Wyner–Ziv Video Coding With Classified Correlation Noise Estimation and Key Frame Coding Mode Selection

Ghazaleh Rais Esmaili, *Student Member, IEEE*, and Pamela C. Cosman, *Fellow, IEEE*

***Abstract*—We improve the overall rate-distortion performance of distributed video coding by efficient techniques of correlation noise estimation and key frame encoding. In existing transform-domain Wyner–Ziv video coding methods, blocks within a frame are treated uniformly to estimate the correlation noise even though the success of generating side information is different for each block. We propose a method to estimate the correlation noise by differentiating blocks within a frame based on the accuracy of the side information. Simulation results show up to 2 dB improvement over conventional methods without increasing encoder complexity. Also, in traditional Wyner–Ziv video coding, the intercorrelation of key frames is not exploited since they are simply intracoded. In this paper, we propose a frequency band coding mode selection for key frames to exploit similarities between adjacent key frames at the decoder. Simulation results show significant improvement especially for low-motion and high frame rate sequences. Furthermore, the advantage of applying both schemes in a hierarchical order is investigated. This method achieves additional improvement.**

***Index Terms*—Correlation channel, distributed source coding, key frame encoding, Wyner–Ziv coding.**

I. Introduction

Motion-compensated predictive coding is a successful method for exploiting interframe correlation and is used in traditional video coding standards such as MPEG-x and H.26x. In this technique, the encoder exploits spatial and temporal correlations and can choose flexibly between different coding modes and encoding parameters. The encoder complexity is much higher than the decoder complexity. For some recent applications, such as sensor networks, video surveillance, and mobile camera phones, many simple and low-cost encoders are required but a high-complexity decoder can be used. Wyner–Ziv video coding which is founded on the Slepian and Wolf [1] and Wyner and Ziv [2] theorems is a promising solution for such applications. In this approach, the complexity is largely shifted from the encoder to the decoder by encoding individual frames independently (intraframe encoding) but decoding them conditionally (interframe decoding).

Manuscript received October 29, 2009; revised April 12, 2010 and October 06, 2010; accepted February 02, 2011. Date of publication March 03, 2011; date of current version August 19, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Antonio Ortega.

The authors are with the Department of Electrical and Computer Engineering, University of California, San Diego, CA 92093-0407 USA (e-mail: gesmaili@ucsd.edu; pcosman@ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2121079

As a first implementation of distributed video coding (DVC), Puri and Ramchandran [3] and Puri *et al.* [4] introduced a syndrome-based video coding scheme which deployed block-level coding primitives, and no feedback was required. The algorithms proposed in [5]–[9] require feedback which became the basis for considerable further research. In [10], Brites *et al.* outperformed [6] by adjusting the quantization step size and applying an advanced frame interpolation for side information generation. Later, in [11]–[14], enhanced techniques of side generation were proposed to achieve better performance. In [15] and [16], blocks were differentiated to use intra- or Wyner–Ziv coding.

In most Wyner–Ziv coding schemes, the decoder needs some model for the statistical dependence between the source and the side information to make use of the side information. Accurate modeling of correlation has a strong impact on performance by exploiting the statistics between source and side information [17]. The dependence between source and side information is modeled by $Y = X + Z$ where Y denotes the side information and X denotes the source. Z is called the correlation noise. In [18], the correlation noise was modeled by different distributions, and the relationship between the compression ratio and sensitivity of the estimated channel model parameter was investigated.

In most approaches, the probability density function of Z is approximated by a Laplacian distribution and its corresponding parameters are estimated by plotting the residual histogram of several sequences. In these methods, the estimated Laplacian parameter is the same for all blocks within a frame, even though the accuracy of the side information varies based on the motion compensated frame interplation (MCFI) success. In [19], a method was proposed to estimate the pixel domain correlation noise by online adjustment of the Laplacian parameter for each block. In [20] and [21], some methods at frame, block, and pixel levels were suggested for online parameter estimation of pixel and transform-domain Wyner–Ziv (TDWZ) coding. Their proposed method for transform-domain correlation noise estimation was improved by Huang and Forchhammer in [22] by utilizing cross-band correlation. In this paper, we propose a simple and effective method to differentiate blocks within a frame to estimate the correlation noise based on MCFI success at the decoder.

Exploiting the temporal correlation of key frames is another contribution of this paper. As mentioned, key frames are usually intraencoded and decoded, so the interframe correlation between them is not exploited. Extending Wyner–Ziv

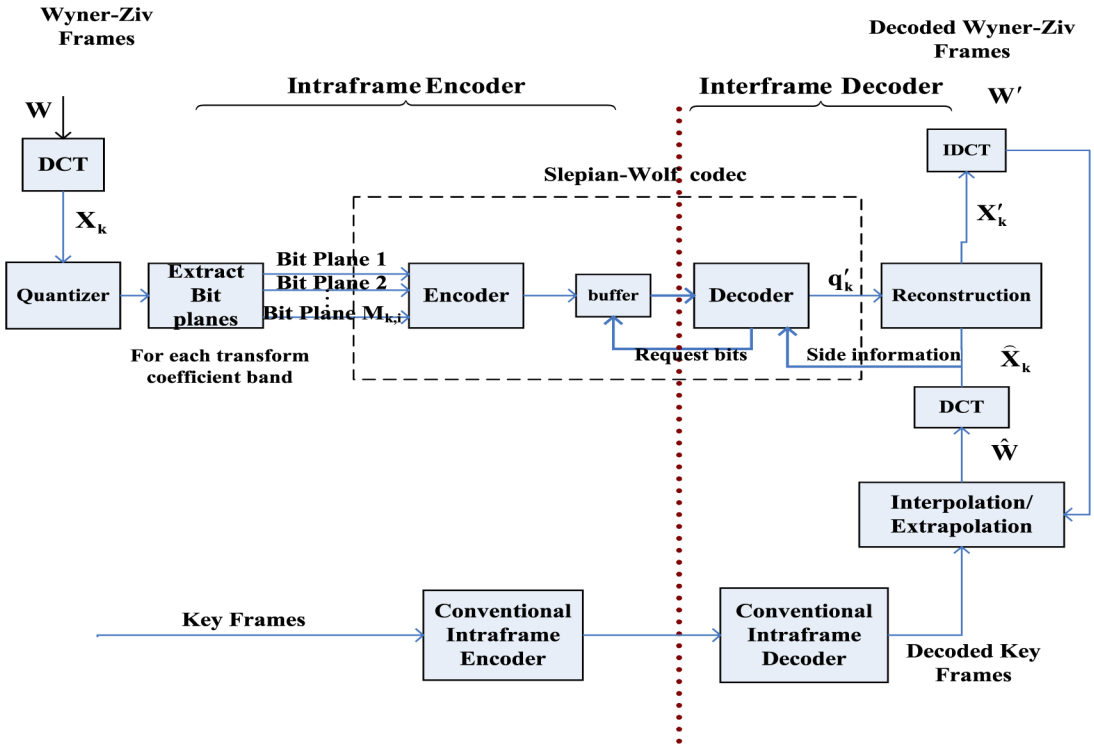


Fig. 1. TDWZ video codec.

coding to key frames as well can help to exploit the temporal correlation and improve the rate-distortion performance. In [23], Wyner–Ziv coding was applied for key frames and the previously decoded key frame was considered as the pixel domain side information for the next key frame to be decoded. Their results showed improvement for two low-motion sequences. However, as shown in [24] and [25], directly applying Wyner–Ziv coding on key frames can degrade the overall performance since Wyner–Ziv coding is capable of outperforming intracoding only when the side information is accurate enough. Using the previously decoded key frame as the side information for the next key frame to be encoded is usually not accurate enough, especially for high-motion sequences. We extend the Wyner–Ziv coding method to key frames by applying a coding mode selection technique that tries to select the proper coding method (Intra or Wyner–Ziv) based on the correlation characteristics of the low- and high-frequency bands of each frame to the past. In this method, the decoder decides the coding mode and no complexity is added to the conventional Wyner–Ziv encoder. After decoding low bands, a new method is used to refine the side information corresponding to the remaining frequency bands.

Finally, we propose and study a hierarchical coding structure applying both of the proposed methods of noise classification and key frame coding. We examine sequences with different motion characteristics at different frame rates. The rest of this paper is organized as follows. In Section II, TDWZ coding is reviewed. In Section III, correlation noise classification based on matching success is described in detail. Key frame encoding based on frequency band classification and side information refinement is explained in Section IV. After presenting hierarchical coding in Section V, the performance of different proposed methods is evaluated in Section VI.

