

# Improving Hash-Based Wyner-Ziv Video Coding Using Genetic Algorithms

Charles Yaacoub<sup>(1,2)</sup>, Joumana Farah<sup>(1)</sup>

<sup>(1)</sup> Faculty of Sciences and Computer Engineering, Holy-Spirit University of Kaslik, Jounieh, Lebanon.

{charlesyaacoub, joumanafarah}@usek.edu.lb

Béatrice Pesquet-Popescu<sup>(2)</sup>

<sup>(2)</sup> Signal and Image Processing Department, Telecom ParisTech, 46 Rue Barrault, 75634 Paris, France.

beatrice.pesquet@telecom-paristech.fr

## ABSTRACT

This work aims at improving the quality of the side information in distributed video coding. A novel technique for the fusion of multiple side information is developed for this purpose. Based on genetic algorithms, the proposed frame fusion technique incurs no additional complexity at the encoder side, in case of hash-based distributed video coding. Simulation results show a significant improvement in the side information quality compared to other interpolation techniques available in the literature, which greatly improves the rate-distortion performance of a distributed video codec, where the gain in PSNR can reach 7 dB.

## Categories and Subject Descriptors

I.4.2 [Image Processing and Computer Vision]: Compression (coding).

## General Terms

Algorithms, Performance, Design, Experimentation.

## Keywords

Distributed video coding, frame interpolation, genetic algorithms, side information, Wyner-Ziv coding.

## 1. INTRODUCTION

In traditional video coding techniques [1], motion estimation is performed at the encoder side, which yields very complex encoders. At the decoder, only motion compensation is performed, thus yielding simple decoders. Recent studies based on the Slepian-Wolf [2] and Wyner-Ziv [3] theorems showed that, with distributed video coding (DVC) [4-9], it is possible to shift most of the computation burden from the encoder to the decoder side. This configuration is suitable for applications requiring simple encoders, such as wireless video sensor networks, due to their limited resources (e.g. power and memory).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Mobimedia'09, September 7-9, 2009, London, UK. Copyright 2009 ICST 978-963-9799-62-2/00/0004 ... \$5.00

In practical DVC systems, a subset of frames, known as key frames, is usually intra-coded using a traditional video encoder. One or more frames following each key frame, known as Wyner-Ziv (WZ) frames, are then compressed by appropriately puncturing the parity bits at the output of a channel coder. At the receiver, the side information (SI) is generated by interpolating previously decoded (key or WZ) frames [4-9].

The quality of the side information greatly affects the rate-distortion (RD) performance of a distributed video codec. On one hand, it is used as a first estimate of the transmitted WZ frame in the decoder. Therefore, improving the side information leads to a reduction of the bit rate necessary for the compression of the WZ frame. On the other hand, the SI is used for the reconstruction of the quantized symbols after source-channel decoding. As a result, improving the side information reduces the distortion at the decoder output.

Several interpolation techniques for generating accurate side information have been presented in the literature. Aaron et al. first proposed average interpolation (AVI) and motion-compensated interpolation (MCI) in [4]. In [5], Ascenso et al. presented an improved motion-compensated interpolation using spatial motion smoothing. In hash-based DVC [6,7], hash bits were used to improve the quality of the side information. Edge-based and mesh-based interpolations were proposed in [10] and [11], respectively. Multiple hypothesis techniques were developed in [12] and [13], where two different frames were used as side information for the decoding of a single WZ frame. Each of these methods outperforms some of the others in particular situations (e.g. background motion, moving objects, etc...). Besides, several fusion-based approaches were proposed in [14-16] to generate side information in multiview DVC.

In our study, we aim at improving the generation of side information in Wyner-Ziv video coding. For this purpose, we first present a frame fusion approach that combines several interpolated frames in a hash-based DVC system. Then, we develop a novel method based on genetic algorithms to further improve the frame fusion technique.

Genetic Algorithms (GAs) [17] are well suited for searching and optimization problems. They are based on the principles of evolution and natural genetics and have rarely been used in video applications. In [18], a GA approach was proposed in order to obtain an up-sampled image sequence from a low-resolution image sequence. Genetic motion search and block matching algorithms were proposed in [19] and [20], respectively. To our knowledge, GAs have not been used in DVC applications up to this date.

This paper is organized as follows. Section 2 presents a simple technique for the fusion of multiple side information. In Section 3, frame fusion is modeled with a genetic representation, and genetic algorithms are developed to combine several frames obtained using different interpolation techniques. Simulation results are discussed in Section 4, and finally, conclusions are drawn in Section 5.

