Efficient Frame Interpolation for Wyner-Ziv Video Coding

Çağatay Dikici[a], Thomas Maugey[b], Marie Andrée Agostini[c], and Olivier Crave[bd]

[a] LSS, CNRS - Supélec, Gif-sur-Yvette, France;
[b] LTCI, CNRS - TELECOM ParisTech, Paris, France;
[c] I3S Laboratory, CNRS - University of Nice-Sophia Antipolis, France;
[d] Project TEMICS, IRISA/INRIA Rennes, France

ABSTRACT

In the framework of Wyner-Ziv Coding of Video, the coding efficiency depends on the quality of the side information (SI) at the decoder, where the side information is constructed from the key frames available at the decoder. In this paper, we propose a novel frame interpolation method for Wyner-Ziv coding, where the motion compensation is bidirectional and allows pixelwise estimation. The proposed interpolation method allows to obtain better SI quality than the one obtained by state-of-the-art interpolation methods.

Keywords: Distributed video coding, Wyner-Ziv, frame interpolation, motion vector estimation, motion compensation

1. INTRODUCTION

Distributed Video Coding (DVC) is a new and very interesting paradigm in video coding which proposes to move the computation complexity from the encoder to the decoder.[1,2] This can be useful in many industrial applications such as video compression on mobile devices, multi-sensor system, etc. Contrary to classical video coding schemes (H.263, MPEG-4,[3] H.264[4]), DVC performs intra-frame encoding of correlated frames (without exploiting any correlation between frames at the encoder), and inter-frame decoding (by exploiting the temporal frame correlation at the decoder). In other words, motion estimation is not realized anymore at the encoder as classical video coding schemes do, but at the decoder. This motion information extraction is performed to build an estimate, called side information (SI), of some frames of the sequence. The quality of this SI has a strong impact on the coding performance of the system. Slepian-Wolf[5] showed that the compression rate of separate encoding of correlated sources is the same for joint encoding/decoding in lossless case. Wyner and Ziv[6] extended the Slepian and Wolf theorem to the lossy case.

Distributed source coding has been recently brought into practice in video coding. One of the existing architectures has been proposed by Girod et al.[2] and works as follows: first the frames are separated in two subsets, the key frames (KFs) and the Wyner-Ziv frames (WZFs). In order to obtain two correlated sources, the groups of pictures (GOP) are made up of one KF and n WZFs[2] (in this paper $n = 1$). These two sources are encoded independently. The KFs are intra encoded/decoded. The WZFs encoding process is based on error correcting codes and is less complex than the KF encoding. The first step consists on splitting each frame into blocks which are transformed using e.g. a discrete cosine transform .[7] Then, the WZFs are quantized and finally channel encoded. The channel encoding step produces redundant bits (syndrome bits or parity bits), and only these bits are sent to the decoder. At the decoder, the redundant bits and the key frames are jointly decoded. The SI is generated from the already decoded KFs using the previously described method. This estimation is corrected by the channel decoder thanks to the redundant bits sent by the encoder. The reconstructed frames are finally inverse transformed. DVC efficiency strongly depends on the quality of the side information construction at the decoder. The SI construction consists of computing a frame estimation, for example with an interpolation between two existing frames. Coding efficiency strongly depends on the quality of the interpolation method. In this paper we propose a novel interpolation method which performs bidirectional motion estimation and uses pixelwise motion compensation by allowing overlapped motion vectors. This technique surpasses one of the best existing solution proposed by Ascenso et al.[8]

Corresponding author: cagatay.dikici@lss.supelec.fr. Çağatay Dikici was also with INSA de Lyon, France.

The paper is organized as follows: first, the existing interpolation methods are briefly described in Section 2, then, in Section 3, we introduce the details of the proposed solution. Section 4 gives the experimental results of the proposed scheme for lossless and lossy coding of KFs, and shows an improvement of the side information quality. In Section 5, conclusion and future work are drawn.

