On Side Information Generation for Wyner-Ziv Video Coding

Eduardo Peixoto, Ricardo L. de Queiroz and Debargha Mukherjee

***Resumo*— A codificação distribuída de vídeo (DVC) é um novo paradigma de codificação de vídeo, que permite a reversão de complexidade de um codec, na qual a complexidade é transferida do codificador para o decodificador. Muitas implementações de DVC precisam realizar uma predição do quadro codificador no lado do decodificador, utilizando apenas os outros quadros como referência, em um processo chamado geração de informação lateral. Este processo é crucial para qualquer implementação DVC - uma melhor informação lateral irá levar a um melhor desempenho em relação taxa-distorção. Neste trabalho, propomos um modelo de movimento linear para produzir informação lateral associada, e modos para lidar com os problemas comuns deste modelo. Resultados são mostrados para verificar o desempenho do método proposto.**

***Palavras-Chave*— Informação Lateral, Codificação de Vídeo Wyner-Ziv, Interpolação de Quadros por Compensação de Movimento.**

***Abstract*— Distributed video coding (DVC) is a new video coding paradigm, which allows a reverse complexity codec in which the complexity is shifted from the encoder toward the decoder. Many DVC implementations have to make a prediction of the encoded frame at the decoder side using only the other frames as reference, in a process called side information (SI) generation. This process is crucial to any DVC framework - a better SI will lead to a better rate distortion performance. In this work, we propose a linear motion mode to produce SI associated with, and a way to deal with the common problems of this model. Results are presented to support the proposed method.**

***Keywords*— Side Information, Wyner-Ziv Video Coding, Motion Compensation Frame Interpolation.**

I. Introduction

The traditional video compression solutions are based on the hybrid architecture of block-based motion compensated prediction and on a spatial transform followed by entropy coding of the residue and related information [1]. The goal of these tools is the reduction of spatial and temporal redundancy. Codecs that uses this scheme are more complex at the encoder side, where most of the computation is done to decide the best way to encode each macroblock [2]. The decoders, however, are less complex, just performing entropy decoding, motion compensation and inverse transforms.

Based on the fundamental theorems of Slepian-Wolf [3] and Wyner-Ziv [4], distributed video coding (DVC) has been the focus of many studies in the recent years. DVC allows us to explore the video statistics at the decoder side, resulting in a less complex encoder and more complex decoder. Even though it has been acknowledged that "it is unlikely that a distributed video coding algorithm will ever beat conventional video schemes in rate-distortion performance" [5], DVC is suitable for other kinds of applications, such as mobile video communications and surveillance wireless cameras. In these, conventional solutions fail to achieve an optimal performance.

In order to explore temporal redundancy, traditional codecs use motion estimation and compensation techniques to form a prediction of the current frame based on the previously reconstructed frames (past and/or future frames in exhibition order, provided they have been already encoded). Advanced techniques of motion estimation, such as quadtree partitioning, multiple reference frames and sub-pixel spatial interpolation filters, were successfully used by state of the art video compression standards to significantly improve the performance of these codecs.

A common and simple architecture for a pixel-domain DVC codec is described here. The odd frames are called key frames and are encoded with a regular intra encoder, while the even frames are called Wyner-Ziv (WZ) frames and are encoded with a channel coding technique, such as systematic turbo codes [6]. However, only the parity bits are sent to the decoder. At the decoder, the key frames are used to generate a prediction to the current WZ frame, called side information (SI), which is fed to the turbo decoder along with the parity bits sent by the encoder. The turbo decoder then uses the parity bits to correct eventual errors between the actual WZ frame and its SI. The SI can be generated using motion estimation and compensation techniques on neighbor frames of the current frame, thus exploring temporal correlations among them. A better side information might result both in a lower rate (less parity bits need to be sent) and in a better quality, which is why SI generation is a crucial process to any DVC codec.

The main focus of the paper is to develop a new method to generate side information, and to evaluate this method against current established methods.

II. Wyner-Ziv Codec used

To fully investigate the performance of SI generation methods, we used a simple pixel-domain WZ codec [7], whose architecture is shown in Figure 1.

At the encoder, the odd frames are labeled as key frames and encoded with a regular intra-frame encoder (H.263 INTRA) [8]. The even frames are labeled as WZ frames and are encoded as follows. The difference between the current WZ

Eduardo Peixoto and Ricardo L. de Queiroz are with the Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, Brasil. Debargha Mukherjee is with Hewlett Packard Laboratories, Palo Alto, USA. E-mails: eduardo@image.unb.br, queiroz@ieee.org, debargha.mukherjee@hp.com. This work was supported by HP Brasil and by CNPq under grant 47.3696/2007-0.