II. TDWZ Coding

The TDWZ video codec architecture proposed in [6] is our reference. As depicted in Fig. 1, key frames are encoded and decoded by a conventional intraframe codec. The frames between them (Wyner–Ziv frames) are also encoded independently of any other frame, but their decoding makes use of other frames. In the following, the term decoder refers to the entire interframe decoder of Fig. 1, whereas the term Slepian–Wolf decoder refers to the decoder module inside the Slepian–Wolf codec.

At the encoder, a blockwise 4×4 discrete cosine transform (DCT) is applied on Wyner–Ziv frames. If there are N blocks in the image, X_k (for $k = 1$ to 16) is a vector of length N obtained by grouping together the kth DCT coefficients from all blocks. To have the same quality for both Wyner–Ziv and intra modes, $Q(x)$ is used to quantize DCT coefficients where

$$Q(a_{i,j}) = \text{round}\left(\frac{a_{i,j}}{QP \times c_{i,j}}\right) \tag{1}$$

and $a_{i,j}$ is the unquantized coefficient at position (i, j). $c_{i,j}$ is the element of the quantization matrix at position (i, j) and QP is the quantization parameter. The quantization matrix applied in our simulation is the initializing quantization matrix borrowed from H.264 JM 9.6, as follows:

$$C = \begin{bmatrix} 6 & 12 & 19 & 26 \\ 12 & 19 & 26 & 31 \\ 19 & 26 & 31 & 35 \\ 26 & 31 & 35 & 39 \end{bmatrix}.$$

TABLE I
LOOKUP TABLE OF α PARAMETERS FOR 16 DCT BANDS OF DIFFERENT CLASSES

class	$f_{1,1}$	$f_{1,2}$	$f_{1,3}$	$f_{1,4}$	$f_{2,1}$	$f_{2,2}$	$f_{2,3}$	$f_{2,4}$	$f_{3,1}$	$f_{3,2}$	$f_{3,3}$	$f_{3,4}$	$f_{4,1}$	$f_{4,2}$	$f_{4,3}$	$f_{4,4}$
1	1.10	1.28	1.43	1.70	1.31	1.86	1.99	2.24	1.70	2.18	2.34	2.60	2.07	2.63	2.73	3.08
2	1.03	1.13	1.14	1.15	1.05	1.37	1.39	1.58	1.23	1.55	1.70	2.01	1.49	1.89	2.08	2.37
3	0.83	0.98	0.99	1.01	0.89	1.05	1.06	1.22	0.97	1.13	1.25	1.47	1.15	1.45	1.59	1.87
4	0.64	0.89	0.91	0.93	0.66	0.87	0.89	1.00	0.70	0.88	0.97	1.17	0.85	1.09	1.23	1.51
5	0.37	0.55	0.58	0.60	0.44	0.60	0.62	0.68	0.48	0.58	0.63	0.75	0.56	0.69	0.78	0.95
6	0.24	0.36	0.41	0.45	0.32	0.44	0.48	0.54	0.38	0.45	0.48	0.56	0.44	0.52	0.55	0.67
7	0.17	0.28	0.33	0.38	0.24	0.32	0.37	0.43	0.30	0.34	0.38	0.45	0.36	0.41	0.44	0.51
8	0.10	0.16	0.19	0.25	0.15	0.18	0.22	0.28	0.19	0.22	0.23	0.29	0.24	0.27	0.29	0.33
Unique	0.20	0.31	0.36	0.41	0.28	0.36	0.41	0.50	0.34	0.39	0.44	0.52	0.40	0.47	0.51	0.63

The coefficients of X_k are quantized to form a vector of quantized symbols q_k. That is, q_k is the vector of quantization step indices for the elements of X_k. After representing the quantized values in binary form, bit-plane vectors $M_{k,i}$ ($i = 1$ to I_k) are extracted, where I_k is the maximum number of bit planes for frequency band k. The maximum number of bit planes for frequency band k is calculated by

$$I_k = \begin{cases} \lfloor log_2 |v_k|_{\max} + 1 \rfloor & \text{if } k = 1 \\ \lfloor log_2 |v_k|_{\max} + 1 \rfloor + 1 & \text{otherwise} \end{cases} \quad (2)$$

where $|v_k|_{\max}$ is the highest absolute value within frequency band k. The encoder lets the decoder know the maximum number of bit planes for each frequency band within a frame. Each bit-plane vector then enters the Slepian–Wolf [Turbo or low-density parity-check accumulate (LDPCA)] encoder. The parity bits (or accumulated syndrome bits) generated by the Turbo (or LDPCA) encoder are stored in the buffer and sent in chunks upon the decoder request through the feedback channel until a desired bit error rate is met. Our simulation setup assumed ideal error detection.

At the decoder, $\hat{W}$ is the estimate of W (Wyner–Ziv frame) which is generated by applying extrapolation or interpolation techniques on decoded key frames. For a group of pictures of size 2, a motion compensation interpolation technique that will be briefly explained in Section III is applied on previous and next key frames to estimate the Wyner–Ziv frame in between. A blockwise 4×4 DCT is applied on $\hat{W}$ to produce $\hat{X}$. $\hat{X}_k$, the side information corresponding to X_k, is generated by grouping the transform coefficients of $\hat{X}$. When all the bit planes are decoded, the bits are regrouped to form a vector of reconstructed quantized symbols q_k. At the end, the reconstructed coefficient band X_k is calculated as $E(X_k|q_k, \hat{X}_k)$.

The Slepian–Wolf decoder and reconstruction block assume a Laplacian distribution to model the statistical dependence between X_k and $\hat{X}_k$. Although more accurate models such as generalized Gaussian can be applied, Laplacian is selected for good balancing of accuracy and complexity. The distribution of d can be approximated as

$$f(d) = \frac{\alpha}{2} e^{-\alpha |d|} \quad (3)$$

where d denotes the difference between corresponding elements of X_k and $\hat{X}_k$. In existing approaches [5]–[10], a different α parameter is assigned for each frequency band. These α parameters are estimated by plotting the residual histogram of several sequences using MCFI for the side information. For example, for frequency band k the differences between corresponding elements in X_k and $\hat{X}_k$ of several sequences are grouped to form a set F_k. The α parameter is calculated by $\sqrt{2}/\sigma_k$, where σ_k is the square root of the variance of the F_k values. In this way, we have a 16-element lookup table at the reconstruction block and Slepian–Wolf decoder. An example of it is shown in the last row of Table I where each element represents the α parameter of the corresponding DCT band.

III. CORRELATION NOISE CLASSIFICATION BASED ON MATCHING SUCCESS

The main usage of correlation noise estimation is in the calculation of the conditional probability of the Slepian–Wolf decoder which, in our case uses the regular degree 3 LDPCA codes proposed in [26]. More accurate estimation of the dependence between source and side information means that fewer accumulated syndrome bits need to be sent, resulting in improved rate-distortion performance. Traditional estimation of Laplacian distribution parameters treats all frames and blocks within a frame uniformly, even though the quality of the side information varies spatially and temporally. General MCFI methods are based on the assumption that the motion is translational and linear over time among temporally adjacent frames. This assumption often holds for relatively small motion but tends to give a poor estimation for high-motion regions. The general approach to estimate a given block B in the interpolated frame F_t is to find the motion vector of the colocated block in F_{t+1} with reference to frame F_{t-1}, where t, $t-1$ and $t+1$ are time indexes. In [27], the motion vectors obtained by block matching in the previous step are refined by a bidirectional motion estimation technique. A spatial smoothing algorithm is then used to improve the accuracy of the motion field. If $v = (v_x, v_y)$ is the final motion vector, where v_x and v_y are the x and y components of v, then the interpolated block is obtained by averaging the pixels in F_{t-1} and F_{t+1} pointed to by $v/2$ and $-v/2$. These blocks of pixels in F_{t-1} and F_{t+1} which are called forward and backward interpolations, FMCFI and BMCFI, respectively can be calculated as

$$\text{FMCFI}(x,y) = F_{t-1}\left(x + \frac{v_x}{2}, y + \frac{v_y}{2}\right) \quad (4)$$

$$\text{BMCFI}(x,y) = F_{t+1}\left(x - \frac{v_x}{2}, y - \frac{v_y}{2}\right). \quad (5)$$

The interpolated block is calculated by

$$F_t(x,y) = \frac{\text{FMCFI}(x,y) + \text{BMCFI}(x,y)}{2}, \quad (x,y) \epsilon B. \quad (6)$$

