ADAPTIVE HASH-BASED SIDE INFORMATION EXPLOITATION FOR EFFICIENT WYNER-ZIV VIDEO CODING

João Ascenso[1], Fernando Pereira[2]

[1] joao.ascenso@lx.it.pt, [2] fp@lx.it.pt
[1] Instituto Superior de Engenharia de Lisboa, [2] Instituto Superior Técnico, [1,2] Instituto de Telecomunicações

ABSTRACT

Wyner-Ziv video coding is a lossy source coding paradigm where the video statistics are exploited, partially or totally at the decoder. The side information represents a noisy version of the original frame and is generated at the decoder with time consuming motion estimation and compensation tools. This paper proposes a novel bidirectional hash motion estimation framework which enables the decoder to choose between past and/or future reference frames for frame interpolation. New features include the coding of DCT hash with zero-motion, combination of trajectory-based motion interpolation with hash-based motion estimation and adaptive selection of the DCT bands which are sent to the decoder in order to guide the motion estimation procedure. Gains up to 1.2 dB compared to previous motion interpolation approaches may be reached.

Index Terms— hash-based motion estimation, Wyner-Ziv video coding

1. INTRODUCTION

One of the most studied and high performance Wyner-Ziv video coding architectures [1] applies the DCT transform to the original frame X_i (i as frame number), followed by uniform quantization and turbo coding of the resulting DCT bands after sliced in bitplanes. At the decoder, an estimate of the current frame, the side information Y_i, is used by the turbo decoder and in the reconstruction process to obtain the decoded frame. The quality of Y_i is critical to obtain a high-performance Wyner-Ziv video codec, since Y_i higher quality means better correlation between Y_i and X_i which leads to a reduction in the bitrate required to achieve a target quality.

Currently, there are two major approaches to create the side information frame: trajectory-based motion interpolation (TMI) and hash-based motion estimation (HME). In the first case, the Y_i frame is inferred by tracking the motion trajectories between two (past and future) reference frames X'_B and X'_F and the estimated motion vectors are used to interpolate the Y_i frame in between. In this approach, it is common to make assumptions about the type of motion in the sequence, e.g. smoothness of the motion vector field, in order to improve the quality of the interpolated frame [2]. However, this approach has several drawbacks, notably the difficulty to interpolate frames when high and badly behaved motion occurs, when the reference frames are temporally far from each other (long GOP sizes), and when severe quantization noise occurs in the reference frames. This causes a significant decrease in the side information quality and leads to an overall RD efficiency loss.

On the other hand, it was proposed in [3-5] that the encoder sends to the decoder some auxiliary/helper information about the current frame, in order to aid the decoder in the motion estimation (ME) process. In this approach, the encoder transmits a signature or hash $H_i[l,m]$ for each (or some) block(s) with coordinates $[l,m]$, where $H_i[l,m] = t(X_i[l,m])$ and t is a function transforming the block $[l,m]$ of the original frame X_i into the hash $H_i[l,m]$. In some previous work on hash-based motion estimation, it has been proposed using CRC (*Cyclic Redundant Check*) codes [3] and high pass filters in the DCT domain [5], as the transform function to generate the hash.

In this paper, the hash function t selects a subset of the DCT coefficients in the frame, both spatially, i.e. for which blocks the hash is sent, and in frequency, i.e. which DCT bands are selected to build the hash. The selection criteria attempts to maximize the quality of the side information (therefore reducing the parity bits to be sent) while minimizing the rate spent on the hash bits. Therefore, the major contributions of this paper, which depart from previous related work published in the literature [4, 5] are:

i) Adaptive DCT band selection (Section 3.1): Since there are some bands which have a higher discriminative power than others, an adaptive scheme is proposed. This technique selects, for each frame, the DCT bands used as hash in order to maximize the quality of the side information while maintaining the hash rate as low as possible.
ii) Bidirectional motion estimation (Section 3.3): In this procedure, it is chosen for each Y_i block one candidate block from the backward or forward frames, or a linear combination of blocks from both references. This is accomplished without sending any coding mode information, as in traditional coding schemes.
iii) Combination of trajectory-based motion interpolation with hash-based motion estimation (Section 4): Since there are strengths in both approaches and it is possible to combine them in an intelligent way by selecting for each block the best interpolation mode, it is also proposed that the decoder reuses the motion vectors calculated in the trajectory-based motion interpolation in the hash-based motion estimation.

