Robust Motion-Compensated Orthogonal Video Coding using EBCOT

Ousmane Barry, Du Liu, Stefan Richter, Markus Flierl
KTH Royal Institute of Technology
Stockholm, Sweden
{ousmane, dul, sgric, mflierl}@kth.se

Abstract

This paper proposes a rate-distortion control for motion-compensated orthogonal video coding schemes and evaluates its robustness to packet loss as faced in, e.g., IP networks. The robustness of standard hybrid video coding is extensively studied in the literature. In contrast, motion-compensated orthogonal subbands offer important advantages and new features for robust video transmission. In this work, we utilize so-called uni-directional motion-compensated orthogonal transforms in combination with entropy coding similar to EBCOT known from JPEG2000. The approach provides a flexible embedded structure and allows flexible rate-distortion optimization. Moreover, it may even permit separate encoding and rate control. The proposed rate-distortion control takes channel coding into account and obtains a preemptively protected representation. Our implementation is based on repetition codes, adapted to the channel condition, and improves the PSNR significantly. The optimization requires an estimate of the packet loss rate at the encoder and shows moderate sensitivity to estimation errors.

1. Introduction

This paper addresses the robustness of video coding with motion-compensated orthogonal subbands. In contrast to subband video coding, the widely used hybrid video coding sends a reference frame and then only the motion-compensated prediction error for a larger number of following frames. While this improves coding efficiency, it also introduces the risk that errors propagate when the reference frame is decoded incorrectly [1]. A number of error-resilience tools have been studied and became part of, e.g., H.264 [2, 3]. A different approach to cope with error propagation is avoiding the reference frame what motion-compensated subband transforms do [4, 5, 6]. Fig.1 depicts the robust video transmission system of this paper which is based on motion-compensated orthogonal transforms (MCOT) [1, 6], adaptive spatial wavelets [7], and EBCOT [8]. A similar system was found efficiently compressing previously [6]. Since subband video codecs operate in open loop fashion and the chosen transforms strictly maintain orthogonality, the rate can be controlled in the compressed domain. This simplifies the design of rate-distortion strategies which anticipate packet loss, a critical aspect of packet-switched networks [2]. We introduce a novel loss-aware rate-distortion optimization which takes a channel model and a channel code into account. It can be rerun on the bitstream, e.g. for each user at a base station of a cellular phone network.

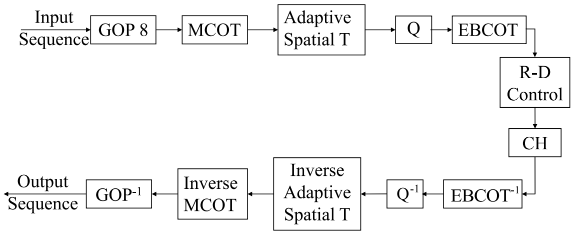

Figure 1. Video transmission system.

2. Robust Orthogonal Video using EBCOT

2.1. Unidirectional Motion-Compensated Orthogonal Transform

The UMCOT is a transform for successive pictures of an image sequence that maintains orthogonality while permitting motion compensation between pairs of pictures [6]. It only allows for unidirectional motion compensation and is a special case of MCOT. Orthogonality is an important principle in digital image and video processing. It matches the quantization cell shape due to scalar quantization of transform coefficients and it conserves the total energy of the signal. The UMCOT decomposes a pair of pictures into a temporal low band and high band. Our system operates with a GOP size of 8 frames. The block size of motion compensation is 16x16 with a displacement range of ±12. These parameters have been chosen after several tests (see

Section 3). From the UMCOT, we recover a temporal low band where the energy is accumulated, 7 temporal high bands and the motion vectors. The high bands are quantized and coded directly, whereas the temporal low band is further decomposed by an adaptive spatial transform for efficient coding [7]. The motion vectors are very important for the decoder. They are used to retrieve the scale counters which are required to perform the inverse spatial and temporal transform. A lossless transmission is required to obtain the appropriate inverse transforms. Hence, a Huffman code is used.

