PREDICTOR SELECTION USING QUANTIZATION INTERVALS IN JPEG2000-BASED SCALABLE INTERACTIVE VIDEO (JSIV)

Aous Thabit Naman and David Taubman

School of Electrical Engineering and Telecommunications,
University of New South Wales, Sydney, Australia.

ABSTRACT

The authors have recently introduced the JPEG2000-Based Scalable Interactive Video (JSIV) paradigm. JSIV relies on JPEG2000 format for providing scalability and accessibility, and on motion compensation and conditional replenishment to exploit temporal redundancy. JSIV can provide considerably better interactivity compared to existing video streaming practices, and can adapt immediately to interactive changes in client interests, such as forward or backward playback and zooming into individual frames. This work extends our previous work by providing server and client policies that can exploit the client's knowledge about the quantization intervals of received samples in selecting a favorable predictor in dyadic hierarchical B-frame arrangement that does not employ motion compensation. We also demonstrate the flexibility of the JSIV paradigm by showing an improved client policy working with a non-improved server policy without any negative impact on reconstructed video.

***Index Terms*—** Teleconferencing, video signal processing, image coding, image communication

1. INTRODUCTION

Existing video compression techniques, such as MPEG1 through MPEG4, have focused on improving reconstructed video quality for a given data rate while paying little attention to interactivity. This limited interactivity and the diverse client needs motivated research in scalable video coding; the aims is to produce an embedded bitstream that can provide better interactivity options, serve a wider variety of client needs, and adapt to changing network conditions by gracefully degrading reconstructed video quality. Research in this field have produced some promising results [1,2] and recently a scalable video coding (SVC) extension to H.264/AVC [3] has been approved to provide enhanced scalability options. Even with scalable video coding, interactivity is still restricted by the design of the encoder. As an example, if a remote client is particularly interested in just one frame, the server must send sufficient bits to reconstruct a larger number of frames in order to invert the motion compensated transform used during compression.

The authors have recently introduced a new paradigm [4–6] that gives considerably better interactivity and coined the term JPEG2000-Based Scalable Interactive Video (JSIV) [6] for it. JSIV relies on JPEG2000 format to independently store individual video frames and to provide for spatial scalability and accessibility. To reduce temporal redundancy, JSIV exploits motion compensation and conditional replenishment.

Central to the JSIV paradigm is the concept of loosely-coupled client and server policies; neither the server nor the client should drive the video streaming interaction, but rather the server dynamically selects and sends the pieces that, it thinks, best serve the client needs and the client makes most of the pieces of information it has. This makes it possible for the server and client policies to evolve independently with little or no negative effect on video reconstruction when a newer policy is introduced at one side but not the other. We demonstrate one such example in this work. An interested reader can refer to our earlier work [4–6] for other advantages of JSIV.

In this work we employ the dyadic hierarchical B-frame arrangement similar to the one proposed in the SVC extension of AVC, and we choose not to employ motion compensation in our prediction model[1] since many applications, such as surveillance still benefits from such a model. In this context, there are two main *contributions* of this work. Firstly, we show that the client's knowledge about quantization intervals of received quantized samples can be utilized in selecting a more favorable predictor among possibly more than one. Secondly, we propose a way of encoding and delivering side information needed by the client; this information helps the client make reasonable decisions in case of ambiguity.

Many researchers have realized the limited interactivity provided by existing techniques and have devised different approaches to address it [7–9]. Cheung and Ortega [7] propose flexible video decoding that provides forward and backward playback by utilizing distributed video coding techniques. Devaux et al. [8] investigate a problem similar to JSIV in some aspects; however, they stop short of investigating the flexibility that such a paradigm can provide. Mavlankar et al [9] propose a way of dynamically providing pan, tilt, and zoom features in video playback to different clients with varying regions of interest by breaking a high resolution video into tiles and streaming them simultaneously employing H.264 compression. JSIV adapts some of its concepts from a work about interactive browsing of 3D scenes [10] and shares, in some sense, the idea of decode-time selection of a predictor with that of [7].