Finally, this paper also describes in detail the motion estimation procedure, namely the matching criteria (Section 3.2) used to select from the candidate blocks the most similar one. The experimental results (Section 5) show improvements up to 1.2 dB compared to the previous TMI approach [2] when these techniques are used.

2. HASH-BASED WYNER-ZIV VIDEO CODEC

The hash-based coding solution proposed in this paper enhances a previous solution proposed by the same authors in [2] where an advanced motion interpolation framework is used to generate the side information (with no hash data). Figure 1 illustrates the new overall coding architecture. The shaded modules correspond to new blocks that were added to perform hash-based motion estimation for all or some blocks of the side information frame.

The coding process starts by dividing the video frames into keyframes and Wyner-Ziv (WZ) frames. The keyframes are the first frames sent from the encoder to the decoder for each GOP and are encoded using the H.264/AVC Intra coding scheme.

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

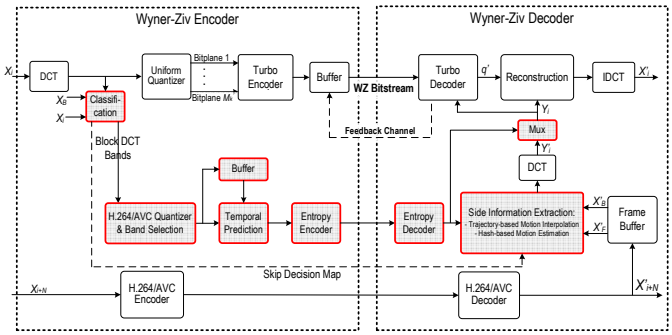

Figure 1 – *Architecture of the hash-based WZ codec.*

At the encoder, each WZ frame is coded by applying the 4×4 H.264/AVC DCT transform and the DCT coefficients of the entire frame X_i are grouped together in DCT bands. Each band is uniformly quantized and bitplanes are formed and sent to the turbo encoder. The turbo coding procedure for the DCT coefficients band starts with the most significant bitplane and generates the respective parity sequences which are stored in the buffer and transmitted in small amounts upon decoder request. A simple classifier detects regions where significant motion has occurred based on the backward frame X_B and selects for which blocks the DCT hash should be calculated and sent to the decoder. Then, for the chosen blocks, DCT bands are constructed with the quantized DCT coefficients. Some of the bands are selected to construct the hash and sent before the WZ bitstream in order to help the side information creation at the decoder. In order to reduce the hash bitrate, a temporal prediction module which performs DPCM coding of the hash with respect to the previous hash sent to the decoder (i.e. zero coding of DCT bands) and entropy coding of the residual DCT hash at the band level are used. Since it is also necessary to signal to the decoder for which blocks the hash is sent, a binary map signaling the respective positions is encoded with run length encoding (RLE) and entropy coding. These new encoder operations to construct the DCT hash do not significantly increase the Wyner-Ziv encoder complexity when compared to the H.264/AVC Intra encoder.

At the decoder, the first step is to perform entropy decoding of the hash bits sent from the encoder. Then, the side information extraction module can generate the side information Y'_i, an estimate of the X_i frame, based on two references, one temporally in the past X'_B and another in the future X'_F. Each Y'_i block can be generated in one of two ways: if a DCT hash is available to guide the motion estimation process, HME is performed; otherwise, TMI is used. In order to further improve the Y'_i quality, the Y'_i frame is DCT transformed and the DCT bands for which hash bands are available are substituted at the multiplexer by the corresponding bands sent from the encoder, generating Y_i. This enables to reduce the bitrate needed to correct the errors (in this case only quantization errors can occur) in the coefficients for which a DCT hash has been received. Then, the iterative turbo decoder uses the received parity bits to correct Y_i errors and generate the decoded (almost error free) quantized symbol stream q'. Y_i is also used in the reconstruction module, together with q', to obtain the decoded DCT bands; finally, IDCT is applied to generate the WZ decoded frame X'_i.