## 2. FUSION OF MULTIPLE SIDE INFORMATION

Our technique for the enhancement of SI generation in monoview DVC is based on the following idea: since different SI frames can be obtained using different interpolation techniques, the decoder can take advantage of this diversity by selecting the best among the different versions of the side information. This decision can be made on a frame-level basis (i.e., by selecting the best SI frame) or on a region-level basis (i.e., separately for every region within the frame). However, since only key frames are available at the decoder, some information about the missing WZ frame needs to be transmitted in order to evaluate the quality of the side information. For example, statistical information (e.g., mean, variance) about the WZ frame, or a binary edge image [10], can be transmitted for this purpose. Since the encoding complexity is a major concern in DVC, the computational burden required to construct this additional information should be minimized. Additionally, the transmission overhead required to deliver this information to the receiver should be negligible in order not to degrade the system's RD performance.

In hash-based DVC, a discrete cosine transform (DCT) is performed on the WZ frames, and a subset of the DCT coefficients is transmitted [6,7] as a hash word to aid the decoder in improving the system's RD performance. The hash word can be used during the interpolation process in a hash-based motion compensated interpolation (HMCI), where the minimal distance measure (e.g. square error) is performed between the received hash word and the one corresponding to the candidate SI block, instead of the block matching technique used in MCI. Therefore, the received hash bits can be used, in our frame-fusion technique, to evaluate the quality of the different SI frames.

In our study, the construction of the hash word is inspired from [6] and [7]. After computing the DCT coefficients for a block of dimensions  $B \times B$ , a subset of  $n$  coefficients ( $n < B^2$ ) that packs most of the energy within the DCT block is kept, while the remaining coefficients are replaced with zeros. Once this has been done for all the blocks within a frame, either the resulting DCT frame is compressed using entropy coding (e.g. Huffman coding), or only the non-zero coefficients along with their relative positions are compressed and transmitted to the decoder.  $B$  and  $n$  are tunable parameters; they can be fixed or dynamically varied throughout the sequence.

At the decoder side, several interpolation techniques (e.g. AVI, MCI and HMCI) can be used to generate different SI frames. The key frames involved in the interpolation process can also be considered as candidate SI frames. For every block within a frame, a hash word is computed for all SI candidates as was done at the encoder side for the WZ frame. In order to evaluate which of the candidates better represents the missing WZ frame, the mean square error is computed between the received hash and

each of the hash words extracted from the candidate SI blocks, after replacing the missing DCT coefficients with zeros. Then, the block that minimizes the Mean Square Error (MSE) is selected as the final SI block.

Even though the MSE computed in the transform domain may not be an ideal measure, the proposed technique guarantees that the energy distribution of the DCT bands in the final SI frame is as close as possible to the original WZ frame.

## 3. GENETIC ALGORITHMS FOR ENHANCING THE SIDE INFORMATION

In the previous section, the proposed frame-fusion approach is performed at the block level. In other words, different blocks in the final SI are obtained from different SI candidates. However, all the pixels within the same block belong entirely to the same SI candidate frame generated at the decoder. In order to perform frame fusion at the pixel level, i.e. in order to have different pixels originating from different SI candidates within the same block, we propose to use Genetic Algorithms. For this purpose, different SI frames are first generated using previously developed interpolation techniques. Actually, any SI generation technique can be considered as an input to our algorithm. The GA then combines two or more SI frames, in a fusion-based approach, in order to improve the side information generated at the decoder.

In order to develop a GA, a genetic representation of the solution domain and a fitness function that allows evaluating each possible solution need to be defined. Initially, for a given block in the WZ frame, each of the co-located blocks in the available SI frames represents a possible solution. A candidate solution is referred to as a *chromosome* which consists of a sequence of *genes*. Therefore, a chromosome is defined as 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*. In hash-based DVC [6,7], the received hash bits can be used for fitness evaluation in the decoder. In our proposed GA, a hash word is computed for every chromosome in the population. The fitness of a chromosome is then evaluated as:

$$F = \frac{K_1}{K_2 + K_3 D}, \quad (1)$$

where  $K_1$ ,  $K_2$  and  $K_3$  are tunable scaling factors, and  $D$  is a distance measure, such as the sum of absolute differences (SAD) or the sum of squared errors (SSE), between the received WZ hash word and the one computed for the current chromosome. As a result, our genetic algorithm can be implemented in a hash-based Wyner-Ziv video codec without any complexity increase at the encoder side.

