Low-Complexity TTCM Based Distributed Video Coding Architecture

J.L.Martínez[1], W.A.C. Fernando[2], W.A.R.J. Weerakkody[2], J. Oliver[3], O. López[4],
M. Martinez[4], M.Pérez[4], P.Cuenca[1] and F.Quiles[1]

[1] Albacete Research Institute of Informatics
Universidad de Castilla-La Mancha
02071 Albacete, Spain
{joseluismm,pcuenca,paco}@dsi.uclm.es

[2] Center for Communications Research
University of Surrey
Guildford GU2 7XH, United Kingdom
W.Fernando@surrey.ac.uk; rajithawee@yahoo.com

[3] Dep. of Computer Engineering.
Technical University of Valencia. Spain
joliver@disca.upv.es

[4] Dept. of Physics and Computer Engineering.
Miguel Hernández University. Spain
{mmrach,otoniel,mels}@umh.es

Abstract. Distributed Video Coding (DVC) is a promising coding solution for some emerging applications, where the encoder complexity, power consumption or memory requirements are constraint the system resources. Current approaches to DVC focus on improving the performance of the Wyner-Ziv coding by improving the quality of the reconstructed side information or by improving the quality of channel codes. Up to date, no attention has been paid to the problem of key frames coding where a low-encoding complexity scenario is also needed. This work focuses on key frames coding in its effect to the Wyner-Ziv frames decoding aiming to implement a very low-complexity Turbo Trellis Coded Modulation (TTCM) based DVC architecture. In this paper, we propose a new key frame coding scheme which has very low complexity and memory requirements for the TTCM based distributed video codec. Results show that the proposed intra frame codec for key frame coding outperforms the JPEG2000 and the Intra H.264 AVC codecs in terms of encoding-time and memory requirements, with better RD performance.

Keywords: Distributed Video Coding, Low Complexity, TTCM codes.

1 Introduction

Nowadays, with emerging applications such as multimedia wireless sensor networks, wireless video surveillance, disposable video cameras, medical applications

and mobile camera phones, the traditional video coding architecture is being challenged. For all the applications mentioned above there is need to have a low complexity encoder probably at the expense of a high complexity decoder. For these emerging applications, Distributed Video Coding (DVC) seems to be able to offer efficient and low-complexity encoding video compression.

DVC is a new video coding paradigm which allows among other things shifting complexity from the encoder to the decoder. The theoretical framework and the guidelines for DVC were established by Slepian-Wolf [1] and the current work in this field is based on the work by Wyner-Ziv [2]. Based on this theoretical framework, several turbo coded DVC codecs have been proposed recently [3,4,5]. In [3,4] the authors have proposed a turbo coded based Wyner-Ziv codec for motion video using a simple frame interpolation. In [5] the authors proposed a more sophisticated motion interpolation and extrapolation techniques [5] to predict the side information. The majority of these well-know research works on DVC have been carried out using a Turbo Wyner-Ziv codec. However, recent experimental results [6] show that the Turbo Trellis Coded Modulation (TTCM) based DVC codecs can improve the PSNR up to 6dB at the same bit rate with less memory compared to the Turbo Coded DVC codecs.

Current practical schemes developed for DVC are based in general on the following principles: the video frames are organized into two types; Key frames and Wyner-Ziv frames, while the key frames are encoded with a conventional intraframe codec, the frames between them are Wyner-Ziv encoded. At the decoder, the side information is obtained using previously decoded key frames and Wyner-Ziv frames.

In this context, most of the contributions given in the literature focus on improving the performance of the Wyner-Ziv coding by improving the quality of the reconstructed side information [5] or by improving the quality of channel codes [7]. Up to date, no attention has been paid to the problem of key frames coding where a low-encoding complexity scenario is also needed. The most current approaches to DVC rely on key frames available at the decoder perfectly reconstructed (lossless-compression) or encoded with conventional intra-frame codecs (lossy-compression). Recently, in the DISCOVER European project, JPEG2000 and Intra AVC have been proposed as technologies for the key frames coding [8]. However, these conventional intraframe encoders are too complex to be implemented in a DVC low-complexity scenario.

For this reason, this paper present a DVC architecture based on TTCM codes for the Wyner-Ziv frames as proposed in [6] and on LTW for the key frames as proposed in [9]. This paper is an integration and evaluation of these two architectures. In particular, the main objective of the paper is to propose a DVC codec with very low complexity and memory requirements for the non-DVC portion of an integrated TTCM based DVC architecture with very low complexity.

