A DISTRIBUTED VIDEO CODER BASED ON THE H.264/AVC STANDARD

Simone Milani and Giancarlo Calvagno

Dept. of Information Engineering, University of Padova, Italy
email: {simone.milani,calvagno}@dei.unipd.it

ABSTRACT

One of the latest innovations in the world of multimedia technologies is the application of Distributed Source Coding (DSC) theory to the robust transmission of video sequences. However, these DSC-based video encoders are usually characterized by a lower compression gain with respect to their hybrid counterparts. In this work, we investigate achieving H.264-like high compression efficiency with a DSC-based approach without constraints on encoding complexity. In this way, highly non-stationary video data are modelled through coarse low-cost Motion Estimation, and this allows us to obtain a good compression efficiency together with a certain robustness to channel errors and losses. Experimental results show that in presence of losses the presented algorithm permits a better quality with respect to H.264/AVC.

1. INTRODUCTION

During the last years the aim of providing video and audio applications almost anywhere and anytime has lead to the widespread of multimedia services over heterogeneous networks. In these scenarios mobile communications have played a significant role since they allow end users to interact with great flexibility, but at the same time, they introduce the need of dealing with a higher percentage of losses and errors with respect to the traditional wired communications. As a consequence, the capability of providing reliable video communication in a heterogeneous scenario is the most relevant issue in the widespread and the diffusion of multimedia mobile services.

Unfortunately, current video codecs fail to satisfy this requirement because most of them are based on temporal prediction. Despite temporal DPCM makes possible to obtain high coding gains, it results to be inefficient whenever some of the information is lost. In this case, the state of the encoder can not be recovered until it is refreshed (i.e. the encoder codes a frame without any temporal prediction, called Intra frame). The presence of frequent Intra refresh leads to a waste of the available bandwidth since the compression gain of Intra coding is much lower than that of temporal DPCM.

During the last years, novel coding solutions that cope efficiently with these problems have been found, and the recent literature reports a wide number of different proposal that try to cope with the problems of transmitting a video sequence across a network affected by losses. Among these, a whole class of new coding paradigms based on Distributed Source Coding (DSC) theory were presented [1–3]. All these works utilize capacity-achieving channel codes to approach the Wyner-Ziv bound. This solutions require both a high decoding complexity and a long block length which can be applied to a very large area of the video frame[1]. This contradicts with the highly non-stationary nature of video data. In 2002, an approach known as PRISM (Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding) was proposed by Puri and Ramchandran for multimedia transmissions on wireless networks using syndromes [4]. The major goal of this solution is to join the traditional intraframe coding error robustness with the traditional interframe compression efficiency. After a series of improvements the original PRISM architecture has evolved until providing a coding performance which is comparable with that of H.263 [5]. The obtained compression efficiency, together with the low-complexity and the FEC nature which characterize this coding scheme, make the PRISM architecture an interesting field for investigation. In this work, a DSC-based coder that utilizes the coding units of H.264/AVC is presented. Its structure has been derived from that of PRISM despite the characteristics of H.264/AVC coder require significant modifications in the design of the entropy coder and in the characterization of reference blocks. In fact, the adoption of a 4×4 DCT transform modifies the characteristics of the coded signal in the transform domain. According to [6], a quad-tree based solution proves to be more suitable for coding the signal with respect to the more traditional run-length coding in a loss-free transmission. The current paper investigates how the solution proposed in [6] can be adapted to a transmission environment affected by errors.

Section 2 gives an overall description of the implemented DSC coder, while Section 3 characterizes the implemented decoder and the detection of false positive. Section 4 presents some experimental results for different transmission environments. Conclusions are shown in Section 5.

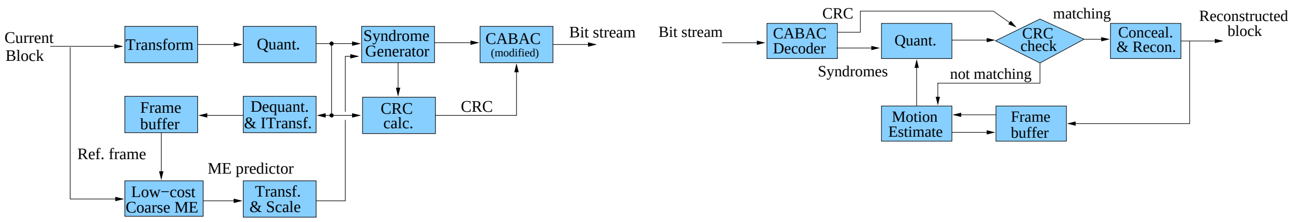
2. STRUCTURE OF THE PRESENTED DSC CODER

The proposed DSC coder relies on the basic structure of the H.264/AVC coder despite some of its building blocks were appropriately modified since the processed signal is not the difference between the current block and the motion-compensated prediction, but the least significant information in the original (quantized) block. Figure 1 depicts the block diagram of the proposed system.

