Complexity Efficient Stopping Criterion for LDPC Based Distributed Video Coding

João Ascenso
Instituto Superior de Engenharia de Lisboa – Instituto de Telecomunicações
Rua Conselheiro Emídio Navarro, 1
1950-062 Lisboa, Portugal
+351 21 8418463
joao.ascenso@lx.it.pt

Fernando Pereira
Instituto Superior Técnico – Instituto de Telecomunicações
Av. Rovisco Pais,
1049-001 Lisboa, Portugal
+351 21 8418460
fp@lx.it.pt

ABSTRACT

In several distributed video coding architectures, a well-known complexity trade-off exists, where the low encoding benefits are paid with a higher decoding complexity. In a feedback channel based DVC architecture, the high decoding complexity is mainly due to the Slepian–Wolf decoding and the repetitive request-decode operation, especially when there is no initial encoder rate estimation or iterative motion refinement is employed. In this paper, an early stopping criterion for the LDPC syndrome belief propagation decoder is proposed that is able to reduce the number of decoding iterations. As a consequence, a significant reduction of the DVC decoder complexity can be observed with negligible losses in RD performance. The experimental results show reductions up to about 4 times in decoding complexity with a maximum of 0.15dB loss at high bitrates while for low and medium bitrates the RD performance loss is negligible.

Categories and Subject Descriptors

E.4 [**Coding and Information Theory**]: Data compaction and compression
I.4.2 [**Image Processing and Computer Vision**]: Compression (Coding) - Approximate methods

General Terms

Algorithms, Performance, Experimentation.

Keywords

Wyner-Ziv video coding, low density parity check codes, belief propagation, early stopping criterion.

1. INTRODUCTION

Distributed video coding (DVC) is currently a hot research topic in video coding since it opens new and interesting opportunities for emerging applications scenarios. In fact, several video applications can benefit from DVC, e.g. wireless digital video cameras, low-power video sensor networks and surveillance systems. Usually, these applications have requirements such as low complexity encoding or a flexible codec complexity distribution, robustness to packet losses, high compression efficiency, and sometimes also low latency/delay as well. In order to fulfill such requirements, a novel video coding paradigm was needed, since predictive video coding mostly targeted one-to-many applications with a high complexity encoder and thus it is not well matching the specific needs of these emerging applications.

The first practical DVC codecs appeared around 2002, following important advances in error correcting codes, especially codes with capacity near the Shannon limit, such as turbo and low-density parity-check (LDPC) codes. Since Slepian–Wolf coding is the core of a DVC codec, it assumes a central role not only in terms of RD performance but also in terms of codec complexity budget. For Slepian-Wolf coding, the two most efficient solutions, turbo codes and LDPC, have a performance close to the channel capacity. The LDPC codes have a wide range of tradeoffs between performance and decoding complexity and a similar encoding complexity can be achieved compared to the turbo codes with a careful LDPC code design [1]. For distributed source coding, regular and irregular LDPC syndrome codes were proposed in [2]; it was shown that the LDPC codes exhibit better results, for Gaussian and binary symmetric channels (BSC) correlation channels, when compared to turbo codes.

When efficient LDPC codes are used in a DVC context, one of the most important issues is how the LDPC code structure can be adapted to the varying statistics of the correlation noise, i.e. to the errors between the side information *Y* and the original data *X*. In some of the DVC codec architectures, it is the decoder responsibility to request for more parity/syndrome bits, when the attempt to decode the source given the available side information (SI) fails. In this case, the encoder replies to each request by sending more parity/syndrome bits which combined with the previous ones allow a higher likelihood of successful decoding. The LDPC codes which support this request-decode architecture are called rate-compatible [3][4]. Since the encoder does not know the correlation between the source and the SI, the encoder sends a small amount of parity/bits in order to achieve the minimum rate and avoid rate overestimation. This usually means that Slepian-Wolf decoding must be run several times for each coding unit (e.g. DCT band bitplane) and thus it is desirable to reduce its complexity.

In other DVC architectures, the encoder is responsible to estimate the syndrome/parity rate and send it at once to the decoder [5][6]. In such scenarios, the Slepian-Wolf decoder is run only once for each coding unit, but can also benefit from a reduced complexity Slepian-Wolf decoder. In this type of architecture [5], it is also critical the capability to iterate over a set of candidate predictors which are used individually as SI, in order to find the one that is able to decode the source successfully. In such scenario, the Slepian-Wolf decoder is run several times, one for each candidate

Mobimedia'09, September 7-9, 2009, London, UK.
Copyright 2009 ICST 978-963-9799-62-2/00/0004 ... $5.00

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7353
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7353

predictor and thus its complexity must be maintained as low as possible.

Considering the highly efficient LDPC codes, there is a need to minimize its decoding complexity, in order to obtain lower latency in a request-decode DVC codec architecture, or allow iterative decoding over a large number of predictors (thus obtaining high coding efficiency), in an architecture where the SI is found in a tentative way. So, the main objective of this paper is to develop techniques that are able to effectively reduce the LDPC codes decoding complexity.

