

ENCODER RATE CONTROL FOR TRANSFORM DOMAIN WYNER-ZIV VIDEO CODING

Catarina Brites¹, Fernando Pereira²

¹catarina.brites@lx.it.pt, ²fp@lx.it.pt

^{1,2}Instituto Superior Técnico – Instituto de Telecomunicações

ABSTRACT

Wyner-Ziv (WZ) video coding – a particular case of distributed video coding (DVC) – is a new video coding paradigm based on two major Information Theory results: the Slepian-Wolf and Wyner-Ziv theorems. Many of the practical WZ video coding solutions available in the literature make use of a feedback channel (FC) to perform rate control at the decoder which implies there must be a FC available in the application scenario addressed. The FC-based DVC solutions also have implications in terms of delay and decoder complexity since several iterative decoding operations may be needed to decode the data to the target quality level. In this context, this paper proposes an encoder rate control (ERC) solution for the transform domain WZ coding architecture previously using a FC driven rate control. Although this is the first solution in the literature, promising results are achieved with the proposed ERC solution without significantly increase the encoder complexity.

Index Terms— Wyner-Ziv, encoder rate control, transform domain

1. INTRODUCTION

Most of the existing video coding schemes, namely the popular MPEG standards, are based on an architecture where the encoder is typically much more complex than the decoder mainly due to the computationally intensive motion estimation task. The DVC approach based on the Wyner-Ziv theorem [1] (an extension of the Slepian-Wolf theorem [2] for the lossy case with side information available at the decoder) enables a flexible allocation of complexity between the encoder and the decoder. The Slepian-Wolf theorem states that it is possible to compress two statistically dependent signals, X and Y , in a distributed way (separate encoding, jointly decoding) using a rate similar to that used in a system where the signals are jointly encoded and decoded, i.e. like in traditional video coding schemes.

One of the most interesting DVC approaches is the turbo-based transform domain Wyner-Ziv (TDWZ) coding scheme presented in [3], where the decoder is responsible to explore the source statistics, and therefore to achieve compression following the Wyner-Ziv paradigm. This solution makes use of a feedback channel to perform rate control at the decoder. Depending on the application scenario addressed, DVC solutions using a FC driven rate control may not be acceptable, notably if there is no FC available. In this context, this paper proposes an encoder rate control solution for the same coding architecture previously using a FC driven rate control; from now on, the FC driven rate control will be called decoder rate control (DRC). In the ERC solution, the WZ bitrate is estimated at the encoder and sent at once to the decoder. Thus, the delay in the system and the decoder complexity are reduced since only one turbo decoding operation is needed to recover the decoded data.

In WZ coding solutions, the side information (SI) created is

interpreted as an attempt made by the decoder to obtain the best estimate of the original frame. The WZ bits are then used to improve the quality of the SI frame, by correcting the SI mismatches/errors, until a target quality for the decoded frame is achieved. Thus, the minimum amount of WZ bits the encoder needs to send to the decoder to achieve a target quality depends mainly on the SI quality and on the accuracy of the correlation noise model used to characterize the residual between the WZ and SI frames. Since the encoder has no access to the SI frame, two key questions arise in an ERC WZ coding scenario: How to estimate the SI at the encoder to derive a correlation model being aware of the encoder complexity increase this operation may bring? How should the WZ rate be allocated in order to, ideally, achieve the same RD performance obtained with DRC (the target RD performance)? The ERC framework proposed in this paper tackles these two questions with promising results. As far as the authors know, this is the first ERC solution made available for turbo-based WZ video coding. This paper is organized as follows: Section 2 presents an overview of the IST-TDWZ codec with an ERC solution. In Section 3, the major contribution of this paper, this means an ERC solution, is described. Section 4 presents experimental results. Finally, conclusions and some future work topics are presented in Section 5.

2. THE IST-TRANSFORM DOMAIN WYNER-ZIV (IST-TDWZ) VIDEO CODEC WITH ERC

Figure 1 illustrates a modified version of the IST-TDWZ video codec architecture proposed in [4] in order to allow an ERC solution; the codec in [4] follows the same architecture as the one proposed by Aaron *et al.* in [3] but uses an advanced frame interpolation (FI) framework for SI generation (for more details, see [4]). Regarding the architecture in [4], the major contribution of this paper is the encoder rate control (ERC) module.