In a hybrid coder, motion estimation identifies the best motion-compensated predictor according to a given distortion metric. Both the location of the predictor and the resulting DFD are transmitted to the decoder. The decoder then reconstructs the coded block generating the corresponding prediction through motion compensation and adding the decoded DFD.

In the DSC scheme, the encoder sends the indicator of side-information, which is a Cyclic Redundancy Check (CRC) equivalent to Motion Vectors (MVs), and the entropy-coded syndromes (generated according to the correlation). The coding units that significantly differ from the original

[1] Typically bit plane encoding is over an entire frame.



(a) DSC coder (b) DSC decoder

Figure 1: Encoder and decoder block diagrams. The key differences between the proposed DSC-based encoder and a H.264/AVC encoder are the syndrome generator and re-designed entropy coder.

H.264/AVC encoder are: (1) a CRC computation unit, (2) a syndrome generator, and (3) a modified entropy coder to better suit the probability distribution of syndrome values. This is to replace the H.264/AVC entropy coders, Context-Adaptive Variable-Length Coder (CAVLC) and Context-Adaptive Binary Arithmetic Coder (CABAC) as they were designed according to the statistics of the quantized transformed DFD.

We now describe these three coding units in more detail.

2.1 Syndrome generation

Given the original 4×4 block x and its motion-compensated prediction x_p, the H.264/AVC coder generates the residual DFD $r = x - x_p$, and transforms it into coefficients through a multiplication-free separable integer transform [7]. In the DSC approach, the original non-predicted block is transformed into coefficients X, which are quantized later into the levels X_q according to the rule

$$X_q(i,j) = sign(X(i,j)) \left\lfloor \frac{|X(i,j)| + O(i,j,QP,mb\_type)}{\Delta(i,j,QP,mb\_type)} \right\rfloor \quad (1)$$

where the quantization step $\Delta(i,j,QP,mb\_type)$ and the offset $O(i,j,QP,mb\_type)$ depend on the coefficient position (i,j) in the block[2], the Quantization Parameter QP, and the macroblock coding type mb_type. For the sake of simplicity, in the following paragraphs we will omit the indices i, j, QP, and mb_type.

For each quantized coefficient, the DSC coder has to compute how many most significant bits (MSBs) are correlated with the previous blocks and how many least significant bits (LSBs) are not. The correlated MSBs of each coefficients are aligned into a binary array and a CRC value is computed on them (see Fig. 2). This provides to the decoder a sort of *"signature"* or *"hash"* of the current block, which allows the decoder to identify a correct predictor among the blocks of the previous frames that have been correctly reconstructed. The LSBs are coded into *"syndromes"* Z, i.e. into pieces of information that allow the computation of those predictor blocks that are to be checked using the CRC. More specifically, given a certain quantizer characteristic (identified by the lattice Λ), each syndrome and its number of bits identifies a sub-quantizer or coset (denoted by the sub-lattice Λ_Z) according to the scheme reported in Fig. 3. The number

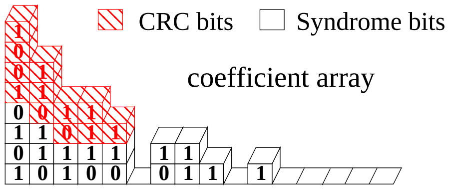


Figure 2: CRC and syndrome bits.

