Distortion Estimation for Optimized Delivery of JPEG2000 Compressed Video with Motion

Aous Thabit Naman[1] and David Taubman[2]

School of Electrical Engineering and Telecommunications,
University of New South Wales,
Sydney, Australia.
[1] aous@student.unsw.edu.au
[2] d.taubman@unsw.edu.au

***Abstract*—A JPEG2000 compressed video sequence can provide better support for scalability, flexibility, and accessibility at a wider range of bit-rates than the current motion-compensated predictive video coding standards; however, it requires considerably more bandwidth to stream. The authors have recently proposed a novel approach that reduces the required bandwidth; this approach uses motion compensation and conditional replenishment of JPEG2000 code-blocks, aided by server-optimized selection of these code-blocks. The proposed approach can serve a diverse range of client requirements and can adapt immediately to interactive changes in client interests, such as forward or backward playback and zooming into individual frames. This work extends the previous work by approximating the distortion associated with the decisions made by the server without the need to recreate the actual video sequence at the server. The proposed distortion estimation algorithm is general and can be applied to various frames arrangements. Here, we choose to employ it in a hierarchical arrangement of frames, similar to the hierarchical B-frames of the SVC scalable video coding extension of the H.264/AVC standard. We employ a Lagrangian-style rate-distortion optimization procedure to the server transmission problem and compare the performance of both distortion estimation and exact distortion calculation cases against streaming individual frames and SVC. Results obtained suggest that the distortion estimation algorithm considerably reduces the amount of calculation needed by the server without enormously degrading the performance compared to the exact distortion calculation case. This work introduces the concepts, formulates the estimation and optimization problems, proposes a solution, and compares the performance to alternate strategies.**

I. Introduction

Scalable video can solve many existing problems in video storage and streaming; as it can accommodate the varying needs of different clients from one base source file; and it can dynamically adapt to available network bandwidth gracefully degrading the streamed video quality. For this reason, it has been an active area of research in the last twenty years with many promising results [1] [2] [3]. The existing video coding standards such as MPEG-1 through MPEG-4 and H.261 through H.264 offer at best limited scalability. Recently, a scalable video coding (SVC) extension to H.264/AVC has been approved within the ISO working group known as MPEG to provide improved scalability options. For all these standards the need to exploit inter-frame redundancy imposes restrictions on video structure that limit accessibility for streaming applications; one example is the need to stream multiple frames even if the client is interested in only one specific frame.

Recently, the authors have presented a new approach for serving scalable video [4] [5] [6]. To provide for quality and spatial resolution scalability, this approach relies on JPEG2000 to independently compress the individual frames. For inter-frame redundancy reduction, the approach relies on motion compensation and optimized selection of JPEG2000 code-blocks. These goals are achieved through the cooperation of loosely coupled server and client policies. The server policy dynamically determines which code-blocks to send, while the client policy determines how best to make use of the data which is received from the server, possibly in conjunction with some motion model.

In [4], we presented the approach and presented one realistic implementation; however, that work is limited to the special case in which redundancy is exploited only within disjoint frame pairs. In [5], the server optimization policy was extended to a sliding window of sequential frames, with the potential to exploit the redundancy between any two frames within the window. In [6] the server optimization policy was extended to a hierarchical group of frames, similar to the hierarchical B-frames dyadic prediction structure used by the SVC extension of H.264/AVC. In this paper, we turn our attention to improving the real-time performance of the server by estimating the distortion associated with the server selection of code-blocks rather than reconstructing the video to calculate the exact distortion. The results presented here are still somewhat idealistic; as it is assumed that the client has all the necessary information to make optimal use of the data it receives from the server. This allows us to focus only on the server policy.

Recently Devaux et al. [7] investigated a problem similar to the one we investigated in [5]; however, that work deals only with sequential prediction and does not employ motion-compensation. Another recent work by Cheung and Ortega [8] is similar to our work in attempting to enable flexible video delivery by dealing with motion information and residual distortion as side data; unlike the approach proposed here, theirs is based on distributed video coding.

The applications of the paradigm under consideration are diverse and we expect real-time interactive applications, such as surveillance video browsing and teleconferencing, to benefit the most. We briefly mention some benefits here; the interested

reader can also refer to [4] [5]. The server can adapt the streamed video to the client's processing power, resolution, region of interest, or motion-compensation capability without the need to recode the video. For lossy transmission environments, the server does not need to retransmit lost packets, instead it can adapt by adjusting its delivery policy for future frames alone. Both the client and the server can easily and dynamically change from video playback mode to individual frame browsing mode and any data in the client cache is readily available for the reconstruction of individual frames. In this paper, we choose to examine a scenario in which the client may be interested in viewing a particular frame from the video, in addition to browsing the streamed video as a whole. Such a scenario is common in video surveillance browsing applications and also interesting for video editing.