The rest of the paper is organized as follows. Section 2 introduces the TTCM based distributed video coding architecture with very low complexity key-frame coding. In Section 3, we carry out a performance evaluation of the proposed architecture in terms of memory requirements, computational complexity and rate-distortion. We compare the performance of our proposal to the JPEG2000 and Intra AVC proposals. Finally, in section 4 conclusions are drawn.
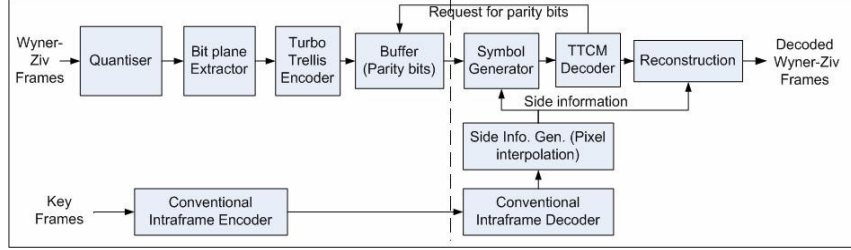
Figure 1. DVC Architecture using Turbo Trellis Coded Modulation (TTCM).

2 LOW-COMPLEXITY TTCM BASED DVC ARCHITECTURE

2.1 Wyner-Ziv Frames Coding

The considered Distributed Video Coding architecture is showed in Figure 1. The odd frames {X1, X3 ...} are the Wyner-Ziv frames which go through the interframe encoder to generate the parity sequence to be transferred to the decoder. The Wyner-Ziv frames are first passed through the 2M level quantizer where the level M is an independently varied parameter based on the expected quality of output and the available channel bandwidth. Next, the Slepian-Wolf based encoder incorporates the bit plane extractor and then the turbo trellis encoder. Each rate ½ component encoder of our implementation has a constraint length K=M+1 = 4 and a generator polynomial of (11 02) in octal form. A Pseudo-random interleaver is used in front of the 2nd constituent encoder. Only the parity bit sequence thus generated is retained in the parity buffers and the systematic bits are discarded.

The decoder generates the side information using the Key-frames employing a pixel interpolation algorithm as below:

$$Y_m(i, j) = \tfrac{1}{2}\left(X_{m-1}(i, j) + X_{m+1}(i, j)\right)$$

(1)

This side information together with the parity bits passed from the encoder, upon request, form the PSK symbols to be processed in the TTCM (Turbo Trellis Coded Modulation) decoder. A multi level set partitioning is done with the constellation mapping of the TCM symbols in order to maintain the maximum Euclidian distance between the information bits. Where ever parity bits are not available due to puncturing being effective, the symbol automatically reduces to a lower modulation level. In the implementation under discussion, a combination of 4 PSK and Binary-PSK is used based on the availability of the parity bits for the constellation mapping.

As commonly understood, Trellis Coded Modulation is conceptually a channel coding technique used to optimize the bandwidth requirements of a channel while protecting

the information bits by increasing the size of the symbol constellation. Our effort is to exploit the high coding gain and the noise immunity inherent in this technique.

A block diagram of the Turbo-TCM decoder implementation is shown in Figure 2. A symbol based MAP algorithm is used in the turbo trellis decoder which is run for 6 iterations as a complexity performance trade-off. A modification was done to the branch metric calculation to take care of the independent distributions of side information and parity bits. The parity bits are supplied to the decoder through an "on-demand" approach using a reverse channel for passing the request to the parity buffer maintained in the encoder. The de-puncturer function in the decoder basically watches the parity availability and manipulates the symbols fed to the SISO based MAP decoder accordingly. A reconstruction function is used to smoothing some adverse effects in the output sequence including some contribution by the quantization noise.