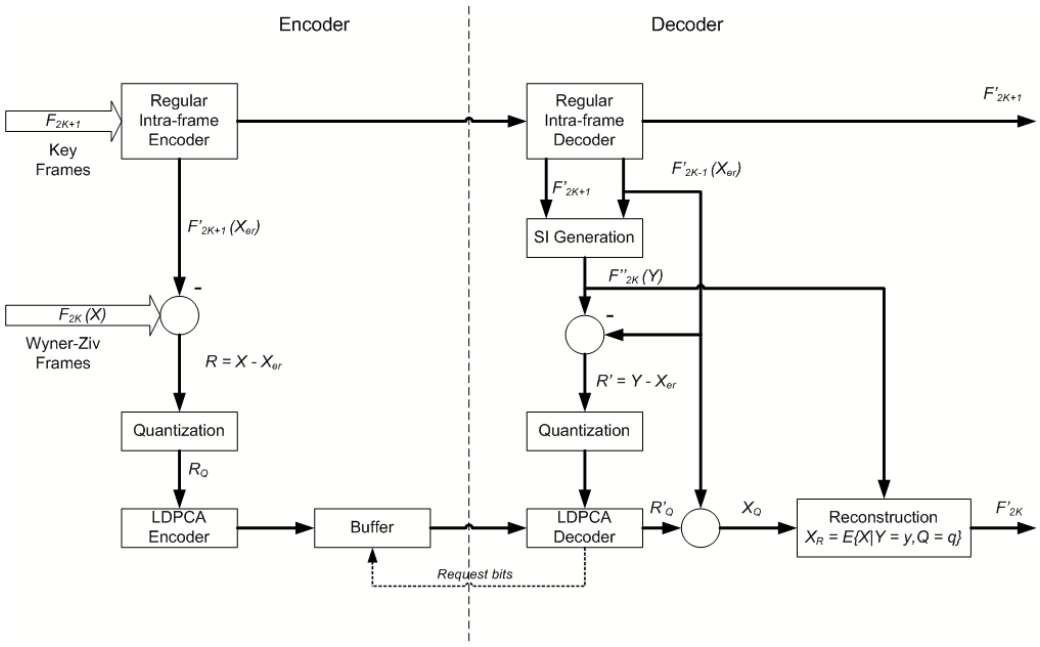


Fig. 1. WZ codec architecture.

frame, X, and the encoder reference frame, X_{er}, is fed to the WZ encoder. X_{er} is taken as the previous reconstructed key frame, which is available both to the encoder and the decoder, in order to avoid drifting errors. The pixels of the residual frame $R = X - X_{er}$ are quantized using an uniform scalar quantizer, generating R_Q, which is then encoded by a low-density-parity-check accumulated (LDPCA) code [9].

At the decoder side, the key frames are decoded with a regular intra-frame decoder. The decoder generates the side information Y with one of the processes described in Section III. The residual side information $R' = Y - X_{er}$ is fed to the LDPCA decoder to generate the residual frame R'_Q, which is then added to X_{er} to find X_Q. The final step of the decoding process is the reconstruction of the pixel values, which is taken as $X_R = E(X|X_Q = x_Q, Y = y)$. The details of the reconstruction process are described elsewhere [10].

The side information quality will influence the codec's performance in two ways. First, it will drive the rate of a WZ frame (a better SI frame will result in a lower $H(X|Y)$), and second, it will help the reconstruction process to get a better estimation of a given pixel.

III. Side Information Generation

Since side information is crucial for any WZ codec, there are many studies about it and its impacts on WZ coding. Currently, there are two main different classes of techniques: hash controlled motion estimation and motion compensated frame interpolation (MCFI). In the former class, the encoder transmits some additional information to help the decoder with the work of generating the SI. Cyclic Redundancy Check (CRC) codes are used at the PRISM codec [11] as the additional information sent by the encoder. This procedure is helpful specially when intense, complex motion occurs, allowing the decoder to generate more reliable SI in this case. However, a rate penalty is paid to send the information. There is a small increase of encoder complexity.

In the later class, the SI frame is generated without any help from the encoder, by tracking the motion trajectories between the reference frames. A comparison between these two types of SI generation was previously made [12]. However, in this paper, we consider the case when the decoder does not have any help to generate the SI, so the hash-controlled motion estimation will not be further considered.

