J. Vis. Commun. Image R. 23 (2012) 229–236

Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Side information generation with auto regressive model for low-delay distributed video coding

Yongbing Zhang [a,b,*], Debin Zhao [a], Hongbin Liu [a], Yongpeng Li [c], Siwei Ma [d], Wen Gao [d]

[a] Harbin Institute of Technology, Harbin, China
[b] Graduate School at Shenzhen, Tsinghua University, Beijing, China
[c] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[d] Peking University, Beijing, China

ARTICLE INFO

Article history:
Received 24 March 2009
Accepted 3 October 2011
Available online 14 October 2011

Keywords:
Distributed video coding
Low delay
Auto regressive model
Side information
Wyner-Ziv
Probability based fusion
Centrosymmetric rearrangement
Least mean square

ABSTRACT

In this paper, we propose an auto regressive (AR) model to generate the high quality side information (SI) for Wyner–Ziv (WZ) frames in low-delay distributed video coding, where the future frames are not used for generating SI. In the proposed AR model, the SI of each pixel within the current WZ frame t is generated as a linear weighted summation of the pixels within a window in the previous reconstructed WZ/Key frame $t-1$ along the motion trajectory. To obtain accurate SI, the AR model is used in both temporal directions in the reconstructed WZ/Key frames $t-1$ and $t-2$, and then the regression results are fused with traditional extrapolation result based on a probability model. In each temporal direction, a weighting coefficient set is computed by the least mean square method for each block in the current WZ frame t. In particular, due to the unavailability of future frames in low-delay distributed video coding, a centrosymmetric rearrangement is proposed for pixel generation in the backward direction. Various experimental results demonstrate that the proposed model is able to achieve a higher performance compared to the existing SI generation methods.

1. Introduction

With the development of high performance computing and channel coding [1], distributed video coding (DVC) has received more and more attentions in recent years due to its desirable properties for some applications such as wireless low power video surveillance, video compression and sensor networks. DVC is based on the principles stated by Slepian–Wolf [2] for the lossless case and Wyner–Ziv (WZ) [3] for the lossy scenario. The majority of Slepian–Wolf and WZ coding systems adopt channel coding principles [4–7], assuming the statistical dependence between the two correlated sources X and Y as a virtual binary symmetric channel or additive white Gaussian noise channel. Compression of the source X can be achieved by transmitting only parity bits using error correcting codes. At the decoder side, with the aid of received parity bits and Y, called the side information (SI) of X, the error correcting decoding is performed, i.e., performing MAP or MMSE estimation of X.

Based on these theorems, some practical DVC systems have been presented. Pradhan and Ramchandran proposed a constructive and practical framework for distributed source coding using syndromes (DISCUS) [4] to perform WZ coding. Puri and Ramchandran proposed a power-efficient, robust, high-compression, syndrome-based multimedia (PRISM) [8] DVC framework. Besides, Aaron et al. provided an asymmetric WZ coding scheme [9] for motion video using intra-frame encoding and inter-frame decoding. In their framework, the key frames are encoded by H.263+ intra frame mode and the WZ frames are encoded by Slepian–Wolf codec based on turbo codes.

One of the most critical aspects in enhancing the compression efficiency of DVC is improving SI quality. According to the Slepian–Wolf theorem [2], the less the conditional entropy $H(X|Y)$ is, the fewer the bits to reconstruct X are required, under the condition that Y can be perfectly reconstructed at the decoder. Intuitively, in practical system, where SI is generated at the decoder side, better SI will result in better performance for the WZ frames. Different from the most existing video compression standards, where the computationally intensive motion estimation is performed at the encoder side, DVC shifts the motion estimation to the decoder side. Consequently, it is very difficult to generate high quality SI without the existence of the original video sequence at the decoder side.

According to the way SI generated, DVC can be categorized into interpolation and extrapolation cases. In interpolation case, similar to the B frame coding in hybrid video coding, SI is generated by the interpolating between the previous and following reconstructed

* Corresponding author at: Graduate School at Shenzhen, Tsinghua University, Beijing, China.

E-mail addresses: ybzhang@jdl.ac.cn (Y. Zhang), dbzhao@jdl.ac.cn (D. Zhao), hbliu@jdl.ac.cn (H. Liu), ypli@jdl.ac.cn (Y. Li), swma@jdl.ac.cn (S. Ma), wgao@jdl.ac.cn (W. Gao).

1047-3203/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2011.10.001

WZ/key frames [10–15]. On the contrary, in the extrapolation case, the SI is generated by referring only the previous reconstructed frame [16–22]. Generally speaking, the SI generated by interpolating has superior performance than that generated by extrapolating, since the former can use the future information to generate SI. However, this only holds if the temporal distance is small enough [20], i.e. the GOP (group of pictures) size is sufficiently small. Besides, the extrapolation DVC is very desirable in the sequential decoding for low latency cases, since the decoding process begins as soon as it receives the previous reconstructed frame, without waiting for the arrival of the following reconstructed key frame.

To improve the compression performance of low-delay DVC, many pioneering works have been done to improve the quality of SI. In Natario's scheme [19], a robust extrapolation module is proposed to generate SI based on motion field smoothening. In this method, the extrapolation is completed by motion estimation, motion field smoothening, motion projection as well as overlapping and uncovered areas. Borchert et al. [20] introduced a true motion based extrapolation scheme considering the 3-D recursive search (3DRS) motion estimation. All these methods resort to conventional motion estimation to extract motion information from the reconstructed video frames at the decoder side. They are all based on a translational motion model, in which it is assumed that the motion in the current frame is a continuous extension of the motion in the previous frame. However, the translation model is not always satisfied, especially for the video sequences with high motion.

