Design of a Highly Efficient Wavelet-based Video Coding Scheme

Detlev Marpe*[a], Thomas Wiegand*[a] and Hans L. Cycon**[b]
[a]Heinrich-Hertz-Institute (HHI); [b]University of Applied Sciences (FHTW Berlin)

ABSTRACT

In this paper, we present a novel design of 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. In addition, we follow the rate-distortion optimizing strategy of using a Lagrangian cost function to discriminate between different decisions in the video encoding process. Finally, we demonstrate that context-based adaptive arithmetic coding is a key element for fast adaptation and high coding efficiency. The combination of overlapped block motion compensation and frame-based transform coding enables 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 2.0–3.5 dB PSNR.

Keywords: Video coding, wavelet-based coding, very low bit rate coding, H.26L, MPEG-4

1. INTRODUCTION

Multi-frame prediction[12] and variable block size motion compensation in a rate-distortion optimized motion estimation and mode selection process[11, 13] are powerful tools to improve the coding efficiency of today's video coding standards. In this paper, we present the design of a video coder (*DVC*), which demonstrates how most elements of the state-of-the-art in video coding as currently implemented in the test model long-term[2] (TML8) of the ITU-T H.26L standardization project can be successfully integrated in a blocking-artifact free video coding environment. In addition, we provide a solution for an efficient macroblock-based intra coding mode within a frame-based residual coding method, which is extremely beneficial for improving the subjective quality as well as the error robustness. We further demonstrate how to improve the efficiency of a wavelet-based residual coder by using appropriately designed entropy coding tools like those, which were introduced in some of our previous publications[7, 8]

In our experiments, we compared our proposed wavelet-based DVC coder against an improved MPEG-4 coder[11] and the H.26L TML8 coder.[2] The coding results for MPEG-4 were obtained by using all coding tools defined in the Advanced Simple Profile (ASP), while those for TML8 were obtained by using all 7 different block shapes, the instrument of low-complexity motion estimation/mode decision and context-based binary adaptive arithmetic coding (CABAC).[5, 6] All competing codecs were operated using a fixed frame rate, fixed quantization step sizes and a motion vector search range of ±32 pels. We obtained coding results for various sequences showing that our proposed video coding system yields a coding gain of 2.0–3.5 dB PSNR relative to MPEG-4 and 0.25–0.8 dB PSNR relative to H.26L . Correspondingly, the visual quality provided by the DVC coder compared to that of the block-based coding approach of MPEG-4 is significantly improved, especially at very low bit rates.

* [marpe,wiegand]@hhi.de; phone +49 30 31002-619/-617; fax +49 30 3927200; http://bs.hhi.de; Image Processing Department, Heinrich-Hertz-Institute (HHI),Einsteinufer 37, 10587 Berlin, Germany; ** hcycon@fhtw-berlin.de; phone +49 30 54699-363; fax +49 30 54699-329; http://www.fhtw-berlin.de; University of Applied Sciences (FHTW Berlin), Allee der Kosmonauten 20–22, 10315 Berlin, Germany

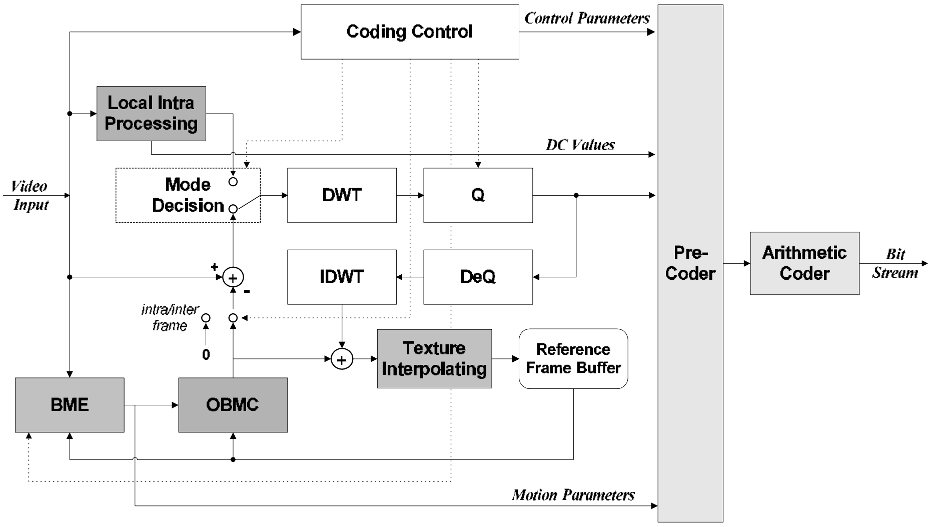


Figure 1: Block diagram of the proposed coding scheme

2. OVERVIEW OF THE DVC SCHEME

Fig. 1 shows a block diagram of the proposed DVC coder. As a hybrid system, it consists of a temporal predictive 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.