of bits n used for the syndrome Z identifies the quantization step $2^n\Delta$ of the corresponding sub-quantizer, while the value Z of the syndrome identifies its offset (see Fig. 3 and [4, 8]). The verification of the CRC allows the reconstruction of the coded block. In the proposed scheme, a syndrome has to be generated for each quantized transform coefficient based on the estimated correlation between the coefficient and its predictor. Previous works [4] adopted a classification rule based on the minimum square error (MSE) between the current block and the corresponding block in the previous frames at the same location. This classification rule proves to be more or less efficient according to the correlation existing between the MSE computed with no Motion Estimation (ME) and the MSE found after ME. In our case, the irregular and ever-changing statistics of transform coefficients in a 4×4 block makes this parameter ineffective for the purposes of classification. Moreover, since all the hybrid video coders which have been designed so far rely on an effective ME performed at the encoder, many DSP board are enabled with a powerful hardware accelerators that allow estimating a predictor block using a limited power supply. Therefore, in the presented coding scheme it is possible to take advantage of full-pel ME at the encoder in order to identify a predictor block x_p that allows the estimate of the correlation statistics. The prediction x_p is transformed into the coefficients X_p which are used to compute the number n of least significant bits for a quantized coefficient X_q through the equation[3]

$$n = \begin{cases} 2 + \left\lfloor \log_2 \left(\dfrac{\left|(X_q \cdot \Delta) - X_p\right|}{\Delta} \right) \right\rfloor & \text{if } d > \Delta \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where $d = \min\left\{\left|(X_q\Delta) - X_p\right|, \left|X - X_p\right|\right\}$, Δ is the quantiza-

[2] The adopted transform matrix is orthogonal but it is not orthonormal, and therefore, a rescaling is needed to compensate the amplification introduced at each frequency [7]. However, in our approach we kept the standard quantization matrix specified within the JM reference software.

[3] In our implementation, $M = 2^{14}/\Delta$ is added to each coefficient value to make it positive. The 2^{14} factor depends on the amplification of the 4×4 transform (6 bits) on the input signal (8 bits). For further details, see [7].

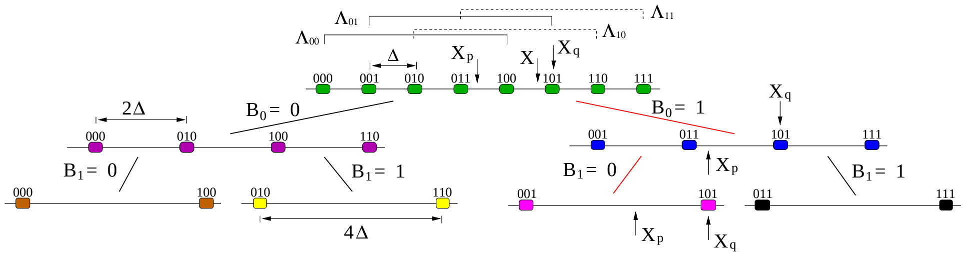


Figure 3: Partitioning of the integer lattice into 4 cosets or sub-quantizer according to the syndrome Z. The parameter Δ identifies the quantization step, X is the source, X_q is the quantized codeword and X_p is the side-information (omitting the spatial coordinates (i, j)). The number of levels in the partition tree depends on the correlation between $X_q\Delta$ and X_p given X.

tion step for that coefficient, and X_p is the corresponding unquantized transform coefficient of the predictor block. Given the value of n, the syndrome Z of X_q is

$$Z = X_q \ \& \ (2^n - 1) \tag{3}$$

where & denotes a bitwise AND operation. Note that when adopting a classification for the correlation structure, the number of the needed least significant bits is assumed known at both the encoder and the decoder according to the correlation class. In our approach, this information is transmitted together with the syndrome value mapping the combination of the length of the syndrome n and the syndrome Z itself to a unique symbol $S = 2^n + Z$.

Given a symbol S and the corresponding side-information X_p from motion compensation, the decoder can reconstruct the original quantized value X_q by selecting the point (codeword) in the sub-lattice Λ_Z that is the closest to the side-information X_p. Since these symbols are not equally likely, they should be entropy coded to achieve high compression efficiency. Here we present a quad-tree based arithmetic coder tailored to the distribution of these symbols.

2.2 Entropy coding of syndromes

Previous works have shown that the probability distribution function (pdf) of coded syndromes differs from the pdf of quantized transform coefficients. The statistics of syndrome S leads to an entropy value greater than that of the corresponding quantized transform coefficients for DFD signals. Therefore, a competitive compression performance is only possible by exploiting the correlation existing between the values of syndromes lying at adjacent frequencies. In [6] it was possible to obtain a good coding gain replacing the traditional run-length coding with a hierarchical quad-tree algorithm. Experimental results showed that the final performance of the DSC coder is comparable with that of the H.264/AVC coder. In our extension of the coder in [6], we have kept the same quad-tree structure but we modified the binarization strategy for syndromes S greater than 7 ($n > 2$). In the original approach, the same number of bits n is used for all the syndromes within a 2×2 sub-block. In this work, the number of bits for each syndrome can vary, a we adopted a VLC table to map each value S into a binary string.