To obtain higher quality SI in low delay DVC, in this paper we propose an auto regressive (AR) model based SI generation based on our previous work [22]. In the proposed AR model, the SI of each pixel within the current WZ frame t is generated as a linear weighted summation of pixels within a window in the previous reconstructed WZ/K frame $t-1$ along the motion trajectory. To capture the variation properties of the current WZ frame, the SI is generated block by block. The motion trajectory of each block is assumed to be that of the co-located block in the previous reconstructed frame and is of integer-pixel accuracy. In order to obtain accurate SI, we use the forward derivation and backward derivation to compute two weighting coefficient sets for each block within the current WZ frame t. In the forward derivation, each reconstructed pixel within the collocated block in WZ/K frame $t-1$ is approximated as a linear weighted summation of pixels within the corresponding window in the reconstructed WZ/K frame $t-2$. The Least-Mean-Square (LMS) algorithm is then employed to derive the first coefficient set of the AR model. In the backward derivation, each pixel in the reconstructed frame $t-2$ can be approximated as the weighted summation of corresponding pixels in the reconstructed frame $t-1$. By the centrosymmetric relation of the backward and forward derivations, the second coefficient set is derived. Finally, a probability based fusion is proposed in which the SI of the processing block within the current WZ frame t is generated as the fusion of the two regression results, generated by using the two derived coefficient sets, as well as the traditional extrapolation result. It should be noted that the proposed AR model employs the pixels centered around the pixel indicated by the motion trajectory to perform extrapolation rather then the pixels centered around the collocated pixel as in [23,24]. In addition to, the proposed AR model exploits the centrosymmetric property of the AR model to further improve the extrapolation accuracy. To verify the superiority of the proposed AR model based SI generation for the low-delay DVC, various experiments are conducted and the simulation results have confirmed that the proposed method is able to achieve SI with much higher accuracy compared with other existing methods.

The reminder of this paper is as follows. The overall architecture of the proposed system is first presented in Section 2. Then the model description and the forward and backward derivations are described in detail in Section 3. The probability based fusion is given in Section 4 followed by the experimental results and analysis in Section 5. Finally the conclusions are provided in the last section.

2. Framework overview

The block diagram of the proposed AR model based low-delay DVC is depicted in Fig. 1. The coding process starts by dividing the input frames into key frames and WZ frames. At the encoder side, the key frames are encoded using the H.264/AVC intra coding scheme. The WZ frames are encoded by applying the 4 × 4 H.264/AVC DCT transform and the DCT coefficients of the entire frame are grouped together in DCT bands. Each DCT band is uniformly quantized and the bit planes are sent to the turbo encoder. The turbo coding procedure for the DCT bands starts with the most significant bit planes and generates the respective parity bits which are stored in the buffer and transmitted in small amount upon decoder request.

At the decoder side, the key frames are decoded using H.264/AVC intra decoding scheme. For the WZ frames, the SI is first generated by the proposed AR model. As shown in Fig. 1, the SI generation is composed of three modules: traditional extrapolation and the interpolations by two AR coefficient sets. In the extrapolation, the motion of each block in the current WZ frame t is derived by performing motion estimation between the reconstructed frames $t-1$ and $t-2$. The first coefficient set of the AR model is computed by the forward derivation and the second coefficient set of the AR model is computed by the backward derivation followed by the centrosymmetric rearrangement. Both the first and second set coefficients are then used to generate the SI through interpolation process. Results of the three modules are then combined by a probability based fusion to generate the final SI. Then the iterative turbo decoder uses the received parity bits to correct the SI errors and generates the decoded quantized symbol stream. Finally, IDCT is applied to generate the WZ decoded frames.

3. Model description and its forward and backward derivations

In this section, we will first give the detail description of the proposed AR model, and then we will present the forward and backward derivations to compute two reliable AR coefficient sets so as to generate high quality SI.

3.1. Model description

In the proposed AR model, the SI of each pixel within the current WZ frame t is generated as a weighted summation of the pixels within a particular window in the previous reconstructed WZ/K frame $t-1$ as shown in Fig. 2. Let $\mathbf{X}_t$ be the current WZ frame at t, and $\mathbf{Y}_t$ be the SI of $\mathbf{X}_t$. For each pixel in $\mathbf{X}_t$, the window, indicated by the circles and the red arrow in Fig. 2, is determined by the integer-pixel accuracy motion field estimated during the motion extrapolation. After the determination of the window, the weighted summation is performed as

$$Y_t(m,n) = \sum_{-R \le (i,j) \le R} \hat{X}_{t-1}(\tilde{m}+i, \tilde{n}+j) \bullet \alpha(i,j). \tag{1}$$

Here $Y_t(m,n)$ represents the SI of the pixel located at (m,n), $\hat{\mathbf{X}}_{t-1}$ represents the previous reconstructed frame $t-1$, $(\tilde{m},\tilde{n})$ represents the corresponding integer-pixel position in $\hat{\mathbf{X}}_{t-1}$ determined by the motion vector of $\mathbf{X}_t(m,n)$, which is obtained during the motion extrapolation, $\alpha(i,j)$ is the forward AR coefficient from frame $t-1$ to frame t. In Eq. (1), R is defined to be the radius of the window, the size of which is $(2R+1) \times (2R+1)$. The proposed AR

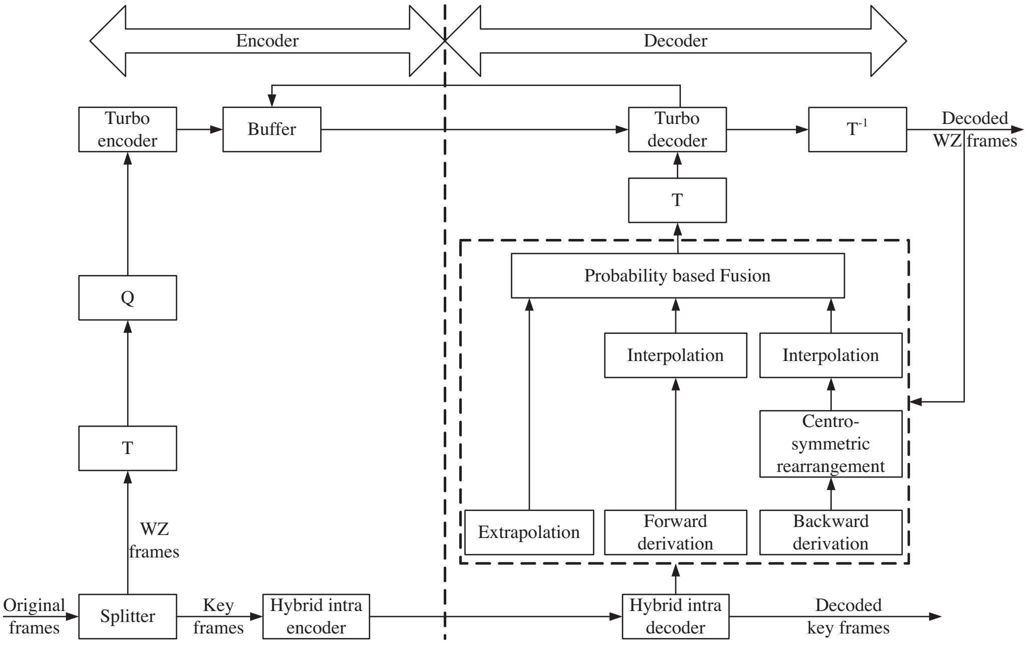


Fig. 1. The block diagram of the proposed AR model based low-delay DVC. The proposed scheme benefits from an improved SI generator (box surrounded by dashed lines).

