Adaptive Frame Interpolation for Wyner-Ziv Video Coding

Savvas Argyropoulos*§, Nikolaos Thomos†, Nikolaos V. Boulgouris‡, and Michael G. Strintzis*§
*Electrical and Computer Engineering Dept., University of Thessaloniki, Thessaloniki, Hellas
†Signal Processing Institute - ITS, Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland
‡ Department of Electronic Engineering, Division of Engineering, King's College London, UK
§Informatics and Telematics Institute, Centre for Research and Technology, Thessaloniki, Hellas
Email: {savvas@ieee.org, nikolaos.thomos@epfl.ch, nikolaos.boulgouris@kcl.ac.uk, strintzi@eng.auth.gr}

***Abstract*— This paper addresses the problem of frame interpolation for Wyner-Ziv video coding. A novel frame interpolation method based on block-adaptive matching algorithm for motion estimation is presented. This scheme enables block size adaptation to local activity within frames using block merging and splitting techniques. The efficiency of the proposed method is evaluated in transform domain Wyner-Ziv video coding. The experimental results demonstrate the superiority of the proposed method over existing frame interpolation techniques.**

I. INTRODUCTION

The proliferation of video surveillance, space exploration applications, and mobile camera phones has motivated recent research on Distributed Video Coding (DVC) [1]. DVC schemes are based on the application of the Slepian-Wolf and Wyner-Ziv (WZ) theorems on source coding, which prove that distributed compression of correlated sources can be as efficient as joint compression. Unlike conventional video compression schemes which utilize computationally intensive encoders, DVC schemes enable low energy consumption encoders.

The first practical implementation of a WZ video coding scheme was presented in [2], where channel coding principles were applied for source coding. Later, transform domain WZ (TDWZ) codec based on Turbo codes was proposed in [3]. These schemes perform intra-frame encoding and inter-frame decoding to reduce encoding complexity. The decoder explores the source statistics to remove redundant information. Frame interpolation techniques are applied to previously decoded frames to obtain an estimation of the encoded frame, which is used as the side information for decoding.

Several frame interpolation techniques have been presented in the literature. Specifically, in [4], bidirectional motion estimation was used for frame interpolation in conjunction with spatio-temporal smoothing and motion compensated interpolation (MCI). Similarly, in [5], a frame interpolation framework employing forward estimation, spatial smoothing, bidirectional motion estimation and compensation was presented. The resulting scheme was evaluated for pixel domain WZ coding. Sub-pel motion-compensated frame interpolation was proposed in [6] for side information generation. In [7], the impact of side information quality on the performance of a WZ decoder was investigated using a rate-distortion (RD) model.

Although the side information generation resembles B-frame coding employed in hybrid video encoders, in WZ coding the decoder has no access to the interpolated frame and motion estimation is performed between neighbouring frames. Apparently, WZ coding aims to estimate effectively the motion field, while in B-frame coding a distortion metric, not necessarily corresponding to the true motion, is minimized. In this sense, the generation of side information is similar to the frame rate up conversion problem. Frame rate up conversion was examined in [8], where motion analysis was used to estimate global motion in the interpolated frame. A reliability factor was assigned to motion vectors and MCI was applied using multiple motion trajectories. Motion vector reliability was also considered in [9] where residual energy information was used to detect unreliable motion vectors.

In this paper, a novel frame interpolation method for WZ video coding is proposed. The algorithm is based on the adaptation of the block size for motion estimation. Instead of using constant block sizes in the block matching algorithm (BMA), the proposed method adjusts dynamically the block size based on motion analysis. This results in more accurate estimation of the motion field and fewer blocking artifacts. Moreover, the proposed method is able to detect smaller objects which can be represented by smaller blocks and capture structured moving objects in a coherent fashion.