In a Slepian-Wolf LDPC decoder, it is often used the popular log-domain sum-product algorithm (SPA) described in [2] for the decoding of LDPC syndrome codes. The SPA is an iterative decoding algorithm which uses an efficient convergence criterion [4] to stop when the source is successfully decoded. However, when the SI does not have enough quality for a given syndrome rate, the source is undecodable (i.e. unsuccessful decoding), and the iterative decoding only stops when fixed maximum number of iterations is reached. Therefore, it is highly desirable to have a stopping criterion which can detect undecodable sources as early as possible and stops the iterative decoding algorithm. In this way, the decoding complexity and delay can be significantly reduced since unsuccessful decoding occurs often, e.g. when not enough syndrome information is sent by the encoder in a request decoding DVC codec architecture. The proposed stopping criterion attempts to predict if there are enough syndrome bits for the SI quality by using decoding convergence metrics; this allows to significantly reduce the decoding complexity.

This paper is organized as follows: in Section 2, a brief overview of the Slepian-Wolf LDPC codec to be used is presented; in Section 3, the novel early stopping criterion is proposed and, in Section 4, the RD performance and the decoding complexity are evaluated. Finally, in Section 5, some final remarks are drawn.

2. SLEPIAN-WOLF CODEC

In DVC, different Slepian-Wolf codes can be used such as the block codes used in the Berkeley PRISM solution [5], the turbo codes used in the Stanford feedback channel based DVC solution [7] and the LDPC syndrome codes used in the solutions described in [4,8]. The LPDC codes are among the most efficient Slepian-Wolf coding solutions due to their capacity approaching performance for several communication channels [1] and for the distributed source coding case [2], outperforming the turbo codes. In this section, the LDPC syndrome codec used is presented.

2.1 LDPC Syndrome Codec Architecture

The architecture of the proposed LDPC based Slepian-Wolf codec is presented in Figure 1. The novelty proposed here relies on the added early stopping criterion module. The rate control is made by the decoder with the help of a feedback channel and the LDPC code makes use of the rate compatible strategy proposed in [4], named check node merging. The proposed LDPC based Slepian-Wolf encoder receives as input the source X_{ik} where i is the bitplane index for coefficient band k. The encoding process starts always with the most significant bitplane array with the frequency bands scanned in zig-zag scan order from the DC band to the high frequency AC bands. The Slepian-Wolf encoder works as follows:

1. ***LDPC Syndrome Encoder***: The LDPC syndrome code matrix H is used to calculate the syndrome S by $S = HX_{ik}$ which is then sent to an accumulator to generate the final syndrome information, in the same way as a LDPCA (accumulate) code [8]. The LDPC code graph H is built according to [4] which allows a higher efficiency when compared to previous state-of-the-art [4].
2. ***Buffer:*** The syndromes are stored in a buffer and sent to the LDPC syndrome decoder upon request, through the feedback channel. The syndromes transmission order is defined by a regular puncturing period Δ, which defines the granularity of the code rates which are incrementally obtained; it also indirectly defines the decoding complexity since a small puncturing period means that the LDPC decoder must run a high number of times to decode the source (i.e. to achieve the necessary bitrate).
3. ***CRC Generator:*** A cyclic redundancy check (CRC) code with a 8 bits polynomial is applied to each bitplane; the checksum is sent to the decoder in order to help the error detection process performed at the decoder.

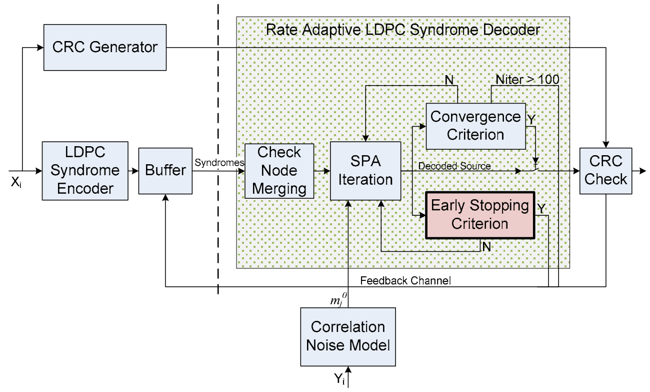


Figure 1. LDPC Syndrome Codec Architecture.

The core of the Slepian-Wolf decoding process is the SPA [9] which will be described in detail in Section 2.2. The decoding process can be described as follows:

1. ***Check Node Merging***: In this module, the LDPC code graph structure (defined by H) is modified, taking into account the number of syndrome bits received. Some check nodes of the LDPC base code graph (compression ratio equal to 1:1) are merged in order to obtain higher compression ratios according to a simple rule: any two check nodes are merged as long as they are connected by a punctured syndrome node (with 2 edges). This step is repeated when more syndrome information is received by the Slepian-Wolf decoder.

2. ***Sum-Product Algorithm (SPA) Iteration***: The SPA in each iteration attempts to decode the source with the received syndromes, the modified LDPC code graph and the soft-input information m_i^0 obtained from the correlation noise model. At the end, soft-output is obtained and the convergence criterion determines if the hard decision output has a small error probability (i.e. it is close to the original data).

3. ***Convergence Criterion***: First, the convergence criterion checks if all LDPC code parity-check equations are fulfilled for the decoded (hard decision) codeword; in the positive case, the LDPC syndrome decoder claims that the source is decoded and stops the iterative process. In the negative case (one or more parity-check equations failed), the iterative process continues and a further SPA iteration is performed if the number of iterations is not greater than a certain threshold; in this case, up to 100 iterations are performed (the same limit as in [4][8]). If this number of iterations threshold has been reached, the

decoder claims that the source is undecodable with the available syndrome bits and more syndrome bits are requested using the feedback channel.

