Side information enhancement using an adaptive hash-based genetic algorithm in a Wyner-Ziv context

Thomas Maugey #, Charles Yaacoub *, Joumana Farah *, Marco Cagnazzo #, Béatrice Pesquet-Popescu #

TELECOM ParisTech, LTCI TSI
{maugey, cagnazzo, pesquet}@telecom-paristech.fr

* *Holy-Spirit University of Kaslik*
{charlesyaacoub, joumanafarah}@usek.edu.lb

***Abstract*—Side information construction in Wyner-Ziv video coding is a sensible task which strongly influences the final rate-distortion performance of the scheme. This side information is usually generated through an interpolation of the previous and next images. Some of the zones of a scene however, such as the occlusions, cannot be estimated with other frames. In this paper we propose to avoid this problem by sending some hash information for these unpredictable zones of the image. The resulting algorithm is described and tested here. The obtained results show the advantages of using localized hash information for the high error zones in distributed video coding.**

I. Introduction

Distributed video coding (DVC) is a promising paradigm which presents numerous applications. This way of coding allows to shift the coding complexity from the encoder to the decoder, which is an important feature for low power systems such as cell phones, video surveillance or sensor networks. In [1], Slepian and Wolf proved that two correlated sources can be independently encoded without affecting the transmission performance, assuming that they are jointly decoded. Wyner and Ziv [2] extended this result to lossy compression such as video coding, where inter-frame correlation can be exploited at the decoder only, without reducing the reconstructed visual quality or increasing the transmission rate.

Two main solutions have been proposed for DVC, thirty years after the theoretical theorems: the PRISM [3] and the Stanford [4] schemes. We adopt here the second approach, which consists in splitting the video sequence into two sets of frames: the key frames (KFs) and the Wyner-Ziv frames (WZFs). The KFs are transmitted using an intra codec (H.264 in this scheme) and are used at the decoder to compute an estimation of the WZFs, called *side information* (SI). At the encoder side, the WZFs are first transformed (DCT) then quantized and finally turbo-encoded. Only parity information is sent for correcting the SI at the decoder side. Finally, an inverse DCT is applied, at the decoder, on the resulting coefficients.

MMSP'10, October 4-6, 2010, Saint-Malo, France.
978-1-4244-8112-5/10/\$26.00 ©2010 IEEE.

Side information quality has a strong impact on the final rate-distortion performance. In order to improve the SI quality at the decoder, a solution proposed by Aaron et al. [5] and by Ascenso et al [6] consists in transmitting a sample of Wyner-Ziv frame information in order to enhance the interpolation precision. The two existing solutions are quite different but propose similar approaches. They both consist in transmitting *hash information* to the decoder for the Wyner-Ziv decoding efficiency improvement. Hash information is related to the original frame; for example, in [6] the bands are quantized and some of them are transmitted (a number which allows to attain a fixed energy). In both approaches, not all the blocks are transmitted. They are selected by thresholding the mean square of the difference with the previous frame at the encoder. At the end, around $10 - 15\%$ of the blocks are selected. This hash information is used to improve the side information extraction when the hash is available (extrapolation in [5] and interpolation in [6]) and to help the turbo-decoding [5]. However, the decision to send hash information or not for each block is performed at the encoder side and using interframe information. Therefore, the paradigm of intraframe encoding is somewhat violated, as stated in [5].

Yaacoub et al. also proposed a hash-based scheme [7], [8], [9]. They use a genetic algorithm (GA) to merge several WZ estimations available at the decoder, where the hash information is used to estimate the quality of candidate solutions. In their previous work, they studied the side information quality aspects without precisely dealing with the rate parameter. In this paper, we propose a new adaptive hash-based side information generation algorithm, where a GA-based fusion is applied only for the blocks that are supposed to be badly estimated with the classical interpolation. The proposed work also considers the rate cost of sending the hash information.