The remainder of the paper is structured as follows. Sections II and III elaborate on our client and server policies, explaining their theoretical aspects. Section IV provides experimental results. Finally, Section V states our conclusions.

II. Client Policy

Motion compensation is widely used to exploit inter-frame temporal redundancy. Here, we choose to arrange the frames in a dyadic hierarchical structure with temporal decimation levels $T_0, T_1, \ldots T_K$. Figure 1 shows the structure for the case of $K = 3$. Each frame belongs to one or more temporal decimation level T_k depending on its position. Frames at level T_K are not predicted from any other frame (intra frames). At temporal levels with $k < K$, the code-blocks can either come from the frame itself (intra code-blocks) or be predicted from the two nearby frames as shown in Figure 1. We write $f^k_{\rightarrow n}$ for the predicted frame given by

$$\begin{aligned} f^k_{\rightarrow n} &= \frac{1}{2}\left(\mathcal{W}_{n-p\rightarrow n}\left(f^k_{n-p}\right) + \mathcal{W}_{n+p\rightarrow n}\left(f^k_{n+p}\right)\right) \\ &= \frac{1}{2}\left(f^k_{n-p\rightarrow n} + f^k_{n+p\rightarrow n}\right) \end{aligned} \tag{1}$$

where $\mathcal{W}_{a\rightarrow b}$ is the motion compensation operator mapping f_a to f_b and $p = 2^k$ is the distance between a frame and the two adjacent frames at the k^{th} temporal level.

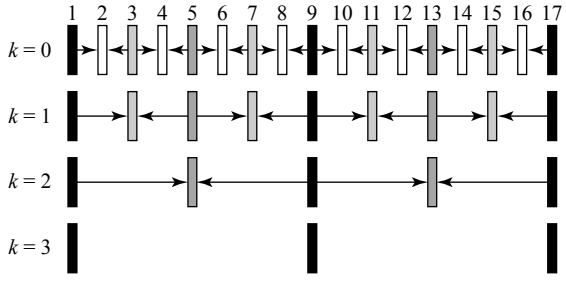

Fig. 1. Two groups of pictures in the dyadic hierarchical structure. The arrows show prediction directions and the numbers at the top are frame indices.

For each code-block C^β_n of each frame f_n the client receives zero or more quality layers q^β_n. Consequently, the dequantized samples of a code-block $\tilde{C}^\beta_n$ have an associated distortion given by $\tilde{D}^\beta_n = \|\tilde{C}^\beta_n - C^\beta_n\|^2$. For each code-block, the client chooses the sub-band samples in $C^\beta_{\rightarrow n}$ obtained from the two-dimensional discrete wavelet transform (2D-DWT) of frame $f_{\rightarrow n}$ if their corresponding distortion error $D^\beta_{\rightarrow n} = \|C^\beta_{\rightarrow n} - C^\beta_n\|^2$ is smaller than $\tilde{D}^\beta_n$; otherwise $\tilde{C}^\beta_n$ is selected. Thus, the distortion in C^β_n is given by

$$D^\beta_n = \min\left\{\tilde{D}^\beta_n, D^\beta_{\rightarrow n}\right\} \tag{2}$$

In this work, we assume that the client has the information required to make decisions on the same basis as the server. This is unrealistic as the client usually receives only partial and quantized information. However, this assumption is more reasonable for the case in which the server uses only distortion estimates, which is the chief feature of this paper.

III. Server Policy

The frames are divided into groups of pictures $\mathcal{G}$, each with 2^{K+1} frames. The last frame in group $\mathcal{G}_s$ is also the first frame in $\mathcal{G}_{s+1}$. Each $\mathcal{G}$ is jointly optimized subject to a transmission budget of $L_{\max}$; as such, frames that belong to T_K have two chances of receiving data. Using an additive model, the distortion in one $\mathcal{G}$ is given by

$$D = \sum_{n\in\mathcal{G}_s}\sum_{\beta\in f_n} D^\beta_n \tag{3}$$

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

$$J_\lambda = \sum_{n\in\mathcal{G}_s}\sum_{\beta\in f_n}\left(D^\beta_n + \lambda\cdot|q^\beta_n|\right) \tag{4}$$

where $|q^\beta_n|$ denotes the number of bytes in q^β_n quality layers of C^β_n. The Lagrangian parameter λ is adjusted until the solution which minimizes J_λ satisfies the length constraint.

Direct minimization of (4) has two difficulties. The first is in selecting q^β_n since D^β_n for $f_n \notin T_K$ depend on other frames whose contribution is also being optimized. The second difficulty is that each C^β_n might contribute to code-blocks in other frames, in which case its distortion contribution should be weighted accordingly; however, this contribution is not known until the decisions for those other frames are made.