The remainder of the paper is structured as follows. Section 2 reveals the client policies. Section 3 discusses the server policies and proposes a way of encoding and sending side information. Section 4 provides experimental results. Section 5 states our conclusions.

2. CLIENT POLICY

Frames are arranged in a dyadic hierarchical structure with temporal decimation levels $T_0, T_1, \ldots T_K$, as shown in Figure 1 for the case of $K = 3$. Each frame belongs to one or more temporal decimation level T_k depending on its position. For each code-block, C_n^β, of each frame, f_n, the client receives zero or more quality layers, q_n^β. Consequently, the de-quantized sub-band coefficients, $\tilde{C}_n^\beta(q_n^\beta)$, of that code-block have an associated distortion given by $\tilde{D}_n^\beta = \|\tilde{C}_n^\beta - C_n^\beta\|^2$. Code-blocks of a frame at temporal level T_K do not employ

---

[1] We have shown the efficacy of JSIV with motion compensation in our earlier work [4–6].

prediction; their coefficients are either received from the server or set to zero. A code-block, C_n^β, in a frame at temporal level T_k but not T_{k+1}, and in the absence of motion compensation, can use code-blocks C_l^β and C_r^β as prediction reference code-blocks, where $l = n - 2^k$ and $r = n + 2^k$.

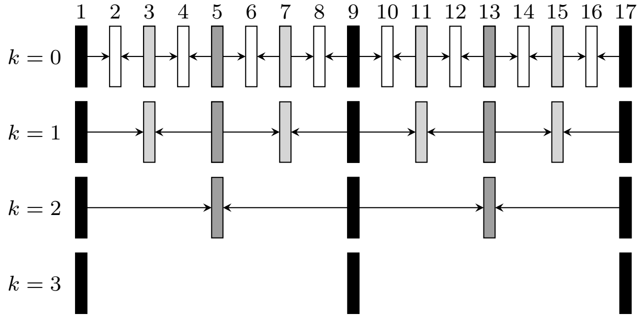

Fig. 1. Two groups of pictures in a dyadic hierarchical structure. Arrows show prediction directions and the numbers at the top are frame indices.

The standard JSIV (JSIV-S) policy uses one predictor given by $C_{\to n}^{\mathrm{S},\beta} = \frac{1}{2}(C_l^\beta + C_r^\beta)$. In general, JSIV employs a precinct-based policy since JPEG2000 code-stream is arranged in packets; a packet is one quality layer, q_n^π, from one precinct, P_n^π, where a precinct is composed of one or more code-blocks. The distortion contribution of one precinct to the reconstructed full-resolution frame is given by

$$D_n^\pi = \sum_{\beta \ni C_n^\beta \in P_n^\pi} G_{b_\beta} \cdot D_n^\beta \tag{1}$$

where G_{b_β} is the energy gain of sub-band b to which C^β belongs. The number of quality layers in a code-block, q_n^β, is equal to the number of packets received and therefore equal to q_n^π.

For JSIV-S, the client also receives a quality layer threshold, $\bar{q}_n^\pi$, that helps the client decide when to use prediction. This way, the client policy becomes

$$P_n^\pi = \begin{cases} P_{\to n}^{\mathrm{S},\pi}, & \text{when } q_n^\pi < \bar{q}_n^\pi \\ \tilde{P}_n^\pi(q_n^\pi), & \text{otherwise} \end{cases} \tag{2}$$

The threshold is the smallest quality layer at which $\tilde{D}_n^\pi(q_n^\pi) < D_{\to n}^{\mathrm{M},\pi}$ where $D_{\to n}^{\mathrm{M},\pi}$, shown in Figure 2, is the distortion associated with $P_{\to n}^{\mathrm{S},\pi}$ when full-quality reference code-blocks are used. In Section 3.1, we have more to say on how this policy attempts to achieve

$$D_n^\pi = \min\left\{\tilde{D}_n^\pi(q_n^\pi), D_{\to n}^{\mathrm{S},\pi}\right\} \tag{3}$$

