FEEDBACK FREE DVC ARCHITECTURE USING MACHINE LEARNING

J.L. Martínez[1], *G. Fernández-Escribano*[1], *H. Kalva*[2], *W.A.R.J.Weerakkody*[3], *W.A.C.Fernando*[3], *A.Garrido*[1]

[1]Albacete Research Institute of Informatics. Universidad de Castilla-La Mancha. 02071 Albacete, Spain
[2]Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL, USA
[3]Center for Communications Systems Research, University of Surrey, GU2 7XH, United Kingdom

ABSTRACT

Most of the reported Distributed Video Coding (DVC) architectures have a serious limitation that hinders its practical application. The uses of a feedback channel between the encoder and the decoder require an interactive decoding procedure which is a limitation for applications such as offline processing. On the other hand, the decoder needs an efficient way to estimate the probability of error without assuming the availability of the original video at the decoder. In this paper we continue with our previous works into a more practical DVC architecture which solves both problems based on the use of machine learning. The proposed approach is based on extracting the relationships that exist between the residual frame and the number of requests over this feedback channel. We apply these concepts to pixel-domain Wyner-Ziv coding demonstrating significant savings in bitrates with a little loss of quality

***Index Terms*—** DVC, Wyner-Ziv coding, Machine Learning, Feedback Channel, Turbo Codes.

1. INTRODUCTION

DVC is a technique used to reduce the asymmetry in video codecs; the processing complexity of the encoders is reduced, leading to a low-cost implementation, while the majority of the computations are taken over by the decoders. The applications that are expected to benefit from this video coding architecture include wireless video surveillance, multimedia sensor networks, disposable video cameras, medical applications and mobile cameras phones. The theoretical framework of DVC was developed by Slepian-Wolf (SW) [1] for lossless Distributed Source Coding (DSC) and by Wyner-Ziv (WZ) [2] for the lossy case.

One of the pioneering DVC approaches is the turbo based WZ coding scheme presented in [3], where the encoder is responsible for exploring the source statistics, and therefore achieving compression following the WZ paradigm. Most of the architectures available in the literature are based on [3] and make use of a reverse channel to the encoder requesting more information for the decoding process; this approach is referred to as *feedback based* architecture in the literature. Moreover, all architectures which are based on [3] use an ideal error estimation at the decoder which is able to measure the distortion between the original frame and the partial reconstructed one. The original frames are used to determine when a sufficient number of parity bits are received, in order to guarantee a residual BER (bit-error rate) below a given threshold, typically set to be equal to 10^{-3}. This feature is also well-known in the literature and is referred to as *rate control at the decoder*. Such feedback based architectures offers significant problems in practical scenarios where the encoded video streams need to be stored for offline processing, and also when a bidirectional communication channel is not available. Feedback mechanics also have implications in terms of delay and decoder complexity since several iterative decoding operations may be needed to decode the data to a target quality level. Moreover, the ideal capacity inherent in these architectures is impractical for real implementations due to the dependence of the decoder on the original frame which is only available at the encoder. Basically, the two main drawbacks of current DVC architectures are the dependence on a feedback channel and the ideal correction detection based on original frames. Therefore, a solution without these disadvantages is essential to develop practical applications using DVC.

In our previous work [4] we demonstrate that Machine Learning (ML) based approaches work well and have the potential to enable feedback free DVC architectures. The previous paper reported a framework that uses decision trees only to determine when a feedback to the encoder is necessary, i.e. a half-feedback approach. In this paper we present a full feedback free DVC architecture focus on eliminating the two main drawbacks mentioned above using a block based turbo encoder in Pixel Domain which exploits the correlation between the residual frame and the number of requests (bits parity) for each block.

Also, in this work we use lossy coding for key frames instead of a lossless case used in the previous work which means a more realistic scenario. It is important to stress that

---

This work has been jointly supported by the Spanish MEC and European Commission FEDER funds under grants "Consolider Ingenio-2010 CSD2006-00046"' and "TIN2006-15516-C04-02"', and by JCCM funds under grant "PAI06-0106".

the proposed video codec does not include any of the limitations which are many times present in WZ papers, notably those adopting this type of WZ architecture [3]. This means, for example, that no original frames are used at the decoder to create the side information or to measure the bitplane error probability.