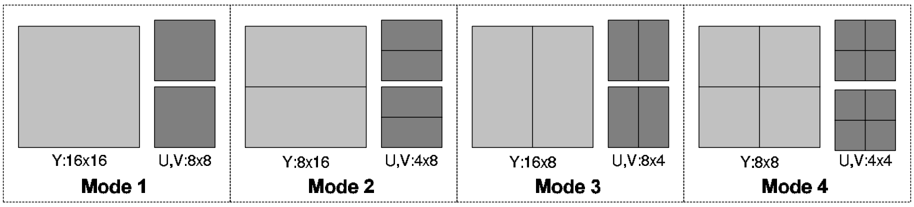


Figure 2: Illustration of the four macroblock partition modes used for temporal prediction

3. MOTION-COMPENSATED PREDICTION

3.1. Motion Model

As already mentioned above, the motion model we used is very similar to that of the H.26L TML8 design.[2] Basically, it relies on a simple model of block displacements with variable bock sizes. Given a partition of a frame into macroblocks (MB) of size 16 $\times$ 16 pels, each macroblock can be further sub-divided 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

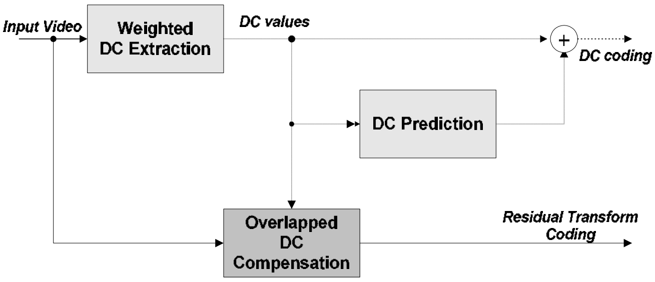


Figure 3: Block diagram of local intra coding mode

macroblock-based intra coding mode in P-frames. This local intra mode is realized by computing DC components for each 8×8 sub-block of all spectral components (Y,U,V) in a macroblock and by embedding the DC-corrected sub-blocks into the residual frame in a way, which is further described in section 3.4.

3.2. Motion Estimation and Mode Decision

Block motion estimation is performed by 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 TML8.[2] 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 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 macroblock partitions and each reference frame, and the cost of the overall best motion vector candidate(s) of all 4 macroblock modes is finally compared against the cost of the intra mode decision to choose the macroblock mode with minimum Lagrangian cost.

3.3. Motion Compensation

The prediction error luminance signal is formed by the weighted sum of the differences between all 16×16 (chrominance: 8×8) 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. As a weighting function w, we used the 'raised cosine', which for a support of $N \times N$ pels, is given by

$$w(n,m) = w_n \cdot w_m, \quad w_n = \frac{1}{2}\left[1 - \cos\frac{2\pi n}{N}\right] \quad \text{for } n = 0, \ldots, N. \tag{1}$$

In our presented approach, we choose $N = 16$ ($N = 8$) for the luminance (chrominance, resp.) in Eq. (1), which results in a 16×16 (8×8) pixel support centered over a "core" block of size 8×8 (4×4) pels for the luminance (chrominance, resp.). Note that for the texture interpolation of sub-pel positions, the same filters as specified in TML8[2] have been used.

3.4. Macroblock-Based Intra Coding Mode

Our macroblock-based intra coding mode 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 corresponding 8×8 core block:

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

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 (if available).

4. WAVELET TRANSFORM

In wavelet-based image compression, the so-called 9/7-wavelet with compact support[3] 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

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

where I is the identity matrix, and

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

are so-called ‘modulation matrices’.