2.3 CRC computation

As it was described before, the CRC value for a given block constitutes a sort of “signature” that allows the identification of a good predictor for the current block independently from which frames have correctly arrived. Therefore, its computation plays an essential role in the decoding process.

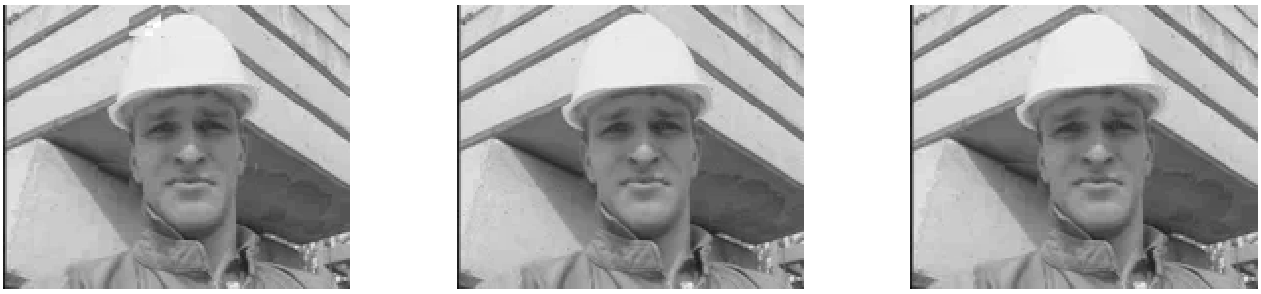
Unfortunately, the coding of CRC values in the transmitted bit stream is extremely expensive in terms of bit rate since CRC values can not be entropy coded easily and it is necessary to specify a sufficient number of CRC bits in order to avoid *“false CRC positive”* blocks. In our approach, we adopted a 16-bits CRC in order to reduce the percentage of false positive blocks that can be found in the search window. Therefore, the significant overhead that is related to this choice must be reduced specifying only one CRC value for each 16×16 block.

3. STRUCTURE OF THE DSC DECODER

One of the main differences that characterize the presented DSC architecture from the traditional H.264/AVC scheme is the inclusion of a ME unit at the decoder. As Figure 1(b) shows, the Motion Estimation block is used to generate a 16×16 prediction for the block that is decoded. Each 4×4 sub-block of the prediction is transformed and quantized according to the transmitted syndrome value. More precisely, given the symbol $S = 2^n + Z$ the decoder generates the reconstruction levels $R = (i + Z)\ \Delta'$, with $i = \ldots, -1, 0, 1, \ldots$ and $\Delta' = 2^n\ \Delta$. Each 4×4 block is then quantized and a corresponding CRC value is computed on the most significant bits of each quantized coefficients. In case the resulting CRC value matches the one signalled in the bit stream, the reconstructed block is dequantized and inversely-transformed. However, the CRC checking can give a positive result even if the current reconstructed block is significantly different from the one reconstructed at the decoder (false positive). In this case, a false-detection unit is necessary in order to discard false positive blocks according to the characteristics of typical video signals. In the following section, some of the techniques that were adopted for the detection of false positive blocks are described.

3.1 Detection of false positive blocks

In the proposed scheme, blocks that result extremely different can present the same CRC value after being quantized according to a given set of syndromes. Therefore, a discriminating algorithm must detect which blocks are to be discarded and which can be kept. In our approach we considered three different solutions that are presented in the following paragraphs.



(a) Simple distortion (PSNR = 30.23 dB) (b) Syndrome bits (PSNR = 36.73 dB) (c) Border distortion (PSNR = 34.08 dB)

Figure 4: Result of false positive detection for frame 1 of sequence `foreman`.

3.1.1 *Discrimination based on simple distortion*

After reconstruction, the decoder computed the distortion between the decoded block and its predictor. The final predictor will be that block that matches the CRC and minimizes the distortion.

3.1.2 *Discrimination based on estimating the correlation structure from syndromes*

The decoder computes again the number of bits that must be used to code each syndrome value S. All the blocks that do not match the CRC and the value n specified in the transmitted bit stream are discarded. Among the conforming blocks the decoder chooses the predictor that minimizes the distortion.