One main contribution of this work is in how the client use its knowledge about the quantization intervals of received samples in selecting a favorable predictor for the case of JSIV with selective prediction (JSIV-SP). JPEG2000 employs a fractional bit-plane code-block encoder such that different sub-band coefficients are quantized with different quantization intervals [11]. These intervals can be easily calculated from the headers of the JPEG2000 file being processed and the state of the decoder during samples decoding. We denote the quantization interval at location $\mathbf{p}$ of code-block C_n^β by $\mathcal{I}_n^\beta[\mathbf{p}]$ and its centroid by $v_n^\beta[\mathbf{p}]$, where $\mathbf{p} = [p_1, p_2]$ is a two-dimensional index.

To select the most favorable predictor for sub-band coefficient $C_n^\beta[\mathbf{p}]$ with the aid of the received quantized coefficient $\tilde{C}_n^\beta[\mathbf{p}]$, we count the number of predicted coefficients in and around $\mathbf{p}$ that are within their quantization intervals, $\mathcal{I}_n^\beta[\mathbf{p}]$; that is, for $C_l^\beta[\mathbf{p}]$, we find

$$c_l^\beta[\mathbf{p}] = \left|\left\{C_l^\beta[\mathbf{i}] \mid C_l^\beta[\mathbf{i}] \in \mathcal{I}_n^\beta[\mathbf{i}], \mathbf{i} \in \mathcal{R}_\mathbf{p}\right\}\right| \tag{4}$$

where $|\cdot|$ is the number of elements in the set and $\mathcal{R}_\mathbf{p}$ is a region around $\mathbf{p}$. Special care should be taken not to cross the code-block boundaries. To preserve count locality, it is advisable to use a small $\mathcal{R}_\mathbf{p}$ region. Experimental results show that making $\mathcal{R}_\mathbf{p}$ larger than 3×3 pixels centered around $\mathbf{p}$ does not produce measurable improvement and therefore we use a 3×3 region in this work.

Thus, the sub-band coefficients in the selectively-predicted code-block, $C_{\to n}^\beta(q_n^\beta)$, that utilizes both the received samples and prediction is given by

$$C_{\to n}^\beta(q_n^\beta) = \begin{cases} C_l^\beta, & C_l^\beta \in \mathcal{I}_n^\beta \text{ and } C_r^\beta \in \mathcal{I}_n^\beta \text{ and } c_l^\beta > c_r^\beta \\ C_r^\beta, & C_l^\beta \in \mathcal{I}_n^\beta \text{ and } C_r^\beta \in \mathcal{I}_n^\beta \text{ and } c_l^\beta < c_r^\beta \\ C_{\to n}^{\mathrm{S},\beta}, & C_l^\beta \in \mathcal{I}_n^\beta \text{ and } C_r^\beta \in \mathcal{I}_n^\beta \text{ and } c_l^\beta = c_r^\beta \\ C_l^\beta, & C_l^\beta \in \mathcal{I}_n^\beta \text{ and } C_r^\beta \notin \mathcal{I}_n^\beta \\ C_r^\beta, & C_l^\beta \notin \mathcal{I}_n^\beta \text{ and } C_r^\beta \in \mathcal{I}_n^\beta \\ v_n^\beta, & \text{otherwise} \end{cases} \tag{5}$$

where the $\mathbf{p}$ index is dropped from all the terms for shortness, and the use of q_n^β in $C_{\to n}^\beta(q_n^\beta)$ helps to identify JSIV-SP and remind the reader that selectively-predicted samples depend on the number of received quality layers, q_n^β. When $q_n^\beta = 0$, the quantization interval, $\mathcal{I}_n^\beta$, is considered an unbound interval and therefore (5) falls back to $C_{\to n}^\beta(0) = C_{\to n}^{\mathrm{S},\beta}$.