Petukhov[10] constructed a one-parametric family of filters h_a, g_a, $\tilde{h}_a$ and $\tilde{g}_a$[*] satisfying Eq. (3)

$$\begin{aligned} h_a(z) &= \frac{1}{\sqrt{2}}(1+z), & (4)\\ \tilde{h}_a(z) &= \frac{(2+a)(z^{-1}+3+3z+z^2)(z^{-1}+b+z)}{4\sqrt{2}(2+b)(z^{-2}+a+z^2)}, & (5)\\ g_a(z) &= \frac{(2+a)(z^{-1}-3+3z-z^2)(-z^{-1}+b-z)}{4\sqrt{2}(2+b)}, & (6)\\ \tilde{g}_a(z) &= \frac{1}{\sqrt{2}}\frac{1-z^{-1}}{z^{-2}+a+z^2}, & (7) \end{aligned}$$

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

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

5. PRE-CODING OF WAVELET COEFFICIENTS

For encoding the quantized wavelet coefficients, we follow the conceptual ideas initially presented in[7] and later refined in.[8] Next, we give a brief review of the involved techniques. 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 sub-source according to the given subband structure.

[*] 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.

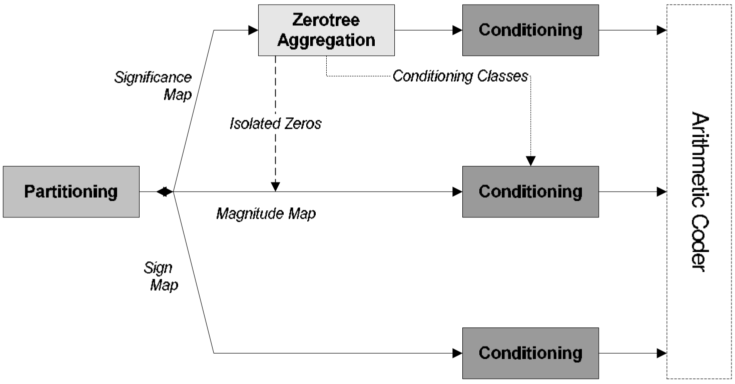


Figure 4: Schematic representation of the pre-coder used for encoding the quantized wavelet coefficients

5.2. 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*[4,7] connect insignificant coefficients, which share the same spatial location along a fixed orientation (vertical, horizontal or diagonal) of the multiresolution pyramid as shown in Fig. 5 (a). However, 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)). If the corresponding coefficient in the lowpass band (*e.g.* L3 in Fig. 5 (a)) is also insignificant, we build a so-called 'integrated' zerotree root, which resides in the lowpass band.

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. In the first part, it consists of a causal neighborhood of the actual coding event C, which depends on the scale and orientation of a given band. Except for the lowest frequency bands, 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. The processing of the lowest frequency band depends on the intra/inter decision. In intra mode, mostly non-zero coefficients are expected in the LL-band, so there is no need for coding a significance map. For P-frames, however, we indicate the significance of a coefficient in the lowpass band 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 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 (cf. 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 of the lowpass band related to an I-frame, the proposed scheme uses a DPCM-like procedure 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

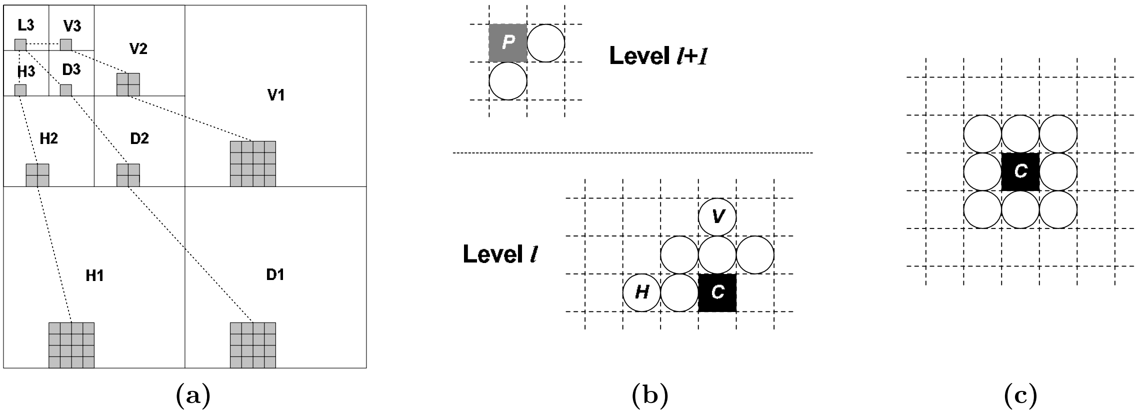


Figure 5. **(a)** Three level wavelet decomposition with illustration of the zerotree structure; **(b)** Orientation dependent design of the two-level template (*white circles*) used for conditional coding of a binary event C of the significance map; V, H: additional elements used for vertical and horizontal 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.

