An Empirical Measurement of the Coding Efficiency in Scalable Video Coding

M.F. López, V.G Ruiz and I. García
Department of Computer Architecture and Electronics
mflopez@ace.ual.es, vruiz@ual.es, inma@ual.es

Abstract

Video compression techniques can be classified into scalable and non-scalable. Scalable coding is more suitable in variable band-width scenarios because it improves the quality of the reconstructed video. On the other hand, the scalability has a cost in terms of coding efficiency and complexity. This paper describes a JPEG2000-and-MCTF-based fully scalable video codec (FSVC) and analyzes a set of experiments to measure the cost of the scalability, comparing two different FSVC encoders, called open-loop FSVC and closed-loop FSVC. In the open-loop version of FSVC, the encoder uses the original images to make the predictions. The closed-loop version generates the predictions with reference images identical to those obtained by the decoder at a given bit-rate. Numerical and visual results demonstrate a small loss of the coding efficiency for the open-loop scheme. Moreover, the inclusion of the closed-loop loop increases the complexity of the encoder and produces poor performance at high bit-rates.

Keywords—**Scalable video coding, coding efficiency, JPEG 2000, MCTF, video streaming.**

1 Introduction

Scalable video coding is a technique which allows us to decode a compressed video stream in several different ways. Users can recover a specific version of a video according to their own requirements: (i) spatial resolution, (ii) image quality, (iii) frame rate and (iv) data rate. *Spatial Scalability* provides a set of lowered resolution reconstructions for each image or region of interest, typically in a dyadic way (that is, dividing vertical and horizontal resolutions by a power of two). The progressive minimization of the distortion of the reconstructed video at the decoder is achieved using *Quality Scalability*. A variation of the frame (or image) rate is obtained by means of *Temporal Scalability*. Finally, these types of scalabilities can be combined together to generalize the idea of scalability with the concept of *Data Rate Scalability*.

Scalable video coding is a major feature for video storage and video transmission systems. For example, in Video-on-Demand (VoD) applications, a server sends a video stream to a set of clients through a number of transmission links. For most of the cases, the quality, resolution, and frame-rate of the visualizations must be adapted to the requirements of the decoder and the band-width available for each link. In this context, the computational requirements of the servers are proportional to the number of different kinds of clients, and non-scalable video coding has two alternatives to minimize them: (i) the creation of a specific copy of the video sequence for each possible type of client or (ii) the use of CPU-intensive real-time transcoding processes to re-encode on-the-fly the video. Scalable video coding addresses this problem by storing only one copy of each video sequence at the server and simplifying the transcoding task. This simple transcoding consists of a re-ordering that can be carried out by the clients retrieving the adequate portions of the data of the compressed video.

This work describes and studies a fully scalable video coding system, called FSVC, specially designed for VoD applications over unpredictable band-width data networks (like the Internet). FSVC is open-loop MCTF-based (Motion Compensated Temporal Filtering) and its data output is a sequence of JPEG2000 packets that are placed in the compressed stream using some ordering (or progression). The decoding ordering of these packets determines the way the video sequence will be displayed when only a part of the compressed stream is decoded. FSVC supports the following kind of scalabilities: (i) fine grain progressive by quality, (ii) dyadic progressive by resolution and (iii) dyadic progressive by frame-rate. The coding behavior of FSVC is examined with the adding and testing of a closed-loop scheme.

The rest of this paper is organized as follows. In Section 2 the open-loop FSVC encoding system is described. The design of closed-loop FSVC is focused on Section 3. Experimental results are shown and analyzed in Section 4. Concluding remarks are given in Section 5.

2 The FSVC Codec

FSVC is a fully scalable video compression system. As it can be seen in Figure 1, the encoder is a differential coding scheme based on open-loop MCTF (Motion Compensated Temporal Filtering) applied to the wavelet do-