3.1.3 *Discrimination based on computing the distortion along borders*

The decoder computes the difference between the pixels of the reconstructed block and those of the neighboring previously-decoded blocks along the borders and chooses the predictor that minimizes the distortion.

Figure 4 reports frame 1 from the sequence `foreman` decoded using different false-positive detection algorithms. It is possible to notice that the approach based on minimizing the distortion between the reconstructed block and its predictor fails for one of the macroblocks in the first row (Fig. 4(a)). The reported PSNR values indicates that the most effective approach is the one that weighs the distortion considering the correlation between the reconstructed block and its prediction (Fig. 4(b)).

4. EXPERIMENTAL RESULTS

At first the performance of the proposed coding scheme was tested in a loss-free transmission environment. Both coders were implemented in the JM10.1 forcing the adoption of 4×4 transform only. The adopted entropy coder is the CABAC algorithm and a random Intra refresh of 11 macroblocks per frame is adopted in order to stop the error propagation in case some information is lost. Note that the Lagrangian optimization and the coefficient cancellation was disabled in order to allow a fair comparison between the two coders since similar algorithms should be designed and optimized for the PRISM coder too. Moreover we also assume that false-detection algorithm does not decrease the quality of the reconstructed sequence. Experimental results showed that the rate increment produced by the DSC coder depends on the adoption of CRC values in place of motion vectors. In fact, the resulting bit rate for DSC coded sequences is comparable with that obtained using H.264/AVC in case Motion Vectors (MVs) information is included (see Figure 5). Replacing MVs with CRC values

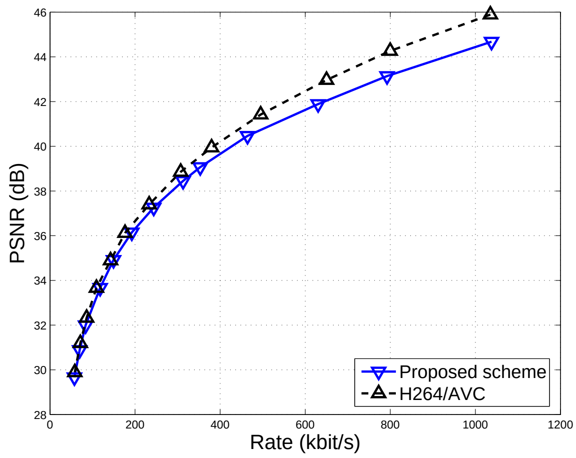


Figure 5: PSNR vs. rate for the sequence `foreman` using MVs instead of CRC values.

leads to a significant increment in the overall bit rate (see Figure 6). However, the adoption of CRC proves to be effective whenever testing these two coding schemes in a loss-prone environment. Figures 7 and 8 report the coding results for different sequences at different quality and loss probability 6%. It is possible to notice that even for significant loss probability the sequence reconstructed by the DSC coder is not affected by the quality degradation experienced by the H.264/AVC decoder. In fact, even if error concealment algorithms proved to be effective in estimating the lost information, a loss of 4 dB in PSNR can be noticed from Figure 7 at 630 kbit/s. Note also that the mismatch between the performance of DSC coder and that of H.264/AVC is significant for sequences that present rapid motion since the traditional concealment techniques fail in estimating the lost signal.

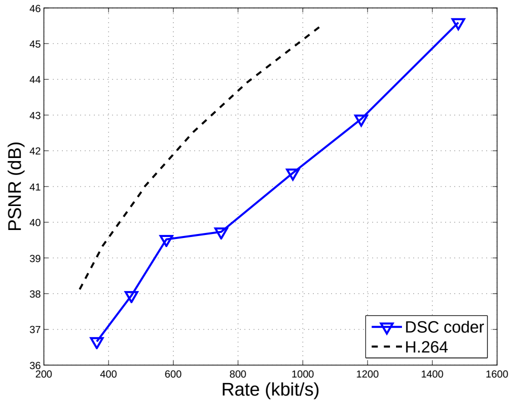


Figure 6: PSNR vs. rate for the sequence `foreman` using CRC over a channel with no losses.

