

# A GENETIC FRAME FUSION ALGORITHM FOR SIDE INFORMATION ENHANCEMENT IN WYNER-ZIV VIDEO CODING

*Charles Yaacoub<sup>1,2</sup>, Joumana Farah<sup>1</sup> and Béatrice Pesquet-Popescu<sup>2</sup>*

<sup>1</sup>Faculty of Sciences and Computer Engineering, Holy-Spirit University of Kaslik, Jounieh, Lebanon  
{charlesyaacoub, joumanafarah} @ usek.edu.lb

<sup>2</sup>Signal and Image Processing Dept., Télécom ParisTech, Paris, France  
beatrice.pesquet @ telecom-paristech.fr

## ABSTRACT

*This paper presents a novel frame fusion algorithm that aims at improving the side information in distributed video coding. Using a genetic algorithm, the proposed technique combines several interpolated frames, obtained by any existing methods, in a fusion based approach. Simulation results show a significant improvement in the side information quality, where the gain in PSNR exceeds 1dB most of the time, and can even reach 20 dB for some frames, compared to classical interpolation techniques available in the literature.*

## 1. INTRODUCTION

In traditional video coding techniques [1], encoding complexity has been a major concern for the research community during the last decade, especially for applications where power and memory resources are scarce at the encoder side, such as in wireless video sensor networks. In such applications, sufficient resources are usually available at the decoder. 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.

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 appropriate puncturing of the parity bits at the output of a channel encoder. At the receiver, 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 (R-D) performance of the distributed video codec. On one hand, it is used as a first estimate of the transmitted WZ frame, at the decoder side. 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 and motion-compensated interpolation 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...).

In this paper, we propose a novel technique for improving the side information in Wyner-Ziv video coding. Based on genetic algorithms (GA) [14], our frame fusion algorithm relies on previously developed interpolation techniques and optimally searches for the best SI candidate in different regions within a frame. In the literature, GAs have rarely been used in video applications. In [15], a GA approach is 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 [16] and [17], respectively. To our knowledge, GAs have not been used in DVC applications up to this date.

This paper is organized as follows. In Section 2, a detailed description of the proposed algorithm is presented. Simulation results are then discussed in Section 3. Finally, conclusions are drawn in Section 4.

## 2. DESCRIPTION OF THE GENETIC FRAME FUSION TECHNIQUE

A genetic algorithm [14] is an efficient searching and optimization technique, based on the principles of evolution and natural genetics. In this study, different SI frames are first generated using previously developed interpolation techniques. Actually, any SI generation technique can be considered as an input for 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. Since, in DVC, only key frames are available at the decoder, some information about the missing WZ frame needs to be transmitted to the decoder so that the fitness of a possible solution can be evaluated. For example, some statistical information (e.g., mean, variance) about the WZ frame, or a binary edge image [10], can be transmitted for this purpose. As stated earlier, the encoding complexity is a main concern in DVC. Therefore, 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 R-D performance. In hash-based DVC [6-7], a subset of the WZ frame's DCT coefficients is transmitted as a hash word to the decoder. The received hash bits can therefore be used for fitness evaluation. In our proposed GA, a hash word is computed (at the decoder side) for every chromosome in the population. The fitness of a chromosome is then evaluated as  $F = 1/D$ , where  $D$  is a distance measure, such as the mean absolute difference (MAD) or the mean square error (MSE), 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.

The proposed GA can operate at the frame level or the block level. In this paper, we consider a block-level GA for it can better adapt to local image statistics. Therefore, 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 given several SI frames obtained from different interpolation techniques. In our system, the initial candidate frames to be fed to the GA are obtained by average interpolation (AVI) [4], motion compensated interpolation (MCI) with symmetric motion vectors [4], and hash-based motion compensated interpolation (HMCI) [7]. In HMCI, the received hash word is used for block matching instead of the minimal distance measure (e.g. square error) used in MCI. 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.