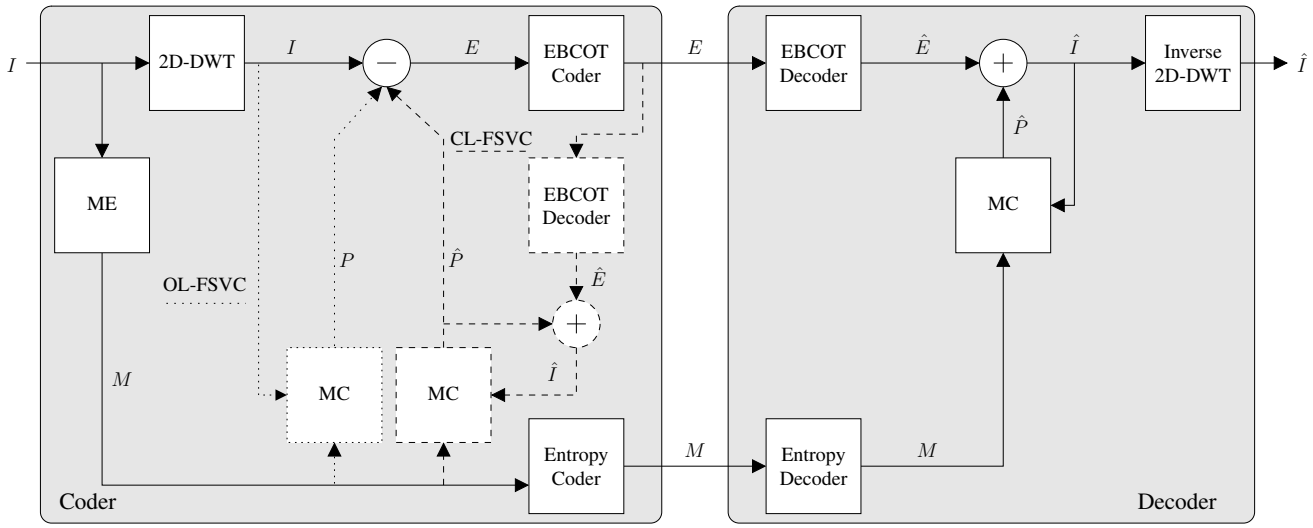

Figure 1: The block diagram of the FSVC codec. MC = Motion Compensation, ME = Motion Estimation, 2D-DWT = 2-Dimensional Discrete Wavelet Transform and EBCOT = Embedded Block Coding with Optimized Truncation.

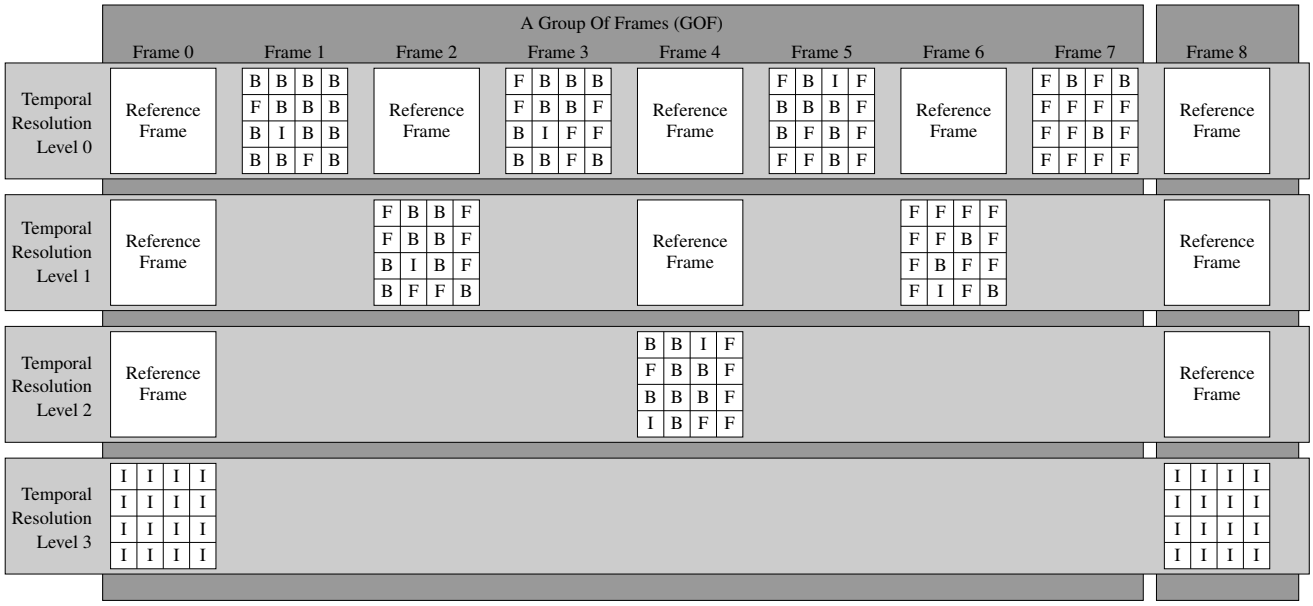

Figure 2: An example of the MCTF-based temporal decorrelation scheme of FSVC for a GOF with 8 frames (only one DWT subband is shown).

main, and EBCOT (Embedded Block Coding with Optimized Truncation) [5, 3, 7] applied to the residues. The compressor uses the motion information computed at the ME module and the original images to generate a sequence of prediction frames P that are subtracted to the original video sequence I. The prediction errors E are progressively encoded using the EBCOT module.