This paper is organized as follows: Section 2 presents a very fully overview of the non-feedback architectures available in the literature. In Section 3 and 4, we present the major contribution of this paper: feedback channel free solution to DVC. Section 5 presents experimental results. Finally, conclusions and some future work topics are presented in section 6.

2. RELATED WORK

In most DVC codecs available in the literature the decoder has the ideal capacity to measure the distortion between the partial reconstructed frame and the original one. This theoretical assumption is going to be impractical since the original frame is only available at the encoder. On the other hand, most of the literature on DVC is based on feedback channel to rate allocation and on a Rate-Compatible Punctured Turbo Code (RCPTC). In this configuration, the turbo encoder generates all the parity bits for the bitplanes to be encoded, saves these bits in a buffer, punctures and divides them into parity bits sets. These sets are sent to the decoder in response to requests via feedback channel. The task of the turbo decoder is to correct the side information mismatches / errors using these parity bits sent by the encoder. To solve some of these problems some approaches have been recently published in the literature.

The first attempt to solve the feedback problem was proposed in 2005 by *Artigas et al* in [5] where the authors try to determine the rate at the encoder based on empirical results obtained by examining different test sequences. *Adikari et al* in [6] propose a Unidirectional DVC codec based on two parallel Wyner-Ziv encoders using different interleaves index for each one in order to scatter the original image and the bit rate is set by the user. This architecture was enhanced employing an iterative decoding with gradually updated side information using spatial-temporal predictions. *Morbée et al* in [7] propose an algorithm based on a faster estimation at the encoder side of the optimum rate calculated in the ideally turbo decoder [3] from the parameters: σ^2 (correlation noise) and P_k (error probability) to be calculate for each bitplane. *Brites et al* in propose a transform domain architecture employing a faster method to calculate a closer approximation of the side information at the encoder side [8].

All these approaches try to determine at the encoder side, the proper number of bits that would be necessary in an ideal feedback based architecture such as [3]. In a different way to these architectures, our approach walks in the line of low-complexity encoder devices and we do not increase the encoder complexity compared to the rest of literature. On the other hand, all these architectures deal with the frame as a whole and they do not take into account the different characteristics of the motion level that can be appear into a frame such as covered / uncovered areas, noise effects or illuminations changes which significantly affect on the performance of the non-feedback based architectures. In order to solve it, our approach is based on block by block basis.

3. PROPOSED ARCHITECTURE

Our proposed architecture is based on the feedback based Pixel Domain architecture proposed in [3] which is depicted in Figure 1. However, the encoder rate control block has been added with respect to the proposed one in [3].

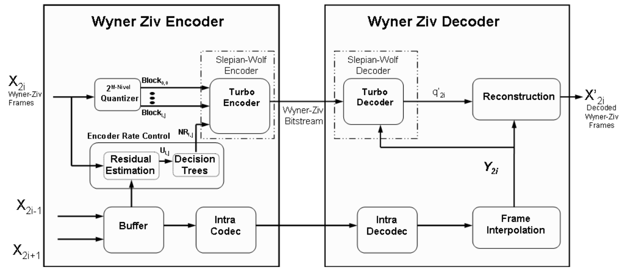

Figure 1. Video Codec Architecture Proposed

The encoder rate control is formed by the Residual Estimation and a decision tree. The Residual Estimation block generates the residual information using the previous and passed key frames and the current WZ frame. The decision tree provides a low complexity method of estimating the proper number of parity bits.

The first modification over the architecture proposed in [3] is the introduction of a block by block basis scheme instead of a frame by frame approach. The motivation behind this decision is to focus on the parity bits to correct the blocks / areas with poor estimates at the decoder (in the side information generation). The traditional architectures assume that the quality of the side information is constant across the whole frame. Because of varying motion characteristics and the amount of detail, the estimation of the original frame at the decoder is not the same for all areas of the frame. The second modification is the use of Machine Learning to exploit the correlation between the residual video frame and the number of requests over an imaginary feedback channel. The Machine Learning tools are only used to get the decision tree form the statistics extracted from the residual frame.

In order to get the residual frame that is used by the decision tree encoders must generate the residual available at the decoder. To reduce the encoder complexity, a fast approximation of the residual can be generated instead. The average operation between adjacent key frames has been adopted in this paper instead of more complex techniques using in the literature [9]. The residual frame is obtained by subtraction operation between the current WZ frame and this faster side information estimation. From this residual frame we need to extract some solid statistics to estimate how good is the side information at the decoder and, therefore, estimate the number of bits (or number of requests) which will be necessary to correct these errors in the side information.