For JSIV-SP, the quality layer threshold, $\bar{q}_n^\pi$, shown in Figure 2, is the smallest quality layer at which $\tilde{D}_n^\pi(q_n^\pi) < D_{\to n}^{\mathrm{M},\pi}(q_n^\pi)$, where $D_{\to n}^{\mathrm{M},\pi}(q_n^\pi)$ is the distortion associated with $P_{\to n}^\pi(q_n^\pi)$ when full-quality reference frames are used. Using this threshold, the client policy can be summarized as:

$$P_n^\pi = \begin{cases} P_{\to n}^\pi(q_n^\pi), & \text{when } q_n^\pi < \bar{q}_n^\pi \\ \tilde{P}_n^\pi(q_n^\pi), & \text{otherwise} \end{cases} \tag{6}$$

In Section 3.1, we have more to say on how this policy attempts to achieve

$$D_n^\pi = \min\left\{\tilde{D}_n^\pi(q_n^\pi), D_{\to n}^\pi(q_n^\pi)\right\} \tag{7}$$

In both JSIV-S and its improved version, JSIV-SP, the quality layer threshold, $\bar{q}_n^\pi$, is the quality layer beyond which it better to use the received samples since prediction produces higher distortion even when full-quality reference precincts are used. The use of loosely-coupled policies requires the use of media properties that are always true regardless of the client cache content, and $\bar{q}_n^\pi$ is one such property.

3. SERVER POLICY

Frames are divided into groups of pictures, $\mathcal{G}_s$, each with $2^K + 1$ frames. Frames at the temporal level T_K are part of two consecutive groups; frame 9 in Figure 1, for example, is part of $\mathcal{G}_0$ and $\mathcal{G}_1$. Frames in each $\mathcal{G}_s$ are jointly optimized subject to a transmission budget of $L_{\max}$; as such, frames in T_K have two chances of receiving data. Using an additive model, the distortion in one $\mathcal{G}_s$ is given by

$$D = \sum_{n \in \mathcal{G}_s} \sum_{\pi \in f_n} D_n^\pi \tag{8}$$

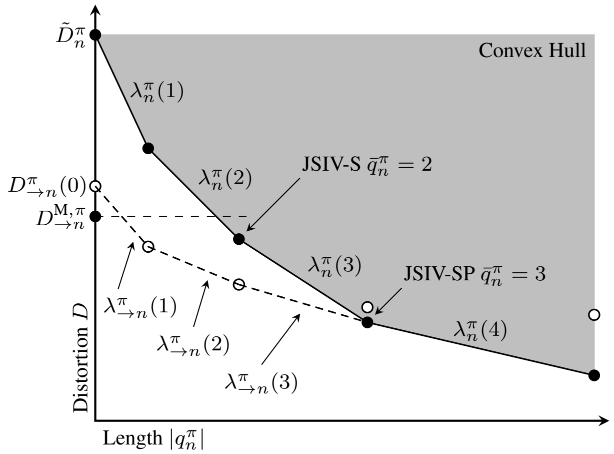

Fig. 2. A typical convex rate-distortion curve for a precinct when $D_{\to n}^\pi(0) < \tilde{D}_n^\pi(0)$. Filled dots are for received samples, $\tilde{D}_n^\pi(q_n^\pi)$, and empty dots are for selectively-predicted samples, $D_{\to n}^\pi(q_n^\pi)$. Each dot represent one quality layer. The figure also shows the quality layer threshold $\bar{q}_n^\pi$ and $D_{\to n}^\pi(0)$ which is equal to $D_{\to n}^{\mathrm{S},\pi}$.

The minimization of D subject to length constraint $L_{\max}$ can be (approximately) recast as the minimization of a family of Lagrangian functionals,

$$J_\lambda = \sum_{n \in \mathcal{G}_s} \sum_{\pi \in f_n} \left(D_n^\pi + \lambda \cdot |q_n^\pi| \right) \tag{9}$$

where $|q_n^\beta|$ denotes the number of bytes in q_n^β quality layers of C_n^β. The Lagrangian parameter λ is adjusted until the solution which minimizes J_λ satisfies the length constraint.