The residual energy between FMCFI and BMCFI is computed by

$$E = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} [\mathrm{FMCFI}(x,y) - \mathrm{BMCFI}(x,y)]^2 \quad (7)$$

where M and N represent the block size (in our case $M = N = 4$). In [21], the residual between forward and backward interpolations was applied to estimate the correlation noise. $R(x,y)$ is the residual frame and is calculated as

$$R(x,y) = \frac{F_{t-1}(x + v_x/2,\ y + v_y/2) - F_{t+1}(x - v_x/2,\ y - v_y/2)}{2}. \quad (8)$$

They define $T(u,v) = \mathrm{DCT}(R(x,y))$. The α parameter for frequency band b and frame s is $\alpha_{b,s} = \sqrt{2}/\sigma_{b,s}$ where $\sigma_{b,s}$ is the square root of the variance of the elements of $|T|$. At the coefficient level, to have more accurate correlation noise estimation, each coefficient of frame $|T|$ was classified into inlier or outlier classes. As explained in [21], inlier coefficient values are close to the corresponding DCT band average value $\widehat{\mu}_b^2$. Outlier coefficients are those whose value is far from $\widehat{\mu}_b^2$. The α parameter for inlier coefficients was taken to be $\sigma_{b,s}$ which was the frame level α parameter. The α parameter for outlier coefficients was taken to be $\sqrt{2/[D_b(u,v)]^2}$, where

$$D_b(u,v) = |T|_b(u,v) - \widehat{\mu}_b. \quad (9)$$

With this approach for blocks/regions where the residual error is high, $[D_b(u,v)]^2$ is used instead of $\sigma_{b,s}$ to give less confidence to areas where MCFI is less successful. But for well-interpolated blocks/regions, coefficient level estimation is not better than frame level estimation. In our method, every block within a frame is classified in order to estimate the correlation noise. By a training stage and offline classification, we are able to estimate the dependence between source and side information based on the residual energy of a given block. By this method, we give different levels of confidence to different blocks based on how well interpolated they are.

In our method, we divide our sample of data into several classes of residual energy. The residual energy between forward and backward interpolation of every block within a frame for all Wyner–Ziv frames of several sequences is calculated to form a set R. We classify elements of this set into m different classes using $m-1$ thresholds T_i where $i \in \{1,\ldots,m-1\}$. Class i is chosen when $T_i < r < T_{i+1}$ where $r \in R$. To help ensure statistically reliable classification, the threshold values are set such that classes have roughly the same number of elements. All coefficients corresponding to frequency band j of all blocks labeled with class i are grouped together to form a set $v_{i,j}$. The α parameter of the set $v_{i,j}$ is calculated by $\sqrt{2}/\sigma_{i,j}$ where $\sigma_{i,j}$ is the square root of the variance of the $v_{i,j}$ elements.

Based on the previous procedure, there are m different classes of correlation estimation for each frequency band. We have, therefore, an m by 16 (since a 4×4 DCT is applied) lookup table of α parameters at the decoder. The component i, j of this table represents the α parameter of frequency band j of class i where $i \in \{1,\ldots,m\}$ and $j \in \{1,\ldots,16\}$. Development of this table is done offline.

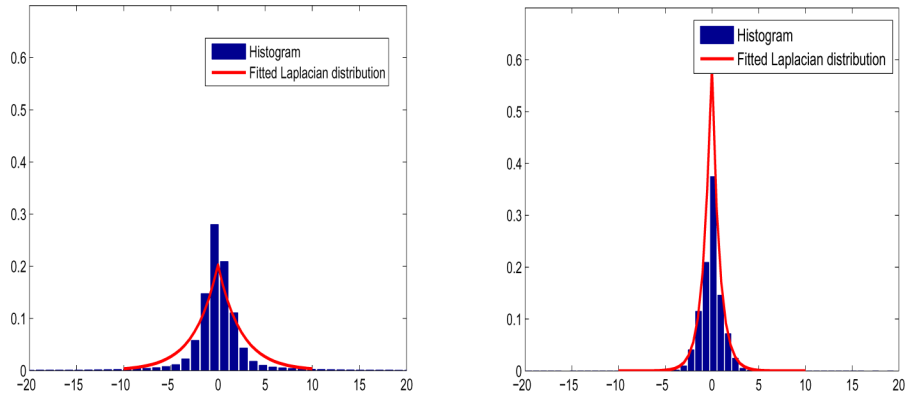
During decoding, for a given block of the Wyner–Ziv frame, the decoder evaluates the matching success of MCFI by calculating the residual energy between forward and backward interpolation and chooses one of the defined m classes by comparing to the threshold values. Once the block class is determined, the α parameter of each frequency band is found through the lookup table.[1] In our simulation, the number of classes is set to 8 since in that case, as discussed below, we can have enough elements in each class to have a reliable distribution model. Threshold values are calculated offline for each quantization parameter, separately. Table I shows the computed lookup table for quantization parameter equal to 0.4. Each row represents the α parameter of different DCT bands of a given class. The last row represents the calculated α parameter of different DCT bands based on the existing method where there is no classification. As we can see, going from class 1 down to class 8 in each column, the α parameter of each DCT band is a monotonically decreasing function of residual energy satisfying our expectation. Also, the α parameter of each class is an increasing function of frequency in each direction meaning that the α parameters of $f_{i,j}, f_{i,j+1}, \ldots, f_{i,j+3}$ and $f_{i,j}, f_{i+1,j}, \ldots, f_{i+3,j}$ are monotonically increasing. This suggests we have sufficient data within each class, since the α parameters follow the same trends as they do when there is no classification. As shown in Table I, the α parameters of the last row (corresponding to no classification) lie between class 6 and class 7. So, for high-motion sequences with most blocks classified to class 6 or higher, we expect less improvement than for low-motion sequences with most blocks classified to class 5 or lower.

Fig. 2(a) and (b) shows the distribution of frequency band $f(1,2)$ corresponding to the traditional method (no classification) and class 1, respectively. As we can see, the width of the approximated Laplacian distribution for frequency band $f(1,2)$ of class 1 is smaller than the width of the distribution for the traditional method meaning that the prediction will be more accurate on average when using the classification.

IV. Key Frame Encoding Based on Frequency Band Classification and Side Information Refinement

In conventional TDWZ coding, key frames are encoded and decoded by a conventional intraframe coder. So, the spatial correlation within a block is exploited by applying a DCT, but the temporal correlation between adjacent key frames is not exploited [24]. To extend the Wyner–Ziv coding idea to key frames to exploit similarities between them, previously decoded key frames can be used as the side information. If the side information is not a sufficiently accurate estimate of the source, Wyner–Ziv coding can do worse than intracoding. So, we need tools to evaluate the quality of the side information to select the proper coding method. Wyner–Ziv coding and intracoding blocks are already part of existing Wyner–Ziv codecs; therefore, applying a method switching between Wyner–Ziv and intracoding to exploit interframe correlation between consecutive

[1]We described this method in a preliminary version in [28]; however, in that work, the training set used to develop the lookup table was the same as the test set. In the current study, training and test data are disjoint.



(a) (b)

Fig. 2. Approximated Laplacian distribution for frequency band (1, 2): (a) without classification; (b) for class no. 1.

key frames does not add complexity to the encoder as long as the decision step is done at the decoder. Since the temporal correlation of low-frequency bands is usually high, Wyner–Ziv coding can often outperform intracoding. For high-frequency bands, measuring the distortion between source and side information of the low-frequency bands at the decoder can help to estimate the accuracy of the side information for high-frequency bands [25]. Side information that is simply a previous decoded key frame can be refined to a more accurate one for high-frequency bands by using decoded low-frequency bands. The Wyner–Ziv coding mode was described in detail in Section II. In this section, after describing the intracoding mode, we present our mode selection scheme with side information refinement.

A. Intracoding

For the intramode, the quantized DCT coefficients are arranged in a zigzag order to maximize the length of zero runs. The codeword represents the run length of zeros before a nonzero coefficient and the size of that coefficient. A Huffman code for the pair (run, size) is used because there is a strong correlation between the size of a coefficient and the expected run of zeros which precedes it. In our simulation, Huffman and run length coding tables are borrowed from the Joint Photographic Experts Group (JPEG) standard.