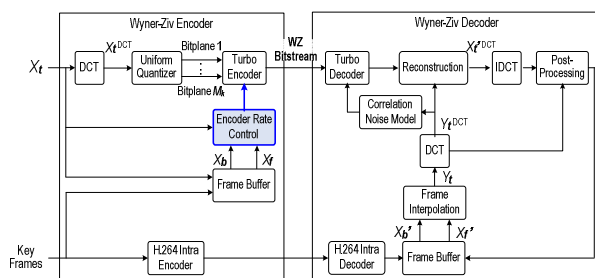


Figure 1 – IST-TDWZ video codec architecture with ERC.

The overall coding architecture illustrated in Figure 1 works as follows: the video sequence is divided into WZ frames and key frames; the key frames are H.264/AVC Intra coded. Over each WZ frame X_t , it is applied a 4×4 DCT. The DCT coefficients of the entire frame X_t are grouped together in DCT bands. Each DCT band is uniformly quantized and bitplanes are extracted and sent to the turbo encoder. The turbo coding procedure for a given DCT band starts with the most significant bitplane. Only a fraction of the parity information generated by the turbo encoder for each bitplane is sent

* The work presented here was developed within DISCOVER, a European Project (<http://www.discoverdvc.org>), funded under the European Commission IST FP6 programme.

to the decoder. This fraction is estimated for each bitplane of each DCT band by the proposed ERC module using a past X_b and a future X_f reference frames of X_t .

At the decoder, the frame interpolation module is used to generate the SI frame Y_t , an estimate of the X_t frame, based on previously decoded frames, X_b' and X_f' . For a GOP length of 2, X_b' and X_f' correspond to the previous and the next temporally adjacent key frames, X_b and X_f , after being decoded but GOPs may be longer. A 4×4 DCT is then carried out over Y_t in order to obtain Y_t^{DCT} , an estimate of X_t^{DCT} . The residual statistics between corresponding coefficients in X_t^{DCT} and Y_t^{DCT} is assumed to be modeled by a Laplacian distribution. The Laplacian parameter is estimated online for each DCT coefficient based on the residual between the frames X_b' and X_f' after motion compensation. Once Y_t^{DCT} and the residual statistics for a given DCT band are known, the decoded quantized symbol stream associated to that DCT band can be obtained through an iterative turbo decoding procedure. After turbo decoding the most significant bitplane of the DCT band, the turbo decoder proceeds in an analogous way to the remaining bitplanes associated to that band. Once all the bitplane arrays of a given DCT band are turbo decoded, the turbo decoder starts decoding the next DCT band. This procedure is repeated until all the DCT bands for which WZ bits are transmitted are turbo decoded.

After turbo decoding the bitplanes associated to a given DCT band, these bitplanes are grouped together to form the decoded quantized symbol stream associated to that band. Once all the decoded quantized symbol streams are obtained, it is possible to reconstruct the matrix of DCT coefficients, $X_t'^{DCT}$. For some DCT bands, no WZ bits are transmitted; at the decoder, those DCT bands are replaced by the corresponding DCT bands of the SI, Y_t^{DCT} . The remaining DCT bands are obtained using a reconstruction function which bounds the error between DCT coefficients of $X_t'^{DCT}$ and X_t^{DCT} (also known as reconstruction distortion) to the quantizer coarseness. After all DCT bands are reconstructed, a 4×4 IDCT is performed. The reconstructed X_t frame, X_t' , is obtained after a simple post-processing operation used to reduce block artifacts that may appear in the decoded frame if the estimated WZ bitrate is not enough or not properly distributed by the DCT bands bitplanes. The post-processing technique proposed here replaces erroneous blocks by the corresponding ones in Y_t ; a block is considered erroneous if encloses at least one pixel value outside the acceptable dynamic range ([0; 255] for 8-bit depth data).

3. ENCODER RATE CONTROL MODULE

Many of the DVC solutions available in the literature use a FC driven rate control scheme. However, some application scenarios, e.g. storage, may not benefit from this architecture since there is no FC available. To avoid the FC usage, it is proposed in this paper an ERC solution to TDWZ video coding which comprises two key modules: 1) low-complexity SI estimation module and 2) parity rate estimation module. Figure 2 illustrates the proposed ERC framework. In a nutshell, the ERC module makes use of X_b and X_f to estimate for each bitplane of each DCT band the parity rate, i.e., the fraction of parity information to be sent to the decoder in order to correct the Y_t bitplane errors. The ERC framework works as follows:

- 1) A low-complexity SI estimate, $\hat{Y}_t$, is generated at the encoder using frames X_b and X_f .
- 2) A 4×4 DCT transform is applied over $\hat{Y}_t$.
- 3) For each DCT band $\hat{Y}_t^{DCT}$ bitplane, the conditional entropy $H_{X|Y}^B$ is computed.
- 4) For each DCT band $\hat{Y}_t^{DCT}$ bitplane, the relative error probability p

is computed; p is called relative probability because it takes into account the location of the errors in previous bitplanes when computing the error probability of the current bitplane.