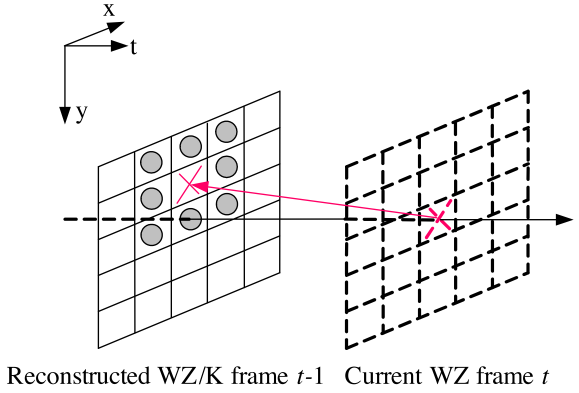


Fig. 2. The proposed AR model.

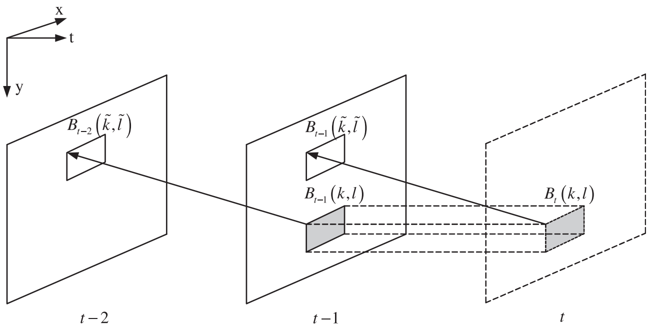

x
t
y
$B_{t-2}(\tilde{k},\tilde{l})$
$B_{t-1}(\tilde{k},\tilde{l})$
$B_{t-1}(k,l)$
$B_t(k,l)$
$t-2$
$t-1$
t

Fig. 3. Coefficient approximation illustration.

interpolation is performed block by block to fully capture the variation properties of the WZ frames.

Obviously, the coefficient estimation plays a critical role for the quality of SI generated by the proposed AR model. Since there is no access to the actual pixel in the current WZ frame $\mathbf{X}_t$ at the decoder, we devise a forward derivation and backward derivation in the following subsections to approximate the AR coefficients for each block within the current WZ frame $\mathbf{X}_t$.

3.2. Forward derivation

The forward derivation comprises the following steps. Firstly, for each block $B_t(k,l)$ located at position (k,l) within $\mathbf{X}_t$ shown in Fig. 3, we find its co-located block $B_{t-1}(k,l)$ in the previous reconstructed WZ/K frame $\hat{\mathbf{X}}_{t-1}$. Secondly, we find the best matched block $B_{t-2}(\tilde{k},\tilde{l})$ for $B_{t-1}(k,l)$ in the reconstructed WZ/K frame $\hat{\mathbf{X}}_{t-2}$ by motion estimation between $\hat{\mathbf{X}}_{t-1}$ and $\hat{\mathbf{X}}_{t-2}$. The displacement between $B_{t-1}(k,l)$ and $B_{t-2}(\tilde{k},\tilde{l})$ is denoted as $v_{t-1,t-2}(k,l)$. Based on the assumption that $B_t(k,l)$ and $B_{t-1}(k,l)$ obey the same motion trends, we conclude that $v_{t,t-1}(k,l)$ is equal to $v_{t-1,t-2}(k,l)$ and we use $v_{t,t-1}(k,l)$ to find the matched block $B_{t-1}(\tilde{k},\tilde{l})$ in $\hat{\mathbf{X}}_{t-1}$ for block $B_t(k,l)$. Thirdly, applying the proposed AR model, each pixel in $B_{t-1}(k,l)$ is approximated as a linear weighted summation of the pixels within a window, which is centered on the corresponding pixel, pointed by the motion vector $v_{t-1,t-2}(k,l)$ in block $B_{t-2}(\tilde{k},\tilde{l})$. In other words, each pixel (m,n) in $B_{t-1}(k,l)$ can be approximated as

$$\tilde{X}_{t-1}(m,n) = \sum_{-R\le(i,j)\le R} \hat{X}_{t-2}(\tilde{m}+i,\tilde{n}+j) \bullet \alpha(i,j). \tag{2}$$

Due to the piecewise stationary characteristics of the frame, we assume that all the pixels within block $B_{t-1}(k,l)$ share the same AR coefficients. The best coefficients can be computed by minimizing the mean squared error (MSE), the most common measure of performance of a predictor, which can be described as

$$\varepsilon_f^2(k,l) = \sum \sum_{(m,n)\in B_{t-1}(k,l)} E\left(\left\|\tilde{X}_{t-1}(m,n) - \hat{X}_{t-1}(m,n)\right\|^2\right). \tag{3}$$

If we pack the $(2R+1)\times(2R+1)$ window of each pixel within $B_{t-1}(k,l)$ into a $1\times[(2R+1)\times(2R+1)]$ row vector, then a matrix C_{t-2} sized $S\times[(2R+1)\times(2R+1)]$, where S denotes the number of pixels within $B_{t-1}(k,l)$, is obtained. According to LMS, the optimal AR coefficients can be computed as

$$\vec{\boldsymbol{\alpha}} = (C_{t-2}^T C_{t-2})^{-1}(C_{t-2}^T \vec{X}_{t-1}), \tag{4}$$

where $\vec{\boldsymbol{\alpha}}$ represents the optimal coefficient vector and $\vec{X}_{t-1}$ represents the pixel vector within $B_{t-1}(k,l)$.

Owing to the fact that there is a high similarity along the motion trajectory within adjacent frames, we assume that the forward AR coefficients for interpolating $B_{t-1}(k,l)$ as the linear combination of pixels in $B_{t-2}(\tilde{k},\tilde{l})$ are the same with those for interpolating $B_t(k,l)$ as the linear combination of pixels in $B_{t-1}(\tilde{k},\tilde{l})$. In other words, Eqs. (1) and (2) utilize the same coefficient $\alpha(i,j)$ to interpolate frames t and $t-1$, respectively. Assuming the first coefficient set derived according to Eq. (4) is $\alpha(i,j)$ $(-R \leqslant i,j \leqslant R)$, then $\alpha(i,j)$ can be used to obtain the SI of $B_t(k,l)$ by Eq. (1).