To deal with these two difficulties, we propose a two pass approach. In the first pass (PASS-1), q^β_n are determined for each C^β_n. In the second pass (PASS-2), the contributions of each C^β_n are evaluated and contribution weights w^β_n are determined as explained in III-B. These weights are used in the next iteration of PASS-1. Multiple iterations of PASS-1 followed by PASS-2 are possible, where last iteration can skip PASS-2. However, we found that very little improvement is obtained beyond the second iteration.

For PASS-1, we notice that f^K_n are independent of any other frames and q^β_n can be easily determined for a given λ as explained next. Once these are determined, the distortions D^β_n for all the f^{K-1}_n are known and their q^β_n can similarly be determined. This process is repeated until all q^β_n are decided.

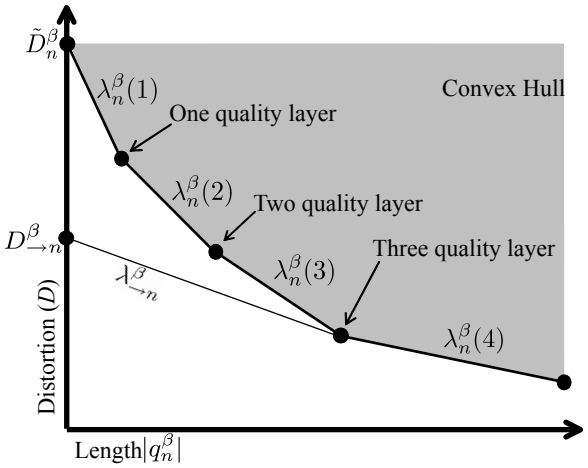

Fig. 2. A typical convex rate-distortion curve for a code-block C_n^β showing the quality layers and $D_{\to n}^\beta$ when $D_{\to n}^\beta < \tilde{D}_n^\beta(0)$.

We turn our attention to determining q_n^β. Figure 2 shows a typical distortion-length curve for a code-block C_n^β, which is guaranteed to be convex by construction [9]. Each circle in the figure represents one quality layer. Also shown in the figure is the distortion $D_{\to n}^\beta$. We define the distortion-length slope associated with each q_n^β by $\lambda_n^\beta(q) = (\tilde{D}_n^\beta(q-1) - \tilde{D}_n^\beta(q))/(|q| - |q-1|)$.

In the absence of any prediction data or when $D_{\to n}^\beta \geqslant \tilde{D}_n^\beta(0)$, q_n^β is optimally determined from

$$q_n^\beta = \max\left\{q \mid \lambda_n^\beta(q) > \lambda\right\} \tag{5}$$

For f_n^k where $k < K$, the existence of prediction sources reduces the effective distortion associated with $q_n^\beta = 0$ to $D_{\to n}^\beta$ when $D_{\to n}^\beta < \tilde{D}_n^\beta(0)$. This reduces the effective distortion-length slope associated with one or more initial quality layers to $\lambda_{\to n}^\beta$, as shown in Figure 2. The optimal choice for q_n^β then becomes

$$q_n^\beta = \begin{cases} 0 & \text{if } \lambda > \lambda_{\to n}^\beta \\ \max\left\{q \mid \lambda_n^\beta(q) > \lambda\right\} & \text{if } \lambda \leqslant \lambda_{\to n}^\beta \end{cases} \tag{6}$$

A. Distortion Estimation

The distortion in the reconstructed video originates from two main sources: motion distortion $D_{\to n}^{\mathrm{M},\beta}$, due to motion modeling difficulties such as occlusion, aperture issues, and parameter quantization; and quantization distortion $D_{\to n}^{\mathrm{Q},\beta}$, arising from the distorted reference frames used for motion compensation. Under the assumption that these are roughly uncorrelated, the distortion in $D_{\to n}^\beta$ can be approximated by [4]

$$D_{\to n}^\beta \approx D_{\to n}^{\mathrm{Q},\beta} + D_{\to n}^{\mathrm{M},\beta} \tag{7}$$

The distortion estimation algorithm separately accounts for these sources of distortion. Motion distortion can be easily pre-calculate and stored in the server, based on unquantized reference frames. At most three motion distortion fields are needed for each predicted frame since each code-block in the 2D-DWT of a frame can be predicted either from the left, the right, or the average of these two references.