II. PROPOSED METHOD

A. Transform-Domain Wyner-Ziv codec

The framework of the employed WZ video coding scheme, originally proposed in [3], is illustrated in Fig. 1. The even frames (key frames) $X_{2i}, i = 0, 1, 2, \ldots$ are encoded using the intra-frame H.264/AVC encoder, while the odd frames (WZ frames) $X_{2i+1}, i = 0, 1, 2, \ldots$ are encoded by the WZ encoder. Since this paper mainly presents a frame interpolation method which is applied in DVC the WZ coding procedure is not described here. Interested readers are referred to [3] for additional details.

The key frames X_{2i} are reconstructed by the H.264/AVC decoder. Bidirectional motion estimation is applied to key frames X_{2i} and X_{2i+2} to obtain an estimation $\hat{X}_{2i+1}$ of the originally encoded WZ frame X_{2i+1}. The estimated frame is

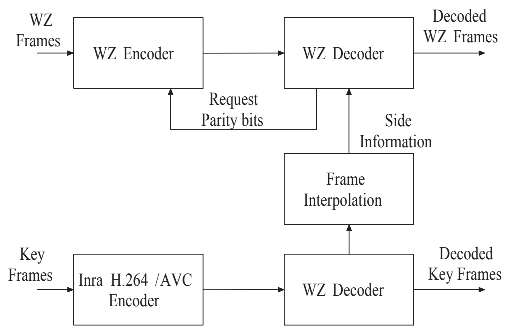


Fig. 1. Architecture of a transform domain Wyner-Ziv codec.

the side information of the WZ decoder. The performance of the WZ scheme depends on the quality of the interpolated frame. Specifically, when the estimated frame $\hat{X}_{2i+1}$ is a good approximation of the original frame, fewer parity bit are required for reconstruction and the LDPC decoder converges after few iterations. Conversely, poor interpolation results in more parity bits requests. The proposed system employs the frame interpolation method presented in [5]. For the sake of completeness, this method is briefly described below.

Initially, the previous and next key frames are low-pass filtered to reduce the probability of erroneous motion vectors selection. Subsequently, forward motion estimation is performed based on the BMA using fixed 8×8 blocks. However, the obtained motion vectors do not intercept the interpolated frame at the center of its non-overlapping blocks, which leads to overlapped and uncovered areas. To tackle this problem, the motion vector of each non-overlapping block of the interpolated frame is defined by the motion vector that intercepts the interpolated frame closer to that block. Thus, every block in the interpolated frame has a single motion vector. To further improve the accuracy of the assigned motion vectors a refinement step is performed. Based on the initial motion vector, the best linear motion trajectory passing through the blocks is estimated. In this case, the search range is confined to a small area around the initial block position. At the final step, the outlier motion vectors are removed using a weighted vector median filter (WVMF).

After the motion field is calculated using the above steps conventional bi-directional MCI methods are applied. Specifically, a block b in the interpolated frame is given by:

$$b(x, y, t) = f_1 \cdot I(x + v_x, y + v_y, t - 1) \\ + f_2 \cdot I(x - v_x, y - v_y, t + 1) \quad (1)$$

where $\mathbf{f} = [f_1 \;\; f_2]$ is the interpolation filter, x and y are the spatial dimensions, t is the temporal dimension, $I(x, y, t)$ is the luminance of the frame in time t, and $\mathbf{v} = [v_x, \; v_y]$ is the motion vector of the interpolated frame. Assuming that the interpolated frame lies in the middle of the next and the previous frames, the filter taps f_1, f_2 are set to 0.5.

B. Adaptive frame interpolation

Although the above frame interpolation method performs well, it has several limitations. Specifically, the displacement vectors are estimated using blocks with no semantic meaning, which prohibits the coherent capture of the structured moving objects in the scene. Moreover, fixed-size blocks are usually inefficient for large static background and areas with sharp moving edges. Additionally, the performance of the BMA with fixed-size blocks relies on the assumption of uniform motion within blocks. Thus, it performs well only when the blocks do not overlap with objects in the image.