On the other hand, the proposed GA can operate at the frame level or the block level. Since the former is a particular case of the latter, we start by presenting the block-level GA. Each frame is first divided into a number of blocks. For each block, the algorithm consists of an iterative procedure represented by the flowchart diagram of Figure 1, where  $i$  represents the iteration number and  $I_{max}$  represents the maximum allowed number of iterations. As shown in the figure, the GA consists of the following steps:

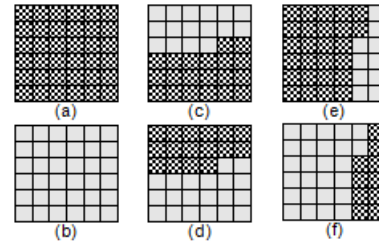
### Step 1: Initialization

An initial population is generated using several SI frames obtained from different interpolation techniques. In our system, the initial candidate frames to be fed to the GA are obtained by AVI, MCI, and HMCI. The key frames involved in the interpolation process are also considered as possible solutions.

Therefore, the initial number of candidates, for each block of the current WZ frame, is five (the co-located block in each of the five candidate frames). The initialization of the GA consists of duplicating each candidate a number of times proportional to its fitness, until the desired population size  $S_p$  is reached. This population then undergoes subsequent genetic operations.

### Step 2: Crossover

The chromosomes of the current population are randomly shuffled and arranged into pairs. Each pair then produces a new pair of child chromosomes (called *offspring*s). Offspring are obtained by combining and exchanging genes between parent chromosomes. This process is illustrated in Figure 2. Given a pair of parent chromosomes (Figure 2-a and Figure 2-b), a crossover position is selected at random and each of the parents is divided into two regions. One offspring is then obtained by combining the first region of the second parent with the second region of the first parent (Figure 2-c, Figure 2-f). Similarly, another offspring is obtained by combining the first region of the first parent with the second region of the second parent (Figure 2-d, Figure 2-e). The crossover can be performed vertically using a horizontal scan of lines within the block as in Figure 2-c and Figure 2-d, or horizontally using a vertical scan of columns as in Figure 2-e and Figure 2-f.



**Figure 2. Crossover of parents (a) and (b) to produce offspring (c) and (d) in vertical crossover or (e) and (f) in horizontal crossover.**

In case of a one-directional crossover (horizontal or vertical), the probability of a crossover to occur is controlled by a tunable parameter  $P_c$ . Therefore, for each pair of parent chromosomes, either a new pair is obtained (with a probability  $P_c$ ), or parents remain unchanged (with a probability  $1-P_c$ ). If the crossover is performed in both (horizontal and vertical) directions, a crossover probability is specified for each direction. In this case, the GA will be referred to as the two-directional genetic algorithm (2D-GA). In the 2D-GA, step 2 either produces one or two new pairs of chromosomes or keeps the two parent chromosomes unchanged. Note that other directional crossovers (e.g. diagonal crossover) can be envisaged, but we limit our study to only two directions, for simplicity.

### Step 3: Mutation

A random change is performed on a given chromosome. In our GA, a gene is first selected at random, and a bit in the selected pixel (gene) is then inverted. Mutation allows the GA to extend its solution space and reduce the possibility of falling into local optima, and it usually occurs with a very low probability  $P_m$  [18].

### Step 4: Fitness evaluation

In our GA, the SSE is first used as a distance measure between the received WZ hash word and each of the hash words computed for all the chromosomes in the current population. Then, the fitness function is evaluated for every chromosome as explained earlier.

### Step 5: Selection

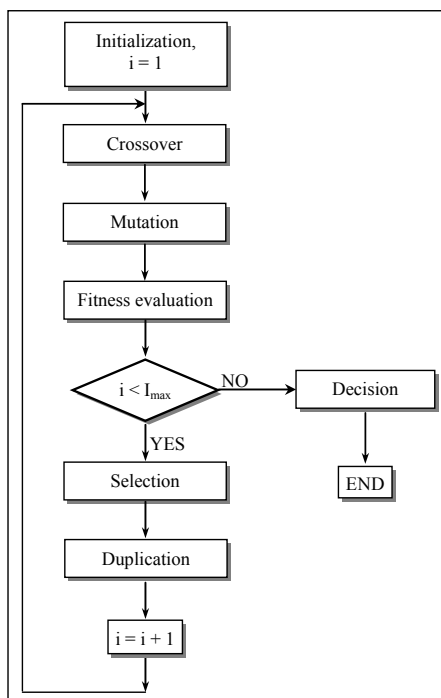
A number  $S_f \leq S_p$  of chromosomes is selected, while  $S_p - S_f$  chromosomes are deleted to make room for new ones. Only the most fit chromosomes survive.  $S_f$  must be chosen not too close to  $S_p$  in order not to give a chance for the least fit chromosomes to duplicate, and not too small in order to allow for a variety of chromosomes to exchange genes and duplicate.

### Step 6: Duplication

Each of the  $S_f$  remaining chromosomes is duplicated a number of times proportional to its fitness, until the population size  $S_p$  is reached.

Steps 2 to 6 are then repeated until the maximum number of iterations is reached. Finally, the fittest chromosome is chosen as the best candidate to be used as side information for decoding the co-located block in the WZ frame.