The propagation of quantization distortion due to motion compensation is more complex and requires elaboration. The error in the reference frame f_r, can be expressed in term of the errors at each location $\mathbf{k}$ in each of its sub-bands b, as

$$\delta f_r = \sum_b \sum_{\mathbf{k}} \delta B_r^b[\mathbf{k}] \cdot S_{\mathbf{k}}^b$$

where $S_{\mathbf{k}}^b$ denotes the relevant synthesis vectors (images). Applying the motion mapping operator, the error in the predicted frame f_n is

$$\delta f_n = \sum_b \sum_{\mathbf{k}} \delta B_r^b[\mathbf{k}] \cdot \mathcal{W}_{r\to n}(S_{\mathbf{k}}^b)$$

since $\mathcal{W}_{r\to n}$ is a linear operator. The error at the $\mathbf{p}^{\text{th}}$ location in the predicted sub-band b' of f_n, due to quantization in the $\mathbf{k}^{\text{th}}$ location of sub-band b in f_r can be obtained by applying the linear analysis operator $A_{\mathbf{p}}^{b'}$ for sub-band b' at location $\mathbf{p}$; that is,

$$\delta B_{r\to n}^{\mathrm{Q},b'}[\mathbf{p}] = \sum_b \sum_{\mathbf{k}} \delta B_r^b[\mathbf{k}] \cdot \left\langle \mathcal{W}_{r\to n}(S_{\mathbf{k}}^b), A_{\mathbf{p}}^{b'} \right\rangle$$

Assuming that the quantization errors in the sub-bands are approximately uncorrelated, the distortion power for some region $\mathcal{R}'$ around $\mathbf{p}$ in sub-band b' can then be approximated by

$$\begin{aligned} &\sum_{\mathbf{p}\in\mathcal{R}'} \left|\delta B_{r\to n}^{\mathrm{Q},b'}[\mathbf{p}]\right|^2 \\ &\quad\approx \underbrace{\sum_b \sum_{\mathbf{p}\in\mathcal{R}'} \sum_{\mathbf{k}} \left|\delta B_r^b[\mathbf{k}]\right|^2 \cdot \left\langle \mathcal{W}_{r\to n}(S_{\mathbf{k}}^b), A_{\mathbf{p}}^{b'} \right\rangle^2}_{D_{\mathcal{R}'}^{b\to b'}} \end{aligned}$$

The fact that both the $\mathcal{W}(S_{\mathbf{k}}^b)$ and $A_{\mathbf{p}}^{b'}$ operators have limited support with decaying boundaries means that $D_{\mathcal{R}'}^{b\to b'}$ depends mainly on the distortion contributions $\delta B_r^b[\mathbf{k}]$ inside and around the region $\mathcal{R}$, the projection of $\mathcal{R}'$ onto sub-band b. If $\mathcal{R}'$ is small enough such that the distortion around it can be approximated by a uniform quantization noise power $D_{\mathcal{R}}^b/|\mathcal{R}|$, we have

$$D_{\mathcal{R}'}^{b\to b'} \approx \frac{D_{\mathcal{R}}^b}{|\mathcal{R}|} \cdot \sum_{\mathbf{p}\in\mathcal{R}'} \underbrace{\sum_{\mathbf{k}} \left\langle \mathcal{W}_{r\to n}(S_{\mathbf{k}}^b), A_{\mathbf{p}}^{b'} \right\rangle^2}_{W_{\mathbf{p}}^{b\to b'}}$$

Here, $W_{\mathbf{p}}^{b\to b'}$ represents a weighting factor which reflects the contribution of the quantization noise power around location $\mathbf{k} \approx \overleftarrow{\mathcal{W}}_{r\to n}^{b\to b'}(\mathbf{p})$ in sub-band b to the distortion at location $\mathbf{p}$ in sub-band b', where $\overleftarrow{\mathcal{W}}_{r\to n}^{b\to b'}$ maps locations in subband b' of frame f_n back to locations in subband b of the reference frame f_r, according to the motion model. Denoting the average quantization noise power $D_{\mathcal{R}}^b/|\mathcal{R}|$ around $\mathcal{R}$ by $\bar{D}_r^b[\mathbf{k}]$ gives

$$\left|\delta B_{r\to n}^{\mathrm{Q},b'}[\mathbf{p}]\right|^2 \approx \sum_b \bar{D}_r^b\left[\overleftarrow{\mathcal{W}}_{r\to n}^{b\to b'}(\mathbf{p})\right] \cdot W_{\mathbf{p}}^{b\to b'} \tag{8}$$

For convenience of implementation, we approximate $\bar{D}_r^b[\mathbf{k}]$ as constant over *grid* blocks $\mathcal{B}_r^b[i]$, such that $\cup_i \mathcal{B}_r^b[i]$ span

subband b of the reference frame f_r, writing $\bar{D}_r^b[\mathbf{k}] = D_r^b[i] / \left|\mathcal{B}_r^b[i]\right|$ for all $\mathbf{k} \in \mathcal{B}_r^b[i]$. Similarly, we partition each subband b' of the predicted frame f_n into *grid* blocks $\mathcal{B}_n^{b'}[j]$, writing $D_{r\to n}^{Q,b'}[j]$ for the predicted total distortion in grid block $\mathcal{B}_n^{b'}[j]$ due to quantization distortion in f_n. Under these conditions, equation (8) can be recast as

$$D_{r\to n}^{Q,b'}[j] \approx \sum_{\mathbf{p}\in\mathcal{B}_n^{b'}[j]} \sum_{i,b\ni \overleftarrow{\mathcal{W}}_{r\to n}^{b\to b'}(\mathbf{p})\in\mathcal{B}_r^b[i]} D_r^b[i] \frac{W_{\mathbf{p}}^{b\to b'}}{\left|\mathcal{B}_r^b[i]\right|} \qquad (9)$$