2. RELATED WORK

In a DVC scheme, there are two kinds of frames: the KFs, available at the decoder, and the WZFs which are estimated by the SI. This SI is corrected at the decoder by the parity bits sent by the WZ encoder. The better the SI, the lower is the bitrate required for the correction. In order to expose what are the existing and the proposed interpolation methods, let us introduced some notations. The frames are now denoted by X_j (where j is the temporal index). Let us assume that the frame to estimate is X_{2i+1}. The interpolation methods will use two KFs: the backward frame X_{2i} and the forward frame X_{2i+2}.

2.1 Averaging

The basic approach of the reconstruction of SI from two neighbor KFs assumes that there is no motion between pixels of neighboring frames. Under this assumption, a simple solution of frame interpolation for DVC scheme (studied in[8]) only consists in averaging the two key frames: $X_{2i+1} = \frac{1}{2}(X_{2i} + X_{2i+2})$.

2.2 Forward and Backward motion estimation

We assume that the motion between the frames X_{2i} and X_{2i+1} is equal to the motion between the frames X_{2i+1} and X_{2i+2}. Then the motion estimation between X_{2i} and X_{2i+2} can be used to interpolate the frame X_{2i+1}. A block matching algorithm can be used to find the best block match of target block $b_{k,l}$ centered at the coordinates (k, l) of KF X_{2i} in the next KF, X_{2i+2}. The parameters that characterize the estimation technique are the block size, the matching criterion, the search range and the precision. Given that the best matching of block $b_{k,l}$ of X_{2i} in X_{2i+2} is $f_{m,n}$ with a block motion of $\vec{w_f} = (m - k, n - l)$, the linear projection of these two blocks onto the frame X_{2i+1} can be calculated as $c = \frac{b+f}{2}$ where c is centered at the location $(\frac{m+k}{2}, \frac{l+n}{2})$. Figure 1(a) describes the forward motion estimation between X_{2i} and X_{2i+2} and their linear projection on X_{2i+1}. When the forward motion vectors are projected on the frame X_{2i+1}, overlapping and uncovered areas will usually appear. The overlapping areas correspond to the multiple motion vectors which pass through a unique pixel, whereas uncovered areas correspond to the absence of the motion trajectory for these pixels.

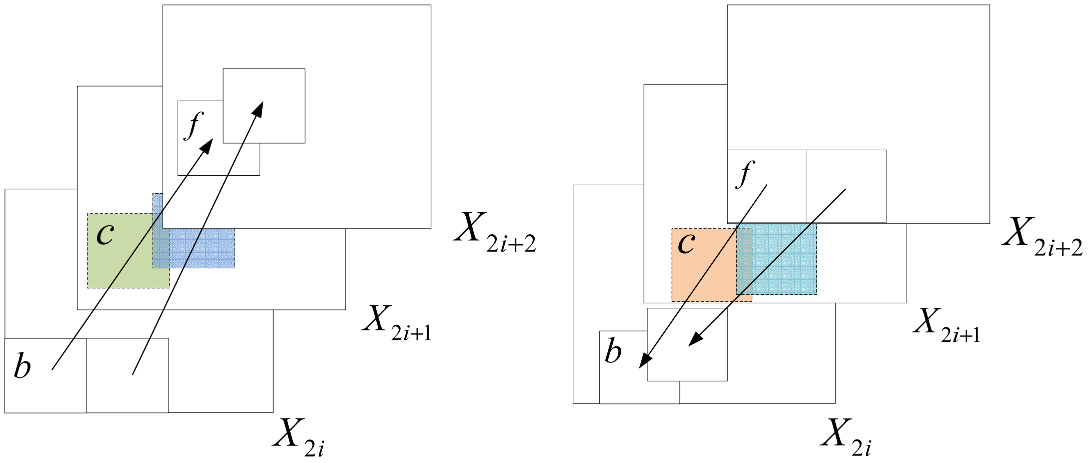
A similar calculation can be done for the backward motion estimation (see Figure 1(b)), where the aim is to find the block $b_{m',n'}$ in X_{2i} which is the best estimation of block $f_{k',l'}$ in X_{2i+2}. Given a backward motion of $\vec{w_b} = (m' - k', l' - n')$, the candidate block c of X_{2i+1} can be calculated similarly as in the forward case $c = \frac{b+f}{2}$, where in this case c is centered at the location $(\frac{m'+k'}{2}, \frac{l'+n'}{2})$.