We noticed that when a very high motion occurs in some parts of the video, the interpolation techniques used to initialize the GA often fail in these parts, which also degrades the performance of the GA. In this case, simply computing the inverse DCT (IDCT) given the received hash word (by first substituting the missing



**Figure 1. Flowchart diagram of the genetic algorithm.**

coefficients with zeros) yields a better result, provided that a sufficient number of coefficients was transmitted. On the other hand, this IDCT block cannot be included as a candidate in the GA since in this case, with the proposed metric for fitness evaluation, the algorithm would converge towards this block all the time. Therefore, to determine whether the GA yielded a good SI candidate, the fitness  $F_{GA}$  of the fittest chromosome can be compared to a threshold  $F_T$ : if  $F_{GA} > F_T$ , the system takes the GA output as side information; otherwise, the system opts for the co-located block in the IDCT frame. This substitution by the IDCT block using the threshold  $F_T$  can also be applied in the simple fusion technique (SFT) presented in Section 2.

The performance of the GA is greatly affected by its different parameters. It can be clearly seen that, by moving from one iteration to the next, possible solutions that have a greater fitness value have a greater chance to survive, whereas candidate solutions having the least fitness values are eliminated. Therefore, the fitness function is a major factor that influences the quality of the final SI. The number of iterations  $I_{max}$  required for the GA to converge depends on the population size  $S_p$  and the number of surviving candidates  $S_f$  in each iteration.  $I_{max}$  defines the number of crossover points, or in other words, the number of regions to be combined from the co-located blocks in the different candidate SI frames initially generated at the decoder.

When the GA operates at the frame level, entire frames are considered in the genetic operations instead of individual blocks. However, since crossover and mutation influence a small number of blocks in this case (in every iteration), the DCT coefficients (necessary for fitness evaluation) need to be updated only for the blocks where a change has occurred. This leads to a simpler algorithm compared to the block-level GA.

From a complexity point of view, it may seem computationally inefficient to run the proposed GA at the decoder. However, with the recent advances in hardware technology and parallel computing techniques, the additional decoding complexity incurred by the GA becomes feasible. Furthermore, in order to limit the decoding complexity, the GA parameters can be tuned accordingly. Additionally, a stopping criterion (e.g. the fitness value of the most fit chromosome remaining constant for several consecutive iterations) can be used to stop the GA before reaching the maximum number of iterations, thus reducing the total number of operations performed to obtain the result.

## 4. SIMULATION RESULTS

In our simulations, we consider 300 frames from the Carphone, Foreman, and News QCIF video sequences, and 150 frames from the Trevor sequence, sampled at a rate of 30 fps, and a WZ GOP size of 2 (i.e. a total of 74 WZ frames for Trevor and 149 for the others). The hash word transmitted to the decoder consists of  $(1/8)^{th}$  of the WZ frame's DCT coefficients, computed for blocks of dimensions  $16 \times 16$  pixels. In the first experiments, the crossover is performed only vertically and the parameters for the fitness function are set as  $K_1 = 1$ ,  $K_2 = 0$ , and  $K_3 = 1$  (i.e., the fitness function is  $F = 1 / D$ ). As for the GA parameters, the following set was determined experimentally after intensive simulations:  $\{S_p = 60, S_f = 40, I_{max} = 10, P_c = 0.8, P_m = 0.01\}$ .

In Figure 3, we show the cumulative density function (CDF) of the difference ( $\Delta_{PSNR}$ ) in peak signal-to-noise ratio (PSNR) between the frame-level GA (without using the IDCT in high-motion areas) on one side, and each of the different interpolation techniques on the other, for Carphone, Foreman, News and Trevor sequences, respectively. We notice that for all these sequences, a significant gain in PSNR is obtained most of the time. For example, with the Trevor sequence, we notice a performance gain for more than 85% of frames. A gain that exceeds 1dB is observed for approximately 70% of the frames compared to AVI, 15% compared to MCI, and 5% compared to HMCI. The percentage of frames with a gain exceeding 1 dB becomes 50%, 25% and 12%, in the Carphone sequence, 78%, 40% and 2% in the Foreman sequence, and 75%, 75% and 40% in the News sequence, compared to AVI, MCI and HMCI, respectively. To improve our system's performance, we used the IDCT instead of interpolated frames in high motion areas, as was explained earlier, with  $F_T = 0.03$ . In these conditions, we noticed that the number of frames with a gain greater than 1 dB increases by 15% to 40% for Carphone, and by 8% to 20% for Foreman. No significant improvement was observed for News and Trevor, since they generally present lower levels of motion compared to Foreman and Carphone.

On the other hand, a slight performance degradation can be noticed with the GA, where a performance loss of less than 1 dB is observed for less than 5% of the frames, for all four sequences.