This might look complicated, but the interpretation is simple. For each location $\mathbf{p}$ in grid block $\mathcal{B}_n^{b'}[j]$, find the corresponding $\mathbf{k}$ in all the sub-bands of the reference frame f_r and add $W_{\mathbf{p}}^{b\to b'} / \left|\mathcal{B}_r^b[i]\right|$ multiplied by the grid block distortion $D_r^b[i]$ to the accumulated distortion, $D_{r\to n}^{Q,b'}[j]$. In practice, we work with grid blocks of size 4×4. More simplifications will be introduced in III-C.

The distortion due to quantization, in code-block C_n^β is readily found by summing the $D_{r\to n}^{Q,b'}[j]$ contributions from all grid blocks which it contains. The reason for estimating distortion in grid blocks, rather than directly at the code-block level, is that this provides a finer description in the event that frame f_n itself becomes a reference frame for motion compensation in finer temporal levels. It is important to note that for such frames, $D_r^b[i]$ can be a combination of motion distortion and quantization distortion effects as these frames are themselves predicted. In this case, we are assuming not only that quantization and motion compensation errors are uncorrelated, so that their squared error distortions add, but also that the motion compensation errors produced at one level of the temporal hierarchy are uncorrelated with those produced at the next level. The validity of this assumption may be questionable, but seems necessary for the development of a workable distortion estimation strategy.

The derivation above is not specific to any particular motion model; however, for the purpose of this paper we choose to use a block-based translational motion model. For such a model, $\left\langle \mathcal{W}(S_{\mathbf{k}}^b), A_{\mathbf{p}}^{b'} \right\rangle$ can be decomposed into a collection of polyphase filters, whose coefficients can be squared and summed to determine values for the weights $W_{\mathbf{p}}^{b\to b'}$, averaged over various locations[1] $\mathbf{p}$, for each possible translational shift. In fact, the process can be further simplified by recognizing that $\left\langle \mathcal{W}(S_{\mathbf{k}}^b), A_{\mathbf{p}}^{b'} \right\rangle$ is a separable function of the horizontal and vertical components of $\mathbf{p}$ and $\mathbf{k}$. Our implementation pre-computes the average values of $W_{\mathbf{p}}^{b\to b'}$ for each shift and stores them in a look-up table. The server also needs to keep code-block quantization distortions D_n^β and distortion-length slope statistics and for all the frames it intends to serve. We then model the quantization distortion power as uniform over each code-block, so that $D_r^b[i] = D_r^\beta \cdot \left|\mathcal{B}_r^b[i]\right| / \left|C_r^\beta\right|$.

[1]Note that the cyclo-stationary nature of the DWT, introduces a dependence of $W_{\mathbf{p}}^{b\to b'}$ on $\mathbf{p}$, even for constant translational motion, whenever b' comes from a higher resolution level than b. However, it is sufficient to use average values for our distortion estimation process.

B. Contribution Weights Estimation

We turn our attention to PASS-2. Using (9), the quantization distortion propagated from f_r to f_n is

$$D_{r\to n}^{\mathrm{Q}} \approx \sum_{j\in\mathcal{P}_n} \sum_{\mathbf{p}\in\mathcal{B}_n^{b'}[j]} \sum_{i,b\ni \overleftarrow{\mathcal{W}}_{r\to n}^{b\to b'}(\mathbf{p})\in\mathcal{B}_r^b[i]} D_r^b[i] \frac{W_{\mathbf{p}}^{b\to b'}}{\left|\mathcal{B}_r^b[i]\right|}$$

where $\mathcal{P}_n$ is the set of all predicted grid blocks $\mathcal{B}_n^{b'}[j]$ in f_n. For convenience, we designate the contribution of $D_r^b[i]$ to $D_{r\to n}^{\mathrm{Q}}$ by $\theta_r^b[i] \cdot D_r^b[i]$; where $\theta_r^b[i]$ is the grid block contribution weight and is given by

$$\theta_r^b[i] = \sum_{j\in\mathcal{P}_n} \sum_{\substack{\mathbf{p}\in\mathcal{B}_n^{b'}[j],\\ \overleftarrow{\mathcal{W}}_{r\to n}^{b\to b'}(\mathbf{p})\in\mathcal{B}_r^b[i]}} \frac{W_{\mathbf{p}}^{b\to b'}}{\left|\mathcal{B}_r^b[i]\right|} \qquad (10)$$