B. Coding Mode Selection and Side Information Refinement

Fig. 3 shows our proposed codec applying coding mode selection for key frames. To separate different frequency bands of the key frame to be encoded, first a DCT is applied. For frequency band k, the kth DCT coefficients from all blocks are grouped to form vector X_k. Low-frequency bands are encoded and decoded by Wyner–Ziv coding. The previously decoded key frame is used to generate the side information for low-frequency bands. To provide the corresponding side information for each frequency band, a DCT is applied on the previously reconstructed key frame, and the kth DCT coefficients from all blocks are grouped to form vector $\hat{X}_k$. Once the decoder receives and decodes all low bands, a block-matching algorithm is used for motion estimation of each block with reference to the previously decoded key frame. In block-matching algorithms, each macroblock in the new frame is compared with shifted regions of the same size from the previous frame, and the shift that results in the minimum error is selected as the best motion vector for that macroblock. Since here only reconstructed low bands of the new key frame are available at the decoder, the best match is found using the mean squared error (MSE) of low-frequency components. The MSE of low bands of two blocks A and B with $n \times n$ pixels is calculated as

$$\mathrm{MSE} = \frac{1}{K_{\mathrm{low}}} \sum_{(i,j)\epsilon \mathrm{Lowfreq.}} (U(i,j) - V(i,j))^2 \qquad (10)$$

where K_{low} is the total number of low bands and U and V are the DCT transform of A and B, respectively. The motion-compensated frame is the new side information for the remaining frequency bands. In our simulation, motion estimation for the refinement step is a full search in a ± 2 pixel search area. To select the proper coding method for high-frequency bands, we need to estimate the accuracy of the side information. At this point, decoded low bands constitute the only available information of the frame to be encoded. Since the side information is a noisy version of the source, measuring the distortion between decoded low bands of the current key frame and those of the motion compensated one at the decoder can help to give an estimation of the distortion for high bands. This distortion is calculated as

$$D = \frac{1}{K_{\mathrm{low}} \times L} \sum_{k\epsilon \mathrm{Lowfreq.}} \sum_{l=1}^{L} (X'_k(l) - \tilde{X}_k(l))^2 \qquad (11)$$

where X'_k denotes the reconstructed X_k at the decoder and $\tilde{X}_k$ denotes a vector formed by grouping the kth DCT coefficient from all blocks of the motion-compensated frame at the decoder. L is the number of elements in each frequency band which is the number of DCT blocks in a frame. If D is less than a threshold T_D, the side information is likely accurate enough that Wyner–Ziv coding can outperform intracoding for high-frequency bands. Otherwise, intracoding is applied for them. The decoder sends a single bit per frame through the feedback channel to indicate the selection. The added effect of sending a single bit per frame through the feedback channel

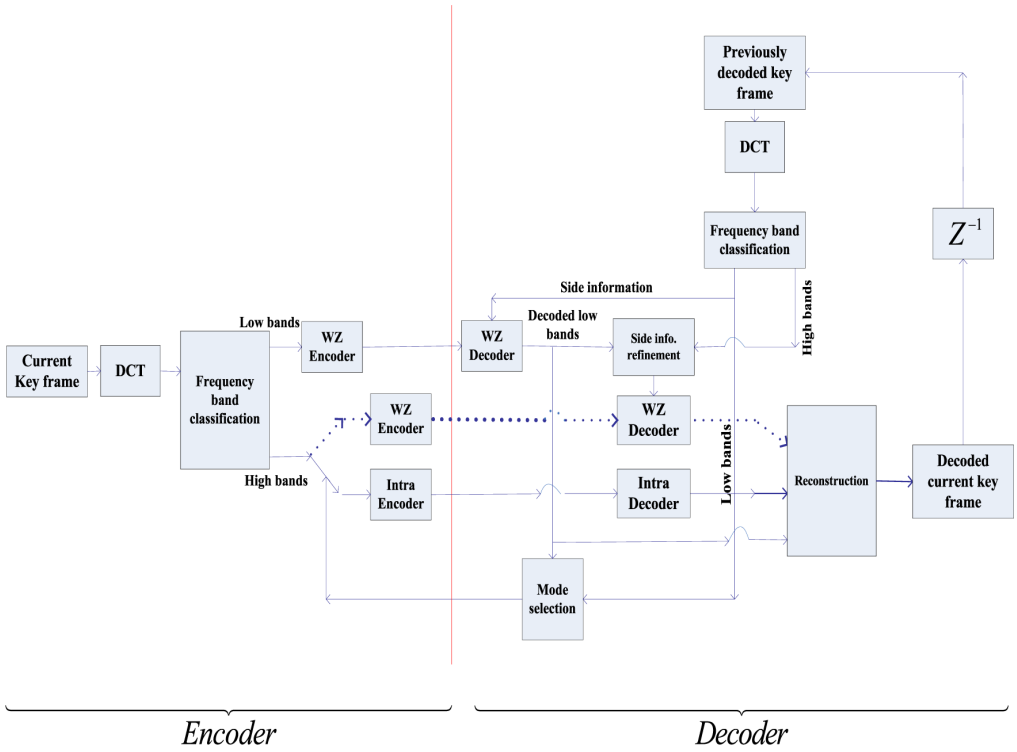


Fig. 3. Proposed video codec with frequency band coding mode selection for key frames.

on the latency of the system is negligible, since in traditional Wyner–Ziv coding, feedback bits might be sent for each bit plane to request more accumulated syndrome bits to meet the desired bit error rate. The conventional DVC decoder allows for all the bands to be decoded in parallel, whereas the proposed scheme essentially cuts in half the amount of parallelization that could be done. So, instead of having a time S in which to decode (in parallel) all the bands, the decoder would have to decode the low bands in $S/2$ and then the high bands in $S/2$. To allow random access and limit error propagation, we can switch OFF our proposed key frame encoding once in a while to use intracoding instead, as is done in conventional IPPP- or IBBP-type coders, where I, P and B denote intracoded, predicted and bidirectionally interpolated frames, respectively. The whole process of Wyner–Ziv coding of low bands, side information refinement, and finding the proper coding method for high bands is called adaptive coding for the rest of this paper.

As more bands are considered to be low, the greater accuracy is expected for the side refinement step in this method, although there would be some exceptions based on video content. But if we increase the number of low bands, fewer bands would be left to take advantage of the improved side information. As depicted in Fig. 4, the performance is improved when $f(1,1)$, $f(1,2)$, and $f(2,1)$ are considered as low bands compared with the case that only $f(1,1)$ is considered. However, the performance is degraded by considering the six lowest frequencies of the 4×4 DCT in zigzag order as low bands. Therefore, in our simulation, $f(1,1)$, $f(1,2)$, and $f(2,1)$ are considered as low-frequency bands, and the rest are considered high-frequency bands.

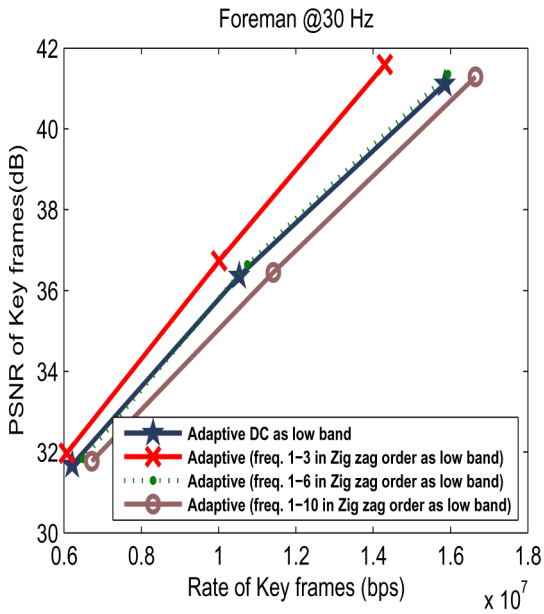


Fig. 4. PSNR of key frames versus rate for different numbers of frequency bands considered as low bands.