4. ***Early Stopping Criterion:*** In parallel with the convergence criterion, the proposed early stopping criterion is also performed with a different objective: to detect the cases of undecodable sources at an earlier stage of decoding, i.e. before the maximum number of iterations is reached. Thus, it is avoided to waste decoding resources for unsuccessful decoding, lowering the overall LDPC decoding complexity. This early detection can also reduce the latency since the decoder can more quickly make a request to the encoder for more additional bits, if necessary. The early stopping criterion is thus able to predict decoding failures, by monitoring changes in the soft-output information of the SPA algorithm, and changes in the number of satisfied check node equations in the LDPC graph at the end of each iteration. When the early stopping criterion detects an undecodable source case, it requests from the encoder more syndrome bits. If this criterion fails, a decodable source is classified as undecodable and the consequence is rate overestimation leading to a reduction of the coding efficiency. Clearly, a tradeoff exists in this case between the decoding complexity reduction and the RD performance losses.

5. ***CRC Check:*** Finally, the source reconstructed by the LDPC syndrome decoder is checked for any residual errors left. For this, the same CRC used by the encoder is applied to the reconstructed bitplane. If the CRC checksum obtained matches the one received from the encoder, no remaining errors are corrupting the bitplane and the decoding of next band or bitplane can start; if there is no match more syndrome bits are requested using the feedback channel. This procedure guarantees a vanishing error probability (≈0) for each decoded bitplane with a small rate penalty (8 bits).

The rest of the decoding process, such as the reconstruction, SI generation, DCT/IDCT proceeds as in [10]. This means that a minimum mean-squared error reconstruction, a motion compensated frame interpolation framework for SI generation, a Laplacian correlation model and a 4×4 integer DCT transform are used (more details in [10]).

2.2 Sum-Product Algorithm

Besides the LDPC code definition, through matrix H, a rate adaptive LDPC syndrome decoder algorithm that provides near-optimal coding efficiency is also necessary. The most used algorithm comes with different names, such as sum-product (SPA), message passing or belief propagation algorithm [9]. For LDPC based syndrome codes, the decoding algorithm for distributed source coding was first presented in [2]. This solution will be described here with the necessary adaptations to DVC, since it provides the necessary basis for the early stopping criterion proposed in this paper.

The SPA algorithm is an iterative algorithm which exchanges messages between two types of nodes defined in the bipartite LDPC graph: the variable nodes (or v-nodes), which represent the codeword bits, the check nodes (or c-nodes), which represent the parity-check equations of the code's parity-check matrix H. The v-nodes are connected by edges to c-nodes according to H and some of the v-nodes are initialized with the syndrome bits receive from the encoder (usually called syndrome nodes). The SPA decoding algorithm operates in the log-domain and exchanges messages along the edges, which correspond to log-likelihood ratios (LLR).

Step 1: Initialization

Before the SPA decoding of a bitplane starts it is necessary to model the correlation noise (CN) between corresponding coefficients of the source X and SI Y. The CN model is used to convert the SI into the virtual channel LLR, m_j^0, the belief of being transmitted the source bit 0 or 1, according to:

$$m_j^0 = \log\left(\frac{P(x_{jk}^i = 1 | y_{jk}, x_{jk}^{i-1} \dots, x_{jk}^0)}{P(x_{jk}^i = 0 | y_{jk}, x_{jk}^{i-1} \dots, x_{jk}^0)}\right) \tag{1}$$

where y_{jk} correspond to the value of the DCT SI coefficient j in band k, x_{jk}^i corresponds to the source bit j in bitplane i of band k and the probability distribution P corresponds to the adopted Laplacian correlation noise model.

Step 2: Check-node operation

In the first half-iteration, the SPA algorithm takes into account the syndrome information received from the encoder. Since it is assumed that this information is received lossless (i.e. with no channel errors), some adaptations are needed for the DVC case when compared to the usual channel coding algorithm (the parity information can also have errors). At this step, it is calculated:

$$m_{lj}^c = \prod_{j'=1,\ j'\neq j}^{s+d_c-1} \alpha_{j'l}\, \phi\left(\sum_{j'=1,\ j'\neq j}^{d_c-1} \phi(\beta_{j'l})\right) \tag{2}$$

where m_{lj}^c is the message sent from c-node l to v-node j, s corresponds to the number of syndromes nodes connected to c-node l (it can be 1 or 2 for the chosen accumulator structure), d_c is the number of v-nodes connected to c-node l, $\alpha_{j'l} = sign[m_{j'l}^v]$, $\beta_{j'l} = |m_{j'l}^v|$, $\phi(x) = -\log(\tanh(x))$, $m_{j'l}^v$ is the message sent from v-node j' to c-node l. In the beginning of the first iteration, all messages $m_{j'l}^v$ are initialized with zero.

Step 3: Variable-node operation

In the second half-iteration of the SPA algorithm, each v-node calculates soft extrinsic information taking into account all the messages sent by its neighboring c-nodes (connected through edges) and the virtual channel LLR calculated in (1). At this step, it is calculated:

$$m_{jl}^v = m_j^0 + \sum_{l'=1,\ l'\neq l}^{d_v-1} m_{l'j}^c \tag{3}$$

where m_{jl}^v is the message sent from v-node j to c-node l, $m_{l'j}^c$ is the message received from c-node l' to v-node j and d_v is the number of edges of v-node j. Note that, in the first iteration, all messages $m_{l'j}^c$ are zero and thus the decoding algorithm only takes into account the virtual channel LLRs, m_j^0.