There are two main factors considered in determining the block size. Higher block lengths yield better turbo coding performance. On the contrary, the use of a smaller block size significantly improves the estimated side information. Moreover, smaller block sizes allow us more granularity in order to estimate different rates for different parts of the frame. The effect of the codec's overall performance with varying block size was analyzed and based on the experiment, a compromised block length of 16 pixels (4 x 4 blocks) is used in the simulations.

4. DATA MINING TO ESTIMATE THE PARITY BITS

In the framework of machine learning, a decision tree is made by mapping the observations about a set of data in a tree made of arcs and nodes. For each 4x4 block, the proposed algorithm uses the mean and the variance of 2x2 sub-blocks, the 4x4 average, the 4x4 mean, and the variance of the 2x2 sub-blocks means. The corresponding number of requests for this block is also used in the file as determined by the ideal feedback based architecture reference [3] using the same block size. The goal is achieved by making use of some residual statistics calculated for each block and exploiting the correlation between the numbers of request and block statistics. The open source WEKA [9] data mining tool is used to discover a pattern of residual information for the number of request over a feedback based architecture. The training set was made with the first frame of some sequences in order to capture the different aspects which can be appear in the video (low / high motion, lower / high definition, etc.) in QCIF format. The number of requests was extracted using an implementation of DVC [3] with block by block coding and a reverse channel with lossless key frames coding. By experimenting with a set of video sequences we found that one frame of each sequence is sufficient to capture the distribution of the residual frame and the decision tree made with this training set was able to make accurate rate control decision for all tested sequences. Based on this training file, the WEKA data mining tool was used to create a decision tree (Figure2) with a set of rules for using the residual information of each block. The J4.8 algorithm is based on the C4.5 algorithm proposed by Ross Quinlan.

The residual images are computer per bitplane and used in training with WEKA. The residual data thus varies depending on the number of bitplanes considered. Thus decision trees are trained and used separately for each bitplane. The first WEKA node is used to check for the ZERO vs. NON-ZERO requests. If a block is classified as NON-ZERO, a second decision level is used for selecting the different levels of puncturing for these blocks. The decision tree works as follows:

Node 1. The inputs for this node are all blocks of the current residual frame. In this node we use a tree decision generated with WEKA to decide whether the block residual is high or a low. The output of this node determines whether the side information is accurate (ZERO requests) or inaccurate (NON-ZERO requests). The tree has a simple decision based on 4x4 mean.

Node 2. The inputs for this node are the NON-ZERO blocks output by node 1. This node evaluates all different puncturing rates and selects the best option using the available residual information. The tree has a more complex decision based on all statistic information available.

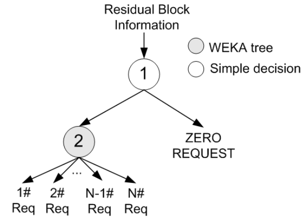

Figure 2. The decision Tree

Due to space constrains we cannot show the rules that are evaluated in the WEKA decision nodes. This decision node implements a decision tree that examines some statistical information of residual frame to arrive at the decision. Since the residual information depends on the quantization parameter used in H.264 Intra Coding for the key frames (which are employed in the residual frame estimation). In this first approximation we used loss key frames coding for the training process, in future extension of this work we will evaluate the tree depending of QP factor applied to key frames.

5. PERFORMANCE EVALUATION

In order to evaluate the proposed parity bits mode decision algorithm, we have implemented the proposed approach based on the DVC reference [3]. The metrics we are interested in are the *% of good choice* and the rate distortion function. Throughout our experiments, we have used various QCIF format video sequences exhibiting different special characteristics. We use a QP factor of 23 for key frames coding. The size of the GOP is 2; the even frames are WZ frames and the odd K frames.

Table 1 shows the *% of good choice* of our algorithm, that is, WEKA was used to get the decision tree and then, we evaluate the % of blocks that are correctly classified compared to the feedback based architecture.

The codec used in our experiments first decomposed each WZ frame into 4 x 4 blocks and then into its 8 bitplanes. Then, the *m* most significant bitplanes are separately encoded by using a RCPTC; the others bitplanes are discarded. In our experiments, *m* is chosen to be 2. The punctured is set to 10 which allows our algorithm to allocate parity bit multiples of 2 bits. Note that in the case of lossy K frames the threshold of the tree must be adjusted. This is however beyond the scope of this paper.