To further improve the accuracy of the generated SI, we derive another set of AR coefficients by backward derivation based on the centrosymmetric relation between the forward and backward derivations in the following two subsections.

3.3. Backward derivation

As shown in Fig. 4, the pixel in the reconstructed frame $t-2$ can be approximated as the weighted summation of the pixels within a window in the previous reconstructed frame $t-1$ as follows:

$$\tilde{X}_{t-2}(\tilde{m},\tilde{n}) = \sum_{-R\le(i,j)\le R} \hat{X}_{t-1}(m+i,n+j)\bullet\beta(i,j), \tag{5}$$

where $\beta(i,j)$ is the backward AR coefficient from frame $t-1$ to frame $t-2$. The optimal coefficient $\beta(i,j)$ corresponding to the backward derivation can be derived by minimizing

$$\varepsilon_b^2(\tilde{k},\tilde{l}) = \sum \sum_{(\tilde{m},\tilde{n})\in B_{t-2}(\tilde{k},\tilde{l})} E\left(\left\|\tilde{X}_{t-2}(\tilde{m},\tilde{n}) - \hat{X}_{t-2}(\tilde{m},\tilde{n})\right\|^2\right). \tag{6}$$

Similar to the forward derivation case, the optimal AR coefficients corresponding to the backward derivation can be computed as

$$\vec{\boldsymbol{\beta}} = (C_{t-1}^T C_{t-1})^{-1}(C_{t-1}^T \vec{X}_{t-2}), \tag{7}$$

where $\vec{\boldsymbol{\beta}}$ represents the optimal backward-derivation coefficient vector, C_{t-1} is a $S \times [(2R+1) \times (2R+1)]$ matrix, with S representing the number of pixels within $B_{t-2}(\tilde{k},\tilde{l})$, and $\vec{X}_{t-2}$ represents the pixel vector within $B_{t-2}(\tilde{k},\tilde{l})$.

We also assume the backward AR coefficients for interpolating $B_{t-2}(\tilde{k},\tilde{l})$ as the linear combination of pixels in $B_{t-1}(k,l)$ are the same with those for interpolating $B_{t-1}(\tilde{k},\tilde{l})$ as the linear combination of the corresponding pixels in $B_t(k,l)$. However, it is noted that our goal is to predict $B_t(k,l)$ rather than $B_{t-1}(\tilde{k},\tilde{l})$. To address this issue, we can exploit the centrosymmetric property between the forward derivation and backward derivation, which will be described in the next subsection, to derive another approximated forward AR coefficient set to predict $B_t(k,l)$.

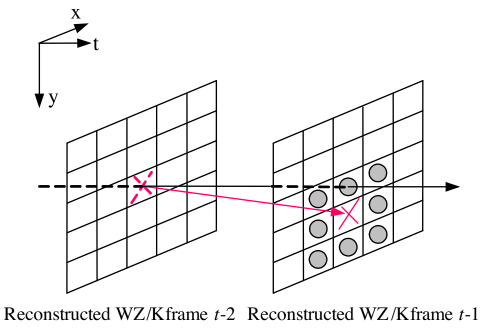


Fig. 4. Backward derivation.

3.4. Centrosymmetric rearrangement

We assume the AR coefficients, obtained by the forward derivation and backward derivation, of the same object are symmetric relative to the center of the AR model, which is depicted in Fig. 5. This assumption can be easily explained in geometry. For example, the motion vector, from $t-2$ to $t-1$ along a certain direction, can be embodied by the relatively larger AR coefficients along the corresponding direction [23]. If we rearrange the coefficients in a centrosymmetric way, the AR coefficients along the reverse direction become relatively larger, which thus in turn is embodied by the reverse motion vector, from $t-1$ to $t-2$, as the former one. Consequently, by the centrosymmetric relation, we can get

$$\beta'(i,j) = \beta(-i,-j), \tag{8}$$

where $\beta(i,j)$ is the backward coefficient from frame $t-1$ to frame $t-2$, which can be derived according to Eq. (7), and $\beta'(i,j)$ is the corresponding rearranged forward coefficient from frame $t-2$ to frame $t-1$. Due to the fact that there is a high similarity among the same objects within adjacent frames, we assume the corresponding forward AR coefficients $\beta'(i,j)$ from frame $t-2$ to frame $t-1$ are the same with the forward AR coefficients from frame $t-1$ to frame t. Therefore, replacing $\alpha(i,j)$ with $\beta'(i,j)$ in Eq. (1), we can get another $Y_t(m,n)$.

4. Probability based fusion

Similar to the fusion method proposed in [24], a probability strategy is employed in this paper to combine the different observations $(o_1,\ldots,o_K)$ of the SI generated by different methods, such as traditional extrapolation, the interpolation by forward derivation coefficients, and the interpolation by the backward derivation coefficients followed by the centrosymmetric rearrangement. The fused result of SI can be generated as the weighted summation of different SI observation o_k, which can be expressed as

$$f(\mathbf{O}) = \sum_{k=1}^{K} \gamma_k o_k, \tag{9}$$

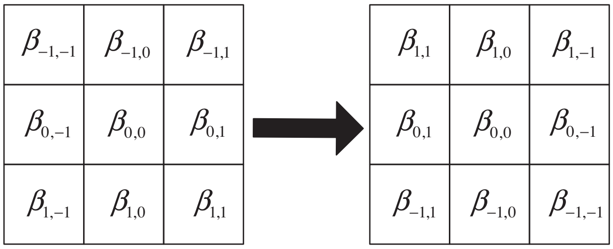
where $\mathbf{O} = (o_1,\ldots,o_K)$ represents the K SI observations generated by different methods, and γ_k represents the corresponding weight of the kth observation o_k. According to Bayesian rule, we have

$$\gamma_k = p(k|f(\mathbf{O})). \tag{10}$$

The posterior probability can be calculated by

$$p(k|f(\mathbf{O})) = \frac{p(f(\mathbf{O})|k)p(k)}{\sum_{l=1}^{K} p(f(\mathbf{O})|l)p(l)}, \tag{11}$$

where $p(k)$ represents the prior probability of the kth observation. For simplicity, the uniform prior $p(k) = 1/K$ is adopted in this paper. From Eq. (11), it is obvious that the conditional probability $p(f(\mathbf{O})|k)$



$\beta_{-1,-1}$	$\beta_{-1,0}$	$\beta_{-1,1}$
$\beta_{0,-1}$	$\beta_{0,0}$	$\beta_{0,1}$
$\beta_{1,-1}$	$\beta_{1,0}$	$\beta_{1,1}$

$\beta_{1,1}$	$\beta_{1,0}$	$\beta_{1,-1}$
$\beta_{0,1}$	$\beta_{0,0}$	$\beta_{0,-1}$
$\beta_{-1,1}$	$\beta_{-1,0}$	$\beta_{-1,-1}$

Fig. 5. The centrosymmetric rearrangement of the AR coefficients between the forward derivation and backward derivation.

plays a significant role to calculate $p(k|f(\mathbf{O}))$. In this paper, $p(f(\mathbf{O})|k)$ is assumed to be Gaussian probability function, which can be expressed as

$$p(f(\mathbf{O})|k) = p(\varepsilon_k) \propto \exp(-\varepsilon_k^2), \tag{12}$$

where ε_k represents the regression or motion compensation error.