Step 4: Tentative decoding

After steps 2 and 3 (one full iteration), it is possible to estimate a novel decoded bitplane with:

$$L_j^{APP} = m_j^0 + \sum_{l=1}^{d_v-1} m_{lj}^c \tag{4}$$

$$\tilde{c}_j = \begin{cases} 0, \text{ if } L_j^{APP} < 0 \\ 1, \text{ if } L_j^{APP} \geq 0 \end{cases} \tag{5}$$

where $L_j^{APP}$is the *a posteriori* probability of the source bit j and $\tilde{c}_j$ corresponds to the estimation of the decoded bit j. With $\tilde{c}_j$, it is possible to evaluate if the source is successfully decoded or not by checking if $\tilde{c}H^T = 0$, i.e. if all parity check equations are fulfilled (see convergence criterion in section 2.1). In case $\tilde{c}H^T \neq 0$, a new iteration starts, i.e. equations (2)-(3) are repeated. The LDPC code performance using the SPA decoding algorithm depends: i) LDPC code structure: irregular graphs with short cycles are preferred; ii) codeword size: corresponds to the bitplane length; iii) code rate: bitplane compression ratio and iv) node degree optimization: number of edges in each c-node or v-node [1][4]. In the next section, the early stopping criterion of the rate adaptive LDPC syndrome decoder is presented.

3. EARLY STOPPING CRITERION

The SPA algorithm is a decoding algorithm which uses a convergence criterion at the end of each iteration to check if the source was successfully decoded or not. Since this may be a rather wasteful process from the decoding complexity point of view, it is proposed here to include an additional criterion to early stop the decoding process and obtain lower decoding complexity and lower latency for undecodable sources, when compared to the usual case where the maximum number of iterations is processed.

The proposed criterion is based on previous work on stopping criteria for turbo [11] and LDPC channel codes [12]. Since most of the existing stopping criteria are only proposed for channel coding (especially turbo codes), it is necessary to study, propose and evaluate a novel early detection criterion for distributed source coding (especially for LDPC syndrome codes). In [12], a LDPC code stopping criterion has been proposed which relies on the variable node reliability (VNR), a summation of the LLR values along all v-nodes. Although good results are obtained for channel coding [12], unsatisfactory results are obtained when the VNR criterion is applied to DVC.

3.1 Decoding Convergence Metrics

In this context, a novel criterion based on the *evolution* of the LLR values between decoding iterations and the number of satisfied parity-check equations at the end of each iteration is proposed here. Using these two novel metrics, it is possible to obtain a more robust criterion when compared to previous state-of-the-art [12] which can: i) early detect undecodable sources, and ii) minimize the number of cases where a source is wrongly classified as undecodable (in the first iterations) when it is in fact decodable (and thus rate overestimation occurs). In each SPA iteration, for decodable sources, the average of the magnitude of L_j^{APP} values increases while the number of unsatisfied parity-check nodes decreases. Thus, the proposed stopping criterion is based on two metrics:

1) The variation of the *a posteriori* log-likelihood L_j^{APP} ratios between successive iterations t and $t-1$, ΔL^{APP}according to:

$$\Delta L^{APP}(t) = \frac{1}{n}\sum_{j=0}^{n}\left|L_j^{APP}(t-1) - L_j^{APP}(t)\right| \tag{6}$$

where n corresponds to the source (bitplane) size. When the iterative SPA algorithm is converging to a successful result, L_j^{APP} increases between decoding iterations, leading to larger values of ΔL_j^{APP}. On the other hand, a source can be classified as undecodable when the SPA algorithm is 'stuck', i.e. with values of ΔL_j^{APP} close to zero (i.e. no variation).

2) The number of parity check equations (or c-nodes) unsatisfied at iteration t:

$$N^{UC}(t) = \sum_{p=0}^{m} a[p], \text{ with } a = \tilde{c}H^T \tag{7}$$

where m corresponds to the total number of c-nodes. The number of unsatisfied c-nodes N^{UC} is a metric which counts the number of 'erroneous' c-nodes, i.e. c-nodes for which the sum of the connected v-nodes hard decision values and syndrome node values is different from zero (equation not fulfilled). In a general way, when the source is decodable, N^{UC} drops between iterations until is equal to zero, at that point the SPA claims that the source is decoded. On the other hand, a source can be classified as undecodable when the value of N^{UC} is kept constant between SPA iterations.

3.2 Early Stopping Criterion Algorithm

With $\Delta L^{APP}(t)$ and $N^{UC}(t)$, it is possible to measure the LDPC syndrome decoding convergence in a robust way, i.e. it is possible to discriminate between a source that is decodable and one that is undecodable for a certain number of syndrome bits. Despite the robustness of these two metrics, there are two situations where they can fail, notably:

i) Oscillation state: in this state L_j^{APP} changes continuously between 0 (no variation, i.e. stuck) and values greater than zero (convergence) and N^{UC} changes between a constant value and an abrupt change. In these cases, it is very difficult to predict the outcome and thus the early stopping criterion should avoid make any decision.

ii) Slow convergence: in this case the SPA only converges after a high number of iterations, i.e. L_j^{APP} is close to zero and N^{UC} is kept constant for a certain number of iterations and then convergence occurs. In this case, the sources can be wrongly classified as undecodable and the stopping criterion should also avoid make a decision too early.