2.3 Motion compensation using rigid motion vectors

In,[8] the authors used, besides forward and bidirectional motion estimation, a spatial motion smoothing algorithm to eliminate motion outliers. After finding the forward motion vectors for non-overlapping blocks in the frame X_{2i+1}, the proposed scheme uses weighted vector median filters, which maintain the spatial coherence of the motion field by looking for candidate motion vectors in neighboring blocks. An extension of this method can be found in.[9]

3. PROPOSED INTERPOLATION METHOD

The proposed method is based on the forward and backward motion compensation between two frames which is explained in the previous section. However, in order to calculate the motion compensation, both forward and backward block matching algorithms are applied to two consecutive KFs X_{2i} and X_{2i+2} for blocks of size n by n that are (r, r) pixel apart, which can be called as overlapped step size. Again, a linear motion is assumed between the key frames and the interpolated frames as in.[8] Once the forward and backward motion vectors ($\vec{w_f}$,$\vec{w_b}$) are calculated as explained in Section 2.2, $\frac{\vec{w_f}}{2}$ and $\frac{\vec{w_b}}{2}$ can be used for the motion compensation step. Furthermore, for each pixel of X_{2i+1}, the proposed bidirectional frame interpolation step is applied as follows.



(a) Forward interpolation.

(b) Backward interpolation.

Figure 1. Classical interpolation tools.

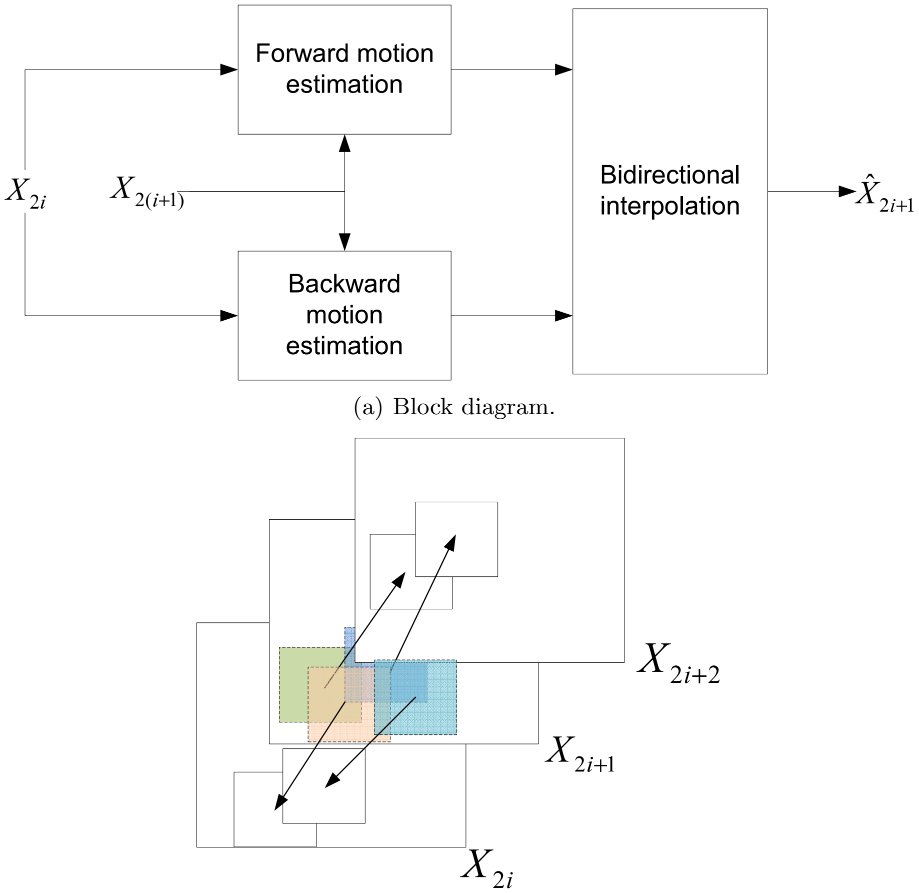
Let $p_i(x, y)$ be the pixel value of the i-th frame located at (x,y). We define the set $\mathcal{C}$ of motion compensated blocks that passes through the pixel $p_{2i+1}(x, y)$ as $\mathcal{C}(p_{2i+1}(x, y))$. Then the interpolated pixel value is:

$$\hat{p}_{2i+1}(x, y) = \begin{cases} \frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} c_i, & \text{if } |\mathcal{C}| > 0, \\ 0.5 \times (p_{2i}(x, y) + p_{2i+2}(x, y)), & \text{else} \end{cases} \quad (1)$$

where $|\mathcal{C}|$ is the cardinal member of $\mathcal{C}$. Hence if the set $\mathcal{C}$ is not empty, corresponding to at least one motion vector that passes through the pixel value $p_{2i+1}(x, y)$, then an averaging of the corresponding pixel values in the motion compensated blocks of the set $\mathcal{C}$ is done. Otherwise, a simple averaging of the pixel values in previous and next KFs is performed. The block diagram of the proposed method and the visualization of the bidirectional estimation can be found in Figure 2. Hence the major difference of the proposed method from existing methods in[2,9] is that the motion compensation is not based on non-overlapping block matching of the SI frame, but a pixel-wise interpolation. Contrary to the non-overlapped block matching approach in,[8] the proposed interpolation allows overlapped block matching and a pixel-by-pixel estimation using real bidirectional motion vectors between consecutive KFs is done in the final step. While small values of overlapped step size result in a more smoothing operation of a pixel value using a set of motion compensation, big values of overlapped size results in pixels with no motion vectors which corresponds simple averaging at the limit. In our experiments, we fix a ratio of 1/2 between the overlapped step size and the block size which gives satisfactory results.

4. EXPERIMENTAL RESULTS

In order to evaluate the proposed interpolation method, we use QCIF resolution sequences with 15 fps such as Foreman, News and Hall for the first 75 frames. Even frames are selected as KFs and their quantized version is available at the decoder, and the odd frames are interpolated from the KFs. We compare our results with average frame interpolation and with the methods proposed in[8,9] available online at.[10] In all our experiments, we use a fixed block size of 8×8 pixels, a search range of ± 16, a step size of 4 pixels for the overlapped blocks, and an integer pixel precision for the forward and the backward motion estimation. The step size determines the shift of the blocks for calculating the next motion vector, hence MV's are calculated for the overlapped blocks in every 4 pixels in height and width. For the generation of the interpolation, we use three different KF types: lossless coding of KFs, H.264 intra-coding of KFs with different visual qualities, and JPEG-2000 coding of KFs with different visual qualities.


Forward motion estimation
Backward motion estimation
Bidirectional interpolation
X_{2i}
$X_{2(i+1)}$
$\hat{X}_{2i+1}$
X_{2i+2}
X_{2i+1}
X_{2i}

(a) Block diagram.

(b) Forward-Backward interpolation.

Figure 2. Proposed interpolation method.

4.1 Lossless Key Frames

In this section, the side information is generated using non-degraded reference frames. We compare the proposed method (ESSOR) to the Discover approach (Section 2.3) and the basic interpolation method (average of the two reference frames, Section 2.1). Experimental results are presented in Table 1. One can see that our approach outperforms the Discover solution up to 1.04 dB.

Table 1. Performance of frame interpolation methods in PSNR for lossless Key Frames.

Sequence	Avg	[8]	[9]	Our method
News	39.76	39.80	39.83	**40.27**
Foreman	27.86	29.42	29.79	**29.90**
Hall	37.84	38.57	38.69	**39.73**

4.2 Lossy Key Frames coded with H.264 Intra

In practical video coding contexts, the KFs are compressed, and the available KFs are not lossless anymore. In many coding schemes in the literature,[11] the coder used to encode the KFs is H.264 intra.[4] In this section, the proposed interpolation is compared to the Discover one, in case of H.264 intra transmission of the KFs. We use three different quantization levels corresponding to low, medium, and high bitrates. The experimental results are presented for the three test sequences in respectively Table 2, 3, and 4. The respective KFs average PSNR values are given in the first row of each table. For each quantization levels, we compare the average PSNR values obtained with our approach to the ones obtained by Discover and by the average method. The results show an improvement of the performance in average PSNR value compared to the Discover approach, of 0.2 dB for News, 0.1 dB for Foreman, and 0.5 dB for Hall. Please note that, for low PSNR values of the KF coding, the interpolation methods can slightly surpass the average PSNR value of the KFs because the motion activity is really low.