Now that we have analyzed the behavior of the proposed frame-level GA, we will evaluate next the block-level GA with  $F_T = 0.02$ . Figure 4 shows the CDF of  $\Delta_{PSNR}$  between the GA output on

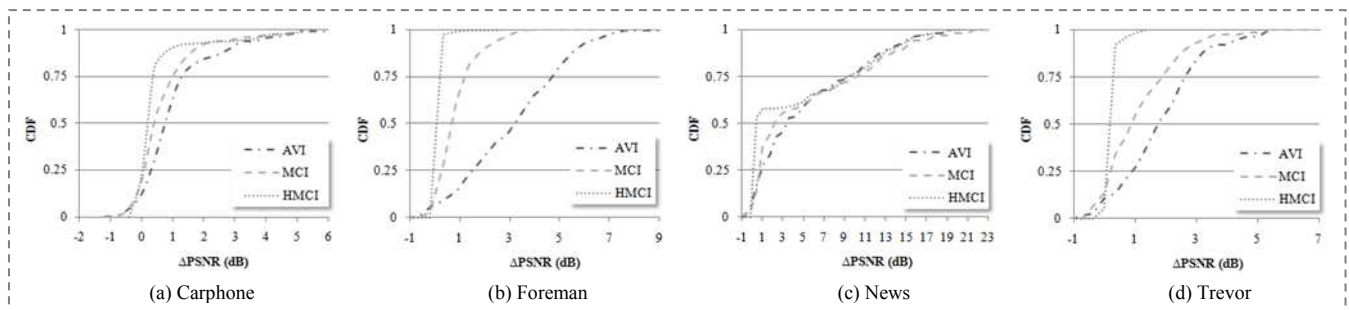
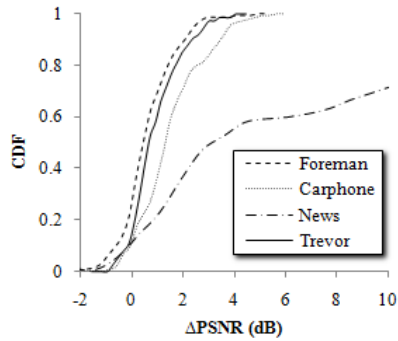


Figure 3. CDF of  $\Delta_{PSNR}$  for the different sequences, with a frame-level GA and  $F_T = 0$ .



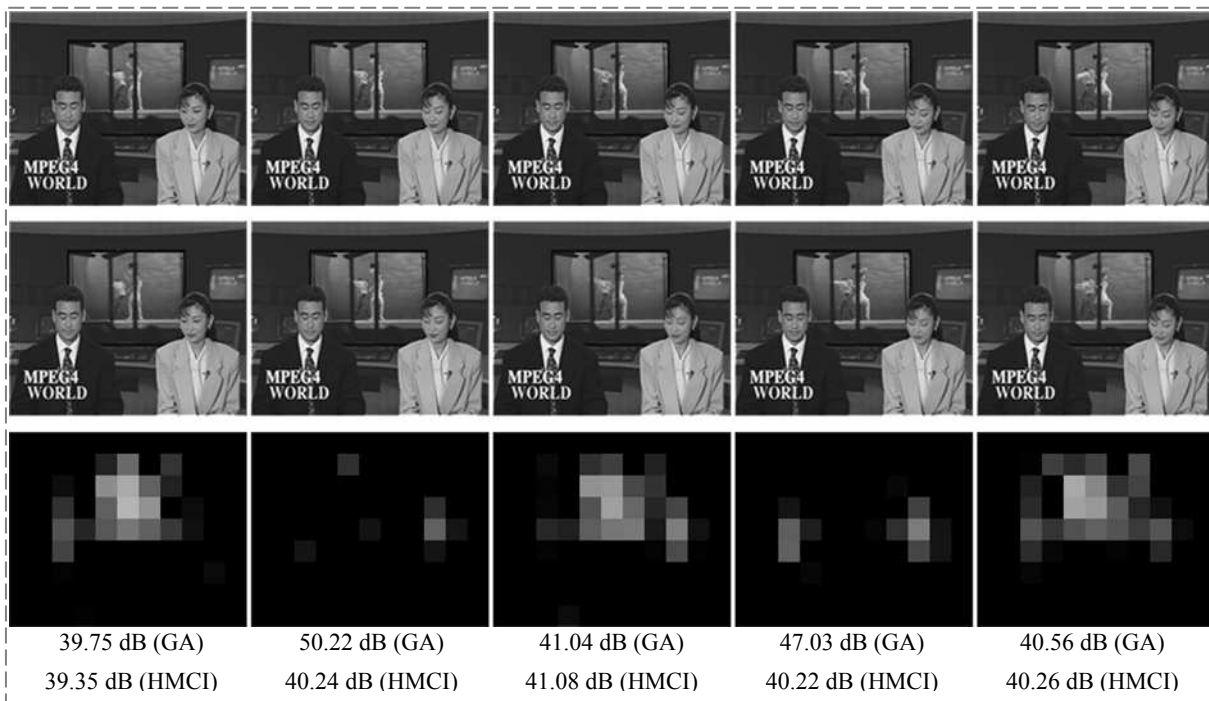
**Figure 4.** CDF of  $\Delta_{PSNR}$  between a block-level GA, AVI, MCI, HMCI and IDCT, for the different sequences, with  $F_T = 0.02$ .