Substituting Eq. (12) into Eqs. (10) and (11), we obtain

$$\gamma_k = p(k|f(\mathbf{O})) = \frac{\exp\left(\frac{-\varepsilon_k^2}{2\sigma_w^2}\right)}{\sum_{l=1}^{K} \exp\left(\frac{-\varepsilon_l^2}{2\sigma_w^2}\right)}, \tag{13}$$

where σ_w^2 is a constant used to control the shape of Gaussian probability function and in this paper it is set to be 20 empirically. It should be noted that since the actual pixels within $\mathbf{X}_t$ are not available at the decoder, the extrapolation error ε_1 is computed by

$$\varepsilon_1 = \frac{1}{M \times N} \sum_{(m,n)\in B_{t-1}(k,l)} \left\| \hat{X}_{t-1}(m,n) - \hat{X}_{t-2}(\widehat{m},\widehat{n}) \right\|^2, \tag{14}$$

where M and N represent the height and width of $B_{t-1}(k,l)$, (m,n) and $(\widehat{m},\widehat{n})$ represent the pixel coordinates in $B_{t-1}(k,l)$ and its matched (best predicted with quarter-pixel accuracy) block in $\hat{X}_{t-2}$, respectively. The regression error ε_2 brought by the forward derivation is computed by

$$\varepsilon_2 = \frac{1}{M \times N} \times \sum_{(m,n)\in B_{t-1}(k,l)} \left\| \hat{X}_{t-1}(m,n) - \sum_{-R\le(i,j)\le R} \hat{X}_{t-2}(\tilde{m}+i,\tilde{n}+j) \bullet \alpha(i,j) \right\|^2, \tag{15}$$

where $\hat{X}_{t-2}(\tilde{m},\tilde{n})$ is the corresponding matched integer-pixel accuracy pixel for each pixel in $B_{t-1}(k,l)$, and $\alpha(i,j)$ is the optimal interpolation coefficient derived by LMS according to Eq. (4). Similarly, the regression error ε_3 brought by the backward derivation is computed by

$$\varepsilon_3 = \frac{1}{M \times N} \times \sum_{(m,n)\in B_{t-1}(k,l)} \left\| \hat{X}_{t-1}(m,n) - \sum_{-R\le(i,j)\le R} \hat{X}_{t-2}(\tilde{m}+i,\tilde{n}+j) \bullet \beta'(i,j) \right\|^2, \tag{16}$$

where $\beta'(i,j)$ is the rearranged coefficient computed by backward derivation. It is easy to see in Eq. (13) that the smaller error ε_k will lead to larger weights in the mixture model, which can better match the assumption that γ_k should reflect the confidence about the kth mixing component. Compared with the non-fusion method, the Bayesian estimation is theoretically more accurate since it subtly combines the mixing components given an appropriate constant σ_w^2.

5. Experimental results and analysis

We have conducted various experiments in this section to evaluate the performance of the proposed AR model based SI generation for low-delay DVC. The proposed AR interpolations are carried out with and without probability based fusion, respectively. Here we use the state of the art work in [19] to perform the motion estimation and use it as the anchor to show the effectiveness of the proposed extrapolation scheme. Two key frames are preceding the first WZ frame in order to derive the motion information of the first WZ frame for its SI generation. The remaining key frame frequency was set to one key frame every two frames and the key frames of the test sequences are encoded by the H.264/AVC intra-frame encoder. The QPs of the key frames are set to be 26, 28 and 30, respectively.

Experimental results are presented on four QCIF@30 Hz video sequences including *Mobile*, *Foreman*, *Bus*, and *Paris*. An RTCP turbo encoder with two identical 4/5 rate constituent convolution encoders and a generator matrix of $\begin{bmatrix} 1 & \begin{matrix} 1+D+D^3+D^4 \\ 1+D^3+D^4 \end{matrix} \end{bmatrix}$ were used [5], and a random puncturing pattern of period 32 was used with a maximum of 30 iterations. The acceptable bit error rate threshold was set to 10^{-3}.

5.1. Frame interpolation without probability based fusion

In this sub-section, we will present the PSNRs of SI without probability based fusion by the proposed AR method and the extrapolation methods [19] and [25]. Fig. 6 shows the PSNR of each interpolated frame generated by the extrapolation methods [19] and [25], forward derivation (FD), backward derivation (BD) as well as forward derivation and backward derivation averaging (FBD_Avg), where the QPs of the key frames are set to be 28. The extrapolation method [19] has slightly better performance than the one in [25]. And the SI generated by the proposed AR model significantly outperforms the extrapolation methods in [19] and [25]. For almost all the frames, the SI generated by FD are able to achieve a significant PSNR gain compared with the extrapolation results. Especially, for *Mobile* sequence, the gain can be up to 4 dB. Although BD has poorer performance than FD does, it still has promising performance compared with the extrapolation method, e.g. for the majority frames BD is able to generate SI with much higher PSNR than the extrapolation does. Besides, when applying FBD_Avg the performance is the best among the four methods. This can verify that when exploiting the centrosymmetric property of the proposed AR model, the SI generated by FBD_Avg is more promising than that generated by only FD or only BD.

5.2. Frame interpolation with probability based fusion

To better illustrate the impact of the probability based fusion method on the quality of generated SI, we present the PSNR of each SI generated by the proposed AR model with and without the probability based fusion in Fig. 7, where the QPs of the key frames are set to be 28. The FD_E_Fusion represents fusion results by applying the fusion method on the interpolation by forward derivation and the extrapolation method [19]. And FBD_E_Fusion represents the fusion results by applying the fusion method on the interpolation by forward derivation, the interpolation by the backward derivation and the extrapolation method [19]. It can be seen that when the probability based fusion is applied, the PSNR of each SI gets improved. Among the four results, the FBD_E_Fusion achieves the best performance, since it elegantly integrates the interpolation results generated by extrapolation [19], interpolation by forward derivation, and the interpolation by backward derivation by adaptively adjusting the weight of each SI result.

