

RESEARCH

Open Access

# Adaptive distributed video coding with motion vectors through a back channel

Kyung-Yeon Min and Dong-Gyu Sim\*

## Abstract

In this paper, a new adaptive distributed video coding (DVC) method with received motion vectors (MVs) through a back channel is proposed. The MVs estimated for side information (SI) generation in the proposed DVC decoder are transmitted to the proposed DVC encoder to reconstruct a predicted SI (PSI) which is the same as the SI at the decoder with minimum computational load. Therefore, the proposed DVC encoder can determine the positions and magnitudes of errors using the PSI. With available error information, the proposed encoder adaptively determines appropriate slice partitions by maintaining fixed per-slice error rates for prevention of channel decoding failure. The encoder also sends a coded SI accuracy map to the decoder and sets the conditional probability of each variable node of the low-density parity-check accumulator (LDPCA) to the correct crossover probability according to the coded map as received on the decoder side. The performance of the LDPCA can also be significantly improved with a minimum number of parity bits and low computational complexity because accurate belief propagation can be carried out using the correct crossover probabilities in the LDPCA decoder. Experimental results show that the rate-distortion performance of the proposed algorithm is twice that of several conventional DVC methods and that the proposed decoder is approximately 24.5% faster than these methods.

## 1. Introduction

As many portable video devices such as wireless low-power surveillance cameras, mobile phones, and multimedia sensor networks have been developed, many people have come to enjoy taking videos and transmitting them easily to friends or Web sites using these portable devices. To improve video quality and delivery speed for portable devices, the demand for low-cost, low-power encoders has been continuously increasing [1,2]. However, conventional video codecs such as MPEG-x and H.26x cannot satisfy these requirements because they were designed according to a philosophy of high-complexity encoding and low-complexity decoding. Several distributed video coding methods have been proposed to satisfy the new demands for low-complexity encoding and high-complexity decoding. The distributed video coding (DVC) technology is based on the migration of computational complexity from the encoder to the decoder and can achieve coding gain with decoder-side predictions [3-5].

DVC was developed to reduce the complexity of the video encoders based on the Slepian-Wolf information theory [1-8]. Two different signals can be coded without prediction between them, and the signals can then be decoded with prediction at the decoder. Slepian and Wolf proved that compression performance with decoder-side prediction can achieve a certain level close to those attained by efficient conventional compression that codes signals with prediction at the encoder side. Therefore, the DVC encoder does not need to perform motion estimation, which enables drastic reduction in the computational complexity of the DVC encoder. Wyner-Ziv (WZ) proposed a lossy DVC approach using side information (SI). The approach codes original input frames using the key frame mode and the WZ mode [4-6]. For the WZ mode, channel coding techniques such as the low-density parity-check accumulator (LDPCA) coder and the turbo coder are often used for distributed video coding [7-12].

As mentioned before, the computational complexity of DVC encoders is significantly lower than that of conventional video encoders. However, the corresponding DVC decoders have high computational complexity because SI generation requires at least as much computation as

\* Correspondence: dgsim@kw.ac.kr  
Department of Computer Engineering, Kwangwoon University, 447-1,  
Wolgye-dong, Nowon-gu, Seoul 139-701, Republic of Korea

conventional motion estimation, and channel decoding also requires many iterative computations. Therefore, several methods have been proposed to reduce the computational complexity of the DVC decoder. One of these estimates error rates to reduce the number of feedback iterations and the computational complexity of channel decoding [4-6]. The performance and speed of the channel decoder can be improved with accurate crossover probabilities which are similar to the actual SI error rates. The crossover probability is defined by the probability that a bit value (0 or 1) will be changed into another value (1 or 0) at each variable node of the LDPCA channel decoder [3]. Because the LDPCA is based on the sparse bipartite graph, errors are corrected by a belief propagation algorithm. In addition, the crossover probabilities influence on convergence of the belief propagation [3]. Crossover probability is used to determine the conditional entropy of a variable node of the LDPCA, which directly impacts on the error-correction capability of the LDPCA. Therefore, the computational complexity of the channel decoder can be reduced by a correct setting of the crossover probabilities in the variable nodes of the LDPCA based on correct SI error rates. However, neither the encoder nor the decoder can estimate error rates of SIs because both the original and SI frames must be available for each side. Therefore, several SI error-probability prediction methods have been proposed [7-9], which can be divided into two groups depending on whether the error-probability prediction is performed on the encoder or the decoder side. Error-probability estimation methods on the decoder side assume that the error rate of a bitplane is correlated with that of a previous frame, a neighboring bitplane, or both. These methods can work well for slow-moving sequences and have no impact on the computational complexity of the encoder side. However, iterative feedbacks are still required because the encoder does not have any information about the SI. Predicted error probabilities are also likely to be different from actual SI error rates for fast-moving sequences. Fast-moving sequences have low correlations in both temporal and bitplane domains. Error-probability estimation methods on the encoder side generate a coarse SI by simple linear interpolation, rough block-based motion estimation, motion compensation, and so on. Although the feedback channel can be removed and adaptive DVC modules are employed with these approaches, the computational complexity of the encoder increases. Furthermore, the estimated error probability is not likely to be accurate, and therefore, coding performance can decrease. The error probability can be estimated with motion vectors received from the decoder side. Motion vector feedback algorithm has been presented to improve the coding efficacy of DVC paradigm [13-15].