3. DCT HASH-BASED MOTION ESTIMATION

The key idea in hash-based motion estimation is to construct the side information with some helper information from the encoder. To accomplish this, it is necessary to find a transform function t which produces a signature or a hash $H_i[l,m]$ that maximizes the side information quality with the lowest possible rate. The transform function t has thus some desired properties such as: *i*) Energy compaction: the function t should generate a hash $H_i[l,m]$ which can be compacted and represented with a low amount of bits; ideally, the hash bits are equivalent to the motion vectors used in traditional video coding and should occupy the same rate for a certain target quality of the side information. *ii*) Robustness: $t(X_i[l,m]) \approx t(X_k[p,q])$ if the block $X_i[l,m]$ is similar to block $X_k[p,q]$; *iii*) Collision avoidance: The hash $t(X_i[l,m])$ should be uncorrelated with $t(X_k[p,q])$ if $X_i[l,m]$ and $X_i[p,q]$ are different from each other. These last two properties enable the use of the hash $H_i[l,m]$ in a full search block based motion estimation process where one block is selected from a high number of candidates blocks. A final constraint should be placed on the complexity of the transform function t which should be as low as possible in order to maintain a low complexity encoder. The DCT transform shares some of these properties since most of the signal information is concentrated in a few low-frequency components, has good robustness in the presence of noise and also low complexity. The other major advantage regards the fact that these coefficients can be reused in the side information transformed frame in order to further enhance its quality.

3.1. Adaptive DCT band selection

The DCT hash proposed in this paper is similar to [4, 5] in the sense that a subset of the original DCT coefficients is sent to the decoder to guide the motion estimation process. In order to minimize the hash rate, coefficients must be quantized and only a continuous subset is transmitted to the decoder, which corresponds to: $t = Q(DCT_{[s,e]}(\,.\,))$, where s and e correspond to the start and end DCT hash bands. Using this proposed DCT hash implies solving critical issues such as which frequencies (or bands s to e) are necessary to be sent in order to achieve a reliable estimation of the side information frame and which quality (or quantization step) should be used to encode the selected DCT coefficients.

Regarding the first question, the authors have investigated several combinations to identify the best DCT hash in the RD sense. Some possibilities are to use the DC band or alternatively some of the AC bands. In the literature, the use of AC bands has been suggested in [5] ([4] omits which bands were used); however, this has several disadvantages. In order for the DCT hash to be strong enough to discriminate among a high number of candidate blocks, it is necessary to quantize the high frequency coefficients with a low step size (when compared to low frequency ones), especially at medium and low bitrates. However, this introduces noticeable artefacts, since the quantization tables would not reflect the human visual perception system, and errors in the low frequencies would be easily perceived. It is also difficult to obtain a good performance for small block sizes transforms (e.g. 4×4) which pack most of the energy in the low frequency coefficients. Additionally, when the reference frames have significant quantization noise, as in the case of low bitrates, the AC bands have a significant amount of zero coefficients and capture all the quantization noise; this leads to low quality side information.

Taking into account these observations, the authors propose to use a low frequency hash, where the DC coefficient is combined with some of the next AC bands in zig-zag scan order (i.e. $s = 0$ is adopted). Regarding the quantization, it is proposed to use uniform quantization (with QP_h) of all DCT hash coefficients, with the same QP as for the keyframes (QP_I), which assures a constant quality between WZ and the Intra keyframes, i.e. $QP_h = QP_I$. Therefore, the DCT hash proposed here can be interpreted as a low pass filter representation of the image with a certain cut-off frequency λ, which is closely related with the number of bands sent. However, an important issue remains: the computation of the "optimal" number of bands, or the cut-off frequency of the low pass filter which should depend essentially on the frequency characteristics of each frame and the quality of the side information necessary to be provided for a

given RD point. Therefore, it is proposed here an adaptive method which selects for each frame the optimal number of bands e according to the energy distribution of each DCT band and the selected QP_h used to quantize the DCT hash bands. In this method, the DC coefficient is always sent; since the DCT block size is small, this coefficient is very important to distinguish each block in a robust way, especially at low bitrates where the hash bitrate must be kept small. For medium to high bitrates, it is necessary to include some AC bands in the hash to achieve a higher precision in the motion estimation. To decide, for each frame, how many AC bands must compose the DCT hash, it is proposed to maximize:

$$\arg\max_{e}\left\{\sum_{b=1}^{e} E[b] \le \delta\right\} \quad (1)$$

where E[b] is the energy of each DCT band $b = 0 \ldots 15$ in zig-zag scan order and