- 5) For each DCT band $\hat{Y}_t^{DCT}$ bitplane, the parity rate $\hat{R}_j$ is computed as a function of p and $H_{X|Y}^B$.

The modules highlighted in Figure 2 are described in the following.

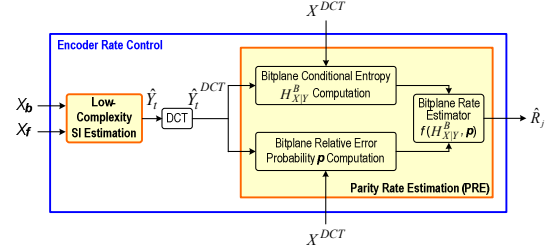


Figure 2 – Proposed encoder rate control framework.

3.1. Low-Complexity SI Estimation

The first step in the ERC framework is to obtain an estimate of SI which should allow the encoder to become more knowledgeable about its noise correlation with the decoder. At the WZ decoder, an advanced FI framework is employed to generate the SI frame [5]. However, such a complex FI process cannot be performed at the WZ encoder if the encoder complexity is to be kept low. Of course, the choice of the technique used to estimate at the encoder the SI frame significantly influences the TDWZ RD performance. The similar the encoder SI is to the decoder SI, the closer the TDWZ RD performance with ERC may be to the one obtained with DRC; of course, if the encoder could know the amount/location of the SI frame errors, it would send the exact amount of parity bits needed to correct those errors (if it could estimate them perfectly). Thus, two low-complexity FI techniques are described in the following.

3.1.1. Average Interpolation (AI)

One of the simplest FI techniques that can be used to estimate the SI at the encoder is to perform bilinear (average) interpolation between frames X_b and X_f as described in (1):

$$\hat{Y}_t(x, y) = \frac{1}{2} [X_b(x, y) + X_f(x, y)] \quad (1)$$

where $\hat{Y}_t(x, y)$ represents the value of the SI estimate at the (x, y) spatial location. Since this technique is simple, it does not provide a very good SI estimate if a complex SI estimation process is used at the decoder.

3.1.2. Fast Motion Compensated Interpolation (FMCI)

If (1) is used to estimate the SI in medium or high motion video sequences, $\hat{Y}_t$ will be a very rough estimate of Y_t since the similarity between the two frames used may be rather low, especially for longer GOPs. In this case, the encoder will send more parity bits when compared to the case where $\hat{Y}_t$ is a closer estimate to Y_t and thus the bitrate will increase for the same PSNR. This observation motivates the need to use a more powerful FI technique at the encoder; however, the encoder FI technique complexity is constrained when a low-complexity WZ encoding scenario is considered.

In this context, it is proposed in this paper a low-complexity FI technique called fast motion compensated interpolation (FMCI); the FMCI technique is inspired in a fast motion estimation algorithm proposed in [6]. In the FMCI search stage, a maximum of 3 search patterns (P) may be evaluated. Each P defines a given number of search points. For a $\hat{Y}_t$ block centered at (x, y) , a search point defines the center location of a candidate block in the reference frame X_b ;

thus, for each search point, a motion vector between $\hat{Y}_t$ and X_b blocks, $MV_b(x, y)$, is established. Assuming linear motion between X_b and X_f , if $MV_b(x, y) = (r, c)$ the corresponding motion vector between $\hat{Y}_t$ and X_f is given by $MV_f(x, y) = (-r, -c)$. The first P, P_1 , defines 5 search points: $(0, 0)$, $(\pm 1, 0)$ and $(0, \pm 1)$. Four search points are defined in the second P, P_2 : $(\pm 2, 0)$ and $(0, \pm 2)$. The third P, P_3 , defines 4 search points around each search point of P_2 . From the 16 search points of P_3 , only 12 don't overlap with prior Ps: $(\pm 3, 0)$, $(0, \pm 3)$, $(\pm 2, \pm 1)$ and $(\pm 1, \pm 2)$; the remaining search points will be skipped since they are evaluated in previous Ps. The FMCI algorithm uses the sum of absolute differences (SAD) metric as the distortion measure for each search point. In order to have low-complexity SI estimation, the FMCI technique is only applied to $T\%$ of the blocks in $\hat{Y}_t$; the remaining blocks are obtained via AI. In summary, the FMCI algorithm works as follows:

- I) The SAD metric is calculated between 8×8 collocated blocks of X_b and X_f frames; one SAD value per block is stored.
- II) The SAD values obtained in I) are sorted in a decreasing order of SAD magnitude.
- III) Motion estimation is performed for the $T\%$ of the blocks with the highest SAD values. For an $\hat{Y}_t$ 8×8 block centered at (x, y) :
 - 1) Four of the 5 P_1 search points are checked. The search point $(0, 0)$ is not checked since it was already evaluated in the step I). For the remaining search points, it is calculated the SAD between a X_b block and a X_f block. If the minimum SAD of the 5 SAD values is located at the $(0, 0)$ position, the final motion vector for the $\hat{Y}_t$ block centered at (x, y) is $(0, 0)$ and the algorithm goes to step 4; otherwise, it goes to step 2.
 - 2) The 4 P_2 search points are checked. If the minimum SAD is located in P_1 , the coordinates of the minimum SAD search point previously found correspond to the final motion vector and the algorithm goes to step 4; otherwise, the minimum SAD search point of P_2 will be the center search point for P_3 and the algorithm goes to step 3.
 - 3) Three search points of P_3 are checked around the minimum SAD search point of P_2 . The minimum SAD search point corresponds to the final motion vector.
 - 4) Once the final motion vectors MV_b and MV_f are obtained, the $\hat{Y}_t$ interpolated block centered at (x, y) is simply filled by using bidirectional motion compensation.

3.2. Parity Rate Estimation (PRE)

After estimating SI, the PRE associated to each DCT band bitplane is the second step in the ERC framework (see Figure 2). The goal of the PRE module is, given $\hat{Y}_t$, to properly allocate the parity rate at the bitplane level in order to correct the Y_t bitplane errors for each DCT band. The PRE algorithm proposed here exploits the correlation between the original bitplanes and the corresponding encoder SI estimate by computing: 1) bitplane conditional entropy (generically represented by $H_{X|Y}^B$ in Figure 2) and 2) relative error probability p . These two steps are described in the following as well as the bitplane parity rate estimator, which corresponds to the third step of the PRE algorithm.

3.2.1. Bitplane Conditional Entropy Computation

As mentioned in Section 1, the target RD performance of the ERC solution is the one obtained with DRC. To achieve this objective, it is important to understand how the decoder works in a DRC scheme.

Before turbo decoding a bitplane, it is necessary to model the residual statistics (correlation noise) between correspondent coefficients in X_t^{DCT} and Y_t^{DCT} (see Figure 1). Since X_t^{DCT} is not available at the decoder, the WZ decoder computes for each bitplane element the conditional probability of being transmitted the bit ± 1 (assuming a BPSK modulation) given Y_t^{DCT} . These conditional probabilities are then provided to the turbo decoder in order to allow the bitplane turbo decoding operation. Depending on those conditional probabilities accuracy, the decoder may request via FC for more parity bits in a WZ coding scenario with DRC; the more accurate the conditional probabilities are, the fewer the number of decoder requests is. Since the number of decoder requests, or the parity bitrate, is related to the conditional probability of transmitting a bit given Y_t , it makes sense to estimate such probabilities at the encoder to properly allocate the parity rate in a WZ coding scenario with ERC. In order to compute the j^{th} bitplane B_j conditional entropy, $H_{X|Y}^{B_j}$, the following steps are made:

- 1) The residual difference between corresponding DCT bands of X_t and $\hat{Y}_t$ is modeled by a Laplacian distribution.
- 2) For the B_j n^{th} bit of X_t DCT band X^{DCT} , $B_j(X^{DCT})$, the conditional probability p_n [7] is computed

$$p_n = \frac{p(B_j(X^{DCT})=1 | B^{j-1}(X^{DCT})=B^{j-1}(x_n^{DCT}), \hat{Y}^{DCT}=\hat{y}_n^{DCT})}{p(B^{j-1}(X^{DCT})=B^{j-1}(x_n^{DCT}) | \hat{Y}^{DCT}=\hat{y}_n^{DCT})} \quad (2)$$

where $B^{j-1}(X^{DCT})$ stands for the $(j-1)$ previous bitplanes of X^{DCT} , x_n^{DCT} represents the n^{th} DCT coefficient of X^{DCT} and $\hat{y}_n^{DCT}$ represents the n^{th} DCT coefficient of $\hat{Y}^{DCT}$. Steps 1) and 2) are similar to what is done at the decoder.