The proposed DVC encoder can generate a predicted SI (PSI) that is the same as the SI of the decoder side with low computational complexity because only motion compensation is performed with received motion vectors (MVs) and reference key frames on the encoder side [13,14]. Because the proposed PSI is identical to the SI for both slow-moving and fast-moving sequences, the correct SI error can be estimated using the PSI, and the WZ frames can be adaptively encoded depending on the target per-slice error rate. After SI error estimation, the proposed encoder performs block-based coding and determines the correct slice partitions to prevent errors from concentrating in certain areas because a channel decoder cannot work for data with a too high error rate. In addition, the proposed DVC encoder sends a coded map indicating whether each block has been coded or not. The coded map leads to faster and more accurate convergence in LDPCA decoding because the proposed DVC decoder marks the variable nodes associated with non-coded blocks as not altered. Therefore, decoding failure rates and delays can be reduced simultaneously.

The rest of this paper is organized as follows: Section 2 introduces several conventional DVC algorithms. Section 3 presents details of the proposed method. In Section 4, experimental results are presented and discussed. Finally, Section 5 gives the concluding remarks.

## **2. Fundamentals of distributed video coding**

As mentioned before, multimedia DVC is a new video coding paradigm which makes it possible to shift complexity from an encoder to a decoder. Two signals can be independently coded in the encoder and then reconstructed with a prediction of the cross-correlation between them in the decoder. Slepian and Wolf proved that coding performance can be improved with decoder-side prediction. Wyner and Ziv developed several lossy DVC systems based on the Slepian-Wolf information theory [1,2]. They also proposed a novel SI generation algorithm to improve DVC coding performance [16-18].

### **2.1. Wyner-Ziv distributed video coding**

In the DVC encoder, key frames are encoded by a conventional intra-frame encoding method such as H.264/AVC intra-frame mode. For the WZ mode, two approaches have been proposed: pixel-based DVC and transform-based DVC. For pixel-based DVC, quantization is carried out as a pre-processing. For transform-based DVC, transform and quantization are performed before a channel coder. After pre-processing, the channel coder produces a message (the original WZ frame) and its parity bits for each bitplane. Among the outputs of a channel coder, only the parity bits are transmitted to the corresponding DVC decoder when a request is received from the decoder.

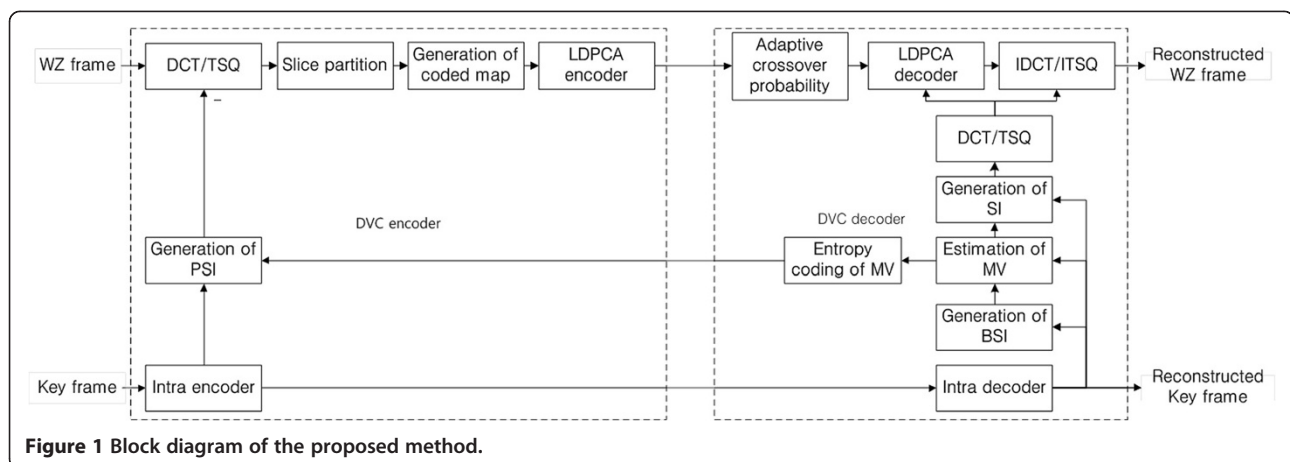
In the DVC decoder, key frames are reconstructed using a conventional intra-frame decoder. Then an SI is generated using the reconstructed key frames. Many SI generation algorithms have been proposed because the rate-distortion (RD) performance of DVC depends highly on SI accuracy [6]. The generated SI is still different from the original WZ frame, but the difference can be corrected by the channel decoder using the received parity information. Because the feedback iterations continue until channel decoding is successful, delays in DVC decoding could not be negligible for real-time applications.