The selection of the block size for motion estimation is a tradeoff between two contradictory parameters. On the one hand, smaller blocks generate low accuracy motion vectors, especially in low motion scenes. They fail to capture coherently the motion of larger objects in the scene creating more visible artifacts. On the other hand, larger blocks perform poorly in complex scenes where different types of motion exist, since it is assumed that the whole block undergoes the same translation.

Ideally, the block size should adapt to motion characteristics. The proposed frame interpolation technique with variable size block matching (FI-VSBM) initially performs motion estimation using fixed-size block. Subsequently, areas of similar motion are detected and grouped into variable-sized blocks with a coding strategy based on motion analysis. Additionally, a block splitting method is proposed to define smaller blocks for motion estimation. The block diagram of the proposed FI-VSBM method is presented in Fig. 2 and discussed in the following subsections.

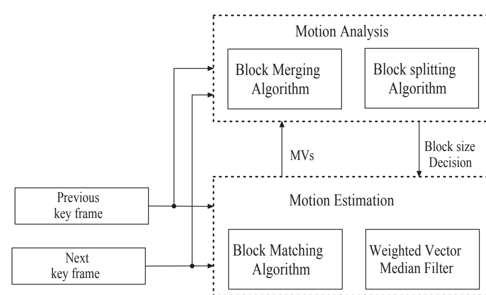


Fig. 2. Block diagram of the proposed frame interpolation method based on block merging and block splitting.

C. Block Merging Algorithm

The proposed block merging algorithm adapts the block size for the BMA based on the motion characteristics. First, the method described in Section II-A is applied. Forward motion estimation is performed using 8×8 blocks. Subsequently, the direction of each motion vector is estimated. In particular, based on the horizontal and vertical components of the estimated motion vectors, the direction of the motion vector of each block is categorized into one of the following directions: *Up*, *Up-Right*, *Right*, *Down-Right*, *Down*, *Down-left*, *Left*, and

Up-Left. An additional *Zero Motion* class is defined, which is selected if both motion vector components are zero. For any other case, the direction of the motion vector is computed according to the following rules:

- if υ_x is zero the direction is classified either to *Up* or *Down* direction, according to the sign of υ_y.
- if υ_x is non-zero, the angle θ of the motion vector direction is calculated for each quarter-plane as:

$$\theta = \arctan \frac{\upsilon_y}{\upsilon_x} \tag{2}$$

As illustrated in Fig. 3, if the angle θ is below a threshold T_{low} the motion vector direction is either *Left* or *Right*. If θ is above a threshold T_{high} the direction of the motion vector is either *Up* or *Down*. If the angle θ is between the two thresholds, the direction of the motion vector is classified to one of the diagonal directions.

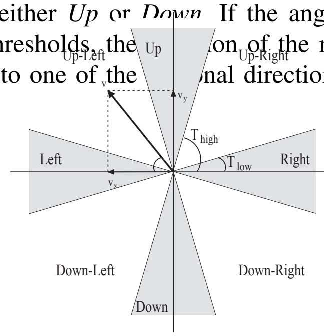


Fig. 3. Classification of motion vector direction.

Next, if four neighbouring 8×8 blocks have the same motion vector direction they are merged into a 16×16 block. Motion estimation is performed again for the 16×16 blocks to estimate the motion vectors of the merged blocks. The same method is applied for further merging 16×16 blocks into 32×32 blocks, *etc*.

D. Block Splitting Algorithm

In some cases, the local motion activity of a frame requires smaller block sizes to capture the true motion. Thus, it is essential to devise an algorithm to determine whether blocks of 8×8 should be split into smaller blocks of 4×4 size. For this reason, after block merging, the direction of motion vectors in a 3×3 window of 8×8 blocks is checked. If block merging has not been applied in the sliding window and the directions of the neighbouring motion vectors are classified to more than four directions, then the central block is split into four 4×4 blocks.

The motivation for this block splitting algorithm arises from the fact that incoherent motion vectors of neighbouring blocks (*i.e.*, different directions of the motion vectors) are indicative of a local translation. Therefore, smaller block sizes in this region can model the true motion more effectively. Motion estimation is performed again to estimate the motion vectors of the 4×4 blocks.