one hand, and the best among the IDCT frame and the different interpolation techniques (AVI, MCI and HMCI) used, on the other. We notice that for all the sequences, a significant gain in PSNR is obtained most of the time. For example, with the Carphone and Trevor sequences, we notice a performance loss that does not exceed 1 dB for less than 10% of the frames, whereas a gain in PSNR is observed for the remaining 90% frames. Approximately 65% of the frames have a gain that exceeds 1 dB, and for some frames, the gain reaches 5.5 dB with Carphone, whereas with Trevor, near 50% of the frames have a gain that exceeds 1 dB, and the gain could reach 4 dB. As for the Foreman sequence, 18% of the frames have a performance degradation that does not exceed 1 dB, and only 2% suffer from a performance loss between 1 dB and 2 dB. On the other hand, a performance gain (up to 5 dB) is observed for 80% of the frames. Similarly, in the News sequence, a small subset (approximately

10%) of the frames suffers from a slight performance loss, while the gain exceeds 3 dB for more than half of the frames. With the News sequence, the gain exceeds 10 dB for 30% of the frames, and can even reach 20 dB (the curve was cropped at 10 dB for clarity).

In our simulations, we notice that the genetic algorithm (whether a frame level or block-level GA), behaves surprisingly well with the News sequence, where the maximum gain exceeds 19 dB. This is due to the very particular nature of the motion in this sequence: the scene contains two almost static (slowly moving) speakers in the foreground, and a couple of dancers in the background, alternating slow and fast movements. Figure 5 shows a sequence of consecutive WZ frames from the news sequence. The top row shows the original WZ frames, and the second row shows their respective side information obtained using the GA. The third row shows the local MSE (for each block) of the SI frame represented as a grayscale image. The last two rows show the SI frame PSNR obtained with the GA and the HMCI technique. Note that block MSE values (in the figures of the third row) are quantized to 256 gray levels (8 bits), where a lighter intensity indicates a greater error. It can be seen that the dominant error (light regions) is mainly concentrated in the background scene (the dancing couple). The error increases every other frame due to the alternating nature of the movement (slow/fast), which can also be seen in the fluctuations of the PSNR (and the gain towards HMCI) that significantly drops every other frame.

In order to evaluate our proposed algorithm in terms of RD performance, we encoded the above mentioned sequences using our WZ video codec presented in [8] (without feedback channel suppression). However, we use a 2D-GA in this case, since it



**Figure 5.** Snapshot from the News sequence, showing WZ frames 111 to 115.

**Top: original frame. Middle: side information obtained with GA. Bottom: error mask (MSE) and SI frame PSNR.**

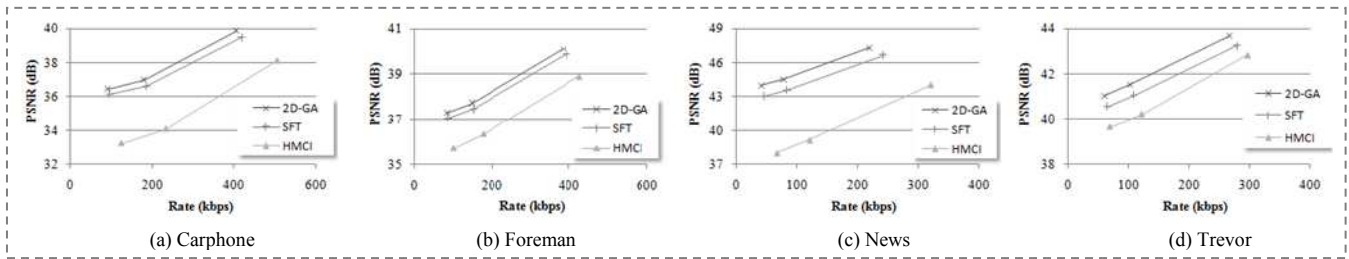


Figure 6. Rate-distortion curves for the different sequences.