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 contexts 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 a source.

For intra and inter frame coding we use separate models. 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.[14]

7. EXPERIMENTAL RESULTS

7.1. Test Conditions

To illustrate the effectiveness of our proposed coding scheme, we used an improved MPEG-4 coder[11] and the H.26L TML8 coder[2] as reference systems. 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, Advanced Simple Profile (ASP).[1] Most remarkable is the fact that this encoder provides PSNR gains in the range from 1.0–3.0 dB, when compared to the MoMuSys reference encoder (VM17).[11] For our experiments, we used the following encoding options of the improved MPEG-4 reference coder:

- $\frac{1}{4}$-pel motion vector accuracy enabled
- Global motion compensation enabled
- Search range of ±32 pels
- 2 B-frames inserted ($IBBPBBP\ldots$)
- MPEG-2 quantization matrix used

For TML8 and our proposed DVC scheme, we have chosen the following common settings:

- $\frac{1}{4}$-pel motion vector accuracy
- Search range of ±32 pels around the motion vector predictor
- Five reference pictures were used for all sequences except for the 'News'-sequence, for which one additional reference frame was used (see discussion of results below)

TML8 was evaluated using low-complexity motion estimation and mode decision and CABAC entropy coding. Note that in contrast to TML8, for which 2 B-frames were inserted ($IBBPBBP\ldots$), DVC was operated using no B-frames.

Coding experiments were performed by using the test sequences 'Foreman' and 'News' both in QCIF resolution and with 100 frames at a frame rate of 10 Hz. Only the first frame was encoded as an I-frame; all subsequent frames were encoded as P-frames or B-frames. For each run of a sequence, a set of quantizer parameters according to the different frame types (I,P,B) was fixed.

7.2. Test Results

Figs. 6–7 show the average PSNR gains obtained by our proposed DVC scheme relative to MPEG-4 ASP and TML8 for the test sequences 'Foreman' and 'News', respectively. In comparison to MPEG-4 ASP, significant PSNR gains of 2.0–2.5 dB on the luminance component have been achieved for the 'Foreman'-sequence (cf. Fig. 6). Figure 8 shows a comparison of the visual quality for a sample reconstruction at 32 kbit/s, both for DVC and MPEG-4.

The results we obtained for DVC using the 'News' sequence show dramatic PSNR improvements of about 2.5–3.5 dB relative to MPEG-4. To demonstrate the ability of using some *a priori* knowledge about the scene content for this particular sequence, we checked in addition to the five most recent reference frames one additional reference frame 50 frames back in the past according to the repetition of parts of the scene content. By using the additional reference frame memory for this particular test case for both TML8 and DVC, we achieved an additional gain of about 1.5 dB PSNR on the average compared to the case where the reference frame buffer was restricted to the five most recent reference frames only.

However, in comparison to TML8, DVC achieved a much lower average PSNR gain of 0.25–0.8 dB for these two test sequences.

8. CONCLUSIONS AND FUTURE RESEARCH

The coding strategy of DVC has proven to be very efficient. PSNR gains of up to 3.5 dB relative to an highly optimized MPEG-4 coder have been achieved. However, it should be noted that in contrast to the MPEG-4 coding system, no B-frames were used in the DVC scheme, although it can be expected that DVC will benefit from the usage of B-frames in the same manner as the MPEG-4 or TML8 coder, *i.e.* depending on the test material, additional significant PSNR improvements might be achievable. Another important point to note, when comparing the coding results of our proposed scheme to that of the highly R-D optimized MPEG-4 encoder used for our experiments, is the fact that up to now, we did not incorporate any kind of high-complexity R-D optimization method. We even did not optimize the motion estimation process with respect to the overlapped motion compensation, although it is well known, that conventional block motion estimation is far from being optimal in an OBMC framework.[9] Furthermore, we believe that the performance of our zerotree-based wavelet coder can be further improved by using a R-D cost function for a joint optimization of the quantizer and the zerotree-based encoder. Thus, we expect another significant gain by exploiting the full potential of encoder optimizations inherently present in our DVC design. This topic will be a subject of our future research.