Table 1 summaries the average PSNR of SIs for different interpolation methods. Here, R represents the radius of the AR model used to generate the SI for the corresponding sequence. It shows that when FBD_E_Fusion is used, the PSNR gains can be up to 4.77 dB, 2.98 dB, 1.33 dB, and 1.5 dB compared to the extrapolation method [19], if the QPs of the key frames are set to be 26, for *Mobile*, *Foreman*, *Bus*, and *Paris*, respectively. This is because the motion vectors derived by the extrapolation method are not very accurate sometimes, whereas the proposed AR interpolation has the superior ability of predicting the future data based on its history observations by adaptively tuning the interpolation coefficients.

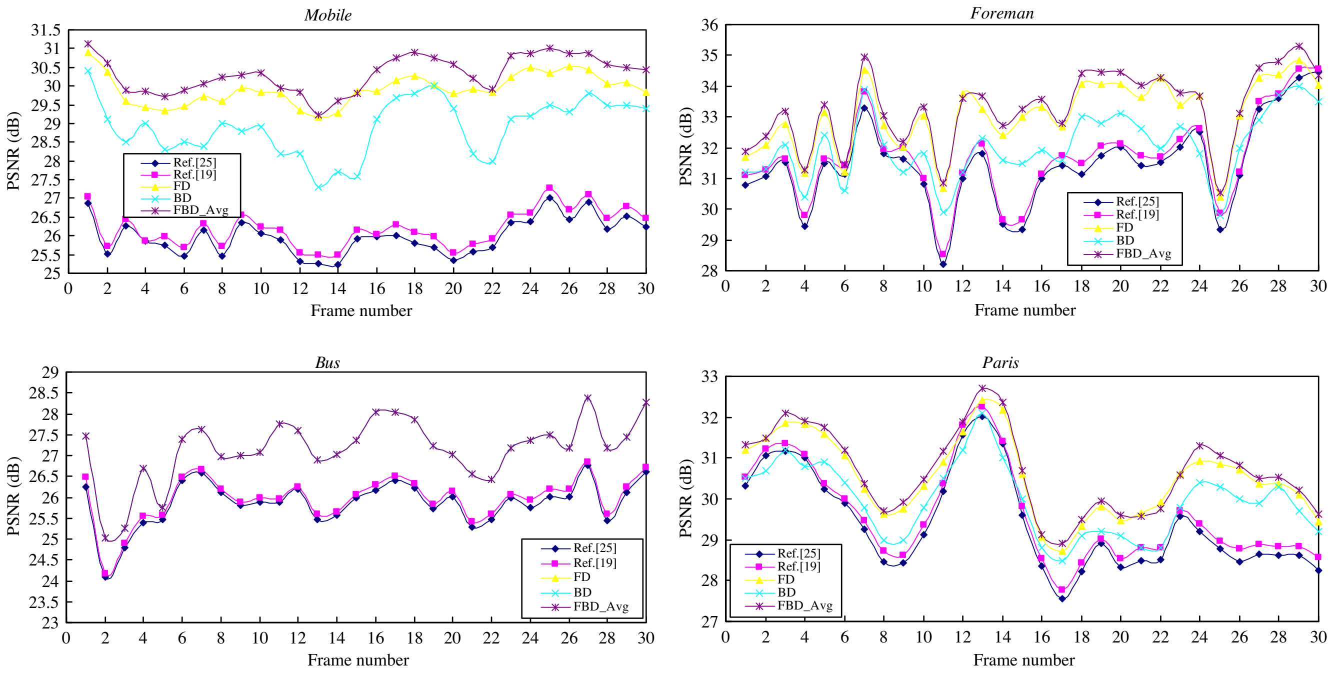


Fig. 6. PSNR of SI by extrapolation method [19], FD, BD and FBD_Avg.

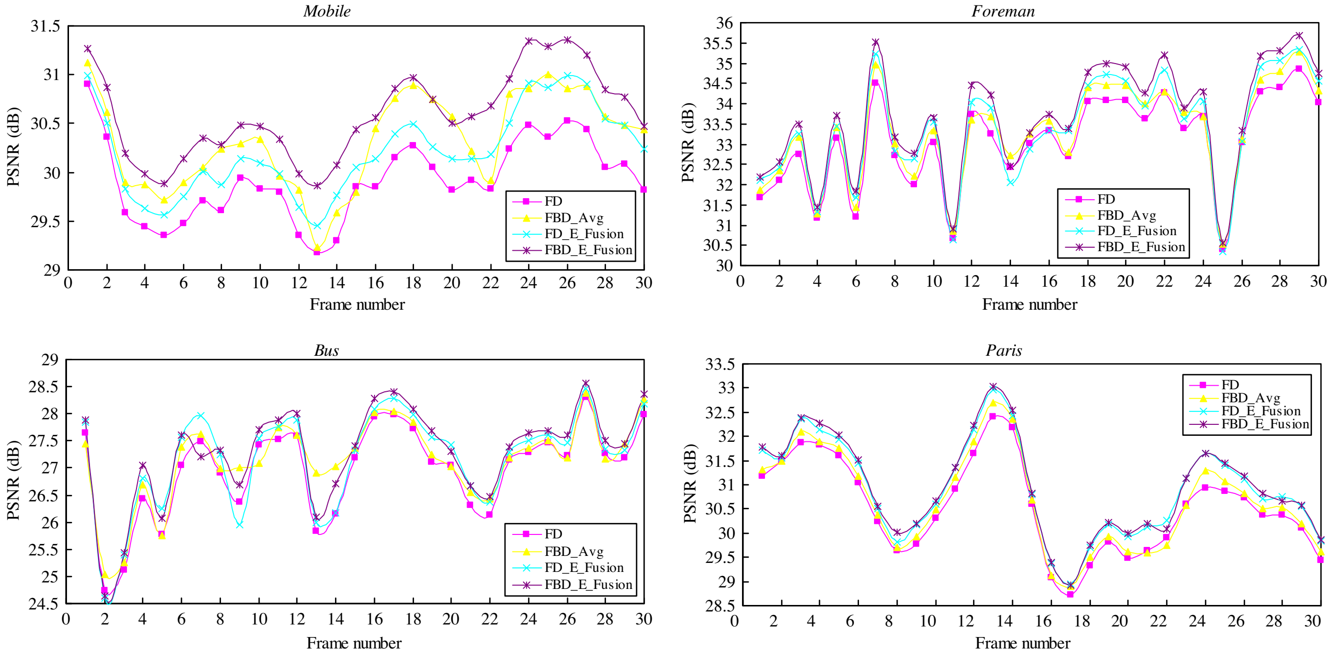


Fig. 7. PSNR of each SI by the proposed AR method with and without probability based fusion.