- 3) The conditional entropy of the X^{DCT} j^{th} bitplane given the $(j-1)$ previous bitplanes of X^{DCT} and $\hat{Y}$ is computed from [7]

$$H(B_j(X^{DCT}) | B^{j-1}(X^{DCT}), \hat{Y}^{DCT}) = H_{X|Y}^{B_j} \approx \frac{1}{N} \sum_{n=1}^N H(p_n) \quad (3)$$

where N is the bitplane length and $H(p_n)$ is given by (4).

$$H(p_n) = p_n \times \log_2\left(\frac{1}{p_n}\right) + (1-p_n) \times \log_2\left(\frac{1}{1-p_n}\right) \quad (4)$$

In (4), $\log_2(\cdot)$ represents the base-2 logarithmic function.

3.2.2. Bitplane Relative Error Probability Computation

Since the decoder has no access to the original frame and uses decoded versions of frames X_b and X_f to generate Y_t , the correlation noise model estimated at the decoder is typically different from the one estimated at the encoder (corresponding to steps 1) and 2) in subsection 3.2.1). This difference may lead to bitplane parity rate underestimation as the correlation noise model is used at the encoder to calculate $H_{X|Y}^B$ (3). Since parity rate underestimation leads to noticeable block artifacts in the decoded frame, another term in the bitplane rate estimator is taken into account to better allocate the parity rate. This term is called relative error probability p and reflects the observation that more parity rate than $H_{X|Y}^B$ is needed when for the j^{th} bitplane errors occur in positions for which in the $(j-1)$ previous (more significant) bitplanes didn't occur. Thus, the j^{th} bitplane p value is given by the ratio between the number of errors presented in the j^{th} bitplane but not present in the $(j-1)$ previous bitplanes and the bitplane length. The inclusion of a term on p (5) allows achieving a better parity rate estimator by reducing most of the parity rate underestimation cases.

3.2.3. Bitplane Rate Estimator $f(H_{X|Y}^B, p)$

As Figure 2 shows, the bitplane rate estimator is a function of the

bitplane conditional entropy and p . The j^{th} bitplane parity rate estimate $\hat{R}_j$, is obtained from (5)

$$\hat{R}_j = \frac{1}{2} H_{X|Y}^{B_j} \times e^{H_{X|Y}^{B_j}} + \sqrt{p} \quad (5)$$

where $e(\cdot)$ is the exponential function. The exponential term in (5) is explained by the fact that, according to experimental results, there is an exponential relation between $H_{X|Y}^{B_j}$ and the parity rate in a DRC scheme. Equation (5) was empirically obtained through exhaustive experiments over video sequences representing different motion contents.

4. EXPERIMENTAL RESULTS

To evaluate the ERC framework proposed in this paper, two QCIF video sequences are considered: Hall Monitor@15Hz and Coastguard@30Hz. In all the experiments, only the luminance data is considered for the RD performance evaluation. A GOP length of 2 is used. The FMCI algorithm is applied to 25% ($T=25$) of the blocks since $T=25$ was considered a good trade-off between encoder complexity and RD performance. The test conditions for the DCT, quantizer, frame interpolation, turbo codec and reconstruction modules are the same as in [5]. Figure 3 shows the RD performance obtained for the IST-TDWZ codec with the proposed ERC framework; it also shows the RD performance obtained if the encoder estimates the SI frame using the complex FI framework present at the decoder (represented by FI in Figure 3).

- 1) As it was expected, the FMCI algorithm proposed in this paper leads to a better RD performance than the simple average; with FMCI, the SI estimate quality is closer to the SI frame and therefore the encoder can more accurately estimate the parity rate needed to correct the SI errors at the decoder.
- 2) Comparing the ERC RD performance against the DRC RD performance, there is a gap up to 1.2dB especially for the highest quality. This gap may be justified by the fact that the proposed rate estimator sometimes overestimates the parity rate needed to correct the SI errors to avoid the undesired underestimation scenario which introduces a significant PSNR penalty.
- 3) The ERC RD performance is typically above the H.264/AVC Intra coding RD performance.
- 4) The SI encoder estimation complexity was measured regarding a DRC scheme to know its impact in terms of encoder complexity. Table 1 shows the encoder complexity increase measured as the average number of comparisons per block, which is a common complexity measure; comparison in this context refers to a block difference/addition operation. As it can be observed, the FMCI is slightly more complex than the AI but is significantly less complex than the advanced FI framework used at the decoder which may represent an acceptable encoder-decoder complexity trade-off for many applications, especially if no FC is available.