On the encoder side, the original frames are available, but the SI is not; on the decoder side, the SI is available, but the original frame is not. As a result, error rates cannot be computed on the decoder and encoder sides. Therefore, conventional DVC systems cannot help to send correct error probabilities to the variable nodes in LDPCA decoding, which leads to a decrease in coding performance and increases the number of feedback iterations in LDPCA decoding. For performance improvement of channel decoding, several conventional algorithms have been proposed to estimate the SI error rate.

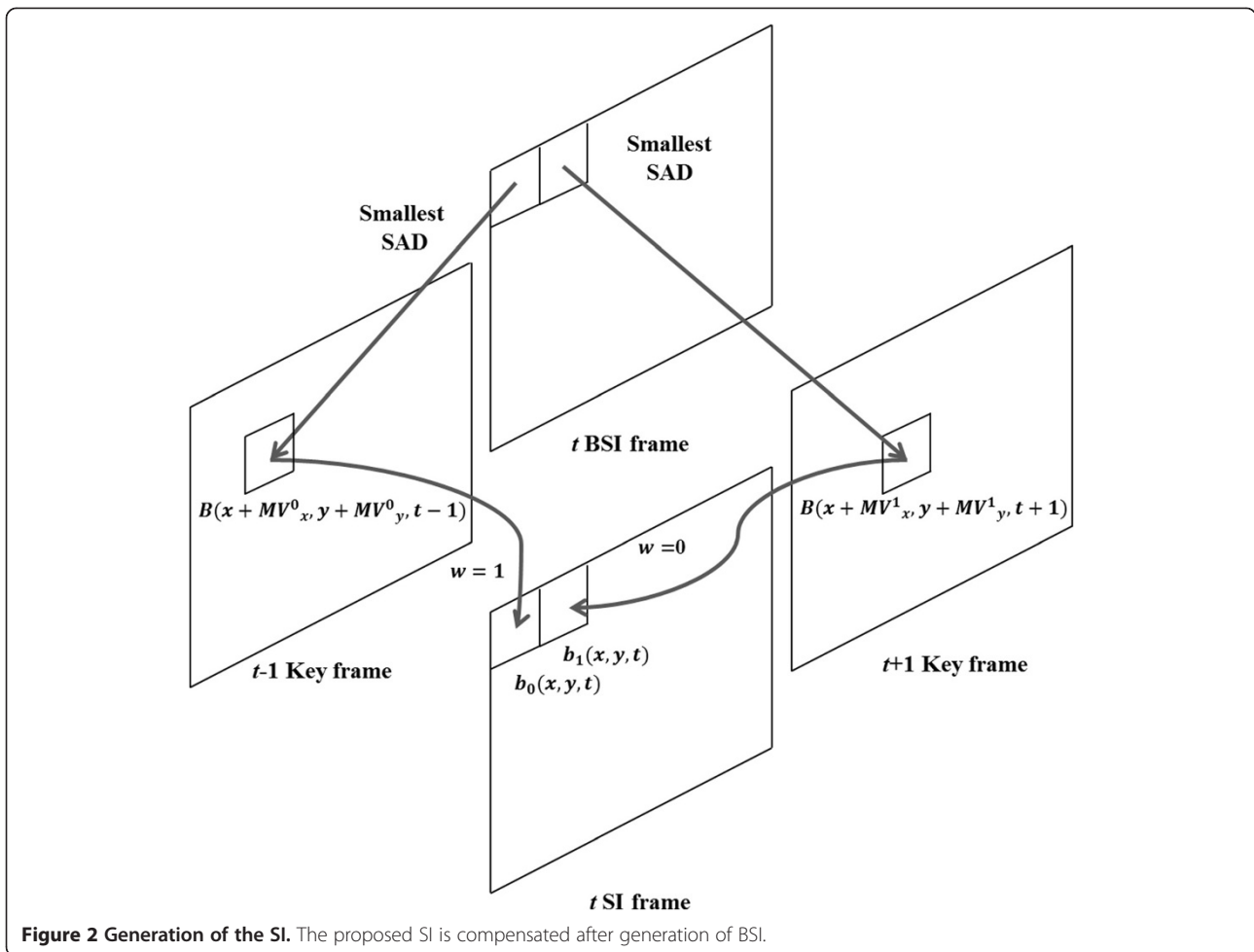
## 2.2. Estimation of error rates

Either hard decisions or soft decisions can be used in a message-passing module for belief propagation in LDPCA [3]. Soft decisions for message passing are known to lead to better coding performance than hard decisions because the soft decision algorithm can use statistical characteristics to set the channel modeling parameters. Note that the log-likelihood ratio can be used to represent conditional entropy in soft decision-making, while only two values  $(-1,1)$  or  $(0,1)$  can be used for hard decision-making. Although LDPCA decoding based on soft decisions requires more computational complexity, the soft decision methods are frequently adopted by many LDPCA systems due to its decoding accuracy.

To determine accurate crossover probabilities for LDPCA decoding, several methods have been proposed to predict SI error probabilities [4-6]. In conventional DVC algorithms, the SI is available on the decoder side, but the original frame is not. Several methods have been proposed to predict error probabilities using neighboring