Table 2. Performance of News sequence when KFs are coded as H-264 intra frames with mean PSNR values 29.3 dB, 34.34 dB, and 40.7 dB.

Average KF Distortion	29.3 dB	34.34 dB	40.7 dB
Averaging	29.614	33.47	37.64
Discover	29.616	33.49	37.72
ESSOR	**29.704**	**33.64**	**37.96**

Table 3. Performance of Foreman sequence when KFs are coded as H-264 intra frames with mean PSNR values 29.5 dB, 33.6 dB, and 39.9 dB.

Average KF Distortion	29.5 dB	33.6 dB	39.9 dB
Averaging	26.43	27.28	27.74
Discover	27.43	28.76	29.64
ESSOR	**27.57**	**28.87**	**29.66**

Table 4. Performance of Hall sequence when KFs are coded as H-264 frames with mean PSNR values 30.9 dB, 34.3 dB, and 40 dB.

Average KF Distortion	30.9 dB	34.3 dB	40 dB
Averaging	29.9	33.31	36.53
Discover	30.05	33.73	37.30
ESSOR	**30.27**	**34.10**	**38.02**

4.3 Lossy Key Frames coded with JPEG-2000

While the Discover approach consists in using discrete cosinus transform (DCT) based method, in the ESSOR project, the adopted DVC scheme is based on the discrete wavelet transform (DWT). Indeed, the intra coder is chosen to transmit the KFs is JPEG-2000.[12] This section provides the results obtained by this setup, and a comparison is given with the existing methods. Similar to the previous section, we produced three different levels of quantization, for the three sequences, which can be seen respectively at Table 5, 6, and 7. One can see that the results of the proposed approach surpass the ones of the two other tested approaches.

4.4 Interpolation error analysis

As presented in the previous section, ESSOR interpolation method outperforms the Discover techniques. In this section we propose to analyze the behavior of the SI error for the different methods.

Figure 3 represents the evolution of the PSNR of the side information along the time for QCIF Foreman test sequence. These plots show that when the motion activity is not important, ESSOR method outperforms the others. This can be explained by the fact that this technique presents a smoothing property. In case of high motion activity, Discover builds an SI of higher quality than ESSOR.

Table 5. Performance of News sequence when KFs are coded as JPEG-2000 frames with mean PSNR values 29.5 dB, 37 dB, and 41.5 dB.

Average KF Distortion	29.5 dB	37 dB	41.5 dB
Averaging	29.48	35.71	38.01
Discover	29.49	35.74	38.04
ESSOR	**29.59**	**35.99**	**38.40**

Table 6. Performance of Foreman sequence when KFs are coded as JPEG-2000 frames with mean PSNR values 31 dB, 35 dB, and 41 dB.

Average KF Distortion	31 dB	35 dB	41 dB
Averaging	26.97	27.70	27.8
Discover	28.11	29.40	29.73
ESSOR	**28.26**	**29.57**	**29.79**

Table 7. Performance of Hall sequence when KFs are coded as JPEG-2000 frames with mean PSNR values 30.9 dB, 39 dB, and 43.4 dB.

Average KF Distortion	30.9 dB	39 dB	43.4 dB
Averaging	30.53	35.78	37.17
Discover	30.72	36.43	37.94
ESSOR	**30.93**	**37.13**	**38.88**

In figure 4, zooms on the side informations for the third frame of News test sequence are represented. Error images are also shown. Looking at these figures, one can clearly see the smoothed aspect of ESSOR estimation, while the SI of Discover presents some blocking artifacts.