$$E[b] = \sum_{l=0}^{N_l}\sum_{m=0}^{N_m}\left(DCT_b[l,m]\right)^2 \quad (2)$$

and N_l, N_m represent the number of blocks per row and column for which the DCT hash is sent. δ in (1) represents the amount of energy necessary to maximize the quality of the side information with the lowest possible rate. After intensive experimentation, it was found that the following model for δ performs well:

$$\delta = \left(QP_h\alpha - \beta\right)\sum_{b=1}^{15} E[b] \quad (3)$$

Above ($QP_h\alpha - \beta$) is always ≤ 1 and represents the fraction of the total energy that should be allocated, given the hash quality QP_h. α and β are two constants found by linear regression with representatives frames of each sequence with the optimal δ found by intensive evaluation. This technique incorporates in a simple way a rate-distortion model, where the rate is represented by the energy of each band and the distortion is measured by αQP_h in a similar way to the rate-distortion models traditionally used in video coding.

3.2. Hash Matching Criteria (Decoder)

One important issue in the design of the hash-based motion estimation algorithm is the matching criteria to measure the similarity between two blocks. There are several possibilities, the most straightforward one is to perform inverse quantization of the DCT hash coefficients to obtain a coarsely reconstructed frame to guide the motion estimation; in this case, the SAD (*Sum of Absolute Differences*) criteria between the reconstructed and candidate pixel blocks should be minimized. Another possibility is to calculate the DCT hash for each candidate block and compare it with the DCT hash received; in this case, ME is performed in the DCT hash domain, where the SAD between the hash received and the hash of each candidate block is calculated; the candidate block with minimum SAD is used to construct the side information. This last approach forces to perform the function t for each candidate block at the decoder which adds complexity to the ME procedure. However, it presents an increased performance when compared to the previous approach. For complexity reasons, the solution here proposed is to perform inverse quantization IQ of the DCT hash $H_i[l,m]$ received from the encoder and then find the motion vector by minimizing:

$$\arg\min_{k}\left\{\left\| IQ\left(H_i[l,m]\right) - DCT_{[s,e]}\left(Z_{ik}[l,m]\right)\right\|\right\} \quad (4)$$

where $Z_{ik}[l,m]$ represents all possible candidate blocks k, for the block [l,m] in frame i, and $\| \cdot \|$ is the SAD criteria. With this new approach, it is possible to avoid the quantization of all candidate blocks, which can save processing time at the decoder. It can also improve the quality of the side information since the SAD criteria works with reconstructed DCT coefficients which will allow better precision when compared to the previous approach where quantized values (in the candidate and reference blocks) are used in the matching criteria.

Another important issue is the block size used in the motion estimation process. Larger block sizes can bring more robust matching but will not account for complex motion, which is more accurately tracked with small block sizes. Therefore, it is proposed here to perform motion estimation with 8×8 block sizes, which is a good balance between robustness and prediction accuracy. Since a 4×4 DCT transform is used, the DCT hash of each 4×4 block inside the 8×8 blocks is used in the following way in the matching criteria:

$$\arg\min_{k}\left\{\sum_{a=0}^{1}\sum_{b=0}^{1}\left\| IQ\left(H_i[l+a,m+b]\right) - DCT_{[s,e]}\left(Z_{ik}[l+a,m+b]\right)\right\|\right\} \quad (5)$$

where a, b are the indexes of each 4×4 DCT block inside the 8×8 block used by the motion estimation procedure.

3.3. Bidirectional Hash-based Interpolation (Decoder)

Previous approaches in the literature [3-5] use the hash bits only in an extrapolation framework where the candidate blocks are only obtained within a search region of the previously decoded reference frame. However, it is expected that if the DCT hash is strong enough i.e. good robustness and collision avoidance, the construction of candidate blocks from both reference frames in an interpolation framework (where two reference frames are available) can further enhance the coding performance. One possibility is to add block modes, as in traditional video coding, especially the most efficient one, the B-mode, where a block can be predicted from the past and/or future reference frames. Therefore, it is proposed here to use the following interpolation modes for a [l,m] 8×8 block:

i) Mode P: Within the search range, obtain 8×8 candidate blocks in the backward and forward reference frames; include all blocks in the set of candidate blocks Z_{ik}.
ii) Mode B1: Find the best blocks B_B, B_F according to (5) for each reference frame available at the decoder X'_B, X'_F, and calculate the linear combination (average) of blocks B_B and B_F; include this interpolated block in the set of candidate blocks Z_{ik}.
iii) Mode B2: In case trajectory-based motion interpolation has been performed, use the motion vector already available and include in the set of candidate blocks Z_{ik} the linear combination of the blocks in X'_B, X'_F to which the motion vector is pointing; include also in Z_{ik} some neighboring blocks within a small search range assuming linear motion, to account for slight displacement errors in the trajectory-based motion interpolation.

The bidirectional ME procedure consists in evaluating (5) for the blocks included in Z_{ik} through the various modes above and use the best block obtained to construct the side information frame. The modes B1 and B2 contribute significantly for the improvement of the side information, especially in occluded areas. It is also important to note that if the hash is strong enough, the inclusion of the block mode B2 guarantees that the block obtained with HME is always a better prediction when compared to the block obtained with the TMI technique.

4. BLOCK MODE SELECTION

In the hash-based motion estimation previously described, hash bits are sent for all the blocks in the current frame in order to perform motion estimation. However, only a small number of blocks have large displacement in most video sequences since many have motion vectors equal to zero (background areas) or with small magnitudes. For this type of blocks, it is not necessary/efficient to send any hash information since the trajectory-based motion interpolation can make a good estimation. The combination of both approaches can bring

major benefits since some rate can be saved by sending the hash only for some blocks. However, the encoder has the additional burden to select the blocks in the frame for which hash codes have to be sent. Ideally, the encoder should generate for each block the side information using both approaches and select the one that minimizes some rate distortion criteria. However, due to complexity constraints, this type of technique cannot be used and therefore it is proposed here to use SAD_o (*Sum of Absolute Differences at the search origin, 0*) between the current frame X_i and the backward reference frame X_B.

$$SAD_o[l,m] = \sum_{(x,y)\in B_i[l,m]} |X_i(x,y) - X_B(x,y)| \quad (6)$$

This metric tries to reflect the degree of motion for a block in a simple way: a large SAD_o indicates a large change between the current and the backward reference frames and thus there is a high probability that the block has a large motion. For each 8×8 block $B_i[l,m]$ of the original frame, if $SAD_o[l,m] \geq T$, the encoder sends a DCT hash for all four 4×4 blocks of the 8×8 block. An adequate threshold T has been found experimentally.

5. EXPERIMENTAL RESULTS

This section evaluates the performance of the novel contributions made in this paper as stated in Section 1. In all experiments, the GOP size is 4: the middle frame in the GOP is always sent first, and uses as references the already decoded Intra frames; the next B frames sent use the adjacent I or B frames.

Assessing the bidirectional hash-based interpolation: The first experiment compares the quality $PSNR_P$ of the side information obtained with the P mode only (candidate blocks from the backward or forward reference frames) with the quality $PSNR_B$ using also the B1 and B2 modes, which includes linear combinations of blocks in both reference frames. In this experiment, a DCT hash with 1 DC + 2 ACs bands was always used and all blocks are HME interpolated; the QP used to quantize the H.264/AVC Intra keyframes is 23 and the WZ frames are quantized according to the last quantization matrix in [6] (highest quality). As observed in Table 1, the quality of the side information increases significantly with the addition of the new B modes, especially for the Coastguard and Foreman sequences. The rate shown includes only the DCT hash rate; it is always the same since what changes here is the number of candidate blocks at the decoder. It is also shown how often the P mode (%P) is selected against both B1 and B2 modes (%B): the B modes are selected very often which justifies the PSNR increase in the side information quality. Based on this conclusion, the next experiments always use bidirectional ME.

Assessing the adaptive DCT band selection: Next, it is evaluated the overall RD performance (including both WZ and hash bitrates) of the adaptive DCT band selection algorithm. It is also presented the case when only the DC band, DC + 2AC bands and the DC + 4 AC bands are sent as hash. Figure 2 shows the importance of each band in the Wyner-Ziv codec performance. As shown, the best performance is achieved by sending DC + 2AC bands for the Foreman sequence, while the proposed adaptive DCT band selection algorithm comes very close. Since for other sequences the "optimal" number of bands is not the same as found for Foreman sequence (DC+2AC), and the adaptive algorithm always performs close to the "optimal" solution, it may be concluded that the adaptive DCT band selection is overall the best approach.