Although the last equation looks complicated, the interpretation is simple. $\theta_r^b[i]$ is a contribution weight associated with each grid block in f_r and reflects that grid block's quantization distortion contribution to predicted frames. For a given value of λ, we identify all the predicted grid blocks $\mathcal{P}_n$ of the predicted frame. For each location $\mathbf{p}$ in $\mathcal{P}_n$, we find the corresponding location $\mathbf{k}$ in each sub-band of f_r and we add $W_{\mathbf{p}}^{b\to b'} / \left|\mathcal{B}_r^b[i]\right|$ to the corresponding $\theta_r^b[i]$. Of course, the accumulation of contributions to the $\theta_r^b[i]$ terms for frame f_r needs to be extended over all target frames which may be predicted using f_r. It is important to mention that for the case of hierarchical frame arrangement being considered here, the reference frame itself might be predicted. In this case, our implementation composes the contribution weights from one predicted frame to the next.

The contribution weight w_r^β for a code-block distortion D_r^β is readily obtained from weighted averaging of the $\theta_r^b[i]$ contributions for all grid blocks it contains; that is,

$$w_r^\beta = \sum_{i\ni\mathcal{B}_r^b[i]\subset C_r^\beta} \theta_r^b[i] \cdot \frac{\left|\mathcal{B}_r^b[i]\right|}{\left|C_r^\beta\right|}$$

These code-block contribution weights modify (4) to become

$$J_\lambda = \sum_{n\in\mathcal{G}_s} \sum_{\beta\in f_n} \left[\left(1+w_r^\beta\right)\cdot D_n^\beta + \lambda\cdot|q_n^\beta|\right] \qquad (11)$$

and the solution for q_n^β, based on (6), becomes

$$q_n^\beta = \begin{cases} 0 & \text{if } \lambda > \lambda_{\to n}^{*,\beta} \\ \max\left\{q \mid (1+w_n^\beta)\cdot\lambda_n^\beta(q) > \lambda\right\} & \text{if } \lambda \leqslant \lambda_{\to n}^{*,\beta} \end{cases} \qquad (12)$$

where $\lambda_{\to n}^{*,\beta} = (1+w_n^\beta)\cdot\lambda_{\to n}^\beta$.

In PASS-1, we start with $w_r^\beta = 0$. In PASS-2 these values are updated according to the decisions made in PASS-1. Then values obtained in PASS-2 are used in the subsequent PASS-1 and so on.

C. Motion Compensation

The most straightforward way to perform motion compensation is to apply the motion compensation operator $\mathcal{W}$ to full resolution reconstructed frames. It can be seen from (9) that the distortion in each sub-band of the reference frame generally propagates to all sub-bands in the predicted frame. To reduce the amount of calculation, we propose the use of the "safe" motion compensation operator $\mathcal{W}^{\text{SAFE}}$ defined in [10], which is briefly described here for completeness.

In a D-level decomposition of a frame, we write $\mathbf{f}^d$ for all the sub-bands required to reconstruct the d-th resolution; thus $\mathbf{f}^0$ is all sub-bands in f, while $\mathbf{f}^D$ is the LL_D sub-band alone. We write H^d, $0 \leq d < D$, for the three detail sub-bands, HL_d, LH_d, and HH_d at the d-th resolution, and $\mathcal{A}_H$ for the one-level (2D-DWT) analysis operator that recovers these sub-bands from $\mathbf{f}^d$. Then, the sub-bands of the predicted frame using "safe" motion compensation can be obtained from

$$\begin{aligned} \mathbf{f}_b^D &= \mathcal{W}_{a\to b}^D(\mathbf{f}_a^D) \\ H^d &= \mathcal{A}_H\left[\mathcal{W}_{a\to b}^d(\mathbf{f}_a^d)\right] \qquad 0 \leq d < D \end{aligned} \tag{13}$$

where $\mathcal{W}^d$ is the motion compensation operator with scaled parameters that operates at the d-th resolution.

For a given sub-band, $\mathcal{W}^{\text{SAFE}}$ limits sources of distortion to only sub-bands in resolutions d through D. This allows us to focus on sources of distortion that are most dominant and at the same time considerably reduce the calculations required for (9). In practice, we recursively estimate the distortion in $\mathbf{f}^d$ of the source frame from distortions in H^d and $\mathbf{f}^{d+1}$. Moreover, $\mathcal{W}^{\text{SAFE}}$ is used only for the distortion estimation process, while motion compensation itself employs full-resolution motion compensation $\mathcal{W}$, with the in-band approach described in [10]. We justify this by the need for the client to reconstruct the best possible video, while a coarse model is sufficient for error estimation.