Without motion compensation, (9) can be broken into a set of independent functionals indexed by π. Even with this simplification, direct minimization is difficult because of the dependencies among the different precincts indexed by the same π as a result of prediction. To deal with these difficulty, we employ two independent passes [4,5]. In pass one, $\mathcal{P}^1$, the contribution of each precinct, P_n^π, to the overall distortion is evaluated and a contribution weight, θ_n^π, is assigned. In pass two, $\mathcal{P}^2$, q_n^π's are determined for each P_n^π as will be explained shortly. The decisions made during $\mathcal{P}^2$ can be used in the next $\mathcal{P}^1$. This iterative process can be repeated until convergence is achieved and no more changes in decisions occur. It can be shown that this iterative approach converges albeit to possibly a local minimum. For the case of $K = 3$, shown in Figure 1, little or no quality improvement is achieved beyond the second iteration.

We turn our attention to determining q_n^π. Figure 2 shows a typical distortion-length curve for a precinct P_n^π which is guaranteed to be convex by construction [11]. Each circle in the figure represents one quality layer. We define the distortion-length slope associated with each q_n^π by $\lambda_n^\pi(q) = (\tilde{D}_n^\pi(q-1) - \tilde{D}_n^\pi(q))/(|q| - |q-1|)$.

For frames at temporal level T_0, distortion-length slopes are readily available. For frames at temporal levels $k < K$, the existence of prediction sources reduces the effective distortion to $D_{\to n}^\pi(q_n^\pi)$ or $D_{\to n}^{\mathrm{S},\pi}$ when $q_n^\pi < \bar{q}_n^\pi$ depending on the server policy . This creates a new distortion-length convex hull that can be easily computed by utilizing an algorithm for convex hull and slope computation similar to those proposed in [11]. Once these new slopes, $\lambda_{\to n}^\pi$, are established; q_n^π can be optimally determined from

$$q_n^\pi = \max\left\{ q \mid (1 + \theta_n^\pi) \cdot \lambda_n^\pi(q) > \lambda \right\} \tag{10}$$

where θ_n^π are the aforementioned contribution weights. This is exactly the server policy for JSIV-SP. For JSIV-S, an added restriction is necessary because the client will uses prediction if it receives $q_n^\pi < \bar{q}_n^\beta$ quality layers; therefore, the server should send $\bar{q}_n^\beta$ or more quality layers for the data to be used. This way the server policy for JSIV-S is

$$q_n^\pi = \max_{q \geq \bar{q}_n^\beta} \left\{ q \mid (1 + \theta_n^\pi) \cdot \lambda_n^\pi(q) > \lambda \right\} \tag{11}$$

Experimental results reveal that this restriction have almost no impact on performance compared to the unrealistic case of a server applying no restrictions and a client somehow correctly predicting the server's intentions.

In real implementation there is no need to calculate exact distortions; it is sufficient to use pre-calculated distortion and approximate the missing distortions at serve-time [5]. The server keeps tables of certain variables such as $\tilde{D}_n^\pi(q_n^\pi)$ and $\bar{q}_n^\pi$ among others. For JSIV-SP, the server also keeps tables of $D_{\to n}^\pi(q_n^\pi)$. To account for imperfections in reference precincts, the server linearly scales $D_{\to n}^\pi(q_n^\pi)$.

3.1. Coding and delivery of quality layer thresholds

Another main contribution of this work is in encoding and transmitting quality layer thresholds, $\bar{q}_n^\pi$, by storing them as an additional component in each JPEG2000-compressed frame. This allows the use of JPIP without any modifications for sending this information to the client. Since there is one threshold per precinct, this thresholds component is heavily sub-sampled. We use the same code-block dimensions, decomposition levels, and number of quality layers, Q, as those of the original frame. Only one sub-band is needed to store all the thresholds for each resolution level; in practice, we use the HL band leaving the LH and HH bands zeros.

The thresholds are encoded using the JPEG2000 block encoder directly. We set the number of missing most significant bits (MSBs) to zero, the number of coding passes to $3 \cdot Q - 2$, and encode $\bar{q}_n^\pi$ as $2^{\mathrm{MSB} - \bar{q}_n^\pi}$. The resulting code-stream is made such that each quality layer stores one whole bit-plane.