In Sec. II, we introduce the general structure of the proposed algorithm and describe each of its operational blocks with an important zoom on the genetic algorithm. Then, in Sec. III, we present several experiments which validate the benefits of our technique, by a comparison of the proposed hash-based algorithm w.r.t. the reference scheme in terms of side information quality and rate-distortion performance.

II. Proposed algorithm

The algorithm presented here proposes to improve the side information quality using some hash information sent by the encoder to perform a genetic algorithm based fusion. The general structure is presented in Sec. II-A, and then a zoom on the hash information coding and the genetic algorithm are proposed respectively in Sec. II-B and Sec. II-C.

A. General structure

The general structure of the proposed system is presented in Fig. 1. The method consists in firstly generating a classical side information (as in DISCOVER [10]) and secondly, for each badly-estimated block, requests some hash information from the encoder so that a hash-based side information estimation can be performed at the decoder side. Therefore, unlike the previous works [6] and [5], the intraframe encoding paradigm is preserved here, since the decision on the need of sending hash information is done at the receiver, instead of thresholding the difference between the two reference (key) frames. The different steps of our algorithm are described in more details in the following.

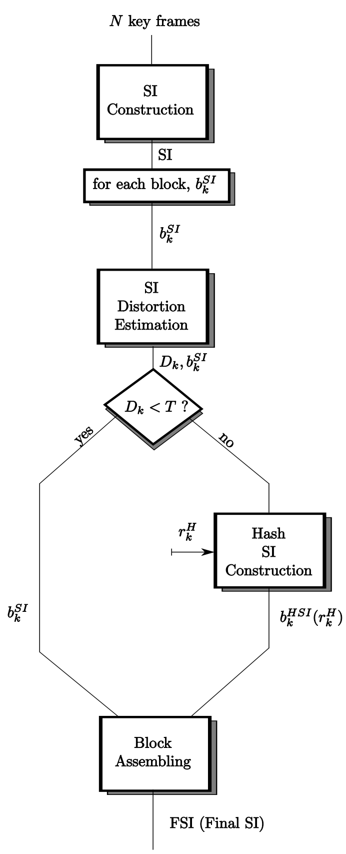
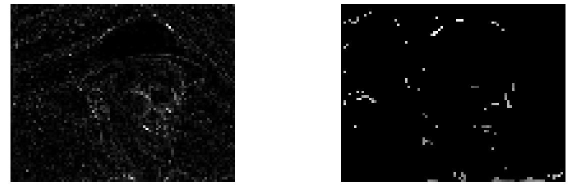


Fig. 1. General structure of the hash-based side information generation algorithm

- **SI construction -** The interpolation algorithm used at this step is the DISCOVER method. It consists of a block-based motion estimation between the two reference frames. The obtained motion vectors are split obtaining one bidirectional field which is refined two times by a bidirectional motion estimation, and which is finally median filtered. The obtained SI is divided into several 4×4 blocks, b_k^{SI}, and each block is processed independently by the subsequent operations of the algorithm.



(a) SI error (b) hash blocks sent

Fig. 2. Foreman 2%

- **SI quality -** For each block, the distortion of the side information is estimated. The technique used for this purpose is the mean square of the difference between the two motion-compensated reference frames. This block distortion is denoted by D_k. We can see in Fig. 2 that the selected blocks are taken from a zone which contains high errors. In other words, the SI quality estimation is quite efficient.
- **Thresholding -** The D_k value is thresholded by a T value which is calculated depending on the percentage of hash blocks sent to the decoder. If the distortion is lower than T, the side information is considered good enough, such that it can be directly turbo-decoded. Otherwise, b_k^{SI} is assumed to be a bad estimation of the original WZ frame, and therefore an *Hash SI construction* is performed.
- **Hash SI construction -** At this step, the side information is re-estimated thanks to some hash information transmitted at a rate of r_k^H. First, several estimations are generated: the average, the DISCOVER interpolation, a simple Motion-Compensated Interpolation (MCI) [11] and the Hash-based MCI (HMCI) [7]. Then, the GA algorithm is performed in order to build the fusion of these candidates. The computed hash-based side information b_k^{HSI} depends on the rate r_k^H.
- **Block assembling** consists in constructing the entire side information by assembling the blocks estimated with (b_k^{HSI}) or without (b_k^{SI}) the hash information. The final side information (FSI) is then turbo-decoded.