IV. Experimental Results

The results presented here are for three test sequences. The first two are the "crew" and "harbour" MPEG test sequences and the third is "speedway"[2]. Only the the Y-component of the first 49 frames of "crew" and "harbour" sequences is used. These frames have a resolution of 704×576 with a bit depth of 8 bits. Similarly, only the Y-component of the first 193 frames of the "speedway" sequences is used. These frames have have a resolution of 352×288 with a bit depth of 8 bits. Motion complexity for the sequences are simple for "speedway", moderate for "crew", and complex for "harbour".

All the sequences are compressed to JPEG2000 format using Kakadu[3], employing five levels of wavelet decomposition, 20 quality layers, and a code-block size of 32×32. Motion compensation employs an advanced hierarchical block-based motion model. Possible grid values range from 128 pixel for the coarsest field down to 4 pixel. Motion is estimated to $1/4$ of a pixel accuracy by employing a 7×7 interpolation kernel obtained from windowing cubic spline functions. For all the sequences, the frame rate is 30 frames/second and the rates given here are for the encoded sub-band samples and encoded motion vectors; they exclude any headers, and signaling to the client, etc.

Four methods are compared here: the proposed distortion estimation method, identified as "APPROX"; the exact distortion method, identified as "EXACT"; the method identified as "INTRA", which is to individually optimize each frame subject to the rate constraint; and the SVC extension of H.264/AVC, identified as "SVC".

For "SVC", the options are as follows. For the "speedway" sequence, two enhancement layers were used with five levels of medium-grain scalability (MGS) giving a total of 11 quality layers. For the "crew" and "harbour" sequences, three enhancement layers were used with three levels of MGS giving a total of 10 quality layers. The "SVC" streams generated had only one resolution, the highest resolution. Although providing more resolution and quality layers makes the comparison fairer as the proposed paradigm can easily provide them; we did not pursue them for two reasons: adding more layers to an SVC stream generally reduces its overall performance for a given bit-rate; and the implementation[4] we have consumes considerable amount of memory and having multiple layers for 704×576 sequences can easily exceed PC hardware design limits. The intra frame period was set to 8 frames and the search range is 16 integer pixels with $1/4$ sub-pixel accuracy.

For the "APPROX" and "EXACT" methods, a K value of three and three iterations of PASS1-PASS2 are used.

The average MSE expressed in terms of PSNR at various bit rates for the "crew", "harbour", and "speedway" sequences are shown in Figure 3, 4, and 5, respectively.

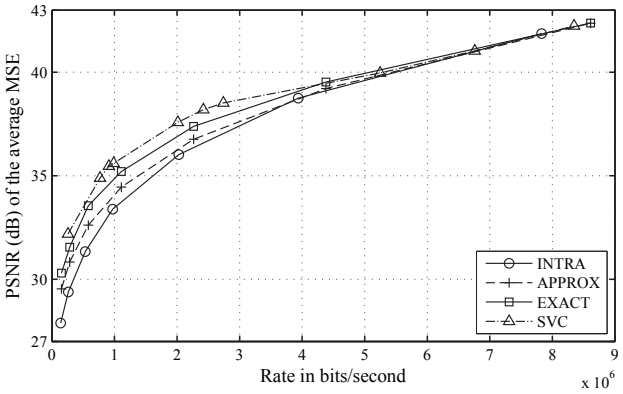

Fig. 3. Performance comparison between the various algorithms for the "crew" sequence at various bit-rates in PSNR of the average MSE.

For the three test sequences, it can be concluded that the proposed approximation reduces the quality of the reconstructed video by anywhere from 0.1 dB to 1 dB compared to the exact case. This reduction depends on bit-rate and motion complexity of the sequence with bigger differences

[2]http://www.openjpeg.org/samples/, OpenJpeg project homepage.

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

[4]JSVM version 9.12.2 obtained through SVC from its repository at garcon.ient.rwth-aachen.de

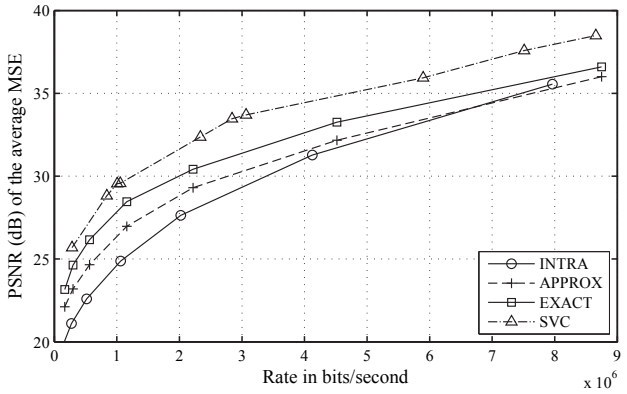

Fig. 4. Performance comparison between the various algorithms for the "harbour" sequence at various bit-rates in PSNR of the average MSE.