5. CONCLUSION

We have presented in this paper an improvement of the state-of-the-art methods of frame interpolation in the framework of DVC. Indeed, in these schemes, the coding efficiency depends on the quality of the estimation of the side information at the decoder. The SI construction is performed thanks to a frame estimation, based on an interpolation between two existing frames. The proposed method is based on a bidirectional motion compensation using pixelwise estimation and allowing overlapped motion vectors, and allows to obtain side information of higher quality than the existing approaches (up to 1.04 dB in PSNR). As a perspective, we want to implement shot detection methods in order to improve the performance of the interpolation method. For further works, we plan to integrate our frame interpolating method in a DVC scheme, in order to increase the whole performances of the coding.

6. ACKNOWLEDGMENTS

The work presented here was funded by the French National Agency (ANR) under the ESSOR project. The authors want to thank Marc Antonini, Christine Guillemot, Michel Kieffer, and Béatrice Pesquet-Popescu for their valuable comments and discussions.

REFERENCES

[1] R. Puri and K. Ramchandran, “PRISM: A video coding architecture based on distributed compression principles,” EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M03/6, 2003.

[2] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, “Distributed video coding,” *Proc. of the IEEE*, vol. 93, no. 71, pp. pp 71 – 83, Jan. 2005.

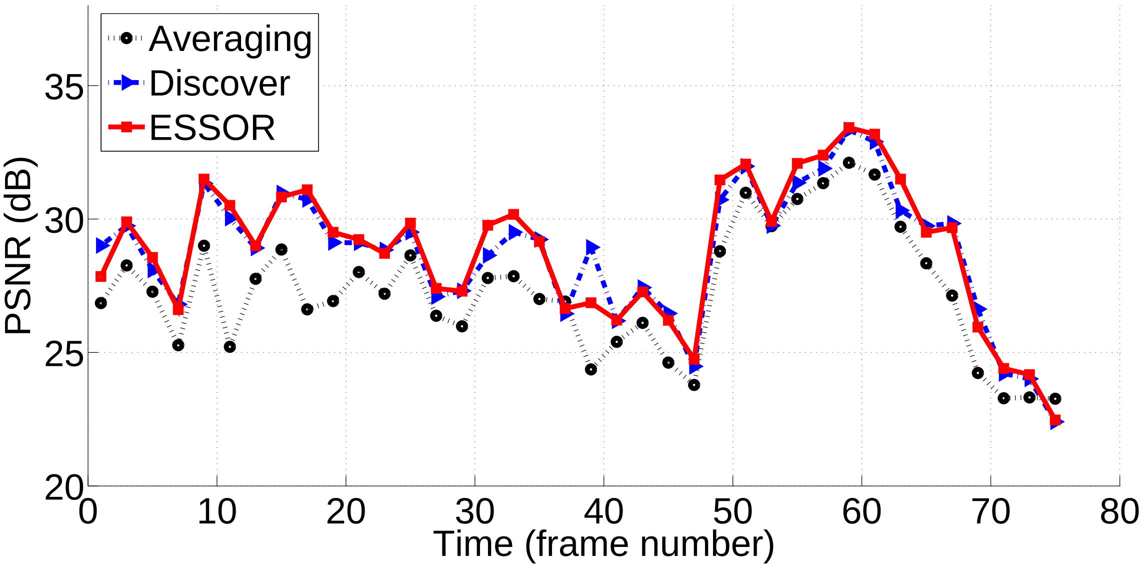
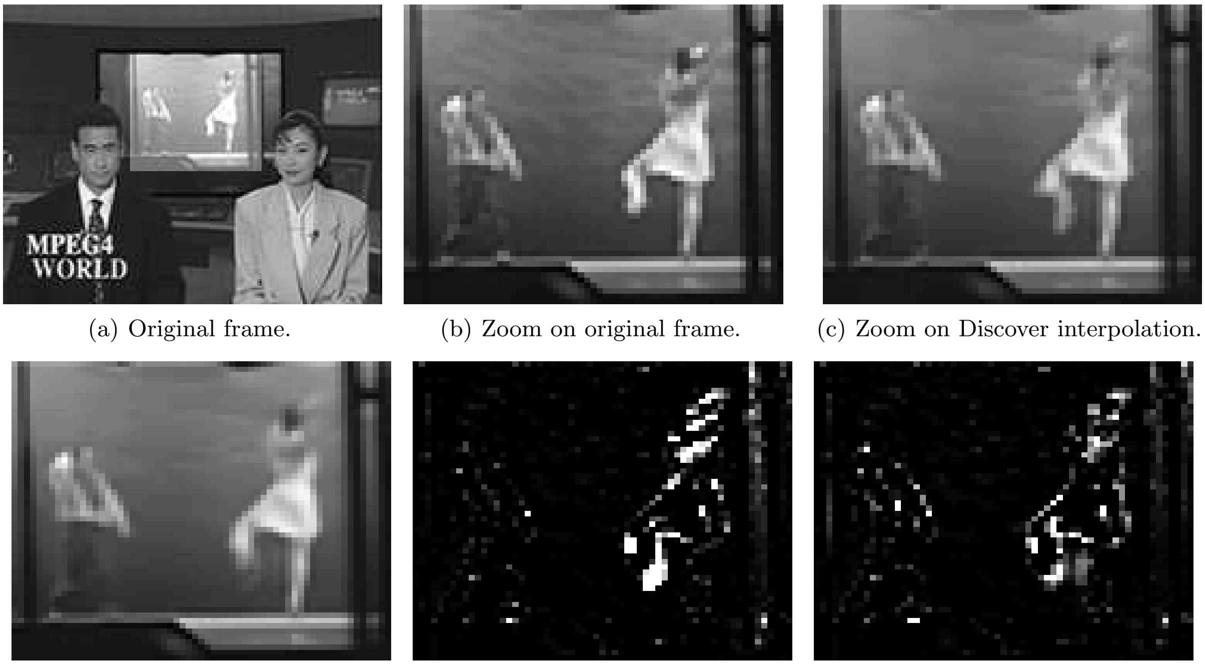