V. Hierarchical Coding

In traditional Wyner–Ziv coding, key frames occur every other frame and are intracoded to provide high-quality side information for the Wyner–Ziv frames in between. Many key frames encoded as intra leads to increasing rate and overall rate-distortion degradation. MCFI methods tend to be less successful when the distance between frames gets higher, so less frequent key frames results in less accurate side information for the corresponding Wyner–Ziv frame. Less accurate side

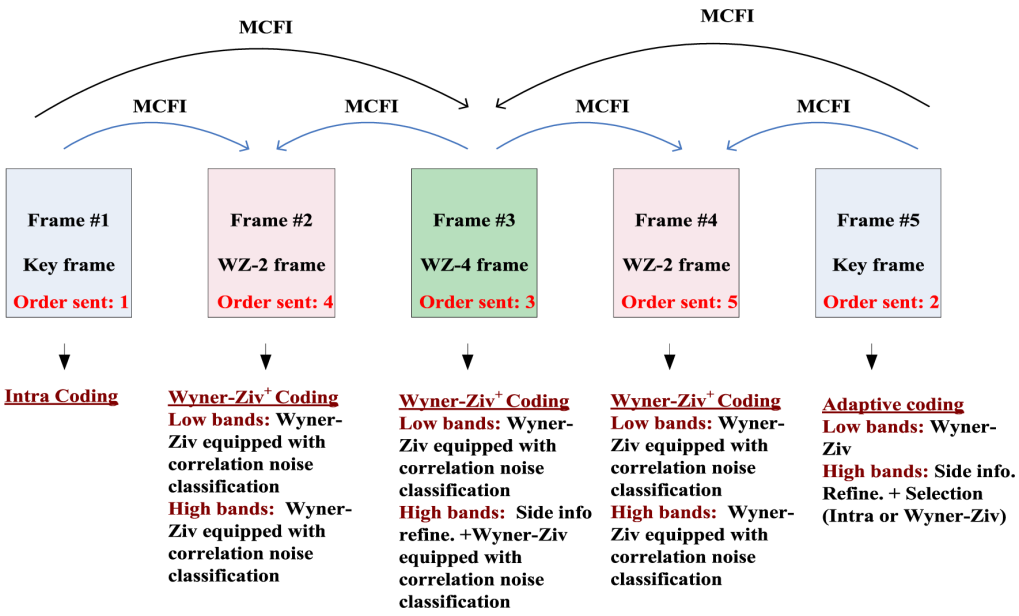


Fig. 5. Proposed hierarchical coding.

information means more accumulated syndrome bits need to be sent to satisfy the bit error expectation. In the previous section, we proposed a method to exploit similarities between key frames. In this section, we propose a more practical structure taking advantage of both adaptive coding and correlation noise classification techniques. Most Wyner–Ziv coders consider key frames every two frames. We started with this spacing and saw what improvement could be obtained by key frame prediction. The next step beyond this is key frame spacing of 4. As shown in Fig. 5, in this hierarchical arrangement, key frames occur every four frames and there are two types of Wyner–Ziv frames: Wyner–Ziv frames with four-frame distance, WZ-4, and Wyner–Ziv frames with two-frame distance, WZ-2, which will be explained in detail. Lookup tables of correlation noise classification for the two types are different and are obtained offline by using several sequences as training data. Compared to the traditional structure with one frame delay, latency in this structure is increased to a delay of three frames. In traditional Wyner–Ziv video coding where key frames occur every other frame, decoding of a Wyner–Ziv frame cannot be started unless the previous and next key frames were decoded.

A. Key Frames

As depicted in Fig. 5, key frames occur every four frames and they are used to generate side information corresponding to WZ-4 frames which will be explained later. The first key frame is intracoded since no other information is available. Applying the proposed adaptive coding method in Section IV will be very helpful to exploit temporal correlation of key frames in high frame rate videos or low-motion sequences. Otherwise, simply applying intracoding would be a better choice. In Figs. 7–9, both methods are applied for key frames, and results for different types of video content and frame rates are compared.

B. WZ-4 Frames

As shown in Fig. 5, these frames are at two-frame distance from key frames and four-frame distance from each other. The MCFI method proposed in [27] is applied on previous and next key frames to generate their corresponding side information. Since here the side information comes from both temporal directions and MCFI is applied, we can apply the proposed correlation noise classification method in Section II. For a given block of a WZ-4 frame, the decoder evaluates the matching success of MCFI by calculating the residual energy between forward and backward interpolation and chooses one of the defined m classes by comparing to the threshold values. Once the block class is determined, the α parameter of each frequency band is found through the lookup table. Once low bands are reconstructed at the decoder, they are used to refine the side information, and the rest of the frequency bands are Wyner–Ziv encoded with the refined side information.

C. WZ-2 Frames

As depicted in Fig. 5, these frames lie between key frames and WZ-4 frames. The MCFI method proposed in [27] is applied on their key frame and WZ-4 frame immediate neighbors which are at one-frame distance from them. For this type of frame also, side information comes from both sides, so the correlation noise classification technique is applicable. Since here the frame distance is only one frame from each side, the obtained side information is more accurate than for WZ-4. Empirically, for WZ-2 frames, having low bands is not very helpful to provide more accurate side information than the one attained by MCFI. So, the refinement step is not applied for them.

VI. Simulation Results

Figs. 7–9(a)–(d) show the rate-distortion performance for the test sequences *Claire*, *Mother-daughter*, *Foreman*, and *Carphone* QCIF (176×144) sequences at 30, 15, and 10 frames/s. Fig. 8(e) shows the rate-distortion performance for the *Soccer* QCIF sequence at 15 frames/s.

In all offline processes such as setting threshold values and correlation noise classification lookup tables, training video

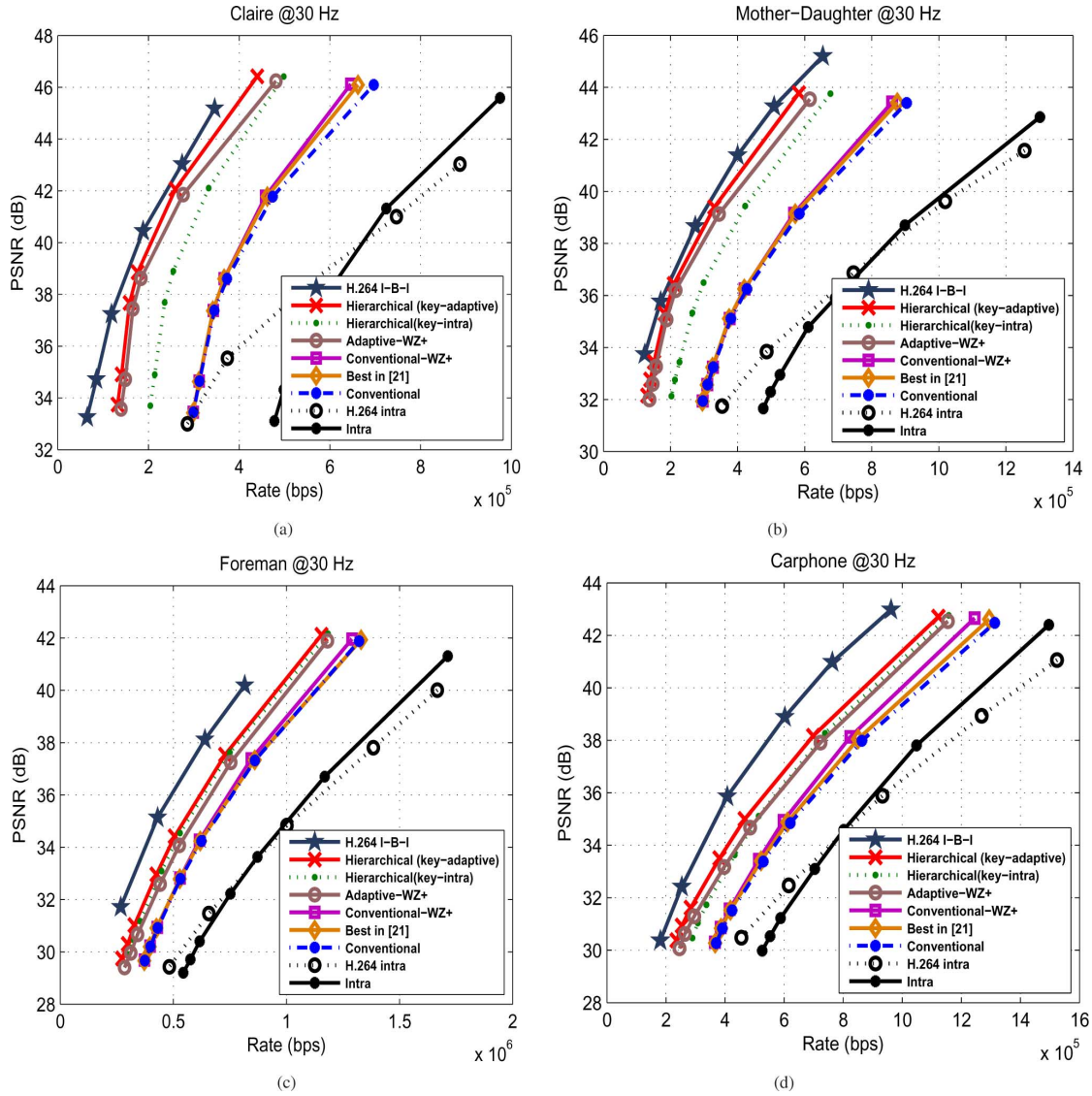


Fig. 6. PSNR versus rate for conventional Wyner–Ziv method applying correlation noise classification.

sequences are different from test video sequences. Our training sequences are *Container*, *Salesman*, *Coastguard*, and *Akiyo*.