information such as previous frames or adjacent bitplanes. These methods assume that changed bit rates in previous bitplanes have an influence on error rates in the target bitplane. The error rates of the target bitplane are also assumed to be similar to those of the same bitplane in the previous frame. Therefore, several conventional algorithms set crossover probabilities equal to the error probabilities of the previous corresponding bitplane or frame using an error model of the difference between the original frame and the SI, such as a Laplacian distribution for DVC [4]. On the other hand, the error probabilities can be estimated on the encoder side. Several existing algorithms have tried to estimate a PSI similar to the SI with minimum computational complexity. For the computational constraint, rough SI generation algorithms are used, for example, temporal linear interpolation or simple block-matching algorithms [7-9,19,20]. If accurate error rates between the original and SI frames are known on the encoder side, the correct number of parity bits to be sent can be determined. As a result, the iterative feedback can be removed and delays can be drastically reduced for practical DVC. In addition, it is easy to decide whether a block can be coded as an intra-coded or a skipped block, given the number of errors in the target block. However, it is not easy to obtain an accurate SI on the encoder side because this entails a computational load as heavy as that of a DVC decoder. If the estimated error probability is far from the correct one, a channel decoder is likely to converge to an incorrect solution.



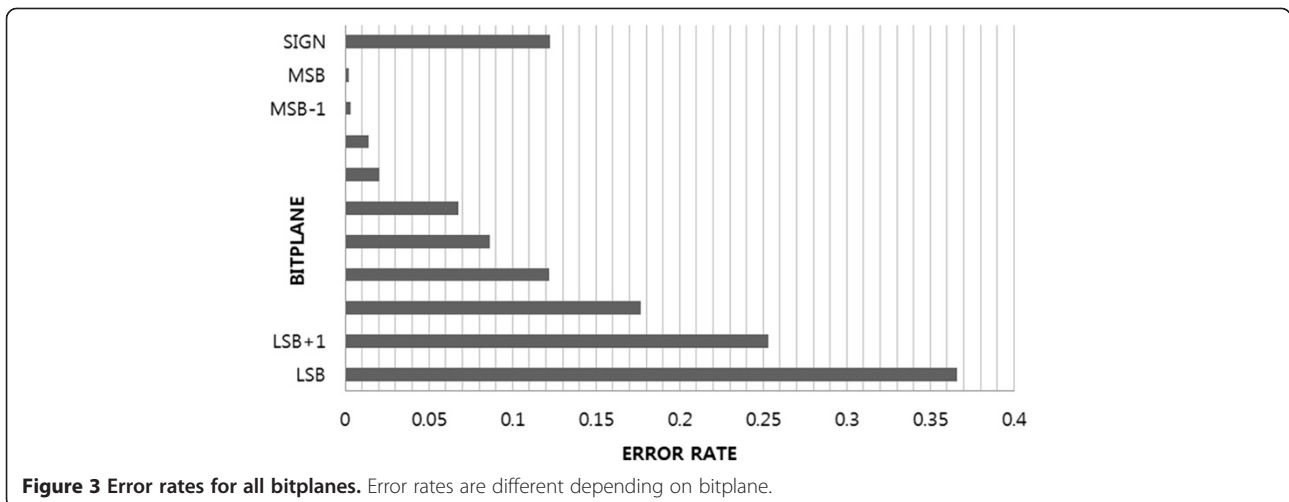
**Figure 1** Block diagram of the proposed method.