In the design of the early stopping criterion, two metrics are used in order to increase the robustness and a source is only considered undecodable when a 'stuck' behavior is observed for a certain amount of iterations. At the end of each SPA iteration t, the following early stopping criterion is performed:

Step 1: At the first SPA iteration ($t = 0$), $m\Delta L = 0$ and $mN = 0$.

Step 2: Calculate $\Delta L^{APP}(t)$ with (6). If $\Delta L^{APP}(t) < \lambda$, increment $m\Delta L$; otherwise, $m\Delta L = 0$.

Step3: Calculate $N^{UC}(t)$ with (7). If $N^{UC}(t) = N^{UC}(t-1)$, increment mN; otherwise, $mN = 0$.

Step4: If $m\Delta L \geq d$ or $mN \geq d$, classify the source as undecodable and stop the iterative SPA algorithm.

The counter $m\Delta L$ measures the number of iterations where small increments in L_j^{APP} are observed and the counter mN measures the number of successive iterations where the number of unsatisfied c-nodes is kept constant. In Step1, $m\Delta L$ and mN are initialized; in Step2 and Step3 the counters are incremented to indicate a 'stuck' decoding behavior; if this behavior is maintained, Step4 terminates the decoding of the source and more syndrome bits are requested.

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7353
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7353

By adjusting the thresholds λ and d, different tradeoffs between performance and decoding complexity can be achieved; experimentally, it was found that $\lambda = 0.5$ and $d = 6$ provide a good compromise between these two conflicting requirements.

4. EXPERIMENTAL RESULTS

In this section, it is evaluated the decoding complexity and the RD performance of the DVC codec in [4] including the early stopping criterion proposed in this paper. The LDPC decoder is evaluated in the context of a well-known DVC codec which follows the Stanford architecture [10]. In order to obtain meaningful results, four test sequences were selected with different types of characteristics (motion and texture): *Hall Monitor*, *Coastguard*, *Foreman* and *Soccer*. All sequences have QCIF spatial resolution and 15Hz. In all the experiments, a GOP length of 2 is used and only the luminance is coded. The key frames are H.264/AVC Intra main profile encoded and the decoded video has almost constant quality for the full set of frames (key frames and WZ frames). To obtain the RD curve, eight RD points are considered, each one corresponding to a 4×4 quantization matrix defined in [10]. The test conditions for the remaining modules are the same as in [10].

4.1 LDPC Syndrome Decoder Complexity

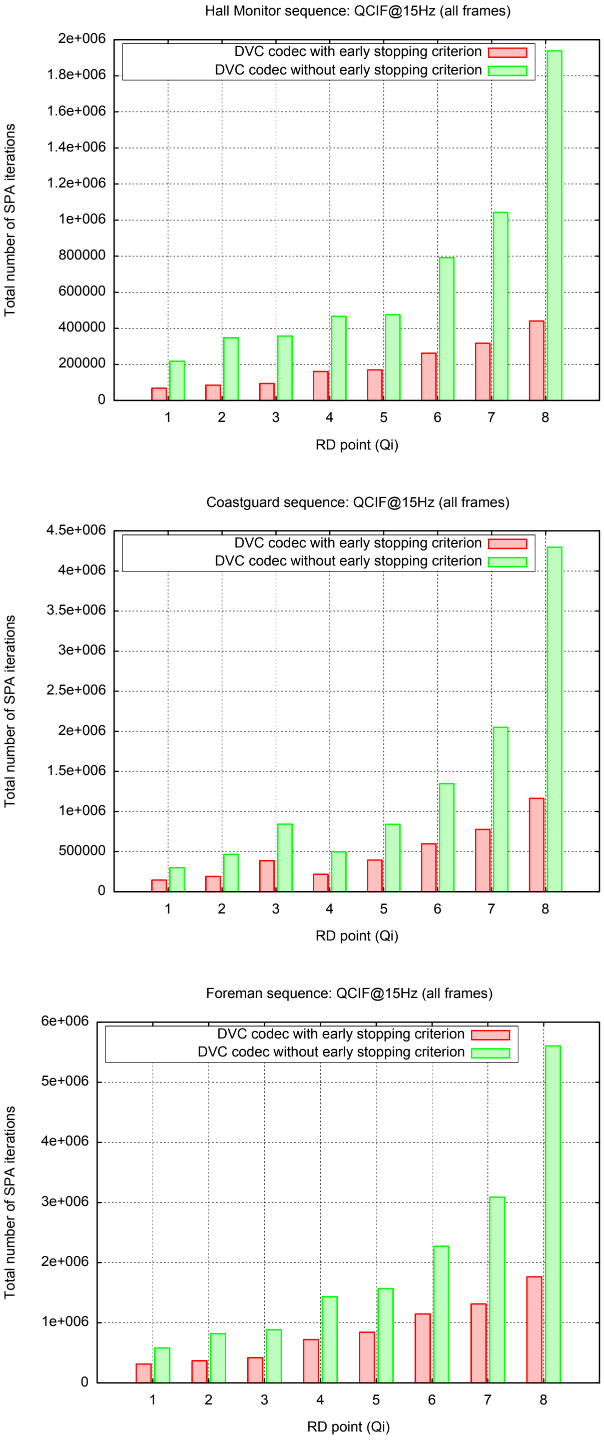
In order to assess the obtained complexity reduction for the Slepian-Wolf decoding process with the early stopping criterion, the total number of SPA iterations for each test sequence has been measured. With this metric, it is possible to assess the impact of the novel early stopping criteria in the LDPC syndrome decoder without any further external complexity contribution such as the SI generation, correlation noise model, reconstruction, etc. In Figure 2, the total number of requests is plotted in a bar chart with and without the proposed early stopping criterion. The x-axis shows the eight RD points considered.