A special spatial wavelet decomposition has been developed for the class of MCOT [7]. This adaptive spatial decomposition exploits the spatial correlation within each temporal low band and maintains the orthogonal representation of the temporal transforms. It has to consider the scale factors that have been used during the temporal decomposition. Our system uses the Haar-like wavelet in [7] and two spatial decomposition stages, resulting in a high energy concentration in the final low band (see Section 3).

After the temporal and spatial decomposition, the subbands are quantized before entropy coding. For this purpose, we used the uniform scalar deadzone quantization in JPEG2000 [9]. The important feature is that it creates an embedded representation when coded as bit planes, i.e., the indices can be dequantized with the available bit planes after truncations. The step size has been set to one and the rate is controlled by the EBCOT algorithm.

2.2. Entropy Coding

EBCOT is used for entropy coding [8]. The important features for our system are the quality scalability, the rate-distortion optimization, and the EBCOT approach to create an embedded bit stream. In our system, EBCOT encodes the four subbands (LL, HL, LH and HH) from the spatial transform of the temporal low band and seven temporal high bands from UMCOT for each GOP. The temporal high bands are coded according to HH rules because motion may be in any direction and no preferred edge direction is expected. We apply pass coding to each of the code blocks and then arithmetic coding is used to obtain the embedded block bit-streams. The optimal block size for pass coding is chosen to be 16x16 (see Section 3). The whole code block is coded without discarding any of the bit-planes, thus the bit-streams contain the complete information before rate-distortion control. Therefore, we only need to encode each video once, which will save a lot of processing time. EBCOT allows a perfect reconstruction of the quantized coefficients from any input image.

The arithmetic coder needs the probabilities of the symbols. A very simple adaptive estimation is used where we count the zeros and ones in each context while encoding. The current counts of zeros n_0 and ones n_1 of the respective context are used to estimate the probability as

$$\Pr(1) = \frac{n_1}{n_1 + n_0} \tag{1}$$

whenever a bit is encoded. We initialize n_0 and n_1 such that we start with equal probabilities. Since the to-be-coded bit is not contained in the estimation, this method can be used in the decoder too. As the context is derived from previously decoded bits, it is no longer possible to decode the entire codeword at once. Hence, the arithmetic decoder has to return single bits.

As a pre-processing step for rate control, we calculate a table of rate distortion points after each pass and for each code block. Instead of decoding at the receiver side, we can simply look up the distortion in the table which leads to a significant increase in simulation speed. We verified that this approach gives the same distortion values as when calculated in the image domain.

2.3. Bitstream

Once codewords and their length are determined (see Section 2.5), they have to be structured for transmission. To reduce overhead, we transmit the codewords of each subband as a superblock. Each superblock starts with the type number of the subband. The next elements are the block number within the subband of the first and the last block in the considered packet, respectively. We are thereby able to split too large packets according to the maximum transfer unit (MTU) of the physical channel. On the other hand, this structure also allows us to put several superblocks in one channel packet, e.g. IP packet. The following number of bit-planes is required to setup the dequantizer appropriately if bit-planes are truncated. Every superblock ends in the ordered codewords prefixed with their length. We determine the maximum number of bits to represent the start and end as well as length and send it to the receiver. The latter accounts for the changing properties of the GOPs. However, the block numbers in fact depend on the video size. It would be more efficient to negotiate, for example, a Huffman code during the setup of the channel. The number of bit-planes is also predisposed to entropy coding since its range changes depending on the subband.

2.4. Channel

Our channel model is a packet erasure channel. With a probability p, transmitted packets are lost in the network due to congestion, link failures, etc. and do not reach the decoder. To improve the reliability, we can send the packets multiple times. These so-called repetition codes have proved their value for hybrid video coding [2]. However, the additional data either increases the data rate or the quality has to be reduced. We can optimize this tradeoff since any prefix of the arithmetic codewords of the EBCOT code-blocks can be decoded.