Table 1
Average PSNR of SIs when the key frames are encoded under different QPs by H.264/AVC intra encoder.

Method	*Mobile* (*R* = 2) QP			*Foreman* (*R* = 2) QP			*Bus* (*R* = 1) QP			*Paris* (*R* = 1) QP		
	26	28	30	26	28	30	26	28	30	26	28	30
Ref. [19]	26.430	26.184	25.811	31.310	30.980	30.583	26.133	25.974	25.761	29.906	29.607	29.059
FD	30.529	29.919	29.060	33.701	33.055	32.248	27.111	26.979	26.657	30.958	30.538	29.922
FD_E_Fusion	30.814	30.216	29.316	34.033	33.403	32.604	27.324	27.196	26.886	31.379	30.920	30.274
FBD_Avg	30.919	30.329	29.503	33.937	33.308	32.490	27.303	27.154	26.878	31.095	30.670	30.061
FBD_E_Fusion	**31.200**	**30.592**	**29.699**	**34.290**	**33.663**	**32.878**	**27.463**	**27.294**	**27.018**	**31.428**	**30.965**	**30.333**

The bold values represent the highest PSNR values of each test sequence under each QP.

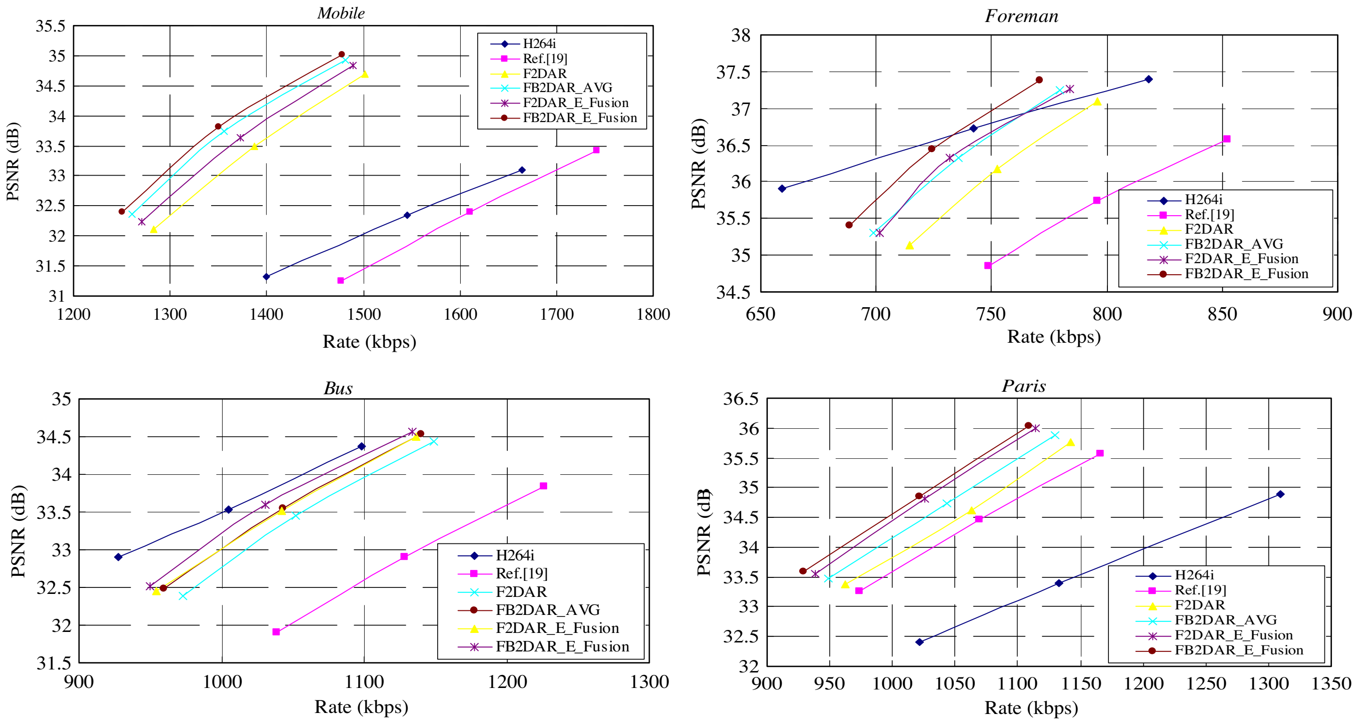


Fig. 8. Rate-distortion curves for H.264/AVC intra, extrapolation method, and the proposed AR interpolation method.

5.3. Rate-distortion performances

Fig. 8 compares the rate-distortion performances, averaged over all the key frames and WZ frames, of the extrapolation methods [19] and [25], the proposed AR interpolation method, and the intra coded results by H.264/AVC reference software version JM 9.8. The proposed AR model strongly outperforms the extrapolation based DVC schemes [19] and [25], confirming the superior ability of the proposed AR model to generate the high quality SI. From Fig. 8, it is observed that the proposed AR model is superior over the whole range of bit rates compared to the extrapolation based low-delay DVC [19] and [25]. For *Mobile* sequence, the extrapolation based DVC schemes have poorer performance than H.264/AVC intra encoder does, however, the proposed AR model has superior performance than H.264/AVC intra encoder and the highest PSNR gain can be up to 3.2 dB for FBD_E_Fusion. For *Foreman* and *Bus* sequences, the extrapolation based DVC schemes still have inferior performance compared with H.264/AVC intra encoder, whereas the proposed AR model is able to reduce the gap between H.264/AVC intra encoder and the extrapolation based DVC. For example, for *Bus* sequence, the gap between the two has been reduced to 0.1 dB from 1.7 dB. Besides, for *Foreman* sequence, the proposed AR model is even able to outperform H.264/AVC intra encoder at higher bit rates. *Paris* is the only sequence among the four test ones whose extrapolation based DVC result is superior to that of the H.264/AVC intra encoder. The PSNR gain of the extrapolation based DVC [19] is about 1.6 dB compared with the H.264/AVC intra encoder does, while the FBD_E_Fusion is able to further improve the performance of the DVC to 2.8 dB compared to H.264/AVC intra encoder. The superior rate-distortion performance of the proposed AR model greatly attributes to its desirable ability to generate the high quality SI and thus reduce the number of parity bits to correct the errors between the SI and the original frame.