During data delivery, we send enough quality layers (or bit-planes) from the thresholds component such that the client is able to deduce $\bar{q}_n^\pi$ for all the precincts in the original frame that has $q_n^\pi \geq \bar{q}_n^\pi$; this is to avoid using prediction for these precincts in the client. We send no less than one layer of this component because this layer tells the client where not to use prediction, $\bar{q}_n^\pi = 0$ wherever a bit is set.

Looking back at JSIV-S, one can see that the threshold delivery policy works with (11) and (2) in attempting to achieve (3). A similar statement can be made for JSIV-SP. A JSIV-S server communicating with a JSIV-SP client should also work properly since the server sends quality layers only when $q \geq \bar{q}_n^\pi$ along with their thresholds. This information is interpreted correctly by the client policy, (6).

4. EXPERIMENTAL RESULTS

The results presented here are for two test sequences, "Speedway"[2] and "Professor". " Speedway" is a 193 frame sequence – it is actually 200 frames but the last 7 frames were dropped to make it more suitable for 3-level hierarchical B-frame arrangement – that has a resolution of 352 × 288 at 30 frames/s and a bit depth of 8 bits per sample. "Professor" is a 97 frame sequence that has a resolution of 3008 × 2000 captured at one frame every approximately three seconds at a bit depth of 8 bits per sample. Only the Y-component

---

[2] http://www.openjpeg.org/, OpenJPEG Library.

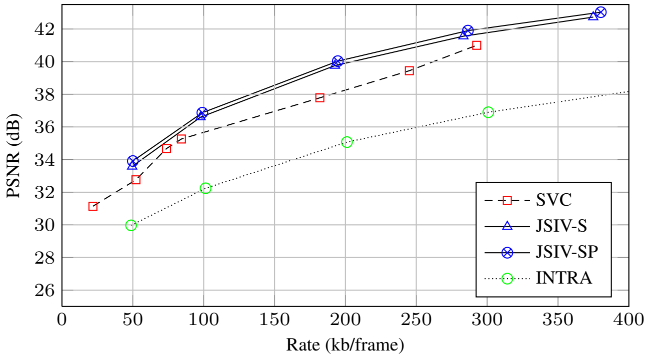

Fig. 3. A comparison of the performance of various schemes for the "Professor" sequence. Note that the x-axis is in (kb/frame).

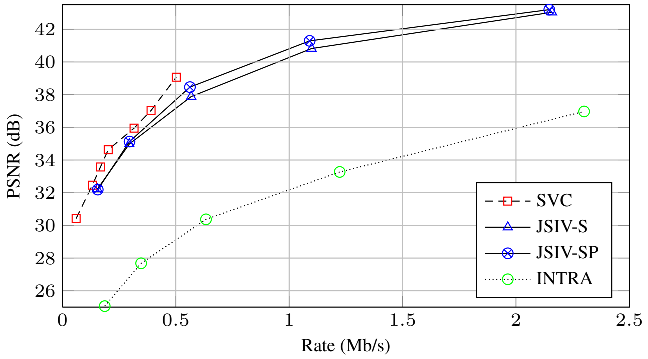

Fig. 4. A comparison of the performance of various schemes for the "Speedway" sequence.

is used for all the tests reported here. These sequence were chosen because they are surveillance footage and therefore they are more suitable in the absence of motion compensation.

For both JSIV-S and JSIV-SP, the test sequences are converted to JPEG2000 using Kakadu[3]. Three levels of irreversible DWT are employed for "Speedway" and five for "Professor". A code-block size of 32×32 and 20 quality layers are used for both sequences. The rates reported here are total rates including side information $(\bar{q}_n^\pi)$ and JPIP protocol overhead. Results reported are obtained with actual policies and 3 passes of $\mathcal{P}^1\mathcal{P}^2$. For SVC, JSVM[4] is used to compress and reconstruct these sequences. Intra-frame period is set to 8 to match that of JSIV. SVC results presented here employ three levels of temporal decimation with two enhancement layers. The enhancement layers use two levels of medium-grain scalability (MGS) between them giving a total of seven quality layers. No spatial scalability option is used for these test. For INTRA, also known as Motion-JPEG2000, each frame is independently optimized. All results are reported in PSNR calculated from the average MSE.