The channel is implemented by dropping packets while obtaining EBCOT blocks from the bitstream. While we determine the distortion at the receiver, the received packets are checked for inconsistent duplications and its bits are checked against the transmitted data for bit errors.

2.5. Rate Distortion Optimization

Our goal is to minimize the distortion D at the decoder subject to a maximum rate R_{max}. Unlike in [8], we further need an error protection and a loss-aware rate control due to the lossy channel.

We use up to $L = 6$ quality layers (i.e. repetitions); each consists of prefixes of the codewords and can thus be decoded on its own. The layers are labeled by the quality they can achieve if received. The *best quality layer* has the longest codewords. The longer the codewords are, the more bitplanes are reconstructed. Hence, the dequantization becomes better and the reconstruction of the subbands improves. On the other hand, the bits of the *worst quality layer* are implicitly repeated several times since the codewords of all other layers begin with these bits. Hence, this layer is best protected and the received quality is determined by the worst quality layer, if any layer is received.

To obtain optimal rate distortion performance, we only use truncation points on the convex hull. They are selected by minimizing the expectation $\mathrm{E}\{D\}$ subject to $R \leq R_{max}$ given the probability of packet loss p. Possibly received shorter prefixes convey no additional information and are discarded since we assume no biterrors. Thus, the distortion is only determined by the length of the longest received codeword. The best quality layer is received with probability $1-p$ and inferior layers only come into action if the l better ones are lost, i.e., with probability $(1-p)p^l$. All repetitions for a code-block are lost with probability p^L. In this case, the block will contribute the distortion of the zero-length codeword. We have therefore a truncated geometric distribution and $\mathrm{E}\{D\}$ is calculated as

$$\begin{aligned}\mathrm{E}\{D\} &= \sum_i \mathrm{E}\{D_i\} \\ &= \sum_i \left[p^L D_i(0) + \sum_{l=0}^{L-1} (1-p)p^l D_i(n_{i,l}) \right], \quad (2)\end{aligned}$$

where we assume additive distortion as in [8] and $D_i(n)$ is the distortion if block i is truncated at n.

Minimizing (2) subject to the constraint

$$R_{max} \geq R = \sum_i \sum_{l=0}^{L-1} R_i(n_{i,l}) \quad (3)$$

is done similar to [8]. We further exploit the concept of *cell problems* in [10]. Hence, the minimization problem simplifies to minimizing $[w_l D_i(n_{i,l}) + \lambda R_i(n_{i,l})]$ for each code-block and repetition where we introduce the weight $w_l = (1-p)p^l$ to account for the different loss probabilities of the repetitions.

Since we have a finite set of possible truncations $m_k, k = 1, 2, 3, ...$, we can simply test them. We update $n_{i,l} = m_k$ whenever

$$\left[w_l D_i(n_{i,l}^{\lambda}) + \lambda R_i(n_{i,l}^{\lambda}) \right] > \left[w_l D_i(m_k) + \lambda R_i(m_k) \right]$$

and after rearranging, we obtain

$$\frac{D_i(n_{i,l}^{\lambda}) - D_i(m_k)}{R_i(m_k) - R_i(n_{i,l}^{\lambda})} > \frac{\lambda}{w_l}. \quad (4)$$

Similar to [8], we can define a subset of feasible truncation points $\tilde{m}_k, k = 1, 2, 3, ...$ with strictly decreasing slopes

$$S_i(\tilde{m}_k) = \frac{D_i(\tilde{m}_{k-1}) - D_i(\tilde{m}_k)}{R_i(\tilde{m}_k) - R_i(\tilde{m}_{k-1})} \quad (5)$$

and select $n_{i,l}^{\lambda} = \max(\tilde{m}_k | S_i(\tilde{m}_k) > \lambda / w_l)$. Note that the slopes do not depend on the weights and thus are determined only once irrespective of the number of repetitions by calculating the convex hull of the rate-distortion points $D_i(n)$.