Comparing the results for the LDPC syndrome decoder with and without the novel early stopping criterion, it is possible to observe significant savings in the total number of iterations (and thus decoding complexity) for all sequences and RD points. As expected, more significant savings are obtained for the last RD points which correspond to high qualities/bitrates, since the LDPC decoder is run for a higher amount of bitplanes and the correlation between the source and the SI is lower (and thus the number of requests is higher [10]). The best result is obtained for the *Hall Monitor* sequence where a 4.4 times reduction is observed for the last RD point. The worst result in terms of complexity reduction is obtained for the Soccer sequence with a 1.7 times reduction in the total number of SPA iterations for the first RD point. In a general way, low (and regular) motion sequences (e.g. *Hall Monitor*) have a more stable decoding behavior (easier to predict) when compared to high (and irregular) motion sequences (e.g. *Soccer*) where a more instable (oscillations) behavior is observed. This implies larger decoding complexity reductions for more stable content and vice-versa.

4.2 Overall DVC Decoder Complexity

The overall DVC decoder complexity was also evaluated with a simple although rather meaningful (if used in a relative sense) complexity metric, the decoding time for the full sequence in seconds. The main disadvantages of using the decoding time as a complexity metric are: i) it is highly dependent on the hardware/software platform, and ii) it is highly dependent on the degree of software optimization. However, in this case there is a minimal impact by these two factors since the goal is to compare the impact of the novel early stopping criterion which as a small footprint (and complexity) when compared to the whole DVC decoder, without changing the hardware platform and the remaining software. For these experiments, the hardware used was an x86 machine with a dual core Pentium D processor at 3.4GHz with 2048MB of RAM and Windows XP operating system.

Figure 3 show the decoder complexity measured in terms of decoding time, distinguishing the DVC decoder with and without the early stopping criteria. As expected, the overall decoder complexity follows the same trend as the Slepian-Wolf decoder complexity presented in the previous section; thus, the conclusions taken before are also valid here.



Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7353
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7353

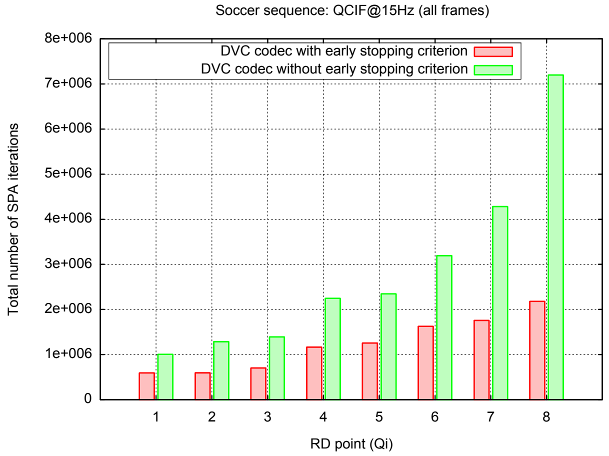
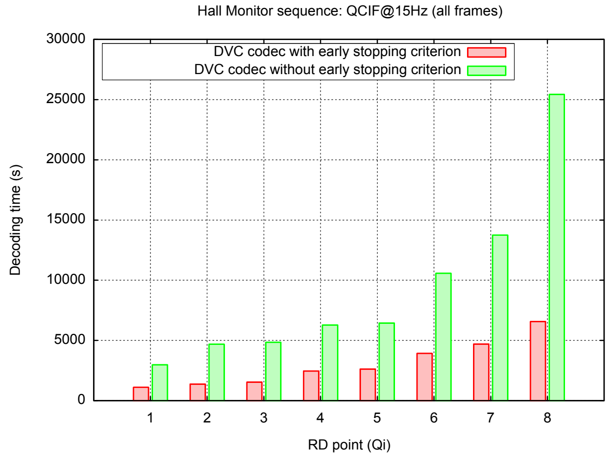
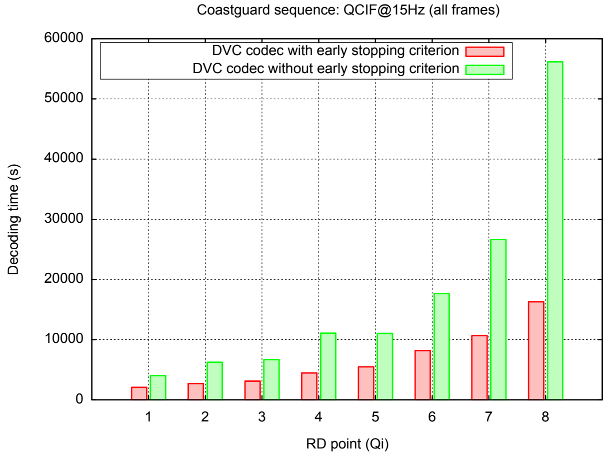
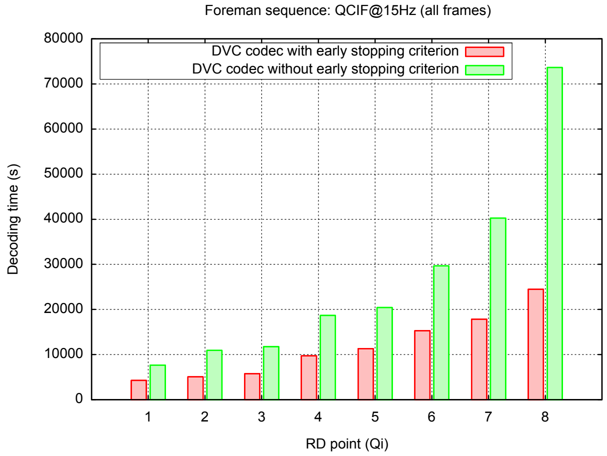