On the other hand, the side information generated by the temporal interpolation of two key frames is assumed to be a form of the original Wyner-Ziv frame subjected to noise. The probability distribution of this noise was a part of the detailed study. It was noticed that both the Gaussian noise distribution and the Laplacian noise distribution resembled the interpolation noise with distinct variance parameters. However, most interestingly, it was noted that our implementation of the codec was not susceptible to error by sub-optimal approximations of the distribution for the purpose of taking the results; an Additive White Gaussian Noise (AWGN) with variance 0.125 was assumed. To obtain more details about this, see [6].
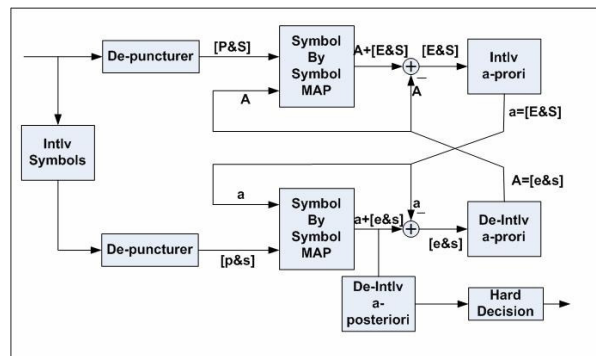

Figure 2. Block Diagram of TTCM Decoder

2.2 Key Frames Coding

Little attention has been paid in the literature to the problem of key frames coding and most of the current approaches to DVC rely on key frames available at the decoder perfectly reconstructed (lossless-compression) or key frame coding using conventional intra frame codecs, such as JPEG2000 or AVC intra. In this work, we propose the use of the LTW (Lower-Tree Wavelet) compression algorithm [9], for Key-Frames encoding in order to be integrated in TTCM based DVC architecture

with very low complexity key-frame coding. LTW is based on the efficient construction of wavelet coefficient lower trees. The main contribution of the LTW encoder is the utilization of coefficient trees, not only as an efficient method of grouping coefficients, but also as a fast way of coding them. Thus, it presents state-of-the-art compression performance, whereas its complexity is lower than the conventional intraframe codecs. Fast execution is achieved by means of a simple two-pass coding and one-pass decoding algorithm. Moreover, its computation does not require additional lists or complex data structures, so there is no memory overhead.

With LTW, the quantization process is performed by two strategies: one coarser and another finer. The finer one consists in applying a scalar uniform quantization, Q, to wavelet coefficients. The coarser one is based on removing the least significant bit planes, *rplanes*, from wavelet coefficients. The use of coefficient trees structure called lower tree reduces the total number of symbols needed to encode the image, decreasing the overall execution time. This structure is a coefficient tree in which all its coefficients are lower than $2^{rplanes}$.

The LTW algorithm consists of two stages. In the first one, the significance map is built after quantizing the wavelet coefficients (by means of both Q and *rplanes* parameters). In Figure 3(b) we show the significance map built from wavelet decomposition shown at Figure 3(a). The symbol set employed in our proposal is the following one: a *LOWER* (L) symbol represents a coefficient that is the roots of a lower-tree, the rest of coefficients in the lower-tree are labeled as *LOWER_COMPONENT* (*) but they are never encoded because they are already represented by the root coefficient. If a coefficient is insignificant but it does not belong to a lower-tree because it has at least one significant descendant, it is labeled as an *ISOLATED_LOWER* (I) symbol. For a significant coefficient, we simply use a symbol indicating the number of bits needed to represent it.

With respect to the coding algorithm, in the first stage (symbol computation), all wavelet subbands are scanned in 2×2 blocks of coefficients, from the first decomposition level to the Nth (to be able to build the lower-trees from leaves to root). In the first level subband, if the four coefficients in each 2×2 block are insignificant (i.e., lower than $2^{rplanes}$), they are considered to be part of the same lower-tree, labeled as *LOWER_COMPONENT*. Then, when scanning upper level subbands, if a 2×2 block has four insignificant coefficients, and all their direct descendants are *LOWER_COMPONENT*, the coefficients in that block are labeled as *LOWER_COMPONENT*, increasing the lower-tree size.

However, when at least one coefficient in the block is significant, the lower-tree cannot continue growing. In that case, a symbol for each coefficient is computed one by one. Each insignificant coefficient in the block is assigned a *LOWER* symbol if all its descendants are *LOWER_COMPONENT*, otherwise it is assigned an *ISOLATED_LOWER* symbol. On the other hand, for each significant coefficient, a symbol indicating the number of bits needed to represent that coefficient is employed.

