.

6 Binarization and Adaptive Binary Arithmetic Coding

All symbols generated by the pre-coder are encoded using an adaptive binary arithmetic coder, where non-binary symbols like magnitudes of coefficients or motion vector components are first mapped to a sequence of binary symbols by means of the unary code tree. Each element of the resulting 'intermediate' codeword given by this so-called *binarization* will then be encoded in the subsequent process of binary arithmetic coding.

At the beginning of the overall encoding process, the probability models associated with all different context models are initialized with a pre-computed start distribution. For each symbol to encode the frequency count of the related binary decision is updated, thus providing a new probability estimate for the next coding decision. However, when the total number of occurrences of symbols related to a given model exceeds a pre-defined threshold, the frequency counts will be scaled down. This periodical rescaling exponentially weighs down past observations and helps to adapt to non-stationarities of the given source.

For intra and inter frame coding separate models are used. Consecutive P-frames as well as consecutive motion vector fields are encoded using the updated related models of the previous P-frame and motion vector field, respectively. The binary arithmetic coding engine used in our presented approach is a straightforward implementation similar to that given in [12].



(a) **(b)**

Figure 6. Rate-distortion curves for a performance comparison of the proposed DVC coder, H.26L and MPEG-4.

7 Experimental Results

7.1 Test Conditions

To illustrate the effectiveness of our proposed video coding scheme, we used for comparison an optimized MPEG-4 coder [3] and the H.26L TML-8 coder [6]. The MPEG-4 coder follows a rate-distortion (R-D) optimized encoding strategy by using a Lagrangian cost function, and it generates bit-streams compliant with MPEG-4, Version 2, operating in *Advanced Simple Profile* (ASP) [4]. Note that this encoder provides PSNR gains in the range from 1–2 dB, when compared to the MoMuSys MPEG-4 reference encoder (VM18) [3]. For our experiments, we used the following encoding options of this improved MPEG-4 coder: quarter-pel accurate motion compensation and global motion compensation were enabled. For motion estimation, a full search with search range of ± 32 pels was performed, and for quantization the MPEG-2 quantization matrix was used. In addition, the recommended deblocking/deringing filter was applied in a post-processing stage.

For the H.26L TML-8 and the DVC coder, the following common settings were chosen: quarter-pel accurate motion compensation was used for QCIF resolution, and eighth-pel accurate motion compensation for sequences in CIF resolution; a full search within a search window of ± 32 pels around the motion vector predictor was performed, and we used five reference frames. For a fair comparison, TML-8 was evaluated using the low-complexity R-D motion estimation and mode decision and the advanced entropy coding mode of context-based adaptive binary arithmetic coding (CABAC) [6, 13].

Coding experiments were performed by using a whole test set of QCIF and CIF sequences. Only the first frame of each sequence was encoded as an intra picture; all subsequent frames were encoded as P-pictures. For each run of a sequence, a set of quantization parameters according to the different picture types (I or P) was fixed.

7.2 Test Results

Figure 6 shows the rate-distortion performance of our proposed DVC scheme relative to MPEG-4 ASP and H.26L TML-8 for the test sequences *Foreman* in QCIF resolution and *Tempete* in CIF resolution. The curves show the average PSNR of the luminance component as the arithmetic mean of the corresponding PSNR values for each frame versus bit-rate averaged over the whole sequence.

As can be seen from the graphs in Fig. 6, the DVC scheme provides significant PSNR gains of 2.0–2.5 dB in comparison to MPEG-4 ASP. At lower bit-rates, the R-D performance of H.26L TML-8 and DVC looks quite similar for the *Foreman* sequence , while for higher rates DVC provides PSNR gains of up to 0.75 dB relative to H.26L for that particular sequence. However, comparing the visual quality of reconstructed video for the DVC and the H.26L coder at those bit-rates that correspond to the same objective quality, an improvement of subjective quality in favor of our coder can be observed.

For an assessment of the visual quality Fig. 7 shows a sample reconstruction of the *Foreman* sequence coded at 32 kbit/sec for all three competing coders. Both DVC and H.26L clearly outperform MPEG-4, but it is also quite visible that the H.26L reconstruction still suffers from some blockiness, especially in the area of the helmet and the person's face. In contrast to that, the DVC reconstruction seems to be more natural and appealing, although some slight ringing artifacts can be observed.