Figure 3. PSNR quality of each interpolated SI frame of Foreman sequence, where KFs are quantized, for the three interpolation methods.



(a) Original frame.

(b) Zoom on original frame.

(c) Zoom on Discover interpolation.

(d) Zoom on ESSOR interpolation performance.

(e) Zoom on Discover interpolation error.

(f) Zoom on ESSOR interpolation error.

Figure 4. Interpolation performance of the News sequence, frame #3, zooming on the center of the frame.

[3] "ISO/IEC 14496-2 (MPEG-4 visual), Information technology - Coding of audio visual objects - part 2: Visual, international standard," 2001.

[4] “H.264: Advanced video coding for generic audiovisual services.” [Online]. Available: http://www.itu.int/rec/recommendation.asp?type=folders\&\#38;lang=e\&\#38;parent=T-REC-H.264

[5] D. Slepian and J. K. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. on Information Theory*, vol. Vol. 19, pp. pp 471–480, July 1973.

[6] A. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the receiver,” *IEEE Trans. on Information Theory*, vol. Vol. 22, pp. pp 1–11, Jan. 1976.

[7] A. Aaron, S. Rane, E. Setton, and B. Girod, “Transform-domain Wyner-Ziv codec for video,” in *in Proc. SPIE Visual Communications and Image Processing*, 2004, pp. 520–528.

[8] J. Ascenso, C. Brites, and F. Pereira, “Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding,” in *EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Smolenice, Slovak Republic, June 2005.

[9] ——, “Content adaptive Wyner-Ziv video coding driven by motion activity,” in *Image Processing, 2006 IEEE International Conference on*, Atlanta, GA, October 2006.

[10] “Discover.” [Online]. Available: http://www.discoverdvc.org

[11] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, “The discover codec: Architecture, techniques and evaluation,” in *Picture Coding Symposium (PCS)*, Lisboa, Portugal, Nov. 2007.

[12] “ISO/IEC FCD 15444-1: JPEG 2000 final comitee draft version 1.0,” 2000. [Online]. Available: http://www.jpeg.org/FCD15444-1.htm