Assessing the combination of TMI with HME: As observed in Figure 3, the TMI has higher efficiency when compared to HME at low bitrates since the bitrate of the hash is significant in these cases; at higher bitrates, the HME has a slight advantage since the hash bits provide an improved side information quality. The best performing solution is the combination of both approaches as proposed which can bring improvements up to 1.2 dB when compared to the TMI approach.

Table 1 – *Improvements with bidirectional hash-based interpolation.*

Sequence	Rate [kbps]	$PSNR_P$[dB]	$PSNR_B$[dB]	%P	%B
Foreman	322.43	33.63	35.43	42	58
Coastguard	322.45	31.73	34.08	33	67
Soccer	378.58	30.86	31.51	42	58
Hall	148.48	37.50	37.89	74	26

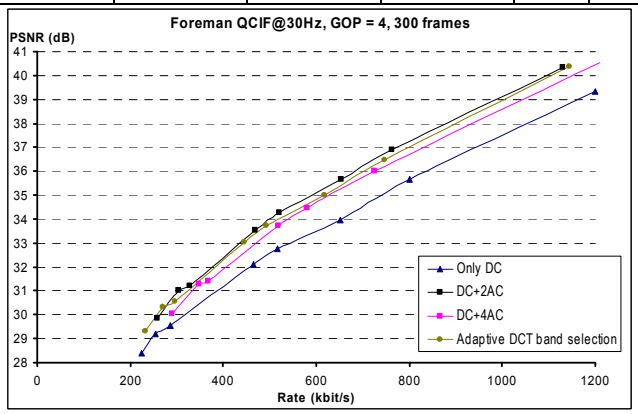

Figure 2 – Adaptive vs. fixed DCT band selection.

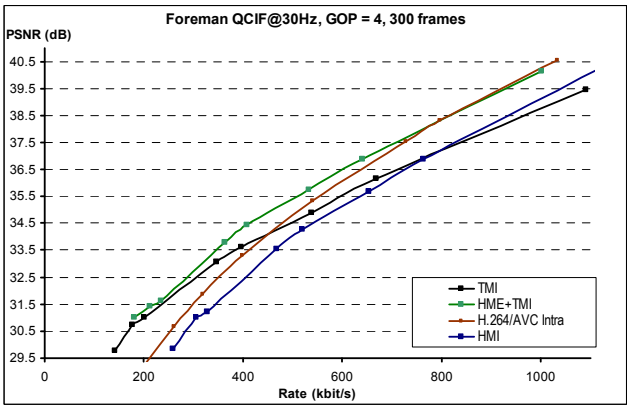

Figure 3 – TMI vs. HME vs. combination of TMI and HME.

6. FINAL REMARKS

In this paper, a bidirectional hash motion estimation framework including an intelligent way to select for which blocks the hash is sent, and how many low frequency bands are necessary to achieve the best performance is proposed. As future work, it is planned to study the effects of sub-pel motion estimation and multiple references on the RD performance of the HME and hybrid schemes.

7. REFERENCES

[1] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed Video Coding", *Proceedings of the IEEE*, Special Issue on Video Coding and Delivery, Vol. 93, No. 1, pp. 71-83, January 2005.

[2] J. Ascenso, C. Brites, and F. Pereira, "Content Adaptive Wyner-Ziv Video Coding Driven by Motion Activity", *ICIP*, USA, October 2006.

[3] R. Puri, K. Ramchandran, "PRISM: A "Reversed" Multimedia Coding Paradigm", *ICIP*, Barcelona, Spain, September 2003.

[4] A. Aaron, S. Rane, and B. Girod, "Wyner-Ziv Video Coding with Hash-Based Motion Compensation at the Receiver", *ICIP*, Singapore, October 2004.

[5] A. Aaron, B. Girod, "Wyner-Ziv Video Coding with Low-Encoder Complexity", *PCS*, San Francisco, USA, December 2004.

[6] C. Brites, J. Ascenso, and F. Pereira, "Improving Transform Domain Wyner-Ziv Video Coding Performance", IEEE ICASSP, May 2006, Toulouse, France.