As it is shown in Figure 2, the input video sequence I is segmented in GOFs (Group Of Frames) of size $\mathcal{G}$ ($\mathcal{G} = 8$ in the example of the figure). Each GOF is divided into $1 + \log_2 \mathcal{G}$ temporal resolution levels to obtain dyadic temporal scalability in each GOF. The lowest temporal resolution level $\boldsymbol{T}_3$ is composed of the $I[\mathcal{G} \cdot i]$ frames (let denote this by $\boldsymbol{T}_3 = \{I[\mathcal{G} \cdot i]\}$), where $i = 0, 1, \cdots$ indexes the frames of the video sequence. The next temporal resolution level is $\boldsymbol{T}_2 = \{I[2^2 \cdot i]\}$. In general, $\boldsymbol{T}_t = \{I[2^t \cdot i]\}$, where $t = 0, 1, \cdots, \log_2 \mathcal{G}$.

$\boldsymbol{T}_j$ depends on $\boldsymbol{T}_{j+1}$ except, obviously, the lowest temporal resolution level $\boldsymbol{T}_{\log_2 \mathcal{G}}$, where all the frames are intra-coded (all of them can be independently decoded). This allows the decoder: (i) to access any GOF of the compressed video without decoding the rest and (ii) to avoid the error propagation when real-time transmissions are carried out over error-prune transmission links.

The MCTF design of FSVC is a Motion Compensated block based system which differs from other common schemes found in the literature [4, 2]. Figure 2 shows how the frames at each temporal resolution level are predicted. Inspecting Figure 1 it can be seen that the Motion Estimation is done in the image domain and the Motion Compensation is performed in the wavelet domain, choosing the correct phase and using the same M motion field for the same location at each spatial resolution. Every transformed frame is decomposed into a set of non-overlapped blocks that are predicted from the previous and the next frame in the lower temporal resolution level. For instance, frame 4 that belongs to $\boldsymbol{T}_2$ is predicted with frames 0 and 8 that belong to $\boldsymbol{T}_3$. Therefore, every block can be backward or forward predicted. The choice between a forward (F-block) or a backward (B-block) prediction is decided according to the MSE (Mean Square Error), and taken into account the minimization of the drift errors. Drift errors propagate over dependencies between predicted frames. Thus, for predicted frame 1, forward predictions have higher priority to be selected than backward predictions, because at the decoder, frame 0 (where every block is intra-coded) will be reconstructed without drift error. After subtracting the prediction frames P to the predicted ones I, a sequence of residue frames E is generated for each temporal resolution level. Note that all the blocks of $\boldsymbol{T}_{\log_2 \mathcal{G}}$ are intra-coded. The intra-coded blocks can be used in other temporal resolution levels when the MSE of the residue is not low enough.

The temporal decorrelation is performed in the wavelet domain with the aim of: (i) avoiding the artifacts in the reconstructions when spatial scalability is used [1] and (ii) minimizing the disgraceful blocking artifacts that are visible at low bit-rates. The motion compensated wavelet blocks are constructed selecting the correct phase (overcomplete DWT) to avoid the shift variability of the DWT [6].

The frame residues are compressed with EBCOT and the motion fields with a static 0-order probabilistic model with a Huffman coder. EBCOT produces a sequence of JPEG2000 packets that are placed in the stream using some ordering. The receiving ordering is important because it determines the way the video will be displayed when only a partial decoding is carried out. In a *progressive by quality* scenario, FSVC decoder must choose the LTRCP progression of JPEG 2000, where L stands for quality layer, T for temporal resolution level, R for spatial resolution level, C for color component and P for precinct. Other useful progressions are RLTCP and TLRCP that allow *progressive by resolution* and *progressive by frame-rate* reconstructions.

3 Closed-loop FSVC

In practical cases, the FSVC decoder decompresses only a part of the stream generated by the encoder, depending on the available bandwidth. Consequently, residues $\hat{E}$ and frames $\hat{I}$ at the decoder are only an approximation of the original residues and frames at the encoder (see Figure 1). As predictions P depend on reconstructions, a drift error appears in the decoder.

By means of the dyadic MCTF scheme of FSVC explained in Section 2, drift is not accumulated along the time. This has two advantages: (i) the number of temporal resolution levels is smaller than the size of the GOF and therefore, the drift is small, and (ii) the drift is spread along the GOF.