For the *Tempete* sequence, DVC achieved PSNR gains of 0.5–0.8 dB relative to H.26L (see Fig. 6 (b)). This corresponds to bit-rate savings of about 10–15% for the same objective quality. In most cases, however, the subjective quality of the DVC coded video appeared to be superior to that of the H.26L coder at R-D points corresponding to the same PSNR value. This kind of results has been observed for all other sequences tested.

(a) **(b)** **(c)**

Figure 7. Reconstructed frame no. 47 of *Foreman* coded at 32 kbit/s: (a) MPEG-4, (b) H.26L, (c) DVC.

8 Conclusions and Future Research

The coding strategy of DVC has proven to be very efficient; PSNR gains of up to 2.5 dB relative to a highly optimized MPEG-4 coder have been achieved. Furthermore, it has been shown that our coder is able to compete with the best video coding technology currently available[2] both in terms of coding efficiency and computational complexity. However, it should be noted that there is still room for further performance improvements of our presented scheme. For instance, it is well known, that conventional block motion estimation is far from being optimal in an OBMC-based framework [14]. Additionally, the performance of our zerotree-based wavelet coder can be enhanced by using a R-D-based cost function for a joint optimization of the quantization and the zerotree-based encoding process. In summary, we expect another significant gain by exploiting the full potential of encoder optimizations inherently present in our DVC design. This will be the topic of our future research.

References

[1] T. Wiegand, X. Zhang, and B. Girod, "Long-Term Memory Motion-Compensated Prediction", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, No. 1, pp. 70–84, 1999.

[2] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-Distortion Optimized Mode Selection for Very Low Bit Rate Video Coding and the Emerging H.263 Standard", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 6, No. 2 pp. 182–190, 1996.

[3] H. Schwarz and T. Wiegand, "Lagrangian Coder Control and Comparison of MPEG-4 and H.26L", 4th International ITG Conf. on Source and Channel Coding, *ITG-Fachbericht 170*, pp. 301–308, 2002.

[2] To the best of our knowledge, the current test model of H.26L represents that kind of technology.

[4] ISO/IEC JTC1, "Coding of Audio-Visual Objects - Part2: Visual", ISO/IEC 14496-2 (MPEG-4 Visual, Version 1), April 1999; Amendment 1 (Version 2), February 2000.

[5] ITU-T, "Video Coding for Low Bitrate Communication", ITU-T Recommendation H.263; Version 1, Nov. 1995; Version 2, Jan. 1998.

[6] G. Bjontegaard, T. Wiegand, "H.26L Test Model Long Term Number 8 (TML8)", ITU-T SG 16 Doc. VCEG-N10, Oct. 2001.

[7] D. Marpe and H. L. Cycon, "Efficient Pre-Coding Techniques for Wavelet-Based Image Compression", *Proc. Picture Coding Symp. 1997*, pp. 45–50, 1997.

[8] D. Marpe and H. L. Cycon, "Very Low Bit-Rate Video Coding Using Wavelet-Based Techniques", *IEEE Trans. on Circ. and Syst. for Video Technology*, Vol. 9, No. 1 pp. 85–94, 1999.

[9] A. Cohen, I. Daubechies, J.-C. Feauveau, "Biorthogonal Bases of Compactly Supported Wavelets", *Comm. on Pure Appl. Math.* 45, pp. 485–560, 1992.

[10] A. P. Petukhov, "Recursive Wavelets and Image Compression", *Proc. Int. Congress of Math. 1998.*

[11] A. Lewis and G. Knowles, "Image Compression Using the 2D Wavelet Transform", *IEEE Trans. on Image Processing*, Vol. 1, No. 2, pp. 244–250, 1992.

[12] I. Witten, R. Neal, and J. Cleary, "Arithmetic Coding for Data Compression", *Communications of the ACM*, Vol. 30, pp. 520–540, 1987.

[13] D. Marpe, G. Blättermann, G. Heising, T. Wiegand, "Video Compression Using Context-Based Adaptive Arithmetic Coding", *Proc. IEEE Int. Conf. on Image Proc. 2001*, pp. 558–561, 2001.

[14] M. T. Orchard and G. J. Sullivan, "Overlapped Block Motion Compensation: An Estimation-Theoretic Approach", *IEEE Trans. on Image Processing*, Vol. 3, No. 5, pp. 693–699, 1994.