Finally, in the second stage, subbands are encoded from the LL_N subband to the first-level wavelet subbands, as shown at Figure 4. Observe that this is the order in which the decoder needs to know the symbols, so that lower-tree roots are decoded before its leaves. In addition, this order provides resolution scalability, because LL_N is a low-resolution scaled version of the original image, and as more subbands are being received, the low-resolution image can be doubled in size. In each subband, for each

2×2 block, the symbols computed in the first stage are entropy coded by means of an arithmetic encoder. Recall that no *LOWER_COMPONENT* is encoded. In addition, significant bits and sign are needed for each significant coefficient and therefore binary encoded.
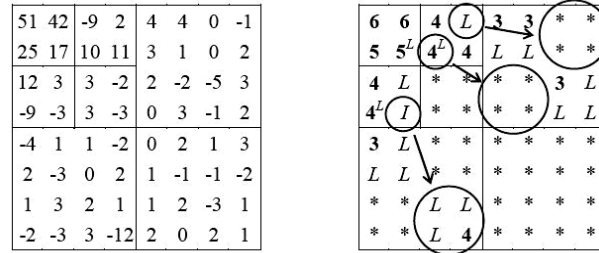

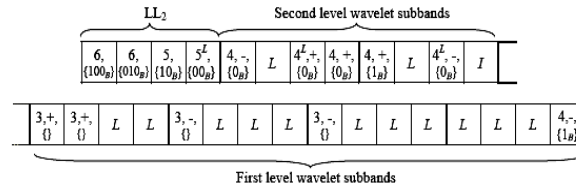**Figure 3. (a) 2-level wavelet transform of an 8x8 example image. (b) Map Symbols**

Figure 4. Coefficient-trees in LTW

3   RESULTS

In this section, we carry out a performance evaluation of the Low-Complexity TTCM Based Distributed Video Coding Architecture proposed in section 2, in terms of memory requirements, computational complexity and rate-distortion. For the purpose of this performance comparative evaluation, even frames were intra coded with LTW, JPEG2000 or H.264 (Baseline Profile, the fastest version of AVC Intra), and decoded while odd frames are coded as Wyner-Ziv frames, as shown in Figure 1.

The bit rate and PSNR are calculated for the luminance of the Wyner-Ziv frames (odd frames), or the Key-Frames (even frames) of the Foreman sequence (300 frames), for a frame size of 176x144 (QCIF) with a Wyner-Ziv frame rate of 15fps.

For a better comparative performance of rate-distortion function, we also show the average PSNR difference (ΔPSNR) and the average bit-rate difference (ΔBitrate). The PSNR and bit-rate differences are calculated according to the numerical averages between the RD-curves derived from LTW encoder, JPEG2000 and H.264 encoders, respectively. The detail procedures in calculating these differences can be found from a JVT document authored by Bjontegaard [10]. Note that PSNR and bit-rate

differences should be regarded as equivalent, i.e., there is either the decrease in PSNR or the increase in bit-rate, but not both at the same time.

For the purpose of our performance evaluation, we first evaluate the key frames coding part of our low-complexity DVC architecture, and then we evaluate the Wyner-Ziv frames coding part of our low-complexity DVC architecture. Finally, global results (taking into account all frames) will be provided

For the key frames coding part, all the evaluated encoders have been tested on an Intel Pentium M Dual Core 3.0 GHz with 1Gbyte RAM Memory. We have selected H.264 [11] (Baseline, JM10.2), JPEG2000 [12] (Jasper 1.701.0) and LTW, since their source code is available for testing. The correspondent binaries were obtained by means of Visual C++ (version 2005) compiler with the same project options and under the above mentioned machine. A further evaluation can be found in [13]

Figure 5 shows the average memory requirements[1] per key frame for all key frame codecs under study and for QCIF and CIF size formats. In both cases LTW needs practically half the memory than JPEG2000 and H.264 needs six times the memory of LTW for QCIF size and eight times for CIF size.
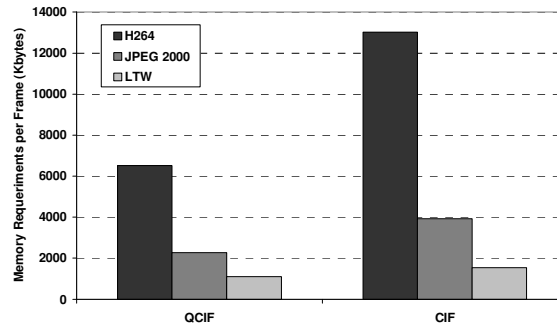
Figure 5. Memory Requirements (Key Frames)

Figure 6 shows the average encoding time per key frame for all key frame codecs under study for QCIF Foreman video sequence. As shown in Figure 6, LTW codec has the lowest complexity of all evaluated codecs and is about 10 times faster than JPEG2000 and 100 times faster than H.264 (Baseline profile, the fastest version of AVC Intra). LTW codec reduces the complexity substantially with respect the other conventional codecs under study showing the effectiveness of the LTW codec in the proposed Low-Complexity TTCM Based Distributed Video Coding Architecture.