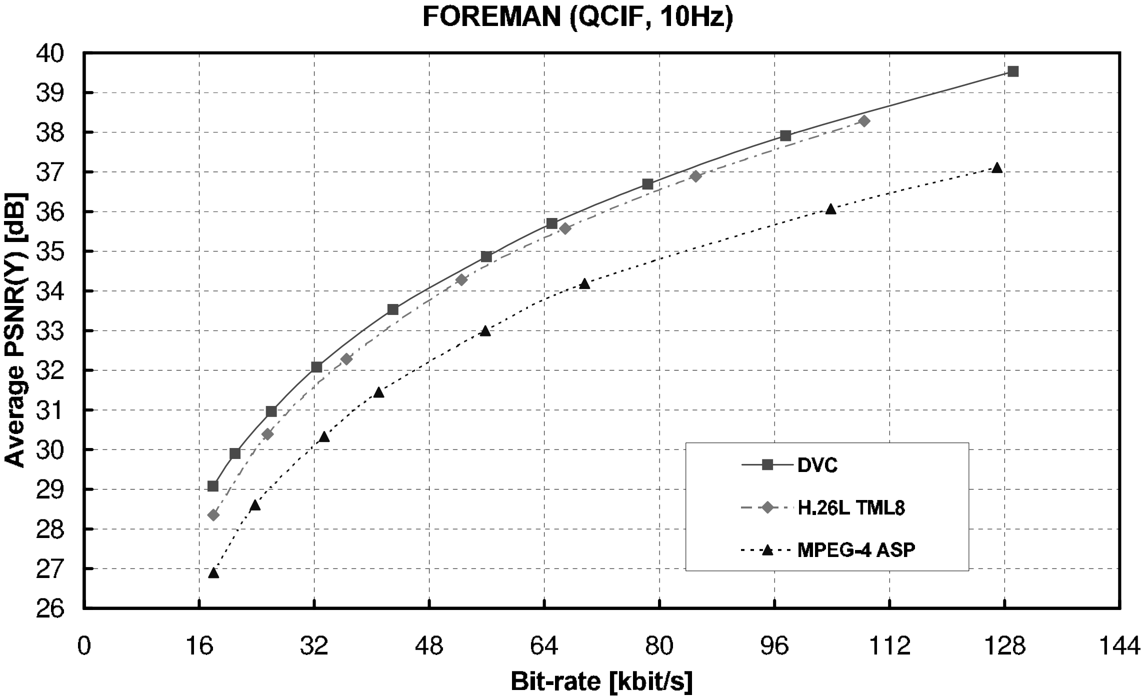


Figure 6: Average Y-PSNR against bit-rate using the QCIF test sequence 'Foreman' at a frame rate of 10 Hz