In our simulation, $QP\epsilon\{0.4, 0.85, 1.5, 2, 3, 3.5, 4\}$. For adaptive coding, which is described in Section IV, for each one of these quantization parameters, a threshold value is set. We tried different values between 50 and 1800 with step sizes 20 to 100 for several video sequences at different quantization parameters. The value of the step size depends on the quantization parameter, with larger step sizes for larger quantization parameters. Threshold values $[T_1, T_2, \ldots, T_7] = [200, 290, 740, 860, 1200, 1500, 1800]$ corresponding to quantization parameters $QP = [0.4, 0.85, 1.5, 2, 3, 3.5, 4]$, were chosen as they work well for the training sequences with different characteristics. Threshold values are obtained for training sequences at 30 frames/s and used for test sequences at frame rates of 30, 15, and 10 frames/s. For correlation noise classification, for each type of Wyner–Ziv frame and each quantization step, a different lookup table is calculated.

Table II shows the average number of times that key frame high bands are Wyner–Ziv coded in the Adaptive coding method. In Figs. 7–9, the results of applying different methods are compared. With "*Intra*," all frames are intraencoded and

TABLE II
AVERAGE FRACTION OF TIME KEY FRAME HIGH BANDS ARE WYNER-ZIV CODED

(a)

	Mother-Daughter		
	@10Hz	@15Hz	@30Hz
QP=0.40	0.84	0.93	0.97
QP=0.85	0.90	0.93	0.97
QP=1.50	0.94	0.97	0.99
QP=2.00	0.98	0.99	0.99
QP=3.00	0.98	0.99	0.99
QP=3.50	0.98	0.99	0.99
QP=4.00	0.98	0.99	0.99

(b)

	Claire		
	@10Hz	@15Hz	@30Hz
QP=0.40	0.92	0.99	1.00
QP=0.85	0.99	0.99	1.00
QP=1.50	0.99	0.99	1.00
QP=2.00	0.99	0.99	1.00
QP=3.00	0.99	0.99	1.00
QP=3.50	0.99	0.99	1.00
QP=4.00	0.99	0.99	1.00

(c)

	Carphone		
	@10Hz	@15Hz	@30Hz
QP=0.40	0.31	0.42	0.64
QP=0.85	0.39	0.53	0.74
QP=1.50	0.55	0.70	0.86
QP=2.00	0.70	0.80	0.92
QP=3.00	0.77	0.90	0.97
QP=3.50	0.81	0.96	0.98
QP=4.00	0.89	0.98	0.98

(d)

	Foreman		
	@10Hz	@15Hz	@30Hz
QP=0.40	0.22	0.29	0.60
QP=0.85	0.22	0.39	0.64
QP=1.50	0.28	0.48	0.71
QP=2.00	0.40	0.57	0.75
QP=3.00	0.58	0.68	0.81
QP=3.50	0.62	0.71	0.83
QP=4.00	0.64	0.76	0.87

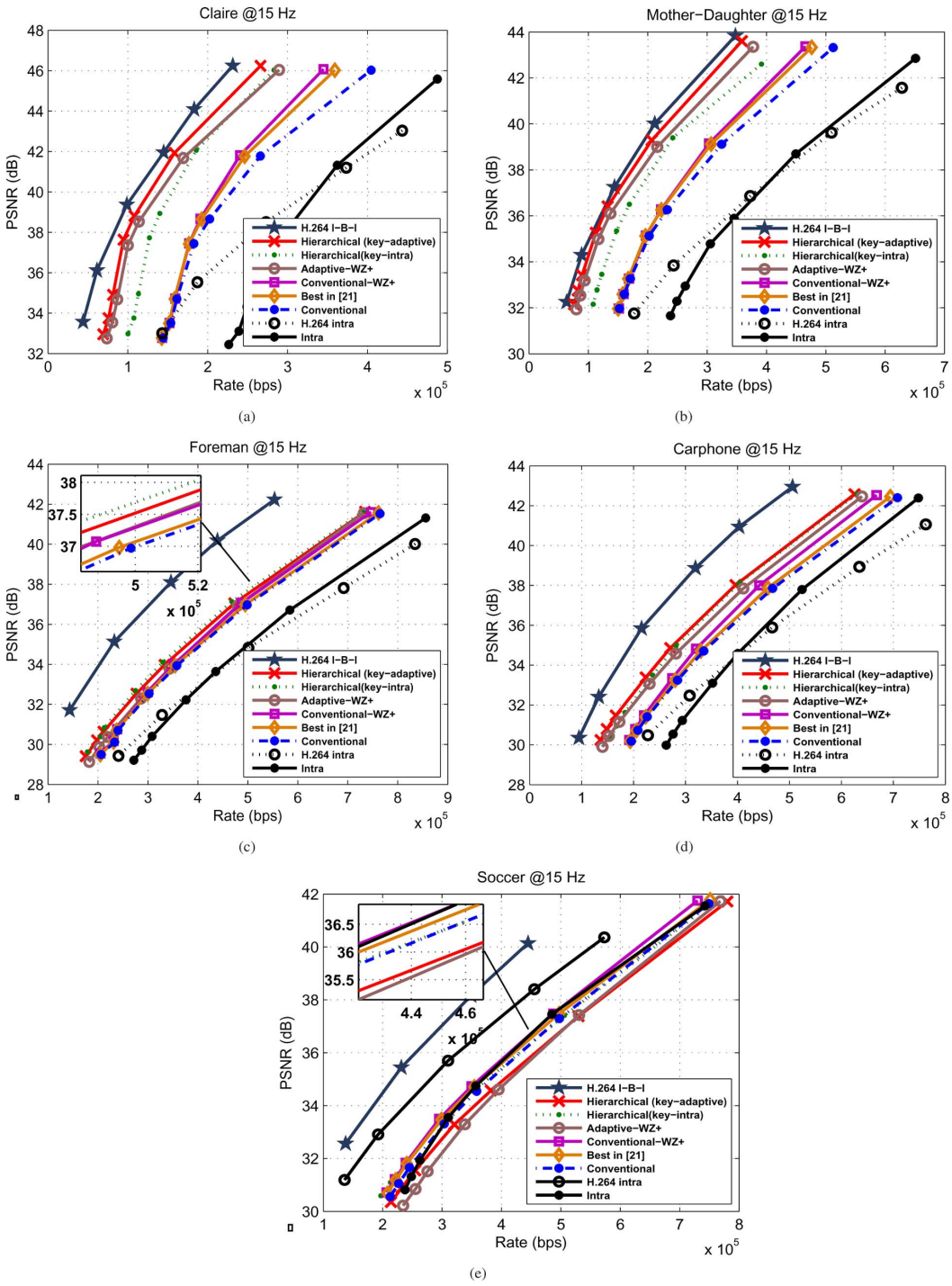


Fig. 7. PSNR versus rate for different coding methods for 30 frames/s sequences.

decoded by using the method explained in Section IV-A. The complexity of this method is as low as JPEG. In this paper, whenever intracoding was needed, this method was used. "*Conventional*" is based on the method in [10], but we modified the algorithm in two ways. First, the assumption of availability of original key frames at the decoder is removed since it is not valid from a practical point of view. Second, the quantization part is replaced with the quantization procedure explained in Section II. Although not depicted in the figures, our simulation results show that this change in quantization method improves the performance of [10]. Our quantization method is applied for all proposed methods. We use the same quantization method for all the approaches in order to highlight the performance improvement due to correlation noise classification and key frame encoding. In the "*Conventional*" method, key frames (odd frames) are encoded and decoded as intra using the method explained in Section IV-A, and even frames are encoded as Wyner–Ziv frames. When Wyner–Ziv coding equipped with correlation noise classification is applied for Wyner–Ziv frames of the conventional method, the