To know how much coding efficiency is lost due to drift, a closed loop has been included in the encoder to ensure that both encoder and decoder use the same predictions, removing completely the drift error at a selected bit-rate. FSVC was designed without update step and preserving the temporal dyadic decomposition of MCTF. This technique allows FSVC encoder to establish open-loop (OL) or closed-loop (CL) prediction step in the lifting scheme. From a block diagram point of view, CL-FSVC is quite similar to OL-FSVC. The MC module of CL-FSVC uses the reconstructed frames at the decoder for a given bit-rate k (see the dashed lines in Figure 1) instead of the original frames used by OL-FSVC (see the dotted lines in Figure 1). Therefore, the drift error disappears when reconstructing video sequence at the bit-rate k (where $\hat{E}$ and $\hat{I}$ are identical at encoder and decoder). The FSVC decoder is

the same for OL-FSVC and CL-FSVC.

4 Experimental Results

A set of experiments have been carried out to analyze the effects of open and closed-loop schemes on the coding efficiency of FSVC. The "progressive by quality" decoding scenario has been chosen because it is the most interesting for most of the VoD applications. The coding parameters used to run OL-FSVC and CL-FSVC are:

- Spatial Filter: Biorthogonal 9/7. Spatial Resolution Levels: 4.
- Temporal Filter: Bidirectional 1/1 (open-loop for OL-FSVC and closed-loop for CL-FSVC). Temporal Resolution Levels: 5.
- Motion Compensation: Fixed block-size with 1/1 Pixel Accuracy.

Each GOF is composed of 16 frames (4 temporal resolution levels). Each color component is encoded using 16 quality layers and 4 spatial resolution levels. Finally, 1 precinct per resolution level has been used. The video codestream has been decompressed using the LTRCP progression at several bit-rates.

The results presented in Figure 3 are for the well-known video test sequences *akiyo*, *bus* and *coastguard*. Similar results, that are not shown in this paper, were obtained for other test sequences. Figure 3 shows the rate-distortion evaluation in order to compare OL-FSVC and CL-FSVC. The Y-axis represents the average PSNR for the luminance component of the complete video sequence. The X-axis shows the decoding bit-rate. The closed-loop prediction of CL-FSVC encoder has been set to $k = 896$ Kbps (Kilobits per second).

Results demonstrate that CL-FSVC obtains slightly better video reconstructions from low bit-rates to the known *a priori* k Kbps. The highest coding gain is obtained at k Kbps and the improvement is smaller than 0.5 dB. At higher bit-rates CL-FSVC performs worse than OL-FSVC because the decoded frames are similar to the original video and the prediction frames have more quality at the decoder than at the CL-FSVC encoder. Finally, Figure 4 shows some reconstructed frames at k Kbps. A subjective comparison indicates that there is no visual difference between the frames decoded with CL-FSVC and OL-FSVC. Note that k Kbps is the decoding bit-rate where CL-FSVC obtains the highest coding gain.

5 Conclusions

This paper describes a fully scalable video codec (FSVC) based on MCTF and JPEG2000. FSVC provides fine granularity on temporal, quality and spatial scalabilities. Two different schemes of FSVC encoder with open and closed-loop have been designed and tested to investigate their coding efficiency and behavior. Experimental results with standard video sequences demonstrate that CL-FSVC only outperforms OL-FSVC around the bit-rate selected for the closed-loop. The coding and visual gain is not significant and CL-FSVC performs worse at high bit-rates. Therefore, it can be concluded that open-loop FSVC performs similar to closed-loop FSVC, and has less complexity because OL-FSVC encoder does not use the decoding closed-loop.

References

[1] Y. Andreopoulos, M. van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens, and J. Cornelis. Fully-scalable wavelet video coding using in-band motion compensated temporal filtering. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 417–420, 2003.

[2] Peisong Chen and J.W Woods. Bidirectional MC-EZBC with lifting implementation. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:1183–1194, 2004.

[3] S.-J. Choi and J.W. Woods. Motion compensated 3-D subband coding of video. *IEEE Transactions of Image Processing*, 8(2):155–167, 1999.

[4] L. Luo, F. Wu, S. Li, Z. Xiong, and Z. Zhuang. Advanced motion threading for 3D wavelet video coding. *Signal Processing: Image Communication, Special Issue on Subband/Wavelet Video Coding*, 19(7):601–616, 2004.

[5] J.-R. Ohm. Three-dimensional subband coding with motion compensation. *IEEE Transactions on Image Processing*, 3:559–571, 1994.