yields better performance. Since in 2D-GA every couple of chromosomes can yield up to 4 offsprings (instead of 2), we doubled  $S_p$  and  $S_f$  ( $S_p = 120$  and  $S_f = 80$ ) while maintaining the same ratio  $S_p/S_f = 2/3$ . Additionally, we noticed that an improved performance can be obtained by modifying the values of  $K_1$ ,  $K_2$  and  $K_3$ . In fact, when  $K_1 = 1$ ,  $K_2 = 0$ , and  $K_3 = 1$  (i.e.  $F = 1/D$ ), a small value of  $D$ , for a given chromosome, results in a very large value of  $F$ , and the chromosome will dominate the next population after the selection and duplication phases. As a result, the GA will quickly converge, which is not desired in our case since the distance measure in use is not an exact indicator of similarity. Even though this phenomenon does not occur very often, it motivated us to investigate different values of the parameters  $K_1$ ,  $K_2$  and  $K_3$ . After a certain number of tests, we found that  $K_1 = 10$ ,  $K_2 = 100$  and  $K_3 = 1$  yield a significant improvement for a certain number of the video frames. For example, in the case of the News sequence, an improvement greater than 0.5 dB is noticed for almost 10% of the frames. In Figure 6, we show the average PSNR of the decoded WZ frames as a function of the average WZ bit rate. Since hash-based DVC [6-7] outperforms traditional WZ codecs, the 2D-GA results are compared with the case where HMCI was used for generating the side information at the decoder. Results obtained with the simple fusion technique (SFT) are also shown. The IDCT is used to improve the SFT and 2D-GA as explained earlier, with  $F_T = 0.02$ . Given that the same hash is transmitted in all cases (HMCI, SFT and block based 2D-GA), the rate overhead for the transmission of the hash was not considered. Otherwise, all curves would be shifted by the same amount, which is almost 12% of the WZ DCT coefficients in case the hash bits are transmitted without compression. As expected, the proposed frame fusion techniques yield significant improvements compared to HMCI. In the case of SFT, the average gain reaches approximately 3 dB with Carphone, 1 dB with Foreman and Trevor, and 6 dB with News. More performance improvements are obtained with the genetic algorithms, where 2D-GA yields additional 0.3 dB for Carphone and Foreman, 0.6 dB for Trevor, and 1 dB for News.

Note that these gains are averaged over the complete sequence and may not reflect the real potential of using a GA in frame fusion. In fact, GAs present exceptional performance in particular situations, as was shown in the case of the News sequence. By analyzing the Trevor sequence, we noticed that it consists of two different scenes, subsequent in time, where the first ends at the 29<sup>th</sup> WZ frame and the second starts at the 31<sup>st</sup>. The 30<sup>th</sup> WZ frame is a transitional frame between the two scenes and is therefore omitted from the following analysis. In the first part of the sequence, we can observe six different sub-windows each showing a different play, whereas in the second part, one play occupies the whole screen. Figure 7 shows a snapshot from each

of these scenes. Figure 8 shows the RD curves independently for each of the two parts of the video sequence. In the former (WZ frames 1 to 29), 2D-GA outperforms SFT where the average gain reaches 1 dB, whereas in the latter (i.e. WZ frames 31 to 74) the average gain with respect to SFT is only 0.3 dB. As a result, it can be concluded that the application of the GA is particularly advantageous in the case of complex video scenes containing parts with different levels of motion. By adding an intelligent step to the proposed technique for the analysis of the varying video content, the decoder would be able to decide, for each block, whether to apply the GA (that would significantly improve the performance towards to the simple fusion approach), or to simply rely on the SFT (to avoid the additional computation burden incurred by the GA).



Figure 7. Snapshots from the Trevor sequence.

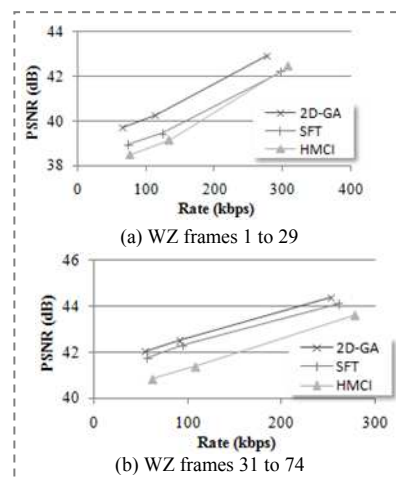


Figure 8. Rate-distortion curves for the two different scenes in the Trevor video sequence.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we developed novel techniques for improving the side information in distributed video coding. The proposed solutions rely on frame-fusion techniques, and especially on genetic algorithms, that make use of different interpolation techniques available in the literature in order to generate an enhanced version of SI. The algorithm runs at the decoder and incurs no additional complexity at the encoder side, in case of hash-based DVC codecs. The superior performance of the GA is especially observed in complex video scenes containing parts with different levels of motion.