Finally, when the motion vectors of the blocks of variable size have been estimated, bi-directional motion estimation, as described in Section II-A, is applied to estimate the interpolated frame.

III. Experimental Results and Discussion

The efficiency of the proposed method for frame interpolation in TDWZ video coding was evaluated by extensive simulations. The two standard QCIF sequences “Foreman” and “Coastguard” were encoded at 30 *fps* with I-WZ-I-WZ GOP structure. Thus, the WZ frame rate is 15 *fps*. The even frames were intra coded using the JM 9.3 version of the H.264/AVC codec. WZ coding was implemented using the coding scheme of [3] with a rate 1/2 LDPC encoder. The search range for the BMA was set to 16 pixels for the forward motion estimation, and 2 pixels for the refinement process. The threshold values T_{low}, T_{high} were set to 25° and 75° respectively after extensive experimental evaluation.

The proposed frame interpolation method FI-VSBM was compared with the frame interpolation method described in section II-A using 8×8 fixed size blocks. The average PSNR of the interpolated frames is presented in Tables I and II when key frames are (a) perfectly reconstructed (infinite PSNR), and (b) quantized with different QPs. The former case is rather unrealistic and is presented merely for comparison reasons with the results reported in the literature. The results clearly show that the proposed method generates better quality interpolated frames. The superior performance is attributed to the adaptation of the block size to the local activity within the frames. It is important to note that larger blocks significantly decrease blocking artifacts during motion compensation. The adaptation of the block size to the local motion characteristics is depicted in Fig. 4 for an instance of the “Foreman” sequence. As illustrated, larger block sizes are used for the static background and large structured moving objects while smaller blocks are utilized in regions with complex motion.

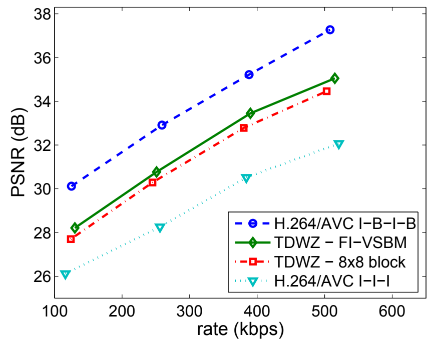
TABLE I
Average PSNR of the interpolated frames for the “Foreman” sequence

Foreman sequence	FI-VSBM	8×8 block size
lossless	37.09	36.54
QP=5	37.06	36.46
QP=15	36.78	36.11
QP=25	35.09	34.45

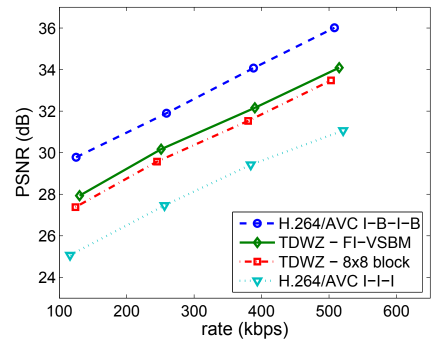
TABLE II
Average PSNR of the interpolated frames for the “Coastguard” sequence

Coastguard sequence	FI-VSBM	8×8 block size
lossless	34.95	34.21
QP=5	34.91	34.12
QP=15	34.86	34.07
QP=25	33.30	32.65

The impact of side information on the performance of a TDWZ codec was also evaluated. The QP parameter of the key frames was adjusted to obtain PSNR values similar to the values of the quantized WZ frames. Fig. 5(a) and 5(b) demonstrate the performance of a TDWZ codec employing the proposed FI-VSBM method compared to a method using fixed block size of 8×8 for the sequences “Foreman” and “Coastguard”, respectively. As seen, the proposed codec yields



(a)



(b)

Fig. 5. Average PSNR results of the TDWZ codec using the two methods for side information generation and comparison with the I-B-I-B and I-I-I H.264/AVC codec for the sequences (a) "Foreman" and (b) "Coastguard".

approximately 0.4 *dB* above the conventional method employing fixed 8×8 blocks. This corroborates the initial statement that better side information improves the performance of the WZ codec. It is worth noting noted that the performance improvements comes at negligible additional computational cost compared to the fixed-block MCI algorithm. Nevertheless, even this small additional complexity is affordable since WZ coding targets at the development of light encoders at the cost of complex decoders.