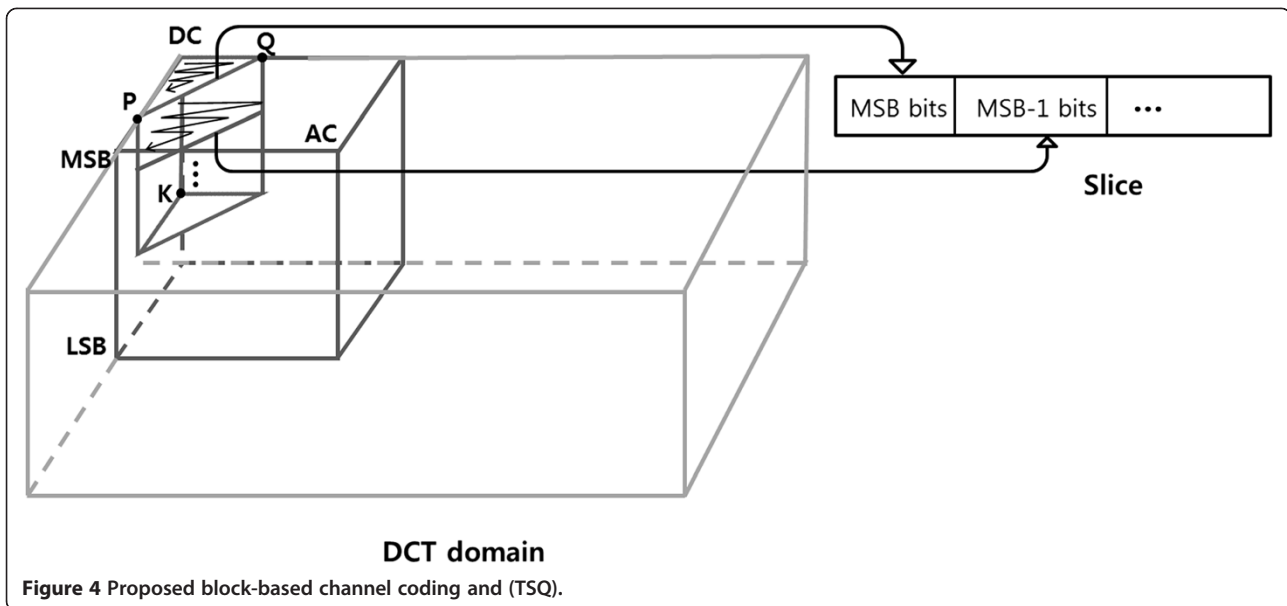


### 2.3. Belief propagation of fixed crossover probability to LDPCA

Although conventional algorithms can estimate a per-slice error probability, they set all variable nodes to a fixed crossover probability for an entire slice. This is one

reason why conventional DVC methods assume that the states of the variable nodes are identically and independently distributed. However, it can be easily seen that SI errors concentrate in certain regions which are commonly boundaries of objects. Therefore, the error