[6] M.J. Shensa. The discrete wavelet transform: weeding the Á Trous and Mallat algorithms. *IEEE Transactions on Signal Processing*, 40(10):2464–2482, 1992.

[7] D. Taubman. High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7):1158–1170, 2000.

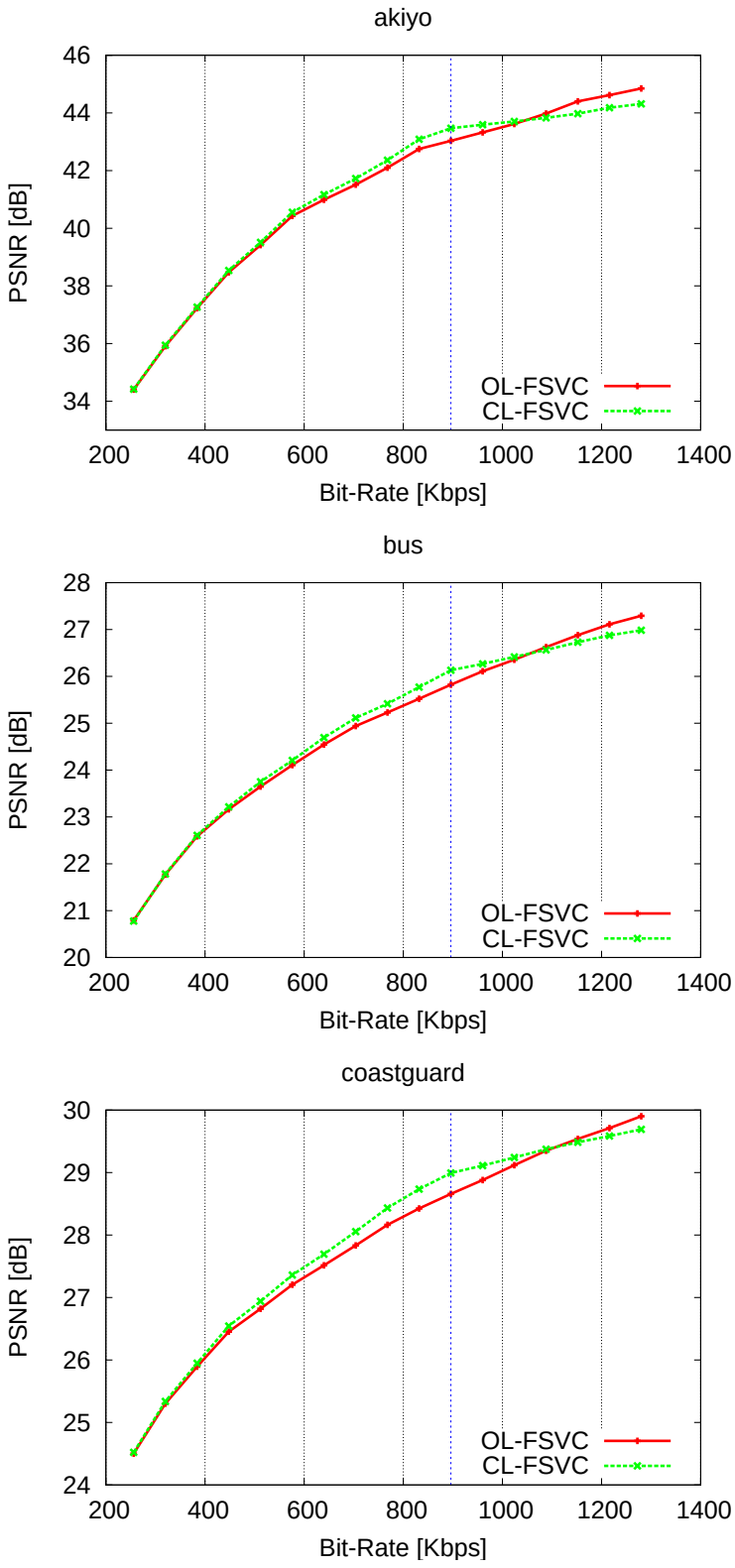

Figure 3: Average PSNR of the luminance component for *akiyo*, *bus* and *coastguard* video sequences. OL-FSVC and CL-FSVC are compared. The reference images in CL-FSVC have been decoded at $k = 896$ Kbps.

Figure 4: Visual results for the third image of the *akiyo*, *bus* and *coastguard* video sequences at 896 Kbps. Left: OL-FSVC. Right: CL-FSVC.