B. Hash information generation

The block size is fixed to 4×4, the same as the DCT block size of the Wyner Ziv frame coding. In the following, we describe how each hash block (a vector of 16 coefficients, one per band) is encoded. Based on the fact that some information regarding the WZ frame (as the dynamic range of the bands) is available at the decoder (transmitted for the Slepian-Wolf decoding), we decide to perform uniform quantization of the hash information, similar to the quantization performed for WZFs encoding (with a dead zone for the AC coefficients). The number of quantization levels is specified by a quantization matrix, showing the number of levels per band for eight rate-distortion points (from low to high bit rate). The matrix [10] is presented in Tab. I.

TABLE I
WZ AND HASH QUANTIZATION MATRIX

band	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
QI 1	16	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
QI 2	32	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
QI 3	32	8	8	4	4	4	0	0	0	0	0	0	0	0	0	0
QI 4	32	16	16	8	8	8	4	4	4	4	0	0	0	0	0	0
QI 5	32	16	16	8	8	8	4	4	4	4	4	4	4	0	0	0
QI 6	64	16	16	8	8	8	8	8	8	8	4	4	4	4	4	0
QI 7	64	32	32	16	16	16	8	8	8	8	4	4	4	4	4	0
QI 8	128	64	64	32	32	32	16	16	16	16	8	8	8	4	4	0

After the quantization process, the hash is converted into bitplanes and transmitted to the decoder. The corresponding rate is given by the sum of the logarithm of the non-zero bands levels at a chosen line of the quantization matrix.

C. *Genetic algorithm*

Based on the principles of evolution and natural genetics, Genetic Algorithms (GAs) [12] are well suited for non-linear optimization problems. In this paper, the use of a GA in a fusion-based approach aims at improving the quality of the side information relying on several initial estimations.

The GA operates at the block level. Initially, for a given block in the WZ frame, each of the co-located blocks in the available SI candidate frames represents a possible solution. A candidate solution is referred to as a *chromosome*, which consists of a sequence of pixels (*genes*)) arranged in a matrix to form a block. A population is a set of chromosomes in the solution space. The similarity between a given chromosome and the corresponding block in the WZ frame represents its *fitness* score, which is evaluated as the inverse of the mean-square-error between the received hash word and a local hash word extracted from the candidate block.

A flowchart diagram of the GA is shown in Fig 3. An initial population is first generated by duplicating each candidate block a number of times proportional to its fitness, until the desired population size S_p is reached. The chromosomes are then randomly shuffled and arranged into pairs. Each pair (parent chromosomes) undergoes a vertical crossover [7], [8], [9] followed by an horizontal crossover to yield a couple of child chromosomes (called *offsprings*). Each of the crossover operations occurs with a probability P_c. In order to extend the solution space and reduce the possibility of falling into local optima, a mutation is performed on offsprings by randomly selecting a gene and inverting one of its bits. Mutation usually has a very low probability of occurrence P_m [13]. The fitness of the resulting chromosomes is then evaluated and a number $S_f \leq S_p$ of the most fit chromosomes is selected, while the others are deleted to make room for new ones. The surviving chromosomes are then duplicated a number of times proportional to their fitness and the whole procedure is repeated until the maximum number I_max of iterations is reached. Finally, the fittest chromosome is chosen as the best candidate to be used as side information for decoding the colocated block in the WZ frame. As for the GA parameters, the following set was determined experimentally after intensive simulations: Sp = 60, Sf = 40, Imax = 10, Pc = 0.8, Pm = 0.01.