The value of λ is obtained in a bisection approach. We start with an interval from zero to maximum slope of all blocks. This interval is split in the middle. We calculate the rate for that λ and thereby determine which half is split again. After several iterations, the smallest value of λ, where $R(\lambda) \leq R_{max}$, is used to calculate the truncation points.

Noting the rather low complexity and given the small summary information of rates and slopes, the algorithm can be rerun at any network node. Thereby it is possible to account for increasing and changing loss probabilities. An example would be a base station which can derive optimal representations for multiple users and their time-varying channels from a single high quality feed.

3. Results

3.1. Energy Concentration

One of MCOT's features is the high energy concentration in the temporal low band. Let E_x, E_y and E_{LL} be respectively the total energy for an input GOP, the energy of the temporal low band, and the energy of the final low band after the spatial decomposition. Let $ratio_1 = \frac{E_y}{E_x}$, $ratio_2 = \frac{E_{LL}}{E_x}$, and $ratio_3 = \frac{E_{LL}}{E_y}$. Tables 1 and 2 show the energy concentration in the temporal and spatial low bands. The results are an average over 200 frames i.e. of 25 GOPs.

We see clearly that we have a better energy concentration for temporal and spatial transforms with a motion block size

	8x8 Motion block	16x16 Motion block
	-8:8 displacement	-12:12 displacement
$ratio_1$	98.45%	99.09%
$ratio_2$	96.11%	97.41%
$ratio_3$	97.63%	98.31%

Table 1. Energy concentration for $Foreman$

	8x8 Motion block	16x16 Motion block
	-8:8 displacement	-12:12 displacement
$ratio_1$	99.20%	99.54%
$ratio_2$	97.46%	98.32%
$ratio_3$	98.25%	98.77%

Table 2. Energy concentration for $Mother\&Daughter$

	8x8 Motion block	16x16 Motion block
	-8:8 displacement	-12:12 displacement
$Foreman$	8.33%	1.19%
$Mom\&Dau$	5.90%	1.15%

Table 3. Contribution of motion vectors for a rate of 1bpp

of 16x16 and a displacement range of -12:12. The slight difference between the two sequences is due to the fact that there is less motion in the $Mother\&Daughter$ sequence.

3.2. Huffman Coding of Motion Vectors

Table 3 depicts the results obtained for Huffman coding of motion vectors. The Huffman code has been established from the 5 training videos $Suzie$, $Salesman$, $Carphone$, $Grandma$, and $Claire$. We used 288 frames of each video sequence. As for the energy concentration, the results are an average over 25 GOPs of the two test sequences. The percentage represents the contribution of the code of the motion vectors for a rate of 1bpp.

The implementation with a motion block size of 16x16 offers a lower bit rate for the motion vectors. For energy concentration and coding efficiency, the motion block size of 16x16 outperforms the motion block size of 8x8.

3.3. Rate Distortion Performance

Figs. 2 and 3 depict the rate distortion performance for the luminance signals of the two test sequences. The motion vector bit streams are taken into account in the rate estimation. The results show the evolution and the improvements brought to our system. Our first prototype was implemented with a motion block (m-blk) and pass coding block (p-blk) size of 8x8. In the end, we arrived at a motion block size and pass coding block size of 16x16 with an improved rate control (RC). The difference between the first and the final system is quite significant and amounts to about 4dB. This is due to the improvement in energy compaction and compression of motion vectors by increasing the block size.