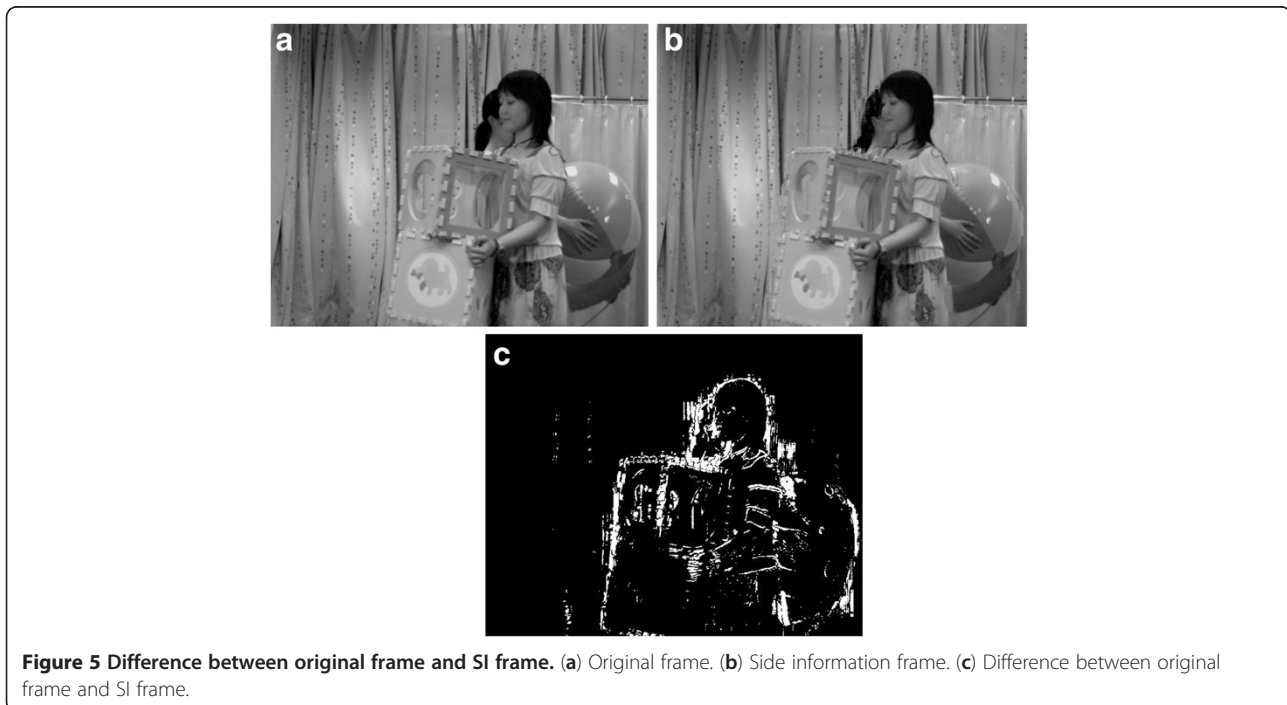




probabilities can vary significantly depending on location. Note that LDPCA decoding based on belief propagation influences all the connected variable nodes if the local crossover probabilities are incorrect. Channel decoding can become slower and error correction performance can be also degraded when using a fixed crossover probability for all the variable nodes. The correction performance of LDPCA decoding can be improved using adaptive conditional probabilities based on local error rates.

### 3. Proposed adaptive DVC with received motion vectors

Odd frames are generally coded as WZ frames in DVC systems. The WZ frames are mainly coded using two key modules: channel decoding and side information generation. For high performance and low computational complexity of the decoder, a new adaptive distributed encoder is proposed, with received MVs to increase channel coding performance. To prevent SI errors from concentrating in certain blocks or slices and to apply a



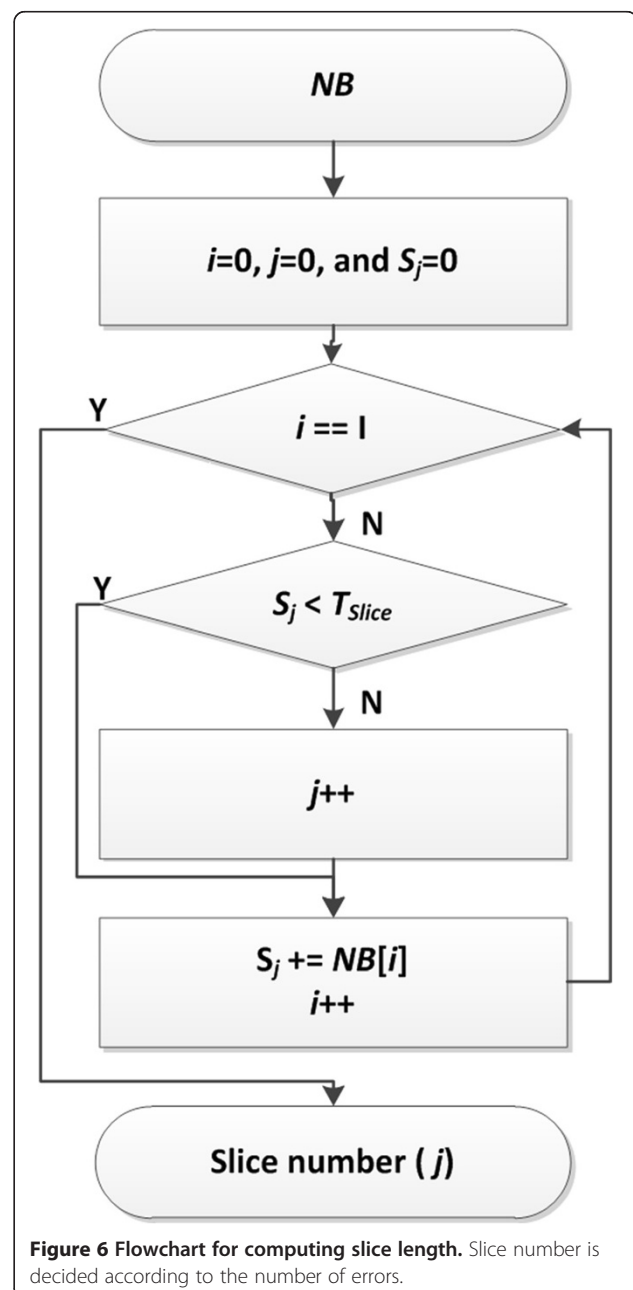
correct crossover probability to each variable node in the LDPCA, the proposed encoder computes an optimum slice partition and transmits one flag for each block to indicate whether the block is coded. In this way, decoding failure rates and channel coding delays can decrease, and the overall performance of DVC can be improved. Figure 1 shows a block diagram of the proposed method. Key frames at  $(t - 1)$  and  $(t + 1)$  are coded and sent using H.264/AVC intra-frame coding. A WZ frame at the time  $(t)$  is coded using the LDPCA channel coder. On the proposed decoder side, the key frames are reconstructed using a conventional H.264/AVC intra-frame decoder. First, base side information (BSI) is generated using the decoded key frames, and then the MVs are again estimated to predict a final SI from the BSI for the two key frames in the decoder. The estimated MVs are transmitted to the proposed DVC encoder. In the proposed encoder, the SI error positions and magnitudes are estimated using the PSI and compensated using the received MVs. Based on the error information, several blocks are grouped to maintain a fixed error rate for each group, which is called a slice. A coded block map indicating which blocks in a slice are coded is generated and transmitted to the proposed decoder. In addition, each variable node in the LDPCA decoder is set to the correct crossover probability based on the received coded block map, and then channel decoding is performed using the SI and the received parity bits.