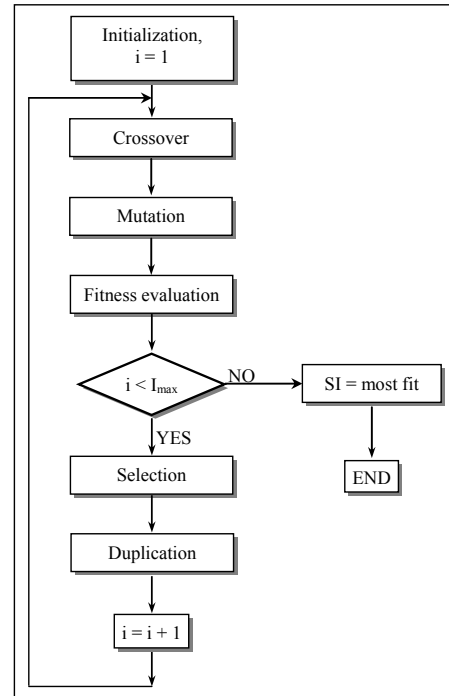


Figure 1 – Flowchart diagram of the proposed genetic frame-fusion algorithm.

#### 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). Offsprings are obtained by combining and exchanging genes between parent chromosomes. This process is illustrated in Fig.2. Given a pair of parent chromosomes (Fig.2-a and Fig.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 (Fig.2-c). Similarly, another offspring is obtained by combining the first region of the first parent with the second region of the second parent (Fig.2-d).

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  $1 - P_c$ ).

#### 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  $P_m$  [15].

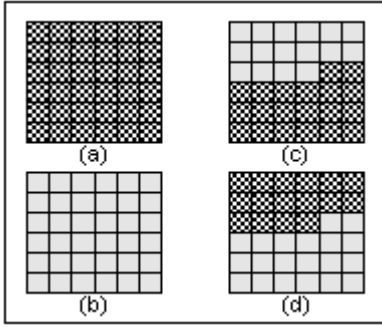


Figure 2 – Crossover example: given parent chromosomes (a) and (b), offsprings (c) and (d) are obtained after crossover.

#### Step 4: Fitness evaluation

In our GA, the MSE is 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. 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.

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.

From a complexity point of view, it may seem computationally inefficient to run the proposed GA at the decoder, since a hash word needs to be computed for every chromosome in each iteration. With the recent advances in hardware technol-

ogy 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  $S_p$  and  $I_{max}$  can be tuned accordingly. Additionally, a stopping criterion (e.g. the fitness value of the fittest 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.

### 3. SIMULATION RESULTS

In our simulations, we consider the Carphone (300 frames), News (300 frames), Foreman (400 frames) and Trevor (150 frames) QCIF video sequences, sampled at a rate of 30 frames per second, and a WZ GOP size of 2 (i.e. the number of WZ frames is half the total number of frames). The hash word transmitted to the decoder consists of  $(1/8)^{th}$  of the WZ frame's DCT coefficients, computed for blocks of dimensions  $16 \times 16$ . 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 this paper, our aim is not to show the improved R-D performance of a WZ codec due to our proposed GA at the decoder. Since, as stated earlier, by enhancing the side information, the WZ R-D performance is obviously improved, and since there are no additional tasks to be performed at the encoder for the GA to be able to run, we measure the performance of the proposed algorithm as in [18], by computing the Peak Signal to Noise Ratio (PSNR) between the SI frame generated by the GA and the corresponding WZ frame.

In Figure 3, we show the PSNR variations along the Carphone sequence obtained with the GA and the different interpolation techniques, for WZ frames 60 to 110 (we limited the plot to this interval for clarity of visualization). The superior performance of the GA can be observed all the time, except in few regions where its PSNR converges toward the best among the other curves.