Figure 7 shows the RD results for key frames for all key frame codecs under study. For a fair comparison, first, the H.264 simulations were carried out by varying the QP factor from 20 to 50. For every simulation the real bit-rate was obtained and then it was introduced to JPEG 2000 and LTW codecs as target bit-rates. As shown, the Rate-Distortion obtained with LTW outperforms the other codecs by 1.2 dB and 1.13 dB on average, with respect to JPEG2000 and H.264 respectively, using less bit-rate, up to 17% and up to 10% with respect to JPEG2000 and H.264 respectively (see Table 1).

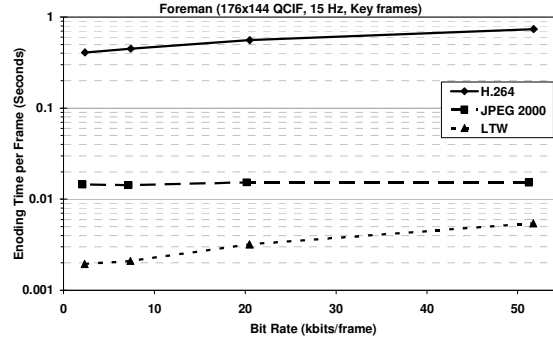[1] Results obtained from Windows XP task manager, peak memory usage column.
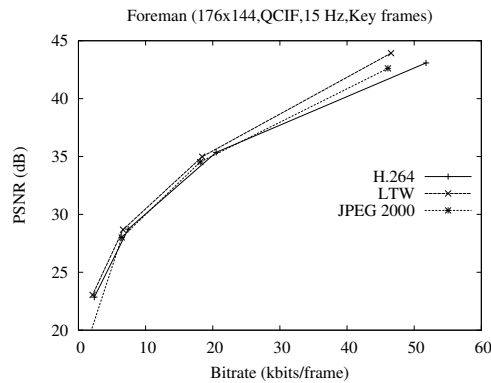
Figure 6. Encoding Time per Key Frames.

Figure 7. Rate-Distortion Results for Key Frames.

Table 1. Comparison for Key Frames Coding

JPEG 2000 vs. LTW		H.264 vs. LTW	
ΔPSNR (dB)	ΔBitrate(%)	ΔPSNR (dB)	ΔBitrate(%)
-1.271	17.11	-1.133	10.70

For the Wyner-Ziv frames coding part, Figure 8 shows the effect on the Wyner-Ziv (WZ) frames decoding when key frames are coded with all key frame codecs under study with QP=20. The best results for Wyner-Ziv frames are obtained when key frames are coded with LTW codec. As seen in Table 2, the Rate-Distortion results obtained on average for Wyner-Ziv frames when key frames are coded with LTW codec outperforms the other codecs by 0.5 dB and 0.6 dB on average, with respect to JPEG2000 and H.264 respectively, using significant less bit-rate, up to 47% and up to 46% with respect to JPEG2000 and H.264 respectively.

Figure 9 shows the effect on the Wyner-Ziv frames decoding when key frames are coded with all key frame codecs under study with QP=50. Again, the best results for Wyner-Ziv frames are obtained when key frames are coded with LTW codec. As shown in Table 3, the Rate-Distortion results obtained for Wyner-Ziv frames when

key frames are coded with LTW codec outperforms the other codecs by 4.5 dB and 0.4 dB on average, with respect to JPEG2000 and H.264 respectively, using significant less bit-rate, up to 2768% and up to 9.6% with respect to JPEG2000 and H.264 respectively. These results may seem erroneous but there is an explication: the side information is generated from the key frames and this side information has an important impact over the general performance of DVC. A lack of PSNR, which is shown by Figure 7, for the key frames denotes a lack of RD performance for the DVC scheme, shown in figure 9.
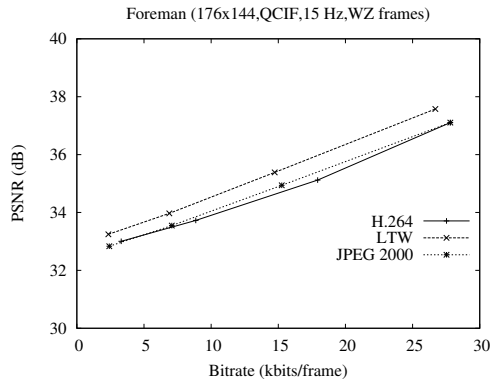
Foreman (176x144,QCIF,15 Hz,WZ frames)