Since in a ERC WZ video coding scenario the parity rate is sent at once (at the bitplane level) to the decoder, the complex turbo decoding operation has only to be performed one time. Hence, the WZ decoder complexity is reduced at the cost of a slight increase of the WZ encoder complexity; this reflects the possibility to flexibly allocate the complexity between the encoder and the decoder characteristic of the DVC paradigm.

5. FINAL REMARKS

The ERC framework proposed here avoids the feedback channel usage in the context of many TDWZ video coding solutions reducing the system delay and decoder complexity. Although the results are

promising, further work is needed to reduce the gap between the ERC and the DRC RD performances. As future work, it is planned to study encoder rate control solutions with hash-based motion estimation at the decoder; the hash bits can be used to estimate the correlation between the original and the side information frames and thus to more properly allocate the parity rate.

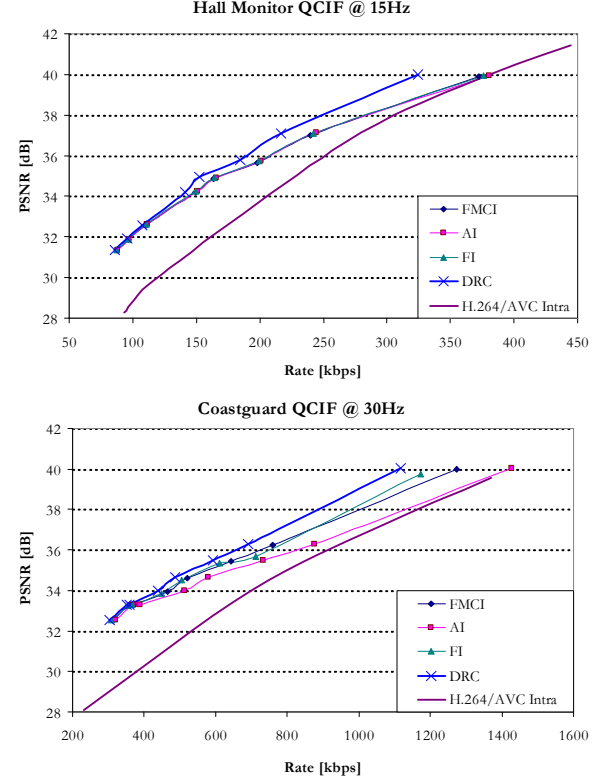


Figure 3 – IST-TDWZ RD performance for the Hall Monitor and Coastguard sequences.

Table 1 – Average number of comparisons per block.

QCIF Sequences	Frame Interpolation Techniques		
	AI	FMCI	Decoder FI
Hall Monitor	1.0	3.23	218.76
Coastguard	1.0	3.97	218.74

6. REFERENCES

- [1] A. Wyner, and J. Ziv, "The Rate-Distortion Function for Source Coding with Side Information at the Decoder", *IEEE Trans. on Inform. Theory*, Vol. 22, No. 1, pp. 1-10, January 1976.
- [2] J. Slepian, and J. Wolf, "Noiseless Coding of Correlated Information Sources", *IEEE Trans. on Inform. Theory*, Vol. 19, No. 4, pp. 471-480, July 1973.
- [3] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform-Domain Wyner-Ziv Codec for Video", *VCIP*, San Jose, USA, January 2004.
- [4] C. Brites, J. Ascenso, and F. Pereira, "Improving Transform Domain Wyner-Ziv Video Coding Performance", *IEEE ICASSP*, Toulouse, France, May 2006.
- [5] J. Ascenso, C. Brites, and F. Pereira, "Content Adaptive Wyner-Ziv Video Coding Driven by Motion Activity", *IEEE ICIP*, Atlanta, USA, October 2006.
- [6] C.-S. Yu, and S.-C. Tai, "Adaptive Double-Layered Initial Search Pattern for Fast Motion Estimation", *IEEE Trans. on Multimedia*, Vol. 8, No. 6, pp. 1109-1116, December 2006.
- [7] S. Cheng, and Z. Xiong, "Successive Refinement for the Wyner-Ziv Problem and Layered Code Design", *IEEE Trans. on Signal Processing*, Vol. 53, No. 8, pp. 3269-3281, August 2005.