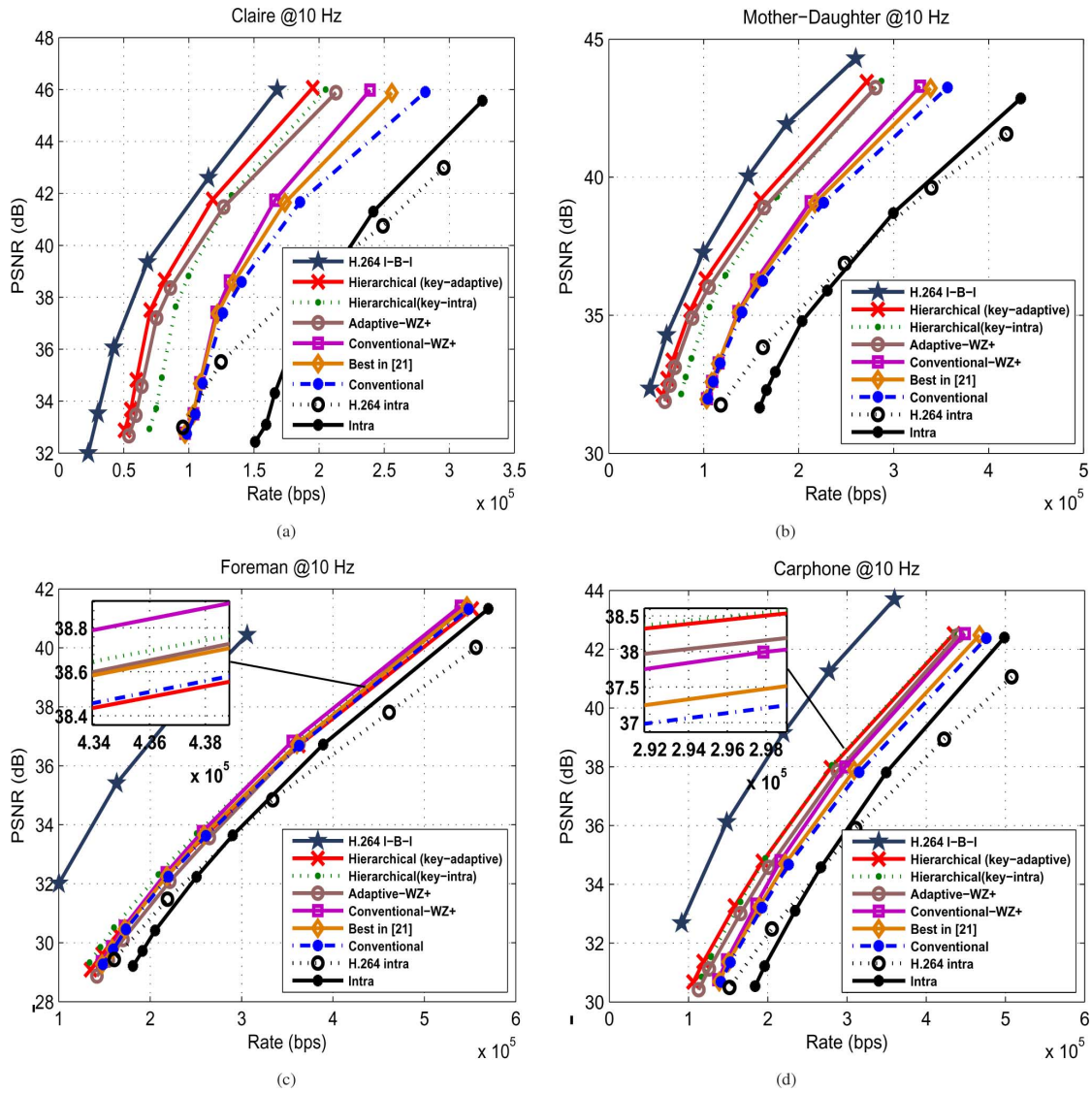


Fig. 8. PSNR versus rate for different coding methods for 15 frames/s sequences.

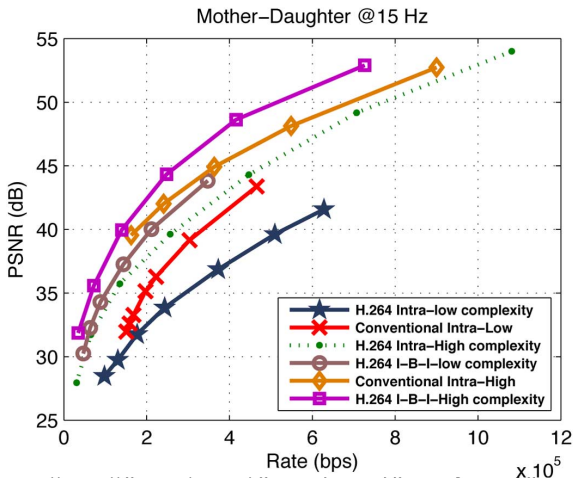


Fig. 9. PSNR versus rate for different coding methods for 10 frames/s sequences.

method is called "$Conventional - WZ+$." The result of this method is compared to the best-proposed method (coefficient level of transform domain) in [21]. When the adaptive method is applied for key frames (odd frames), and the Wyner–Ziv method equipped with correlation noise classification is applied for even frames, the method is called "*Adaptive-WZ+*." "*Hierarchical-key-intra*" and "*Hierarchical-key-adaptive*" are the names of the methods explained in Section V where intra and adaptive are applied for key frames, respectively. Results are also compared to "H.264 intra" and "H.264 I-B-I." In this paper, all methods are using intra method as low complexity as JPEG. It is further explained in Section VI-A.

Simulation results show that applying the correlation noise classification proposed in Section III results in up to 2 dB improvement over "Conventional" and 1 dB improvement over the best proposed method (coefficient level) in [21] (*Claire* 10 frames/s at 240 kb/s). The proposed adaptive method combined with correlation noise classification results in up to 5 dB improvement over "Conventional-WZ+" (*Claire* 30 frames/s at 400 kb/s). The gain is more for low-motion and higher frame rate sequences where the intercorrelation is high. For high-motion sequences at lower frame rate, we do not expect improve-

ment since the intercorrelation is very low. As shown in Figs. 8 and 9(c) for *Foreman*, as a high-motion sequence at 15 and 10 frames/s, the performance of "Adaptive-WZ+" is very close to that of "Conventional-WZ+" but with a slight degradation. For very high-motion sequences like *Soccer* at 15 frames/s where the MCFI method gives a poor side information, the whole idea of Wyner–Ziv coding fails, meaning that intracoding outperforms Wyner–Ziv coding. For such cases, all of these methods for exploiting correlation between consecutive key frames are useless. "*Hierarchical-key-adaptive*" is capable of beating all methods for most cases and results in up to 1 dB additional improvement. The exceptions are *Foreman* at 15 frames/s, 10 frames/s, and *Soccer* at 15 frames/s. For these high-motion and low-frame-rate cases, since in the hierarchical structure, key frames are four frames apart, the temporal correlation between key frames is very low. So, applying intracoding for key frames would be a better alternative. As shown in Figs. 8 and 9, "*Hierarchical-key-intra*" can beat "*Hierarchical-key-adaptive*" for these cases. Although even "*Hierarchical-key-intra*" results in degradation for *Soccer* as the whole idea of Wyner–Ziv coding fails for this sequence.

A. Complexity

Since, in this paper, all methods are using an intra method as low complexity as JPEG to have a fair comparison, intra predictions, Hadamard transform, and context adaptive binary arithmetic coding (CABAC) are turned OFF for I frames of "H.264 intra" and "H.264 I-B-I." Certainly, adding these features can improve the performance of all methods (as partially shown in Fig. 6), at the cost of additional complexity. For example, CABAC entropy coding provides about 15% bit reduction at the expense of a computation and memory increase (up to 30%) compared to universal variable length coding (UVLC) [29]. The use of Hadamard coding results in a complexity increase of roughly 20%, while not significantly impacting the quality versus bit rate [30]. The intra prediction in H.264 employs the rate-distortion optimization technique which remarkably increases the computational complexity. According to Saponara *et al.* [31], motion estimation and entropy coding occupy about 53% and 18% of the encoder computational consumption, respectively. However, it should be noted that the diversity of the operation configuration for motion estimation (subpixel motion estimation and multiple reference frame, etc.) also has a great effect on encoding complexity. For example, motion estimation with quarter-pixel precision typically consumes 60% (with one reference frame) and 80% (with five reference frames) of the total encoding time [32], and the percentage becomes even larger when the search range increases.