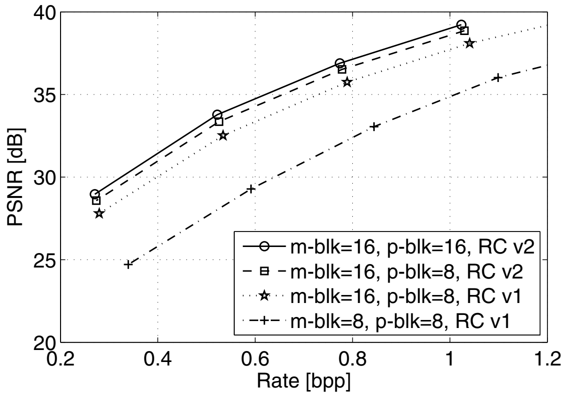

Figure 2. PSNR over the rate for the luminance signal of the QCIF sequence $Foreman$ at 30fps with 64 frames.

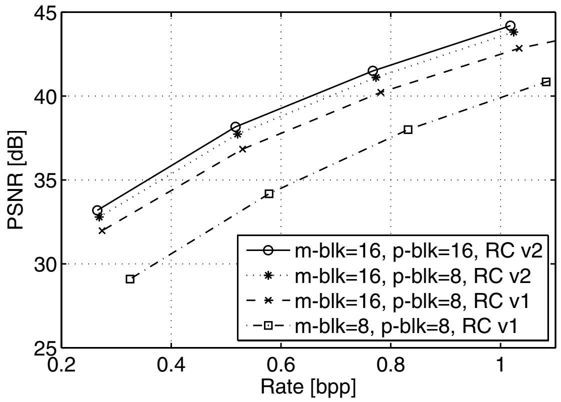

Figure 3. PSNR over the rate for the luminance signal of the QCIF sequence $Mother\&Daughter$ at 30fps with 64 frames.

The choice of a proper pass coding block size is explained as follows. We started with a block size of 4x4, because the size of the smallest LL subband 44x36 is only a multiple of 4. Since there are too few coefficients in one code block, the probability for arithmetic coding is estimated column by column. However, this block size does not exploit much redundancy between the coefficients. Thus, we expect a better performance with a larger block size while arithmetic coding remains the same for simplicity. We found that the PSNR decreases for a block size of 32x32. The main reason is that one code block covers most part of the smallest LL band, which is not optimal for coding. Therefore, we use the block size of 8x8 and 16x16 for our implementation. As a result, the optimal pass coding block size is 16x16 for the two videos, as shown in Figs. 2 and 3 .

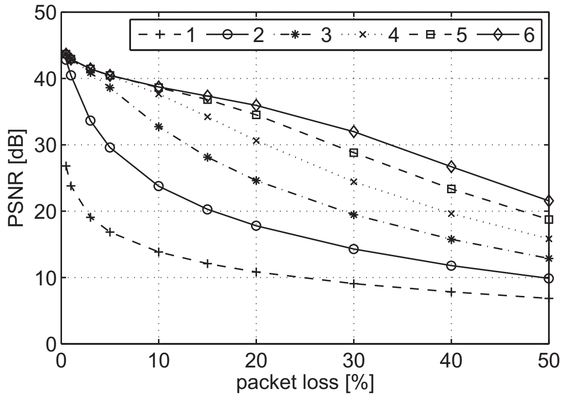

Figure 4. PSNR over packet loss probability for 64 frames for *Foreman* at 2bpp and maximum number of copies limited to between 1 and 6.

3.4. Robustness

The robustness of the proposed video transmission system is assessed using the luminance signal of the first 64 frames of the QCIF sequences *Foreman* and *Mother&Daughter*. We assume that the motion information is highly protected and will always be recovered at the decoder without error.

Fig. 4 shows that our rate control correctly selects the number of repetitions since the curves converge at low loss probability. In fact, the truncation points for some inferior quality layers are set to zero below a certain loss rate and the remaining ones take the same values as if fewer copies were allowed. However, the computational complexity increases and we used at most six copies. Fig. 4 additionally demonstrates the need for protection on the channel since the unprotected transmission is severely degraded even at very low loss rates. This is caused by the spatial low band which introduces considerable distortion if lost. Consequently, it receives the most and largest copies if one allows the rate control to (partially) repeat the codewords. The PSNR improves by about 25dB at 20% packet loss despite the fact that all curves use 2bpp for the entire transmission including all copies, overhead, and motion vectors.