Figure 8. Rate-Distortion Results for Wyner-Ziv Frames.
(When Key Frames are coded with QP=20)

Table 2. Comparison for WZ Frames Coding when Key Frames are coded with QP=20

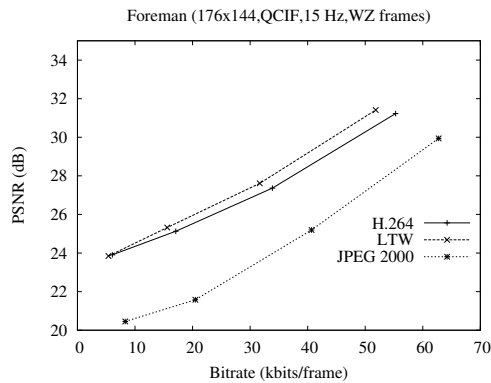| JPEG 2000 vs. LTW		H.264 vs. LTW	
ΔPSNR (dB)	ΔBitrate(%)	ΔPSNR (dB)	ΔBitrate(%)
-0.51	42.07	-0.662	46.52

Foreman (176x144,QCIF,15 Hz,WZ frames)

Figure 9. Rate-Distortion Results for Wyner-Ziv Frames.
(When Key Frames are coded with QP=50)

Finally we present global results taking into account all frames (key frames + Wyner-Ziv frames). Figure 10 shows the effect on all frames decoding for our Low-Complexity TTCM Based Distributed Video Coding Architecture when key frames are coded with all key frame codecs under study with QP=20. The best results are obtained when key frames are coded with LTW codec. As seen in Table 4, the Rate-Distortion results obtained using LTW codec outperforms the other codecs by 1 dB on average, with respect to JPEG2000 and H.264 approximately, using significant less bit-rate, around 20%.

Table 3. Comparison for WZ Frames Coding when Key Frames are coded with QP=50

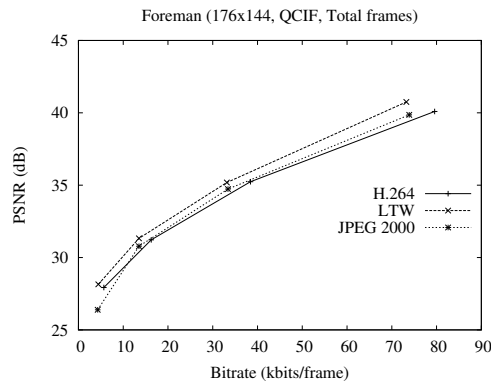| JPEG 2000 vs. LTW		H.264 vs. LTW	
ΔPSNR (dB)	ΔBitrate(%)	ΔPSNR (dB)	ΔBitrate(%)
-4.545	2768.17	-0.435	9.6

Figure 10. Rate-Distortion Results for all Frames.
(When Key Frames are coded with QP=20)

Figure 11 shows the effect on all frames decoding for our Low-Complexity TTCM Based Distributed Video Coding Architecture when key frames are coded with all key frame codecs under study with QP=50. Again, the best results are obtained when key frames are coded with LTW codec. As seen in Table 5, our proposal outperforms the other codecs by 3.9 dB and 0.7 dB on average, with respect to JPEG2000 and H.264 respectively, using significant less bit-rate, up to 105% and up to 10% with respect to JPEG2000 and H.264 respectively.