### 3.1. Generation of SI and PSI

Because the accuracy of the SI has a dramatic influence on DVC performance, many SI generation algorithms have been proposed. However, their accuracy is strongly affected by the characteristics of the sequences involved. As shown in Figure 2, the proposed method generates the SI in two stages: the BSI and SI generation stages. The BSI is first estimated between the reconstructed key frames ( $(t - 1)$  and  $(t + 1)$  frames) with the conventional algorithm based on adaptive search range [21]. In the second stage, the final MV is again estimated from the BSI to the two reconstructed key frames. The motion vector field is delivered to the corresponding encoder. Syntaxes for the motion vector field are not relevant to the first-stage motion estimation (ME) algorithm. At the encoder side, the SI frame can be predicted from the motion vector regardless of the employment of the first-stage ME algorithm. For the second stage, the motion vector is selected by finding the corresponding block having the smallest sum of absolute differences (SAD). The SAD computations are conducted with the same area for search ranges with two adjacent backward and forward frames. The block having the smallest SAD is selected, and we need to send the motion vector and a

flag ( $w_i$ ) indicating the forward or backward frame is selected.

The final SI is reconstructed using the reconstructed key frames and the MVs, and the estimated MVs are transmitted to the encoder. In the encoder, the PSI is compensated with the received MVs without additional motion estimation, and the PSI is set equal to the SI on the decoder side. Regardless of how the first step (generation of the BSI) is performed, the PSI in the encoder is the same as the SI in the decoder because the MVs to be sent are computed in the second stage. However, the proposed encoder knows all the information about the



**Figure 6** Flowchart for computing slice length. Slice number is decided according to the number of errors.

SI because it has the PSI, which makes it possible to send the correct number of bits immediately, without iterative parity bit requests. As a result, the proposed method can reduce time delay and save bitrate. Each block of the PSI at the encoder side is compensated using the following expression:

$$b_i(x, y, t) = w_i B(x - MV_x^i, y - MV_y^i, t - 1) + (1 - w_i) B(x - MV_x^i, y - MV_y^i, t + 1), \quad (1)$$

where  $b_i$  is the  $i$ th predicted block for the target frame,  $(x, y)$  represent the spatial position of pixel resolution,  $B(x, y, t - 1)$  and  $B(x, y, t + 1)$  are the key frames reconstructed using H.264/AVC intra-frame coding,  $t$  is a time index,  $w_i$  is the flag bit to indicate whether a backward or a forward frame is used, and  $MV^i$  is a motion vector that is received from the decoder.

### 3.2. Triangular-shaped quantization and block-based channel coding

In conventional transform-based DVC systems, quantization is conducted on triangular pyramid-shaped discrete cosine transform (DCT) bitplanes [3-5]. LDPC channel coding is performed sequentially from the most significant bit (MSB) plane to the least significant bit (LSB) plane with the assumption that the error probability in any one bitplane is stationary. However, conventional algorithms based on bitplane decoding can easily fail for higher frequency components and lower significance bitplanes because their error probabilities are likely to be significantly higher, as shown in Figure 3. Figure 3 shows the error rate for each bitplane in the DCT domain for six test sequences.