Figs. 5 and 6 depict the distortion introduced by packet losses in the channel for *Mother&Daughter* and *Foreman*, respectively. The loss due to channel coding depends on the quality of the uncoded case. For example, at 5% packet loss, the reduction is smallest with 2.88dB for *Foreman* at 0.5bpp and increases to 6.21dB for *Mother&Daughter* at 2bpp which had the worst and best uncoded results in our experiments, respectively. The convergence of the curves at high loss rate becomes obvious noting that all have to end up in the same point for 100% loss. The low PSNR of this case should improve if lost low band coefficients are not set to zero (i.e. black picture) but the affected pixels are set to gray. With increasing loss rate, it becomes more and more likely that all copies are lost. But the distortion contributed in this case is equal for all bitrates. Hence, the curves approach each other as the influence of lost low bands increases.

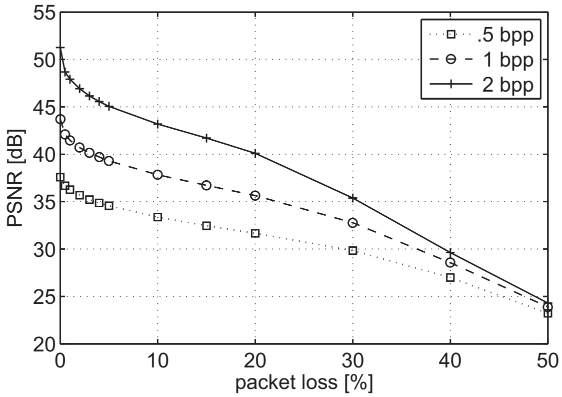

Figure 5. PSNR over packet loss probability for the *Mother&Daughter* sequence.

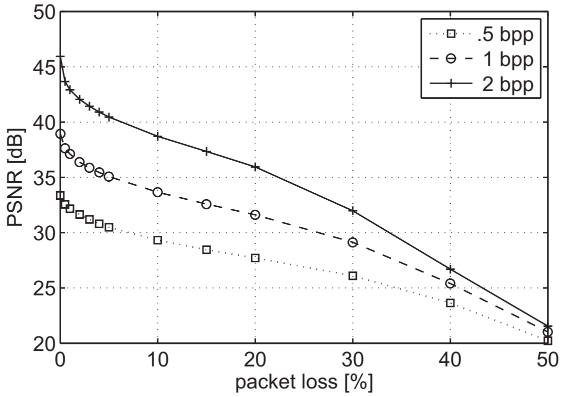

Figure 6. PSNR over packet loss probability for the *Foreman* sequence.

The PSNR of both the best and the worst quality layer is plotted in Fig. 7 along with the result of our algorithm. The quality of both layers converges as the loss rate increases. This is reasonable since it becomes more likely that the decoder has to rely on a repetition which should be of higher quality. The worst quality layer is better than the overall result at 50% packet loss. This is again the severe impact of the low band. It is equally likely lost and taken from the worst quality in this case according to (2). The ripple at the inferior layer is caused when repetitions come into action which were formerly zero.

Fig. 8 evaluates the sensitivity of the rate distortion optimization to errors in the estimation of the packet loss rate. The difference between the resulting PSNR with correct estimation and the PSNR where the estimated loss rate was

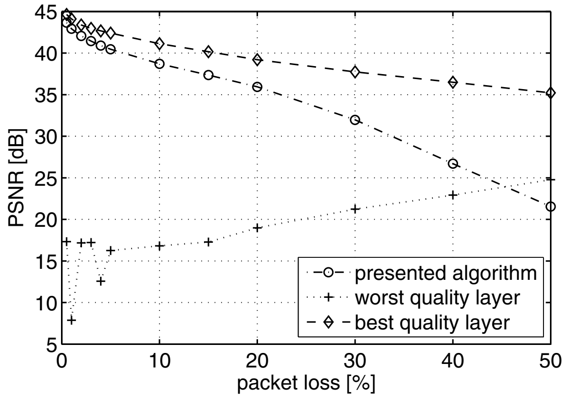

Figure 7. PSNR over packet loss probability investigated for different quality layers and algorithms.