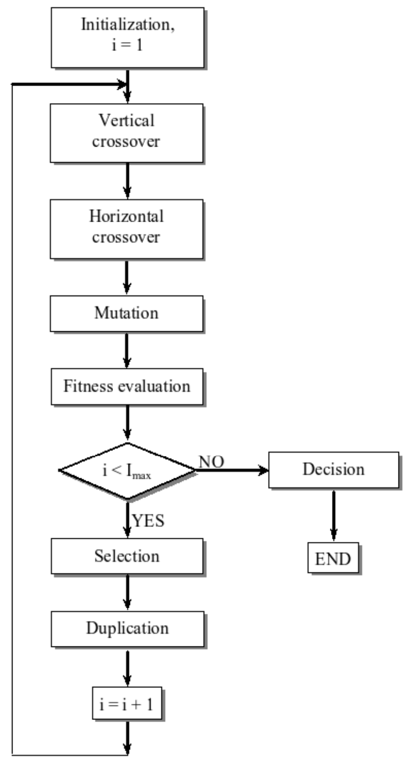


Fig. 3. Flowchart diagram of the genetic algorithm.

III. EXPERIMENTAL RESULTS

The results presented here have been obtained with three CIF (352×288) test sequences: *foreman, mobile, football*, for a GOP (Group Of Pictures) size of 2.

A. *First results*

In this section, we present the first results obtained for the proposed algorithm. These preliminary results have the purpose to set the best parameter values (especially the percentage of hash information to be sent, and the quantization index used for its transmission). For this reason, a set of experiments have been run on the three test video sequences. The quantity of hash information transmitted can vary due to two parameters: the number of blocks which require a hash side information refinement (measured in %) and the quantization level (given as a QI parameter, see Tab. I). We run several experiments in order to adopt the best configuration. We tested all of the couples (%, QI) in the set $\{2\%, 5\%\} \times \{\text{QI } 1, \text{QI } 2, \text{QI } 4, \text{QI } 6\}$. In these tests we measured the rate, r^H (due to the hash transmission), and the PSNR (the quality in dB) of the hash SI. Then we performed the turbo decoding of these obtained hash SI and measured for each couple the number of transmitted parity bits, and the