Finally, the proposed method was compared with the H.264/AVC codec employing the I-B-I-B structure (inter-frame) and the I-I-I structure (intra-frame) as depicted in Figs. 5(a) and 5(b). From these figures it can deduced that the performance of the proposed WZ codec is approximately 3 *dB* above the intra-frame codec and 2 *dB* below the inter-frame codec. The main reasons for the performance of the proposed scheme compared to the I-B-I-B codec is that the channel codes are still not efficient enough to reach the H.264/AVC performance and that motion interpolation can not be as efficient as conventional MCI techniques for the exploitation of redundancy in hybrid video coding. However, it is clear that, using the proposed method, the gap between the performance of the conventional H.264/AVC and WZ coding is closing.

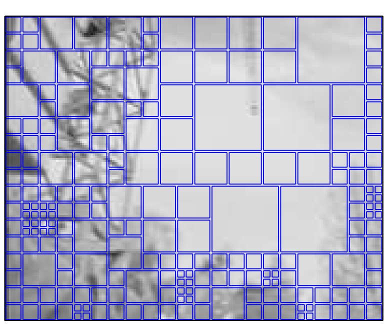


Fig. 4. Dynamic adaptation of block size in frame interpolation.

IV. Conclusions and Future Work

In this paper, a novel frame interpolation method for WZ video coding was presented. The proposed technique is based on variable block size motion estimation depending on the local activity in the key frames. The scheme allows blocks sizes to adapt to local activity within the frame. The number of blocks in any frame can be varied while still accurately representing true motion. Experimental results show that the proposed scheme outperforms existing frame interpolation methods in terms of rate-distortion performance while not significantly increasing computational complexity.

Acknowledgment

This work was supported by the EC under contract FP6-511568 with the acronym 3DTV.

References

[1] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed Video Coding," *Proceedings of the IEEE*, vol. 93, no. 1, p. 71, 2005.
[2] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," in *Proc. Allerton Conference on Communication, Control, and Computing*, Allerton, Illinois, Oct 2002.
[3] A. Aaron, S. Rane, E. Setton, B. Girod, *et al.*, "Transform-domain Wyner-Ziv codec for video," *Proceedings of SPIE*, vol. 5308, pp. 520–528, 2004.
[4] B. Choi, S. Lee, and S. Ko, "New frame rate up-conversion using bi-directional motion estimation," *Consumer Electronics, IEEE Transactions on*, vol. 46, no. 3, pp. 603–609, 2000.
[5] J. Ascenso, C. Brites, and F. Pereira, "Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding," *5th EURASIP, Slovak Republic,*, July 2005.
[6] S. Klomp, Y. Vatis, and J. Ostermann, "Side Information Interpolation with Sub-pel Motion Compensation for Wyner-Ziv Decoder," *International Conference on Signal Processing and Multimedia Applications (SIGMAP), Setubal, Portugal*, pp. 520–528, August 2006.
[7] Z. Li, L. Liu, and E. J. Delp, "Rate Distortion Analysis of Motion Side Estimation in Wyner-Ziv Video Coding," *Image Processing, IEEE Transactions on*, vol. 16, no. 1, pp. 98–113, Jan 2007.
[8] S. Lee, O. Kwon, and R. Park, "Weighted-adaptive motion-compensated frame rate up-conversion," *Consumer Electronics, IEEE Transactions on*, vol. 49, no. 3, pp. 485–492, 2003.
[9] A. Huang and T. Nguyen, "A novel motion compensated frame interpolation based on block-merging and residual energy," in *Multimedia Signal Processing, IEEE 8th Workshop on*, Oct 2006, pp. 395–398.