In our future work, we will evaluate the performance of the proposed GA for GOP sizes greater than 2. Furthermore, we will target the complexity issue at the decoder and seek to optimize the GA for real time applications.

## 6. ACKNOWLEDGMENTS

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 program.

## 7. REFERENCES

- [1] ITU-T and ISO/IEC JTC1, "Advanced Video Coding for Generic Audiovisual Services," ITU-T Recommendation H.264 – ISO/IEC 14496-10 AVC, 2003.
- [2] D. Slepian and J.K. Wolf, "Noiseless Coding of Correlated Information Sources", IEEE Trans. Information Theory, Vol. IT-19, pp. 471-480, Jul. 1973.
- [3] A. Wyner and J. Ziv, "The Rate-Distortion Function for Source Coding with Side Information at the Decoder", IEEE Trans. Information Theory, Vol. IT-22, pp. 1-10, Jan. 1976.
- [4] A. Aaron, R. Zhang, and B.Girod, "Wyner-Ziv Coding of Motion Video", 36th Asilomar Conf. Signals, Systems and Computers, pp. 240-244, Nov. 2002.
- [5] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding", 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, Smolenice, Slovak Republic, Jun. 2005.
- [6] A. Aaron, S. Rane and B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver", IEEE International Conference on Image Processing, ICIP-2004, Singapore, Oct. 2004
- [7] J. Ascenso, F. Pereira, "Adaptive hash-based side information exploitation for efficient Wyner-Ziv video coding", International Conference on Image Processing (ICIP), USA, Sept. 2007.
- [8] C. Yaacoub, J. Farah, B. Pesquet-Popescu, "Feedback Channel Suppression in Distributed Video Coding with Adaptive Rate Allocation and Quantization for Multiuser Applications", EURASIP Journal on Wireless Communications and Networking (WCN), 2008.
- [9] C. Yaacoub, J. Farah, B. Pesquet-Popescu, "New Adaptive Algorithms for GOP Size Control with Return Channel Suppression in Wyner-Ziv Video Coding", International Journal of Digital Multimedia Broadcasting, special issue on Advances in Video Coding for Broadcast Applications, 2009.
- [10] Toan Nguyen Dinh, GueeSang Lee, June-Young Chang, and Han-Jin Cho, "A Novel Motion Compensated Frame Interpolation Method for Improving Side Information in Distributed Video Coding", International Symposium on Information Technology Convergence, Korea, Nov. 2007.
- [11] D. Kubasov, C. Guillemot, "Mesh-based motion-compensated interpolation for side information extraction in distributed video coding", International Conference on Image Processing (ICIP), Atlanta, USA, Oct. 2006.
- [12] K. Misra, S. Karande, H. Radha, "Multi-hypothesis distributed video coding using LDPC codes", 43rd Annual Allerton Conference on Communication, Control, And Computing, Monticello, USA, September 2005.
- [13] D. Kubasov, J. Nayak and C. Guillemot, "Optimal Reconstruction in Wyner-Ziv Video Coding with Multiple Side Information", 2007 IEEE International Workshop on Multimedia Signal Processing, Chania, Crete, Greece, October 2007.
- [14] X. Artigas, E. Angeli, L. Torres, "Side Information Generation for Multiview Distributed Video Coding Using a Fusion Approach", 7th Nordic Signal Processing Symposium, NORSIG'06, Reykjavik, Iceland, June 2006.
- [15] M. Ouaret, F. Dufaux, T. Ebrahimi, "Fusion-based multiview distributed video coding", ACM International Workshop on Video Surveillance and Sensor Networks, Santa Barbara, CA, USA October 27, 2006.
- [16] M. Ouaret, F. Dufaux, T. Ebrahimi, "Multiview distributed video coding with encoder driven fusion", European Conference on Signal Processing (EUSIPCO), Poznan, Poland, September 2007.
- [17] D.E. Goldberg, "Genetic Algorithms: Search, Optimization, and Machine Learning", Addison-Wesley, Reading, MA, 1989.
- [18] P. H. Chang, J. J. Leou, and H. C. Hsieh "A genetic algorithm approach to image sequence interpolation", EURASIP Journal on Signal Processing: Image Communication, Vol. 16, No. 6, pp. 507-520, 2001.
- [19] K. Hung-Kei Chow, M.L. Liou, "Genetic motion search algorithm for video compression", IEEE Trans. Circuits and Systems for Video Technology, Vol. 3, pp. 440-445, Dec. 1993.
- [20] Chun-Hung Lin and Ja-Ling Wu, "Genetic block matching algorithm for video coding", IEEE International Conference on Multimedia Computing and Systems, Japan, Jun. 1996.