In the context of Wyner–Ziv video coding, the main goal is providing a low-cost and low-complexity encoder. Although most of the H.264 encoder complexity is due to motion estimation, the computational requirements of CABAC and intraprediction modes may be still too high for some applications [33]. There is a tradeoff between compression gain and complexity, and based on the application, either one can be sacrificed.

VII. Conclusion

We proposed three new techniques to improve the overall rate-distortion performance of Wyner–Ziv video coding: 1) a new method of correlation noise estimation based on block-matching classification at the decoder; 2) an advanced mode selection scheme for frequency bands of key frames followed by side information refinement; and 3) a hierarchical Wyner–Ziv coding approach including the other two schemes. Simulation results showed that the proposed correlation noise classification results in up to 1 dB improvement over the best method in [21]. With the possible cost of additional buffering at the encoder, the proposed key frame encoding with side refinement combined with correlation noise classification results in up to 5 dB improvement over the Conventional method equipped with correlation noise classification. Experimental results showed that one can achieve up to 1 dB additional improvement by applying the hierarchical method at the cost of extra latency. All the proposed methods keep the encoder low complexity.

References

[1] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 4, pp. 471–480, Jul. 1973.

[2] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 1, pp. 1–10, Jan. 1976.

[3] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," in *Proc. Allerton Conf. Commun., Control, and Computing*, Oct. 2002, pp. 586–595.

[4] R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: A video coding paradigm with motion estimation at the decoder," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2436–2447, Oct. 2007.

[5] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Proc. Asilomar Conf. Signals Syst.*, Nov. 2002, vol. 1, pp. 240–244.

[6] A. Aaron, S. Rane, and B. Girod, "Transform-domain Wyner-Ziv codec for video," in *Proc. Visual Commun. Image Process.*, Jan. 2004, vol. 5308, pp. 520–528.

[7] A. Aaron, S. Rane, and B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2004, vol. 5, pp. 3097–3100.

[8] A. Aaron and B. Girod, "Wyner-Ziv video coding with low encoder complexity," in *Proc. Picture Coding Symp.*, Dec. 2004.

[9] A. Aaron, D. Varodayan, and B. Girod, "Wyner-Ziv residual coding of video," presented at the Picture Coding Symp., Beijing, China, Apr. 2006.

[10] C. Brites, J. Ascenso, and F. Pereira, "Improving transform domain Wyner-Ziv video coding performance," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2006, vol. 2, pp. 525–528.

[11] S. Argyropoulos, N. Thomosy, N. Boulgourisz, and M. Strintzis, "Adaptive frame interpolation for Wyner-Ziv video coding," in *Proc. IEEE 9th Workshop Multimedia Signal Process.*, Oct. 2007, pp. 159–162.

[12] S. Ye, M. Ouaret, F. Dufaux, and T. Ebrahimi, "Improved side information generation with iterative decoding and frame interpolation for distributed video coding," in *Proc. Int. Conf. Image Process.*, Oct. 2008, pp. 2228–2231.

[13] W. A. R. J. Weerakkody, W. A. C. Fernando, J. L. Martinez, P. Cuenca, and F. Quiles, "An iterative refinement technique for side information generation in DVC," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2007, pp. 164–167.

[14] R. Martins, C. Brites, J. Ascenso, and F. Pereira, "Refining side information for improved transform domain Wyner-Ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, pp. 1327–1341, Sep. 2009.

[15] J. Zhang, H. Li, Q. Liu, and C. W. Chen, "A transform domain classification based Wyner-Ziv video codec," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2007, pp. 144–147.

[16] L. Liu, D. He, A. Jagmohan, L. Lu, and E. Delp, "A low complexity iterative mode selection algorithm for Wyner-Ziv video compression," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 1136–1139.
[17] R. Westerlaken, R. Gunnewiek, and R. Lagendijk, "The role of the virtual channel in distributed source coding of video," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2005, vol. 1, pp. 581–584.
[18] R. Westerlaken, S. Borchert, R. Gunnewiek, and R. Lagendijk, "Dependency channel modeling for a LDPC-based Wyner-Ziv video compression scheme," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 277–280.
[19] L. Qing, X. He, and R. Lv, "Distributed video coding with dynamic virtual channel model estimation," in *Proc. Int. Symp. Data, Privacy E-Commerce*, 2007, pp. 170–173.
[20] C. Brites, J. Ascenso, and F. Pereira, "Studying temporal correlation noise modeling for pixel based Wyner-Ziv video coding," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 273–276.
[21] C. Brites and F. Pereira, "Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 9, pp. 1177–1190, Sep. 2008.
[22] X. Huang and S. Forchhammer, "Improved virtual channel noise model for transform domain Wyner-Ziv video coding," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 921–924.
[23] A. B. B. Adikari, W. A. C. Fernando, H. K. Arachchi, and W. A. R. J. Weerakkody, "Low complex key frame encoding with high quality Wyner-Ziv coding," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Aug. 2006, pp. 605–609.
[24] G. Esmaili and P. Cosman, "Low complexity spatio-temporal key frame encoding for Wyner-Ziv video coding," in *Proc. Data Compression Conf.*, Mar. 2009, pp. 382–390.
[25] G. Esmaili and P. Cosman, "Frequency band coding mode selection for key frames of Wyner-Ziv video coding," in *Proc. 11th IEEE Int. Symp. Multimedia*, Dec. 2009, pp. 148–152.
[26] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive distributed source coding using low-density parity-check codes," in *Proc. Asilomar Conf. Signals Syst.*, Oct. 2005, pp. 1203–1207.
[27] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *Proc. 5th Eur. Assoc. Signal Process.*, Jul. 2005.
[28] G. Esmaili and P. Cosman, "Correlation noise classification based on matching success for transform domain Wyner-Ziv video coding," in *IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2009, pp. 801–804.
[29] S. Saponara, K. Denolf, G. Lafruit, and J. Bormans, "Performance and complexity co-evaluation of the advanced video coding standard for cost-effective multimedia communications," *EURASIP J. Appl. Signal Process.*, vol. 2, pp. 220–235, 2004.
[30] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockammer, and T. Wedi, "Video Coding With H.264/AVC: Tools, Performance, and Complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7–28, 2004.
[31] S. Saponara, C. Blanch, K. Denolf, and J. Bormans, "The JVT Advanced Video Coding Standard: Complexity and Performance Analysis on a Tool-by-Tool Basis," in *Packet Video 2003*, Nantes, France, Apr. 2003.
[32] Z. Chen, P. Zhu, and Y. He, "Fast Integer Pel and Fractional Pel Motion Estimation for JVT," in *Proc. 6th meeting, Awaji, JP*, 2002, pp. 5–13.
[33] D. Marpe, H. Schwarz, and T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," in *Proc. SPIE Conf. Wavelet Appl. Ind. Process.*, Oct. 2003.

Ghazaleh Rais Esmaili (S'09) received the B.Sc. degree from the University of Tehran, Tehran, Iran, in 1997 and the M.Sc. degree from Tarbiat Modares University, Tehran, Iran, in 2002, all in electrical engineering. She is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, University of California, San Diego.

Her research interests include video compression and distributed video coding.

Pamela C. Cosman (S'88–M'93–SM'00–F'08) received the B.S. (Hons.) degree from the California Institute of Technology, Pasadena, in 1987, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 1989 and 1993, respectively, all in electrical engineering.

She was an NSF Postdoctoral Fellow at Stanford University and a Visiting Professor at the University of Minnesota during 1993–1995. In 1995, she joined the Faculty of the Department of Electrical and Computer Engineering, University of California, San Diego, where she is currently a Professor. She was the Director of the Center for Wireless Communications from 2006 to 2008. Her research interests include the areas of image and video compression and processing, and wireless communications.

Dr. Cosman is the recipient of the ECE Departmental Graduate Teaching Award (1996), a Career Award from the National Science Foundation (1996–1999), a Powell Faculty Fellowship (1997–1998), and a Globecom 2008 Best Paper Award. She was a Guest Editor of the June 2000 special issue of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS on "Error-resilient image and video coding," and was the Technical Program Chair of the 1998 Information Theory Workshop in San Diego. She was an Associate Editor of the IEEE COMMUNICATIONS LETTERS (1998–2001), and an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS (2001–2005). She was the Editor-in-Chief (2006-2009) as well as Senior Editor (2003-2005, 2010-present) of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. She is a member of Tau Beta Pi and Sigma Xi.