Several techniques were proposed for the MCFI case. Typically, these techniques attempt to find a coherent motion field to express the true motion in the sequence. Some proposed techniques include (i) the use of 3D recursive block matching in an attempt to find the true motion field [13], (ii) using mesh based motion estimation and interpolation to better represent the motion field [14], (iii) using the motion vectors calculated from previously decoded frames as the motion vectors for the current frame [15] and (iv) using a Kalman filtering approach to update these motion vectors for the current frame [16]. Even techniques such as universal prediction were attempted, in which a pixel is predicted by using a weighted average of previous occurrences in the same spatial position.

The technique proposed here belongs to the MCFI class. It will be compared to another consolidated method that will serve as reference to evaluate the SI generation performance.

A. SE-1 and SE-B

First, we consider the approach proposed by Li et al [15]. The main idea of this approach is to model the motion vectors for the current WZ frame using the motion vectors calculated in the previous frames. Several side estimators were proposed in this work. The simplest method, called SE-1, works as

follows: for each macroblock in the current frame F_{2k}, the side estimator uses the co-located block in the previous frame F_{2k-1} as the source and searches the second previous key frame F_{2k-3} for a best match for this block. Then, this motion vector is applied in the previous frame F_{2k-1} to find a reference for the current block. In fact, it models the motion vector for the current frame as the motion vector for the co-located block in the previous frame. Although this is clearly not the best option, it is better than to use the previous frame as the side estimator for the current frame.

In the same work [15], a better side estimator was also proposed, called SE-B, which works similarly to B-frames. This side estimator first uses the previous reconstructed frame F_{2k-1} as the reference and the next (future, but previously decoded) reconstructed frame F_{2k+1} as the source to calculate the forward motion vectors MV_F. Then, it uses F_{2k+1} as the reference and F_{2k-1} as the source to calculate the backward motion vectors MV_B. It then uses $\frac{MV_F}{2}$ on F_{2k-1} to generate a frame P_F, and uses $\frac{MV_B}{2}$ on F_{2k+1} to generate a frame P_B. The final side estimation frame is obtained as the mean between P_F and P_B. In fact, it models the motion vectors for the current frame F_{2k} as half the motion vector for the next frame F_{2k+1} with F_{2k-1} as the reference. The same principle is applied to the backward set. However, as the current frame is not available to decide which motion vector is best, $\frac{MV_F}{2}$ or $\frac{MV_B}{2}$, then both are averaged. This method is depicted in Figure 2.

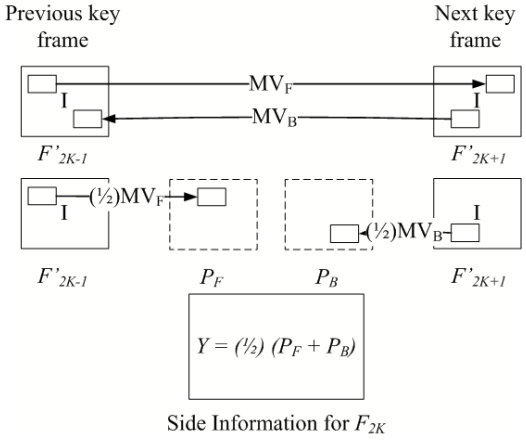


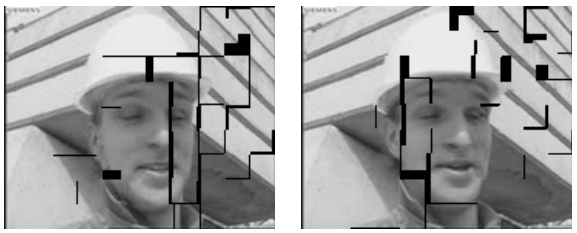
Fig. 2. Illustration of SE-B.

B. Proposed method

The proposed method models the movement between two key frames F_{2k-1} and F_{2k+1} as linear. Thus, the movement between F_{2k-1} and the current frame F_{2k} will be half of the movement between F_{2k-1} and F_{2k+1}. For a given macroblock in F_{2k+1}, it searches the reference frame F_{2k-1} to find the best match, named the reference block. This reference block is kept and translated by $\frac{MV_F}{2}$. This approach will lead to two phenomena that did not happen in the SE-B method: overlapping areas and blank areas. There are three cases for a given pixel:

- It is uniquely defined by a single motion vector;
- It is defined by more than one motion vector (an overlapping occurred); and
- It is not defined by any motion vector (it is left blank).