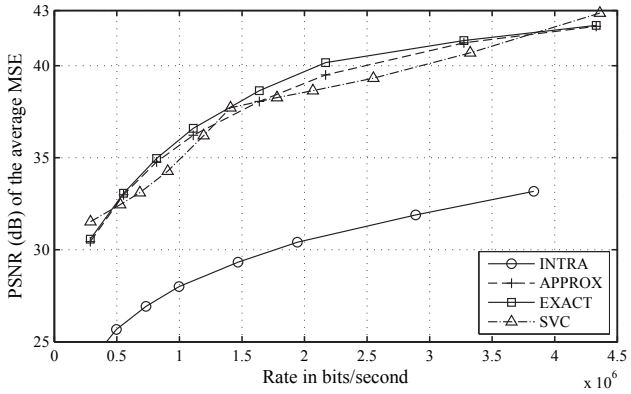

Fig. 5. Performance comparison between the various algorithms for the "speedway" sequence at various bit-rates in PSNR of the average MSE.

for the case of complex motion and for moderate bit-rates. It can also be seen that the proposed paradigm matches or exceeds the performance of "SVC" for the simple-motion sequence "speedway"; while for the complex-motion sequence "harbour", "SVC" outperforms the proposed paradigm by perhaps up to 2 dB. It should be remembered that for cases other than "SVC", the results presented here do not take into account the size of headers and any needed signaling, which negatively impact the outcome.

We turn our attention to the benefits of the proposed approach for a client interested in seeing only one frame. For such a scenario, both EXACT and APPROX methods reduce to the INTRA case and therefore only the INTRA case is shown for all of these three cases. Figure 6 shows the PSNR of frame 10, a frame that belongs to temporal level T_0 and not T_1 of the "harbour' sequence; half of all the frames in the sequence are in level T_0 only. For the "SVC" case, the effective rate shown in the figure is the sum of the rates for frames 9, 10, 11, 13, 15, and 17 as all these frames contribute to frame 10. It can be seen that the savings are enormous.

V. Conclusion

The proposed paradigm provides better support for scalability and accessibility compared to SVC. This extended

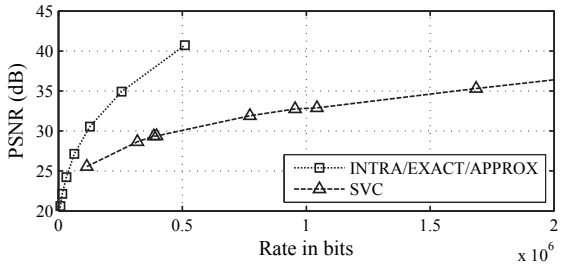

Fig. 6. PSNR of frame 10 of the "harbour" sequence at various bit-rates for the INTRA, EXACT, APPROX, and SVC methods.

flexibility might bring slightly better quality for sequences with simple motion while sequences with complex motion might suffer by a loss in quality of perhaps 2 dB. For a client that wishes to browse individual frames during video playback, the proposed paradigm considerably outperforms SVC. This is hardly surprising, but underlines the benefits that may be provided by the proposed approach in interactive applications, where direct streaming is not the sole objective. The distortion estimation algorithm that has been introduced here considerably reduces the calculations required by the server; however, it also reduces the quality of the reconstructed video by anywhere from 0.1 dB to 1 dB. The hierarchical arrangement of frames provides good exploitation of temporal redundancies for both the proposed method and SVC. More work is needed to improve the approximations and to provide a fully realistic implementation, particularly for client signaling and the client policy.

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] 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.

[3] J.-R. Ohm, "Advances in scalable video coding," *Proc. of the IEEE*, vol. 93, January 2005.

[4] A. Naman and D. Taubman, "A novel paradigm for optimized scalable video transmission based on JPEG2000 with motion," *Proc. IEEE Int. Conf. Image Proc. 2007*, Septmeber 2007.

[5] ——, "Optimized scalable video transmission based on conditional replenishment of JPEG2000 code-blocks with motion compensation," *MV '07: Proceedings of the International Workshop on Workshop on Mobile Video*, pp. 43–48, Septmeber 2007.

[6] ——, "Rate-distortion optimized delivery of JPEG2000 compressed video with hierarchical motion side information," *Proc. IEEE Int. Conf. Image Proc. 2008*, October 2008, accepted for publication.

[7] 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.

[8] N.-M. Cheung and A. Ortega, "Flexible video decoding: A distributed source coding approach," *IEEE 9th Workshop on Multimedia Signal Processing, 2007, MMSP 2007.*, pp. 103–106, 1-3 Oct. 2007.

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

[10] D. Taubman, R. Mathew, and N. Mehrseresht, "Fully scalable video compression with sample-adaptive lifting and overlapped block motion," *SPIE Electronic Imaging (San Jose)*, vol. 5685, pp. 366–377, Jan 2005.