Figure 3 shows the performance of the various schemes for the "Professor" sequence. It can be seen that JSIV-SP performance is comparable to that of SVC and that it performs better than JSIV by somewhere between 0.2 to 0.3 dB in this case. Figure 4 shows the performance of the various schemes for the "Speedway" sequence. It can be seen that JSIV-SP performance is worse than SVC by perhaps 1.5dB at a certain rate and that it performs similar to JSIV at low rates and better by as much as 0.5 dB at higher rates. All the methods performs considerably better than the INTRA case. Experimental results also reveal that JPIP overhead and side information each account for less than 5% of the total bandwidth.

5. CONCLUDING REMARKS

We have presented a method of selecting a favorable predictor utilizing the client's knowledge about the quantization intervals of received samples in the context of JSIV in a dyadic hierarchical B-frame arrangement that does not employ motion compensation, and we have employed it in server and client policies. This method provides only a small improvement in PSNR. We have also demonstrated that it is possible for the client and server policies in JSIV to evolve independently without a negative impact on performance. Side information delivery as a JPEG2000 image proved to be effective, easy to implement, and does not require any changes to the JPIP protocol. Work is still needed to implement a real-time delivery system.

6. REFERENCES

[1] N. Mehrseresht and D. Taubman, "An efficient content-adaptive motion compensated 3D-DWT with enhanced spatial and temporal scalability," *IEEE Trans. Image Proc.*, pp. 1397–1412, June 2006.

[2] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Transactions on Image Processing*, vol. 3, pp. 559–571, September 1994.

[3] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sept. 2007.

[4] A.T. Naman and D. Taubman, "Rate-distortion optimized delivery of JPEG2000 compressed video with hierarchical motion side information," *Proc. IEEE Int. Conf. Image Proc. 2008*, pp. 2312–2315, October 2008.

[5] A.T. Naman and D. Taubman, "Distortion estimation for optimized delivery of JPEG2000 compressed video with motion," *IEEE 10th Workshop on Multimedia Signal Processing, 2008, MMSP 2008*, pp. 433–438, October 2008.

[6] A.T. Naman and D. Taubman, "Rate-distortion optimized JPEG2000-based scalable interactive video (JSIV) with motion and quantization bin side-information," *Proc. IEEE Int. Conf. Image Proc. 2009*, pp. 3081–3084, November 2009.

[7] Ngai-Man Cheung and Antonio Ortega, "Compression algorithms for flexible video decoding," *Visual Communications and Image Processing 2008*, vol. 6822, no. 1, pp. 68221S, 2008.

[8] F.-O. Devaux, J. Meessen, C. Parisot, J.-F. Delaigle, B. Macq, and C. De Vleeschouwer, "A flexible video transmission system based on JPEG2000 conditional replenishment with multiple references," *Proc. IEEE Int. Conf. Acoust. Speech and Sig. Proc.*, April 2007.

[9] A. Mavlankar, J. Noh, P. Baccichet, and B. Girod, "Peer-to-peer multicast live video streaming with interactive virtual pan/tilt/zoom functionality," *Proc. IEEE Int. Conf. Image Proc. 2008*, pp. 2296–2299, October 2008.

[10] Pietro Zanuttigh, Nicola Brusco, David Taubman, and Guido Cortelazzo, "A novel framework for the interactive transmission of 3D scenes," *Signal Processing: Image Communication, Special Issue on Interactive Representation of Still and Dynamic Scenes*, vol. 21, no. 9, pp. 787 – 811, 2006.

[11] D.S. Taubman and M.W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, Boston, 2002.

---

[3] http://www.kakadusoftware.com/, Kakadu software, version 5.2.4.

[4] JSVM version 9.18.1 obtained through CVS from its repository at garcon.ient.rwth-aachen.de