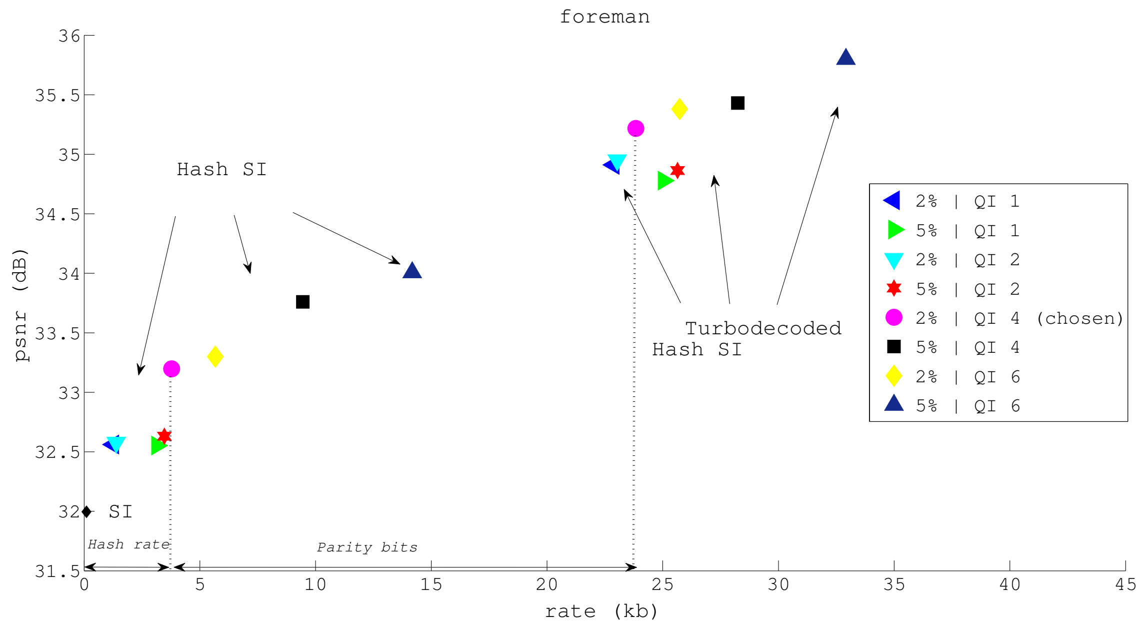


Fig. 4. Tests for several parameters settings (*foreman*, 352×288): percentage of hash information sent from 2% to 5%, and QI from 1 to 6. The best configuration is (2% — QI 4).

quality of the final decoded WZ frame.

Fig. 4 presents the results obtained for *foreman* (average PSNR depending on the average rate of either the hash bits or the parity bits) at a quantization step for the key frames of 31. The different couples of points represent the rate-distortion values respectively for the hash side informations and for the final turbo decoded WZ frames. Note that the final turbo decoded rate is the addition of the hash rate and the required parity bits.

What is noticeable in Fig. 4 and was confirmed for all sequences is that the best couple is a percentage of 2% with a quantization of $QI = 4$. That means that the hash sent has to be quite precise but its rate is quite low.

Our preliminary results also show that the genetic algorithm, in spite of its complexity, brings a real interest compared to a simple direct hash-based fusion or inverse DCT of the received hash. In the next section, we test the performances of the proposed hash algorithm and compare the obtained rate-distortion results to the DISCOVER reference scheme.

B. Rate-distortion results

The rate-distortion curves are shown in Fig. 5 for the three CIF sequences. It can be observed that, at high bitrates, the performance of the hash-based scheme is always better than the reference. This is explained by the fact that at these rates, the hash rate is low compared to the rate of the parity information sent for turbo-decoding. On the contrary, at low bitrates, the hash rate becomes too high and the performance of the hash-based scheme is degraded for *foreman* and *mobile*. To measure the general gain, we use the Bjontegard metric [14]. Though for *mobile* the average gain is almost zero, for *foreman* and *football*, the gains are interesting. Indeed, the decoded quality is improved by 0.14dB for *foreman* and 0.19dB for *football*. Moreover, the rate reduction is around -2.7% for *foreman* and -3.0% for *football*. As was noticed in [7], [8], [9], it is expected that the GA fusion approach will yield even higher performance gains in the case of video sequences presenting more complex scenes (containing parts with different levels of motion).

IV. Conclusion

In this paper, we presented a hash-based side information enhancement technique for distributed video coding. Hash information is sent for a selection of blocks where the known interpolation techniques fail to yield a good estimation. A genetic algorithm is used to exploit hash information and select the best solution among a plethora of combinations of several candidate blocks. Simulation results show interesting gains in terms of rate-distortion performance. Finally, we are currently investigating the extension of the current study to multi-view video and to incorporate larger GOP sizes.

Acknowledgment

This work was partly supported by a research grant from the Lebanese National Council for Scientific Research (LNCSR) and was realized within the Franco-Lebanese CEDRE (08 SCI F2 / L1) program.

References

[1] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Inform. Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