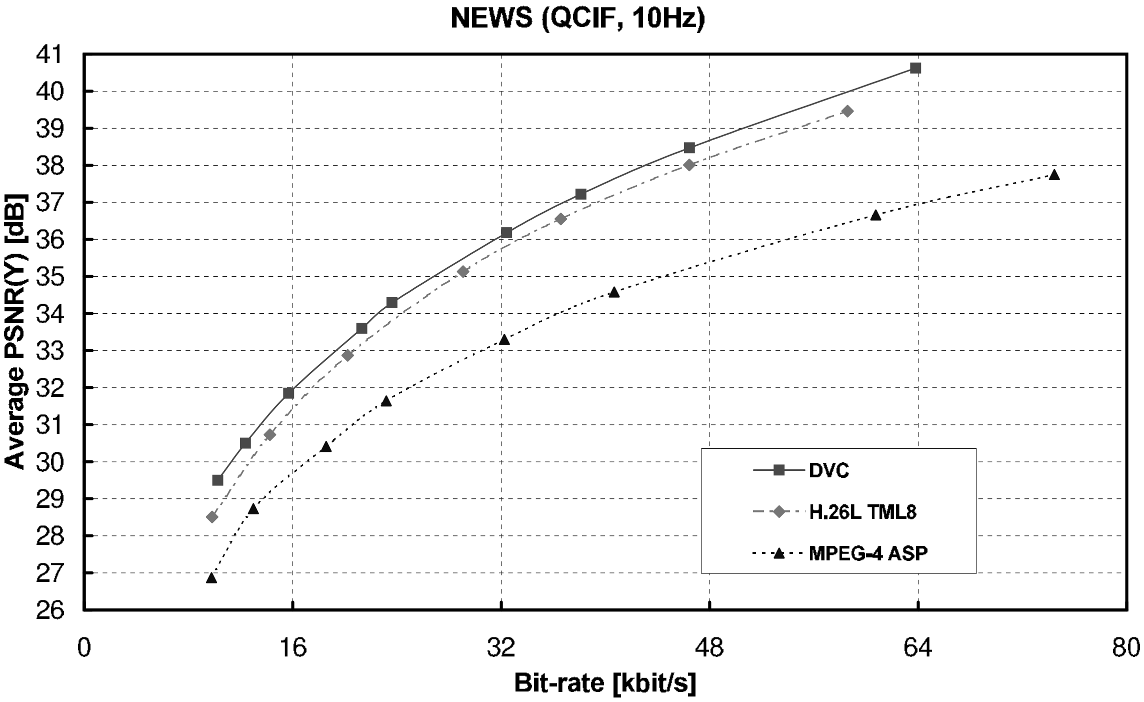
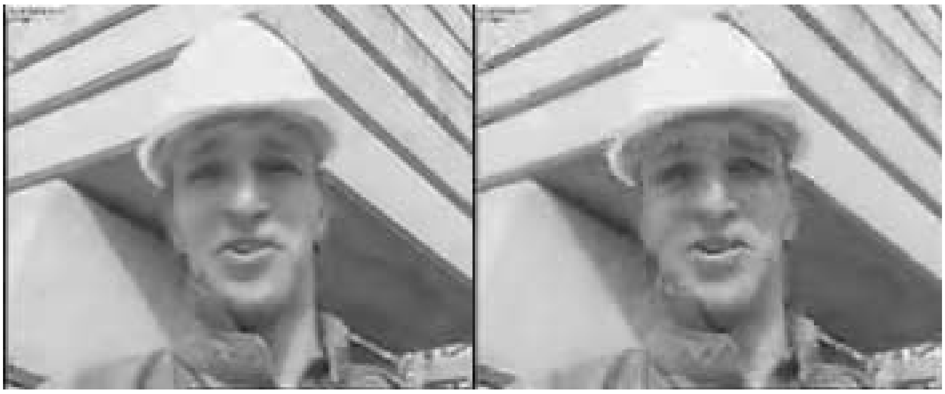


Figure 7: Average Y-PSNR against bit-rate using the QCIF test sequence 'News' at a frame rate of 10 Hz



(a) **(b)**

Figure 8. Comparison of subjective reconstruction quality: Frame no. 22 of 'Foreman' at 32 kbit/s. **(a)** DVC reconstruction **(b)** MPEG-4 reconstruction. Note that the MPEG-4 reconstruction has been obtained by using a de-blocking filter

REFERENCES

1. ISO/IEC JTC1SC29 14496-2 MPEG-4 Visual, Version 2.
2. G. Bjontegaard (ed.), "H.26L Test Model Long Term Number 8 (TML8)", ITU-T SG 16 Doc. VCEG-N10, Oct. 2001.
3. A. Cohen, I. Daubechies, J.-C. Feauveau, "Biorthogonal Bases of Compactly Supported Wavelets", *Comm. on Pure and Appl. Math.* **45**, pp. 485–560, 1992.
4. 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.
5. D. Marpe, G. Blättermann, T. Wiegand, "Adaptive Codes for H.26L", ITU-T SG 16 Doc. VCEG-L13, 2001.
6. 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.
7. D. Marpe and H. L. Cycon, "Efficient Pre-Coding Techniques for Wavelet-Based Image Compression", *Proceedings Picture Coding Symposium 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 Circuits and Systems for Video Technology*, Vol. 9, No. 1 pp. 85–94, 1999.
9. 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.
10. A. P. Petukhov, "Recursive Wavelets and Image Compression", *Proceedings International Congress of Mathematicians 1998.*
11. H. Schwarz and T. Wiegand, "An Improved MPEG-4 Coder Using Lagrangian Coder Control", ITU-T SG 16 Doc. VCEG-M49, 2001.
12. 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.
13. 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.
14. I. Witten, R. Neal, and J. Cleary, "Arithmetic Coding for Data Compression", *Communications of the ACM*, Vol. 30, pp. 520–540, 1987.