Figure 2. Total number of SPA iterations for the *Hall Monitor*, *Coastguard*, *Foreman* and *Soccer* sequences.

The maximum total decoding time reduction observed was for the last RD point of the Hall Monitor sequence with a 3.9 times reduction due to the inclusion of the early stopping criterion. The minimum time reduction occurred for the first RD point of the Soccer sequence with a 1.7 times reduction of the DVC decoder complexity. So, the overall decoder complexity is significantly reduced and, thus, the main objective of the proposed early stopping criterion is fulfilled.







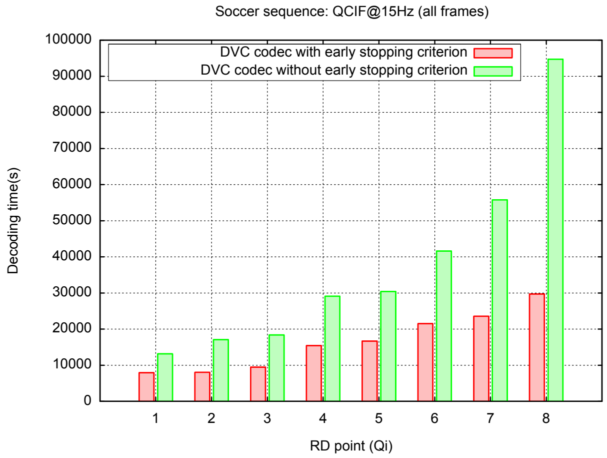
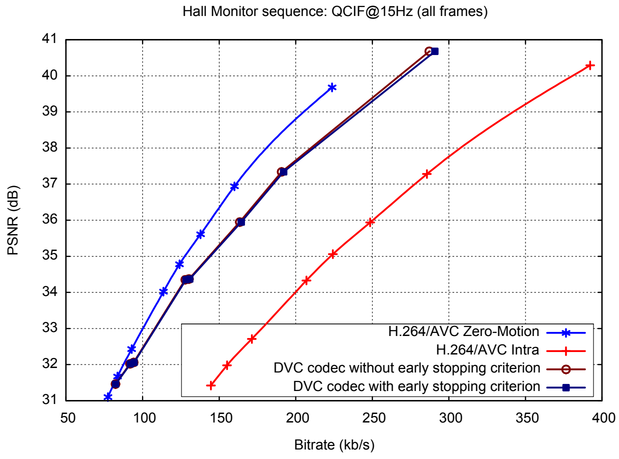


Figure 3. Decoding time(s) for the *Hall Monitor*, *Coastguard, Foreman and Soccer* sequences.

4.3 RD Performance

In Figure 4, the RD performance for the four test sequences are presented. As shown, the RD performance of the DVC codec with and without the proposed early stopping criterion is quite similar, especially for low and medium bitrates. It is only possible to observe a minor loss in RD performance for the last RD point (high bitrate/quality) which goes up to 0.15dB for the *Hall Monitor* sequence (the worst result for all four test sequences).



Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7353
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7353