In order to assign a motion vector or filling process for every pixel, the first case is trivial. For the second case, when more than one option for a pixel exist, a simple average might solve the problem. The last case is more challenging. If no motion vector points to a pixel, then it is not easy to guess its value. One could use the co-located pixel in the previous frame. However, it may not be very efficient since it might be that the motion vector of that block is not zero. In Figure 3(a) it is shown the second frame of the *Foreman* sequence using $\frac{MV_F}{2}$. In this case, the key frames were coded with H.263 INTRA with $QP = 4$. The overlapping areas were averaged and, as expected, there are some blank areas. However, in Figure 3(b) it is shown the frame using $\frac{MV_B}{2}$. Again, there are some blank areas, but most of them are in different places.



(a) SI Frame generated with $\frac{MV_F}{2}$ (b) SI Frame generated with $\frac{MV_B}{2}$

Fig. 3. Generating the SI frame.

So, combining the frame generated by the forward estimation with the one generated by backward estimation will result in a frame with less blank areas, which is depicted in Figure 4.

Fig. 4. Combining the frames in Figures 3.

After the motion estimation and compensation, and after averaging the overlapping areas, the SI frame already looks like a frame, with some blank areas. At this point, there is enough information available about the current frame to perform motion estimation using the current SI frame and the previous frame F_{2k-1}. The current frame is divided into blocks 16×16 pixels. Then, if there is a blank area in a macroblock, a motion estimation will be performed for this macroblock. The blank area will not be considered when calculating the SAD.

Once the new reference block is found, it will be the block in the previous frame that best represents the current information we have about that area. So, the pixels in the reference block will fill the blank area in the current macroblock. This is shown in Figure 5.

Fig. 5. Performing motion estimation using the current SI frame.

To improve the method, bi-directional motion estimation is performed. To fill the blank areas, a reference block will be searched in both the previous and next frames. The result for this single frame is shown in Figure 6.

Fig. 6. Final SI frame. PSNR = 33.38 dB.

Note that, in the proposed method, the reference block found using the motion estimation process is kept and translated to the SI frame by a motion vector that is half the original motion vector. In SE-1 and SE-B methods mentioned in Section III-A, the reference block is changed while the motion vector is kept. In another technique [17], the reference block is also kept, but, in order to prevent the uncovered and overlapping areas, motion vectors are changed to point to the middle of the current block in the SI frame. In the proposed method, both the motion vector and the reference block are kept. Nevertheless, we have to deal with the problems derived therefrom.

IV. Experimental Results

In a first test, the key frames were encoded with a regular H.263 intra encoder, using different $QP's$. Then, we generate estimations through different SI generation methods. The generated SI would be compared against the original frames. This led to the curves in Figure 7 and 8, using the first 151 frames of the popular *Foreman* and Salesman sequences, respectively.

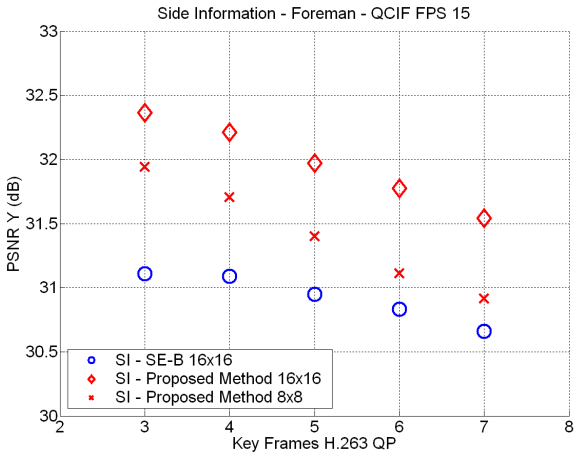


Fig. 7. Results regarding the SI for the *Foreman* sequence.

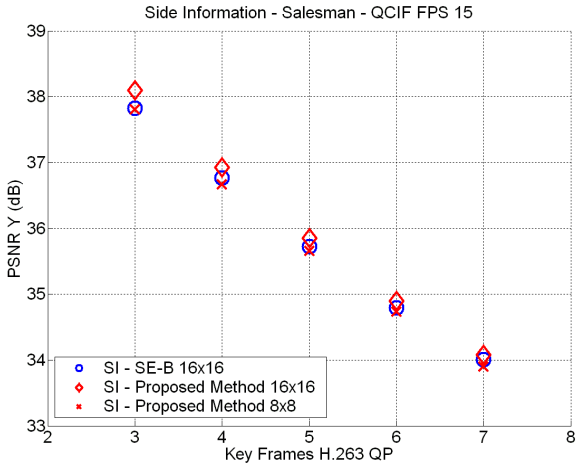


Fig. 8. Results regarding the SI for the *Salesman* sequence.