6. Conclusions

In this paper, we have explored the benefits of the AR model for the SI generation in low-delay DVC. In the proposed AR model, the SI of each pixel in the current WZ frame t can be generated as a weighted summation of pixels within a special window in the previous reconstructed WZ/K frame $t-1$. To obtain high quality SI, we use the forward derivation and backward derivation to derive two weighting coefficient sets. In the forward derivation, each reconstructed pixel within the frame $t-1$ is approximated as a linear weighted summation of pixels within the corresponding window in the reconstructed frame $t-2$. Applying the LMS, the first coefficient set is derived. In the backward derivation, each pixel in the reconstructed frame $t-2$ can be approximated as the weighted summation of corresponding pixels in the reconstructed frame $t-1$. By the centrosymmetric relation of the backward and forward derivations, the second coefficient set is derived. The ultimate SI is generated by applying probability based fusion on the extrapolation, interpolation by forward derivation, and interpolation by backward derivation. The proposed method achieves significantly better results compared to the extrapolation method in terms of PSNR values. In addition, the rate-distortion performance of the proposed method has confirmed that the proposed AR model is able to reduce the gap between the low-delay DVC and H.264/AVC intra encoder or even outperform H.264/AVC intra encoder.

Acknowledgments

This work was supported in part by the National Science Foundations of China: 60736043 and the Major State Basic Research Development Program of China (973 Program 2009CB320905).

References

[1] R. Gallager, Low-Density Parity-Check Codes, MIT Press, Cambridge, MA, 1963.
[2] D. Slepian, J.K. Wolf, Noiseless coding of correlated information sources, IEEE Trans. Inf. Theor. IT-19 (1973) 471–480.
[3] A.D. Wyner, J. Ziv, The rate distortion function for source coding with side information at the decoder, IEEE Trans. Inf. Theor. IT-22 (1) (1976) 1–10.
[4] S. Pradhan, K. Ramchandran, Distributed source coding using syndromes (DISUC): design and construction, IEEE Trans. Inf. Theor. 49 (3) (2003) 626–643.

[5] A. Aaron, B. Girod, Compression with side information using turbo codes, presented at the IEEE Int. Data Compression Conf., 2002.

[6] J. Garcia-Frias, Y. Zhao, Compression of correlated binary sources using turbo codes, IEEE Commun. Lett. 5 (10) (2001) 417–419.

[7] T. Tian, J. Garcia-Frias, W. Zhong, Compression of correlated sources using ldpc codes, presented at the IEEE Int. Data Compression Conf., 2003.

[8] R. Puri and K. Ramchandran, "PRISM: a new robust video coding architecture based on distributed compression principles," Allerton Conference on Communication, Control and Computing, 2002.

[9] A. Aaron, R. Zhang, B. Girod, "Wyner-Ziv coding of motion video," in: Proceedings of the Asilomar Conference on Signals and Systems, Pacific Grove, CA, November 2002.

[10] A. Aaron, D. Varodayan, B. Girod, "Wyner-Ziv Residual Coding of Video," in: Proceedings of the International Picture Coding Symposium, Beijing, PR China, April 2006.

[11] J. Ascenso, C. Brites, F. Pereira, "Content Adaptive Wyner-Ziv Video Coding Driven by Motion Activity," in: IEEE International Conference on Image Processing, Atlanta, USA, October 2006.

[12] Wei-Jung Chien, L.J. Karam, G.P. Abousleman, "Distributed video coding with 3D recursive search block matching," in: IEEE International Symposium on Circuits and Systems, Island of Kos, Greece, May 2006.

[13] M. Tagliasacchi, S. Tubaro, A. Sarti, "On the Modeling of Motion in Wyner–Ziv Video Coding," in: IEEE International Conference on Image Processing, Atlanta, USA, October 2006.

[14] Z. Li, L. Liu, E.J. Delp, Rate distortion analysis of motion side estimation in Wyner–Ziv video coding, IEEE Trans. Image Process. 16 (1) (2007) 98–113.

[15] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, "Exploiting Spatial Redundancy In Pixel Domain Wyner–Ziv Video Coding," in: IEEE International Conference on Image Processing, Atlanta, USA, October 2006.

[16] A.B.B. Adikari, W.A.C. Fernando, H. Kodikara Arachchi, W.A.R.J. Weerakkody, Sequential motion estimation using luminance and chrominance information for distributed video coding of Wyner–Ziv frames, Electron. Lett. 42 (7) (2006) 398–399.

[17] Z. Li, L. Liu, E.J. Delp, Wyner–Ziv video coding with universal prediction, IEEE Trans. Circ. Syst. Video Technol. 16 (11) (2006) 1430–1436.

[18] A. Aaron and B. Girod, "Wyner-Ziv video coding with low-encoder complexity," in: Proceedings of the Picture Coding Symposium, San Francisco, CA, December 2004.

[19] L. Natrio, C. Brites, J. Ascenso, F. Pereira, "Extrapolating Side Information for Low-Delay Pixel-Domain Distributed Video Coding," in: International workshop on Very Low Bitrate Video Coding, Sardinia, Italy, September 2005.

[20] S. Borchert, R. P. Westerlaken, R. K. Gunnewiek, R. L. Lagendijk, "On extrapolating side information in distributed video coding," in: Proceedings of Picture Coding Symposium, PCS 2007, Lisbon, Portugal, November 2007.

[21] J. Ascenso, C. Brites F. Pereira, "Motion Compensated Refinement for Low Complexity Pixel Based Distributed Video Coding," in AVSS 2005, in: IEEE Conference on Advanced Video and Signal Based Surveillance, 15–16 September 2005, pp. 593–598.

[22] Yongbing Zhang, Debin Zhao, Siwei Ma, Ronggang Wang, and Wen Gao, "An auto-regressive model for improved low-delay distributed video coding," in: Proceedings of SPIE Conference on Visual Commun and Image Processing, San Jose, California, USA, January 2009, pp. 18–22.

[23] X. Li, "Least-square prediction for backward adaptive video coding," EURASIP Journal on Applied Signal Processing, 2006, special Issue on H.264 and Beyond.

[24] X. Li, Video processing via implicit and mixture motion models, IEEE Trans. Circ. Syst. Video Technol. 17 (8) (2007) 953–963.

[25] A. Aaron, S. Rane, E. Setton, B. Girod, "Transform-domain Wyner–Ziv codec for video," in: Proceedings of SPIE Conference on Visual Commun and Image Processing, San Jose, California, USA, January 2004, pp. 520–528.