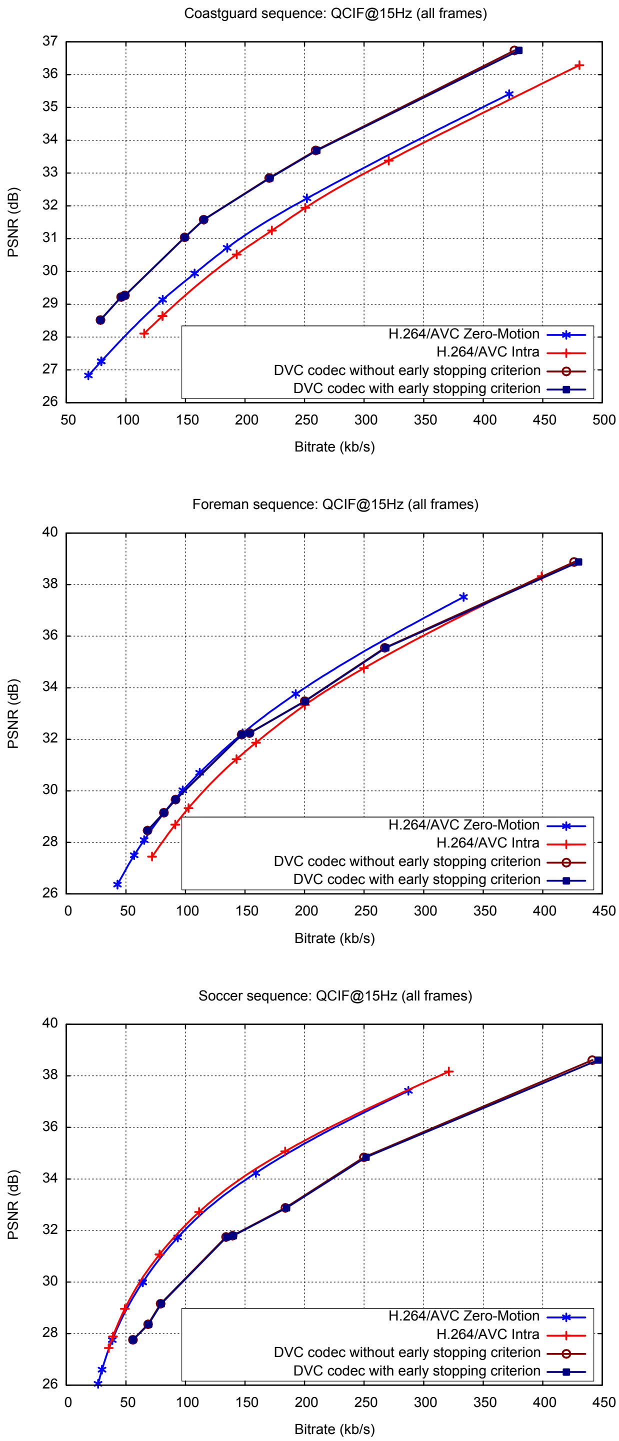


Figure 4. RD performance for the *Hall Monitor, Coastguard, Foreman* and *Soccer* sequences.

Thus performance loss is expected, since it corresponds to the RD point where more significant savings in decoding complexity was achieved (4.4 times in the total number of SPA iterations). This suggests that by choosing different values for the thresholds λ and d, a different balance between the decoding complexity reduction and the RD performance losses can be achieved with the proposed early stopping criterion algorithm. Another conclusion is that the proposed stopping criterion is sequence independent, since it is obtained similar performance with the same threshold parameters.

5. CONCLUSIONS

In this paper, an efficient early stopping criteria algorithm for distributed video coding is proposed. A minor loss in coding efficiency (0.15 dB maximum) is observed while major reductions (1.7 to 4.4 times) in the decoding complexity can be achieved, depending on the amount of the correlation between the side information and original data. The criterion is based on two relevant metrics computed at the end of each step of the sum-product algorithm decoding process. As shown, it is possible to achieve significant reductions in the decoding complexity with negligible losses in the RD performance. As future work, the proposed metrics could be used to indicate the most probable error locations, guiding the refinement of the side information quality; this could lead to better RD performance and a tight integration between syndrome decoding and side information creation processes.

6. REFERENCES

[1] Richardson, T.J. and Urbanke, R.L. 2001. Efficient encoding of low-density parity-check codes. IEEE Trans. Inform. Theory. 47, 2 (February 2001), 638–656.

[2] Liveris, A.D., Xiong, Z., and Georghiades, C.N. 2002. Compression of binary sources with side information at the decoder using LDPC codes, IEEE Commun. Lett. 6, 10 (October 2002), 440–442.

[3] Chen, J., Khisti, A., and Malioutov, D.M. 2004. Distributed source coding using serially-concatenated-accumulate codes. In IEEE Information Theory Workshop (San Antonio, TX, USA, October 2004).

[4] Ascenso, J. and Pereira, F. 2008. Design and performance of a novel low-density parity-check code for distributed video coding. In IEEE International Conference on Image Processing (San Diego, CA, USA, October 2008).

[5] Puri, R., Majumdar, A., and Ramchandran, K. 2007. PRISM: a video coding paradigm with motion estimation at the decoder, IEEE Trans. Image Process. 16, 10 (October 2007), 2436–2448.

[6] Brites, C. and Pereira, F. 2007. Encoder rate control for transform domain Wyner-Ziv video coding. In IEEE International Conference on Image Processing (San Antonio, TX, USA, September 2007).

[7] Girod, B., Aaron, A., Rane, S., and Rebollo-Monedero, D. 2005. Distributed video coding. Proc. IEEE. 93, 1 (January 2005), 71–83.

[8] Varodayan, D., Aaron, A., and Girod, B. 2005. Rate-adaptive distributed source coding using low-density parity-check codes. In 39th Asilomar Conference on Signals, Systems and Computers (Pacific Grove, CA, USA, October/November 2005).

[9] Kschischang, F.R., Frey, B.J., and Loeliger, H.-A. 2001. Factor graphs and the sum-product algorithm. IEEE Trans. Inform. Theory. 47, 2 (February 2001), 498–519.

[10] Brites, C., Ascenso, J., Pedro, J.Q., and Pereira, F. 2008. Evaluating a feedback channel based transform domain Wyner–Ziv video codec. Signal Process-Image. 23, 4 (April 2008), 269–297.

[11] Zhai, F. and Fair, I. J. 2003. Techniques for early stopping and error detection in turbo decoding. IEEE Trans. Commun. 51, 10 (October 2003), 1617-1623.

[12] Kienle, F. and Wehn, N. 2005. Low complexity stopping criteria for LDPC code decoders. In IEEE Vehicular Technology Conference (Stockholm, Sweden, June 2005).

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7353
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7353