As shown in Figures 7 and 8, the proposed method works best with higher block sizes. This is consistent with prior results found in the literature [15] regarding SI generation through MCFI. Note that the final motion estimation to fill the blank areas is always performed using 16×16 blocks. Also, the proposed method outperforms the method $SE - B$ for both sequences.

In another test, the SI methods were used in the WZ codec described in Section II. The PSNR-curves are shown in Figures 9 to 12. The first 151 frames of the popular *Foreman, Coastguard, Carphone* and Salesman sequences were used in this test. The performance of the WZ codec using the two SI generation methods are compared with H.263 standard in *INTRA* mode and with a GOP length of 2 (*IPIP...*) because this is the codec used to encode the key frames.

It can be seen that, for high-motion sequences, the proposed method consistently outperforms $SE - B$. For low motion sequences, such as $Salesman$, despite the fact that the proposed SI method generates slightly better estimations (as shown in Figure 8), the WZ codec performs equally with both SI methods. Also, the margin by which the WZ codec

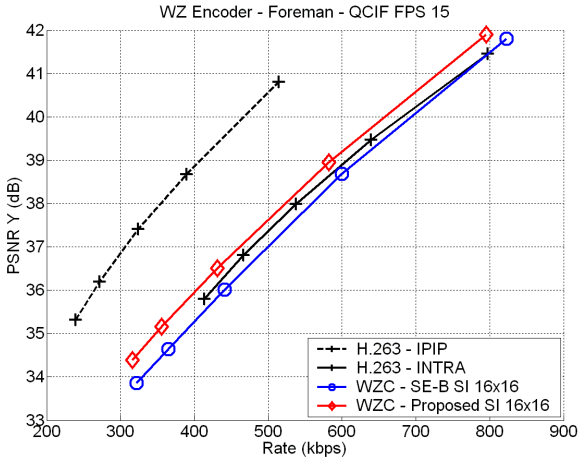


Fig. 9. Results for the WZ codec for the *Foreman* sequence.

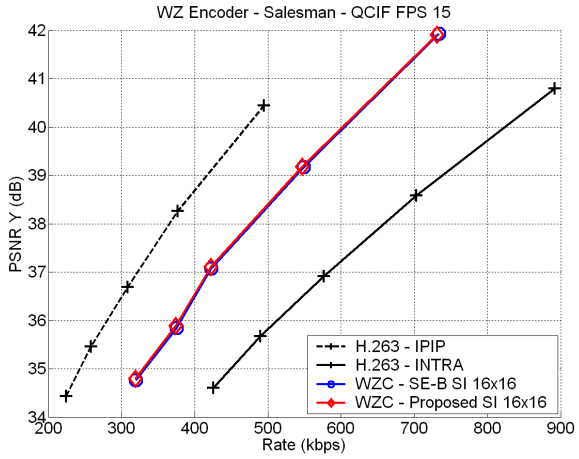


Fig. 10. Results for the WZ codec for the *Salesman* sequence.

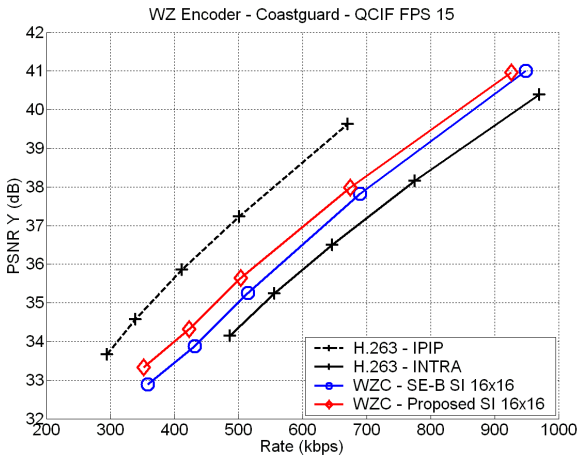


Fig. 11. Results for the WZ codec for the *Coastguard* sequence.