Figures 4 through 7 show the cumulative density function of the difference in PSNR ( $\Delta_{PSNR}$ ) between the GA on one side, and each of the different interpolation techniques on the other, for Trevor, Foreman, Carphone and News 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 97% of frames. A gain that exceeds 1dB is observed for approximately 85% of the frames compared to AVI, 62% compared to MCI, and 14% compared to HMCI. The percentage of frames with a gain exceeding 1 dB becomes 86%, 78% and 62%, in the Carphone sequence, 77%, 42% and 10% in the Foreman sequence, and 80%, 80% and 55% in the News sequence, compared to AVI, MCI and HMCI, respectively.

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 in Trevor and News. No significant loss was observed with Carphone. For some frames in the Foreman sequence, the loss can sometimes reach 2dB. In fact, the percentage of frames with

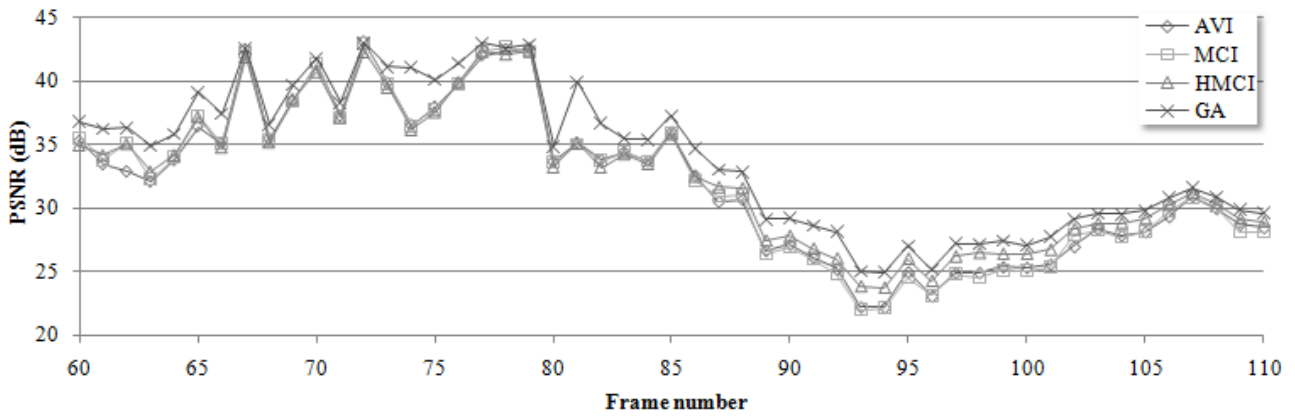


Figure 3 – PSNR variations along the Carphone sequence.

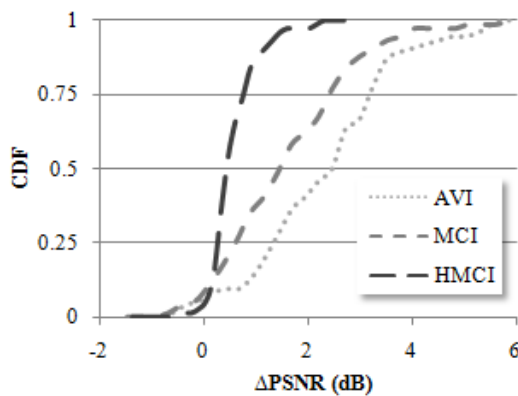


Figure 4 – CDF of  $\Delta_{PSNR}$  for the Trevor sequence.

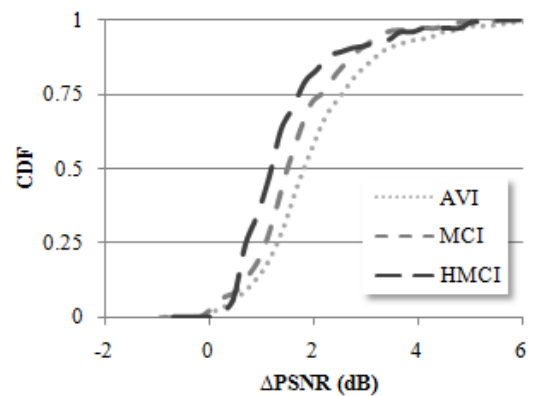


Figure 6 – CDF of  $\Delta_{PSNR}$  for the Carphone sequence.