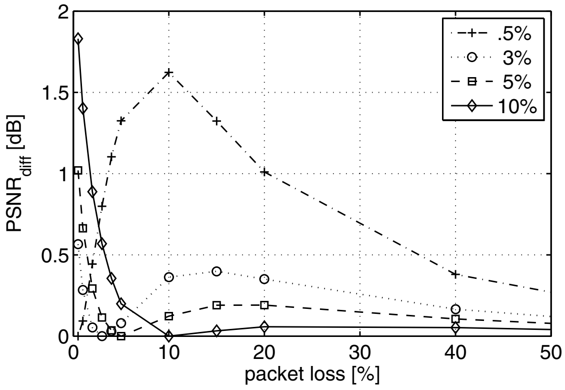

Figure 8. Loss in PSNR over actual packet loss when estimated packet loss is 0.5, 3, 5 or 10%.

fixed is plotted. The algorithm seems to be sensitive to the relative error since the error is especially large if either the assumed or the actual packet loss is low. This relates to the initial drop in Figs. 5 and 6 since the rate distortion optimization trades unnecessarily much quality for robustness. However, the error stays below 0.1dB for an interval of about ±1% around the actual value. The decay of the PSNR difference towards high loss is once more caused by the low band and the common value at 100% loss.

4. Conclusion

Motion-compensated orthogonal transforms combined with adaptive spatial wavelets provide high energy concentration while maintaining orthogonality for any motion field. In our system, the motion vectors are efficiently compressed using Huffman codes. For the coefficients of the orthogonal subbands, EBCOT achieves high compression performance without sacrificing flexibility. The resulting video coding scheme is able to accomplish both an efficient rate-distortion performance and a robust video representation. The latter is achieved by a novel loss-aware rate distortion optimization and basic channel coding with repetition codes. A single encoding is sufficient to obtain any desired quality and protection up to a maximum as determined by quantization. A simple system for packing the resulting coefficients was implemented and tested. Future work includes a more complete implementation of EBCOT and the use of more advanced channel codes.

References

[1] M. Flierl and B. Girod, "Half-pel accurate motion-compensated orthogonal video transforms," in *Proc. of the IEEE Data Compression Conference, 2007. DCC '07*, Mar. 2007, pp. 13 –22. 1

[2] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645 – 656, July 2003. 1, 2

[3] T. Stockhammer, M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657 – 673, July 2003. 1

[4] A. Secker and D. Taubman, "Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting," in *Proc. of the IEEE International Conference on Image Processing*, vol. 2, Oct. 2001, pp. 1029 –1032. 1

[5] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 2001, pp. 1793 –1796. 1

[6] M. Flierl and B. Girod, "A motion-compensated orthogonal transform with energy-concentration constraint," in *Proc. of the IEEE International Workshop on Multimedia Signal Processing*, Oct. 2006, pp. 391 –394. 1

[7] M. Flierl, "Adaptive spatial wavelets for motion-compensated orthogonal video transforms," in *Proc. of the IEEE International Conference on Image Processing (ICIP)*, Nov. 2009, pp. 1045 –1048. 1, 2

[8] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158 –1170, July 2000. 1, 2, 3

[9] M. Marcellin, M. Gormish, A. Bilgin, and M. Boliek, "An overview of JPEG-2000," in *Proc. of the IEEE Data Compression Conference*, Mar. 2000, pp. 523 –541. 2

[10] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Oper. Res.*, vol. 11, pp. 399 – 417, 1963. 3