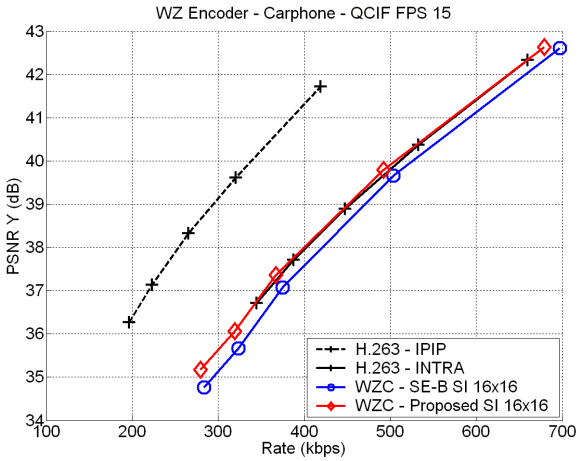


Fig. 12. Results for the WZ codec for the *Carphone* sequence.

using the proposed SI method outperforms the WZ codec using $SE - B$ is not the same margin by which the proposed SI method outperforms $SE - B$ in the first test (Figures 7 and 8). Thus, the first test is not as reliable as one might think. The integrated test seems to be more dependable to evaluate the SI generation method performance.

V. Conclusions

The SI generation scheme is crucial to any DVC framework. This work proposes a new way of generating SI frames, which outperforms a popular method for high-motion scenes. For low-motion sequences, both methods perform equally. The complexity only increased by a small amount compared to method $SE - B$. This is a desired quality for a SI generation method, since the WZ decoder is usually very complex.

For future work, block sizes can be made adaptive. Smaller block sizes better capture the scene for low-motion sequences. However, it misses features where high-motion occurs.

References

[1] J. D. Gibson, T. Berger, T. Lookabaugh, R. Baker and D. Lindbergh, *Digital Compression for Multimedia: Principles and Standards*, Morgan Kaufmann, 2006.

[2] T. Wiegand, G. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC video coding standard", *IEEE Transactions on Circuits and Systems for Video Technology 13*, pp 560-576, July 2003.

[3] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources", *IEEE Transactions on Information Theory 19*, pp 471-480, July 1973.

[4] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory 22*, pp 1-10, January 1976.

[5] B. Girod, A. Aaron, S. Rane and D. Rebollo-Monedero, "Distributed Video Coding", *Proceedings of the IEEE 93*, pp. 71-83, January 2005.

[6] A. Aaron and B. Girod, "Compression with side information using turbo codes", *Proc. IEEE Data Compression Conference*, pp. 252-261, Snowbird, USA, April 2002.

[7] A. Aaron, D. Varodayan and B. Girod, "Wyner-Ziv residual coding of video", *Proc. International Picture Coding Symposium*, Beijing, China, April 2006.

[8] G. Cote, B. Erol, M. Gallant and F. Kossentini, "H.263+: Video coding at low bit rates", *IEEE Transactions on Circuits and Systems for Video Technology 8*, pp. 849-866, November 1998.

[9] D. Varodayan and B. Girod, "Rate-adaptive distributed source coding using low-density parity-check codes", *Proc. Asilomar Conference on Signals, Systems and Computing*, Pacific Groove, USA, November 2005.

[10] D. Mukherjee, "Optimal parameter choice for Wyner-Ziv coding of laplacian sources with decoder side-information", *HP Labs Technical Report*, HPL-2007-34, 2007.

[11] R. Puri and K. Ramchandran, "PRISM: a "reversed" multimedia coding paradigm", *IEEE International Conference on Image Processing*, Barcelona, Spain, September 2003.

[12] J. Ascenso and F. Pereira, "Adaptive hash-based side information exploitation for efficient Wyner-Ziv video coding", *IEEE International Conference on Image Processing*, Saint Antonio, USA, September 2007.

[13] W. J. Chien, L. J. Karam and G. P. Abousleman, "Distributed video coding with 3D recursive search block matching", *IEEE International Symposium on Circuits and Systems*, Island of KOS, Greece, May 2006.

[14] D. Kubasov and C. Guillemot, "Mesh-based motion-compensated interpolation for side information extraction in distributed video coding", *IEEE International Conference on Image Processing*, Atlanta, USA, October 2006.

[15] Z. Li, L. Liu and E. J. Delp, "Rate-distortion analysis of motion side estimation in Wyner-Ziv video coding", *IEEE Transactions on Image Processing 16*, pp. 98-113, January 2007.

[16] M. Tagliasacchi, S. Tubaro and A. Sarti, "On the modeling of motion in Wyner-Ziv video coding", *IEEE International Conference on Image Processing*, Atlanta, USA, October 2006.

[17] 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, Image Processing, Multimedia Communications and Services*, Smolenice, Slovak Republic, July 2005.