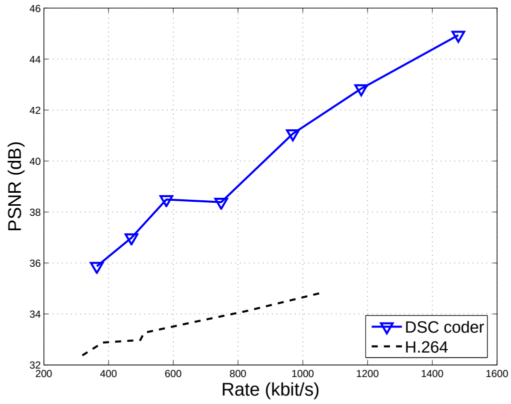


Figure 7: PSNR vs. rate for the sequence `foreman` using CRC over a channel with loss probability 6%.

5. CONCLUSIONS

The paper has presented an effective implementation of a DSC coder using the building blocks of the H.264/AVC coder. The proposed solution reuses the ME unit to classify each transform block and generates blocks of syndromes according to the chosen quantization step. Each syndrome is efficiently coded mapping subblocks of values into variable length strings through a hierarchical quad-tree strategy that has already proved to be significantly effective in previous works. In addition the DSC coder computes a CRC value on the most significant bits of quantized transform coefficients in order to characterize the coded block with a sort of "signature" or "hash". At the receiver, the DSC decoder takes into consideration different predictor blocks and discards some of them checking their CRC and minimizing the distortion introduced in the reconstructed sequence. Experimental results showed that the approach allows a robust transmission of video signals even if the channel is significantly affected by losses.

6. ACKNOWLEDGEMENTS

The work of Simone Milani was partially supported by the Foundation "Ing. Aldo Gini". The authors would like to thank Jiajun Wang for her precious advices and the details about PRISM coding architecture.

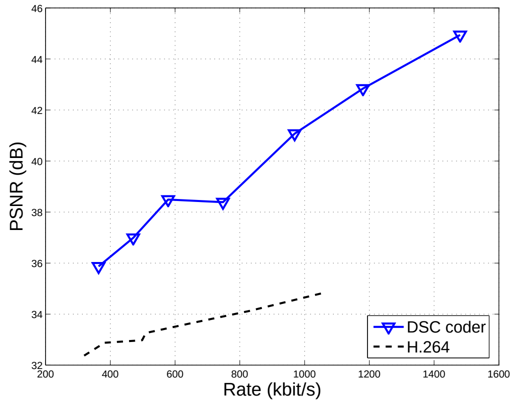


Figure 8: PSNR vs. rate for the sequence `news` using CRC over a channel with loss probability 6%.

REFERENCES

[1] A. Sehgal, A. Jagmohan, and N. Ahuja, "Scalable video coding using Wyner-Ziv codes," in *Proc. of the Picture Coding Symposium 2004*, San Francisco, CA, USA, Dec. 15–17, 2004.

[2] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed Video Coding," *Proc. of the IEEE, Special Issue on Video Coding and Delivery*, vol. 93, no. 1, pp. 71–83, Jan. 2005, invited Paper.

[3] Q. Xu and Z. Xiong, "Layered Wyner-Ziv video coding," in *Proc. of VCIP'04*, Jan. 2004.

[4] R. Puri and K. Ramchandran, "Prism: A new robust video coding architecture based on distributed compression principles," in *Proc. of the* 40^{th} *Allerton Conference on Communication, Control and Computing*, Allerton, IL, USA, Oct. 2002, pp. 402–408.

[5] A. Majumdar, R. Puri, P. Ishwar, and K. Ramchandran, "Complexity/performance trade-offs for robust distributed video coding," in *Proc. International Conference on Image Processing (ICIP)*, 2005.

[6] S. Milani, J. Wang, and K. Ramchandran, "Achieving H.264-like compression efficiency with distributed video coding," in *Proc. of VCIP 2007*, San Jose, CA, USA, Jan. 28 – Feb. 1 2007.

[7] A. Hallapuro, M. Karczewicz, and H. Malvar, "Low complexity transform and quantization - part I: Basic implementation (jvt-b038)," in *Joint Video Team,* 2^{nd} *Meeting*, Geneva, CH, Jan. 29 – Feb. 1 2002. [Online]. Available: ftp://ftp.imtc-files.org/jvt-experts/2002_01_Geneva/

[8] A. Majumdar, J. Chou, and K. Ramchandran, "Robust Distributed Video Compression based on Multilevel Coset Codes," in *Proc. of the Asilomar Conference on Signals, Systems, and Computers*, Nov. 2003.