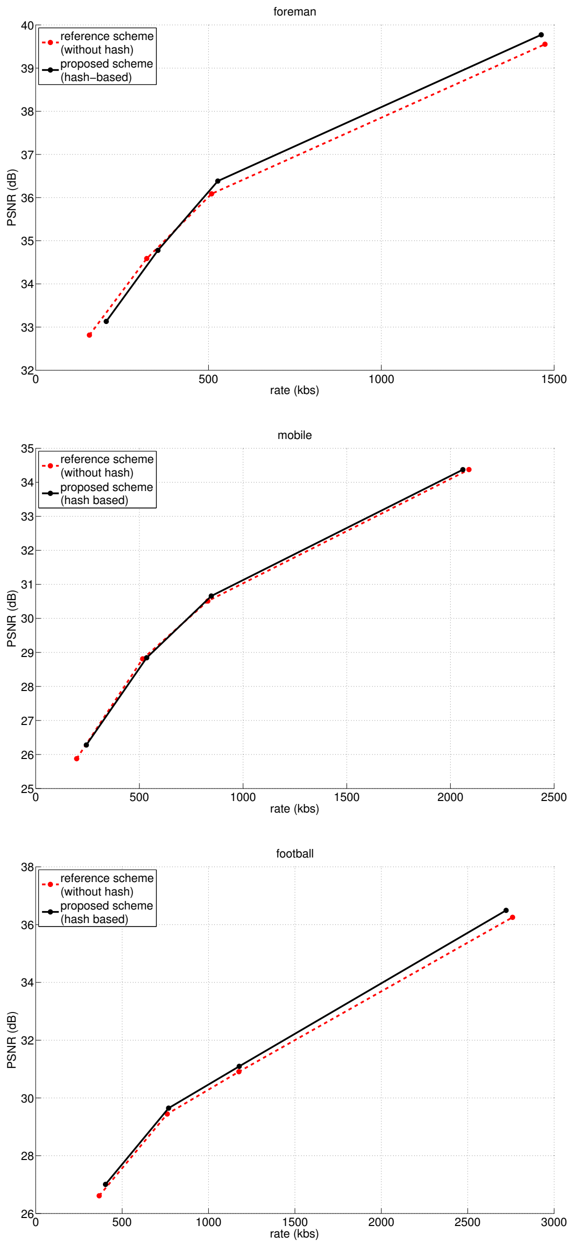


Fig. 5. RD performances for three CIF test sequences. In dashed red lines, the DISCOVER reference scheme, in plain black, the proposed adaptative hash-based algorithm.

[2] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the receiver," *IEEE Trans. on Inform. Theory*, vol. 22, pp. 1–11, Jan. 1976.

[3] R. Puri and K. Ramchandran, "PRISM: A video coding architecture based on distributed compression principles," EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M03/6, 2003.

[4] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[5] A. Aaron, S. Rane, and B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver," in *Proc. Int. Conf. on Image Processing*, vol. 5, Singapore, Oct. 2004, pp. 3097–3100.

[6] J. Ascenso and F. Pereira, "Adaptive hash-based side information exploitation for efficient Wyner-Ziv video coding," in *Proc. Int. Conf. on Image Processing*, vol. 3, San Antonio, TX, Oct. 2007, pp. 29–32.

[7] C. Yaacoub, J. Farah, and B. Pesquet-Popescu, "A genetic algorithm for side information enhancement in distributed video coding," in *Proc. Int. Conf. on Image Processing*, Cairo, Egypt, Nov. 2009.

[8] ——, "Improving hash-based wyner-ziv video coding using genetic algorithms," in *Int. Mobile Multimedia Commun. Conf. (MOBIMEDIA)*, London, UK, Sep. 2009.

[9] ——, "A genetic frame fusion algorithm for side information enhancement in wyner-ziv video coding," in *Proc. Eur. Sig. and Image Proc. Conference*, Glasgow, Scotland, Aug. 2009.

[10] C. Brites, J. Ascenso, and F. Pereira, "Improving transform domain Wyner-Ziv video coding performance," in *Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 2, Toulouse, France, May 2006, pp. 525–528.

[11] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Proc. Asilomar Conference on Signals, Systems and Computers*, vol. 1, Nov. 2002, pp. 240–244.

[12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1st ed. Addison-Wesley Professional, January 1989. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0201157675

[13] P. Chang, J. Leou, and H. Hsieh, "A genetic algorithm approach to image sequence interpolation," vol. 16, no. 6, pp. 507–520, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/B6V08-425F6M5-1/2/6bd1a9cbf0292cb62a69fbacf823698c

[14] G. Bjontegaard, "Calculation of average PSNR differences between RD curves," 13th VCEG-M33 Meeting, Austin, TX, USA, Tech. Rep., Apr. 2001.