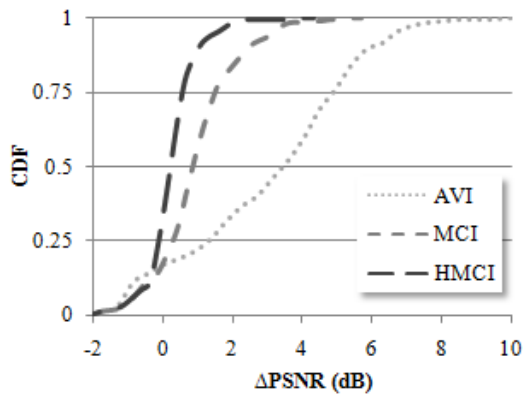


Figure 5 – CDF of  $\Delta_{PSNR}$  for the Foreman sequence.

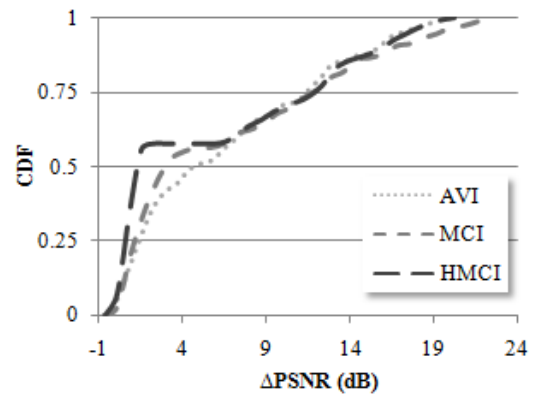


Figure 7 – CDF of  $\Delta_{PSNR}$  for the News sequence.

performance degradation of less than 1dB varies between 10% and 30%, whereas 3.5% to 5% suffer from a loss ranging from 1dB to 2dB. This is mainly due to an insufficient amount of received hash bits, given the high motion in this scene, compared to the other sequences.

In Table 1, we show the average and maximum gains obtained with the GA for the different video sequences. Minimal values can be observed with HMCI, since it offers the best average performance compared to AVI and MCI. We notice that the GA behaves surprisingly well with the News sequence, where the maximum gain exceeds 19 dB. This is

Table 1 – Average and maximum PSNR gains (in dB) for the different video sequences.

	AVI	MCI	HMCI
<b>Carphone</b>	2.0 - 6.8	1.6 - 5.5	1.4 - 5.4
<b>Foreman</b>	3.0 - 9.8	0.9 - 5.2	0.2 - 3.8
<b>News</b>	6.5 - 19.7	6.6 - 21.9	5.7 - 19.7
<b>Trevor</b>	2.3 - 5.5	1.6 - 5.6	0.6 - 2.2

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.

It is important to note that these PSNR gains for the side information may not necessarily be the same in terms of the R-D performance of a distributed video codec. However, they reflect, to a large extent, a significant improvement in the SI quality, which is expected to yield an improved R-D performance too.

#### 4. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel approach for improving the side information in distributed video coding. Based on the principles of natural genetics, the proposed genetic algorithm combines several frames obtained with previously developed interpolation techniques, in a fusion based approach. In hash-based Wyner-Ziv coding, the GA incurs no additional complexity at the encoder. Simulation results showed that most of the time, the GA yields a significant improvement in the side information generated at the decoder. While our aim was only to show the potential of using a GA in DVC, the R-D performance of a DVC codec with our proposed GA will be considered in our future work, especially 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.

#### 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*, 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*, 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*, 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] D.E. Goldberg, *Genetic Algorithms: Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] Z. Li, L. Liu, and E.J. Delp, "Wyner-Ziv video coding with universal prediction", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, pp. 1430-1436, 2006.