Table 1. % of good choice

Sequence	*1st MSB*	*2nd MSB*
CONTAINER	94.6372	93.123
SALESMAN	98.8013	95.4574
HALL MONITOR	96.0252	93.817
MOBILE	89.5899	77.918
AKIYO	99.4322	99.1167
CARPHONE	96.2776	92.429
CLAIRE	97.4763	96.2145
COAST	96.7823	91.5457
MISS	99.1798	97.8549
MOTHER	98.8644	97.2871
MEAN	96.70662	93.47633

Table 2 shows the bitrate savings of our approach compared to the feedback based [3] working in 4 x 4 blocks. The rate saving is due to the absence of feedback and requests for the blocks which are labeled as ZERO request; our approach does not spend any bits while the feedback based method needs to send the first chunk to determine if more requests are needed. On the other hand, the quality achieved by our approach is similar for 1 BP and for the next higher BPs, the quality drops slightly because of the block artifact caused by misclassified blocks. The blocks which the puncturing rate is overestimated for our decision tree offer an unnecessary increase of bitrate without a corresponding increase of quality. On the other hand the blocks which are lower estimated generate a block artifact problem which is depicted in Figure 3. This problem can be solved using fewer bins (each one which contains more requests) in the node 2 in the Figure 2 in order to do an overestimation of each block.

Table 2. Saving Rate

Saving the rate (kbits/frame)						
Sequence	Proposed		Feedback Based		% saving	
	1 BP	2 BP	1 BP	2 BP	1 BP	2 BP
akiyo	141.22	56.17	197.12	96.81	39.58%	72.34%
news	177.74	64.00	230.33	104.47	29.59%	63.23%
miss	77.85	37.74	140.80	81.25	80.87%	115.26%
claire	155.18	82.40	209.75	120.84	35.17%	46.65%

6. CONCLUSIONS

In this paper we present a feedback free architecture for DVC. Receiver feedback and the assumption of original data at the decoder are two key problems that make DVC impractical. This approach is based on constructing a faster encoder estimate of the decoder residual and determining the number of parity bits necessary. The parity bits are determined without any decoder feedback thereby reducing the bitrate and latency. Encoders use decisions trees obtained from supervised machine learning to determine the number of parity bits required by the decoders. Results obtained demonstrate significant bitrate savings with a little loss of PSNR. We are currently working on hierarchical binary trees to eliminate the artifacts due to misclassification.

Figure 3. Block Artifacts

7. REFERENCES

[1] D. Slepian, J.K. Wolf, “Noiseless coding of correlated information sources,” IEEE Trans. on Inform. Theory, vol. IT-19, pp. 471–480, July 1973.

[2] Wyner and J. Ziv, “The Rate-Distortion Function for Source Coding with Side Information at the Decoder”. IEEE Trans.on Inform. Theory, vol. IT-22, pp. 1–10, Jan. 1976.

[3].B. Girod , A. Aaron, S. Rane, D.R. Monedero, "Distributed Video Coding”. Proc. of IEEE Special Issue on Advances in Video Coding and Delivery, Vol.93, No. 1, pp.1-12, 2005.

[4] J.L. Martinez, C. Holder, G. Fernandez-Escribano, H. Kalva and F. Quiles, “DVC Using a Half-Feedback Based Approach”. Submitted to ICME 2008.

[5] X. Artigas, L. Torres, “Improved signal reconstruction and return channel suppression in Distributed Video Coding systems”, 47th International Symposium ELMAR-2005, Multimedia Systems and Applications, June 2005.

[6] A.B.B. Adikari, W.A.C. Fernando, W.A.R.J. Weerakkody, “Iterative Wyner-Ziv Decoding for Unidirectional Distributed Video Coding,” IEE Electronics Letters. Vol. 43, Issue 2, pp. 93 – 95, January 2007.

[7] M. Morbée, J. Prades-Nebot, A. Pizurica, and W. Philips, "Rate allocation algorithm for pixel-domain distributed video coding without feedback channel", ICASSP, April 2007, Volume 1, pp. I-521-I-524.

[8] C. Brites, F. Pereira, “Encoder rate control for transform domain Wyner-Ziv video coding”, ICIP, September, 2007.

[9] Ian H. Witten and Eibe Frank, “Data Mining: Practical machine learning tools and techniques”, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.