Table 4. Comparison for ALL Frames when Key Frames are coded with QP=20

JPEG 2000 vs. LTW		H.264 vs. LTW	
ΔPSNR (dB)	ΔBitrate(%)	ΔPSNR (dB)	ΔBitrate(%)
-1.011	24.76	-1.188	20.41

Although the results presented in this paper are only shown for QCIF format and for Foreman sequence, similar behavior was obtained for CIF format and for other video sequences.
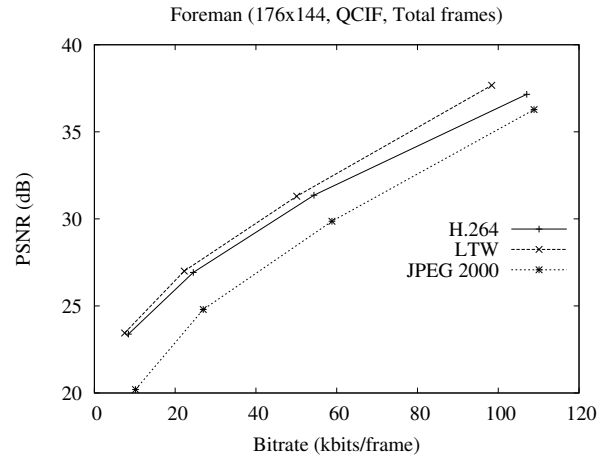

Foreman (176x144, QCIF, Total frames)

Figure 11. Rate-Distortion Results for all Frames.
(When Key Frames are coded with QP=50)

Table 5. Comparison for ALL Frames when Key Frames are coded with QP=50

JPEG 2000 vs. LTW		H.264 vs. LTW	
ΔPSNR (dB)	ΔBitrate(%)	ΔPSNR (dB)	ΔBitrate(%)
-3.913	105.67	-0.744	10.65

4 CONCLUSIONS

In this paper, we have proposed a very low-complexity Turbo Trellis Coded Modulation based DVC architecture. In particular, we have proposed the use of a fast intra frame codec, with very low complexity and memory requirements, in order to be implemented for the non-DVC portion of a TTCM based DVC codec. Results clearly indicate that the use of LTW intra frame codec on a TTCM based DVC architecture outperforms the same architecture when JPEG2000 or the Intra AVC codecs are used in terms of encoding-time and memory requirements, showing very similar RD performance.

References

1. Slepian, D and Wolf, J.K.: Noiseless Coding of Correlated Information Sources. Information Theory, IEEE Transaction on **19** (1973), 471–480.
2. Wyner, D and Ziv, J.: The Rate-Distortion Function for Source Coding with Side Information at the Decoder. Information Theory, IEEE Transaction on **22** (1976) 1–10.
3. Aaron, A., Zhang, R. and Girod, B.: Wyner-Ziv Coding of Motion Video. Proceeding of Asilomar Conference on Signals and Systems, Pacific Grove, USA (2002).
4. Girod, B., Aaron A., Rane, S. and Monedero, D.R.: Distributed Video Coding. Advances in Video Coding and Delivery, IEEE Special Issue on **93** (2005) 1–12.
5. Ascenso, J., Brites, C. and Pereira, F.: Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding. 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, (2005).
6. Weerakkody, W. A. R. J., Fernando, W. A. C., Adikari, A. B. B. and Rajatheva, R. M. A. P.: Distributed video coding of Wyner-Ziv frames using Turbo Trellis Coded Modulation. Proceedings of International Conference on Image Processing, ICIP'06, Atlanta, USA (2006).
7. Wang, H., Zhao, Y. and Wang, A.: Performance Comparisons of Different Channel Codes in Distributed Video Coding. First International Conference on Innovative Computing, Information and Control, ICICIC'06, Vol 2, Beijing, China (2006) 225–228.
8. Pereira, F., Guillemot, C., Leonardi, R., Ostermann, J., Ebrahimi, T. and Torres, L.: Distributed Coding for Video Services. DISCOVER Project Deliverable 7 (2006) .
9. Oliver, J., and Malumbres, M. P.: Low-Complexity Multiresolution Image Compression Using Wavelet Lower Trees. Circuits and Systems for Video Technology, IEEE Transactions on **1051–8215** (2006) 1437–1444.
10. Bjontegaard, G.: Calculation of Average PSNR Differences between RD-Curves. 13th VCEG-M33 Meeting, Austin, USA (2001).
11. ISO/IEC International Standard 14496-10:2003: Information Technology – Coding of Audio - Visual Objects – Part 10: Advanced Video Coding.
12. JPEG2000 Image Coding System. ISO/IEC 15444–1, 2000.
13. López, O., Martínez-Rach, M., Piñol, P., Oliver, J. and Malumbres, M.P.: M-LTW: A Fast and Efficient Non-Embedded Intra Video Codec. Pacific-Rim Conference on Multimedia, PCM'07, (2007). Accepted – publication pending.