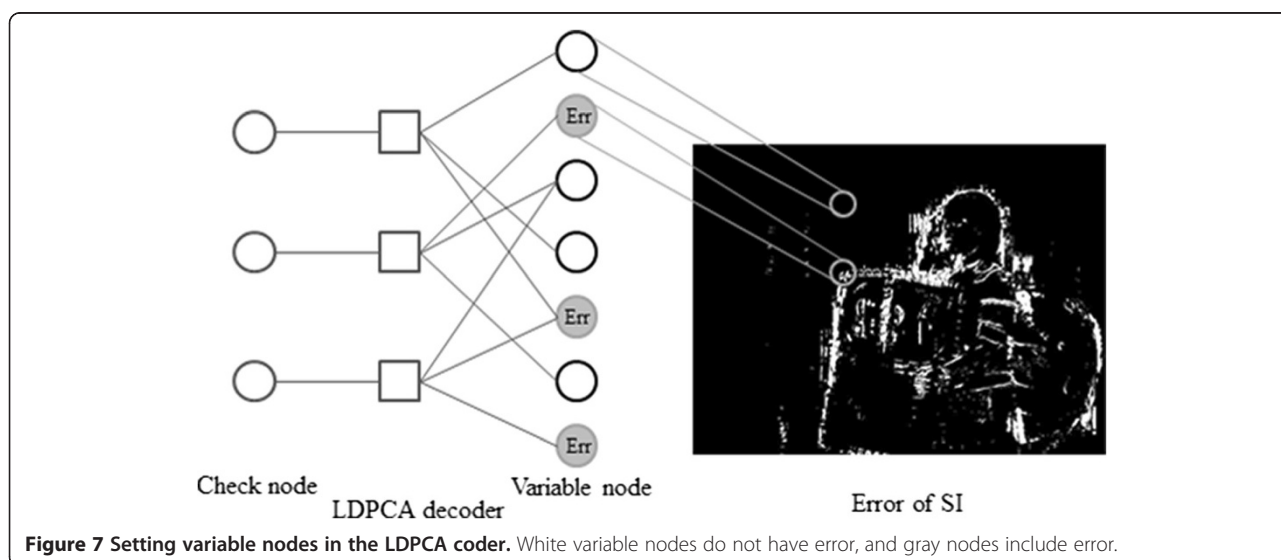
It is helpful to control and balance the bit errors with the aid of a new bit geometry. The proposed method

uses isosceles triangular-shaped quantization (TSQ) of the DCT domain [6] to correctly distribute the error rates. As a result, the proposed algorithm can reduce the probability of decoding failure, and the target bitrate can be gradually adjusted. In other words, error rates can be counterbalanced by combining lower frequency components (which have lower error probabilities) and higher frequency components (which have higher error probabilities) of a block. Therefore, the error rate of a quantized block as determined by the proposed TSQ is not as high as that of the LSB bitplanes or the high-frequency components. The proposed method can also control target bitrates by adjusting quantization parameters and error rates.

Given a target bitrate for each slice, the number of parity bits to be sent is determined. The number of parity bits then determines a target error rate in one slice, which is decoded by an LDPCA in a single pass. According to the target error rate, the TSQ quantization parameters, represented as  $X$ ,  $Y$ , and  $K$  in Figure 4, can be determined. Note that the target error rates are transmitted to the decoder side in the header of the first frame and the target error rate is used as the conditional entropy of the LDPCA decoder. The proposed method performs block-based channel coding that scans and codes quantized frequency components using TSQ, as shown in Figure 4. Figure 4 shows how to gather the DCT bitplanes for a block. The frequency components in block-wise scanning are fed into the LDPCA channel coder, while conventional algorithms use the bitplanes from MSB to LSB in frame-wise raster scanning.

### 3.3. Slice partitioning

Figure 5 shows the difference between an original frame and its associated SI frame. A concentration of errors is



apparent around the edge boundaries of objects in the SI frame. The white regions represent areas with a large number of errors. However, the number of errors varies significantly depending on location within a frame, so that it is necessary to send a large quantity of data for certain regions. Even with a large number of parity bits for DVC, the LDPCA decoder could fail to converge because of an overload of errors. In the proposed algorithm, adaptive-size slices are introduced for slice partitioning. The size of each slice will be determined to maintain a fixed error rate which should be the same as the target error. Because the exact number and positions of errors are easily obtained in the proposed encoder, the correct length of slice to satisfy the target error rate

can be determined. The number of bit errors in a block is defined by

$$NB[i] = \sum_{p=0}^P \sum_{q=0}^Q \sum_{k=0}^K |PSI_i[p, q, k] - ORI_i[p, q, k]|, \quad (2)$$

where  $i$  represents the  $i$ th block index of a frame,  $(p, q)$  are frequency indices, and  $k$  is the depth of the DCT domain. Figure 6 shows the flow chart for computing a slice length. In the flow chart,  $j$  is the slice index, and  $T_{\text{Slice}}$  is the threshold for the target number of errors in a slice. If the accumulated error  $S_j$  is larger than the threshold  $T_{\text{Slice}}$ , an additional slice is created for the subsequent blocks, including the current block. The threshold  $T_{\text{Slice}}$  ( $=700$ ) is

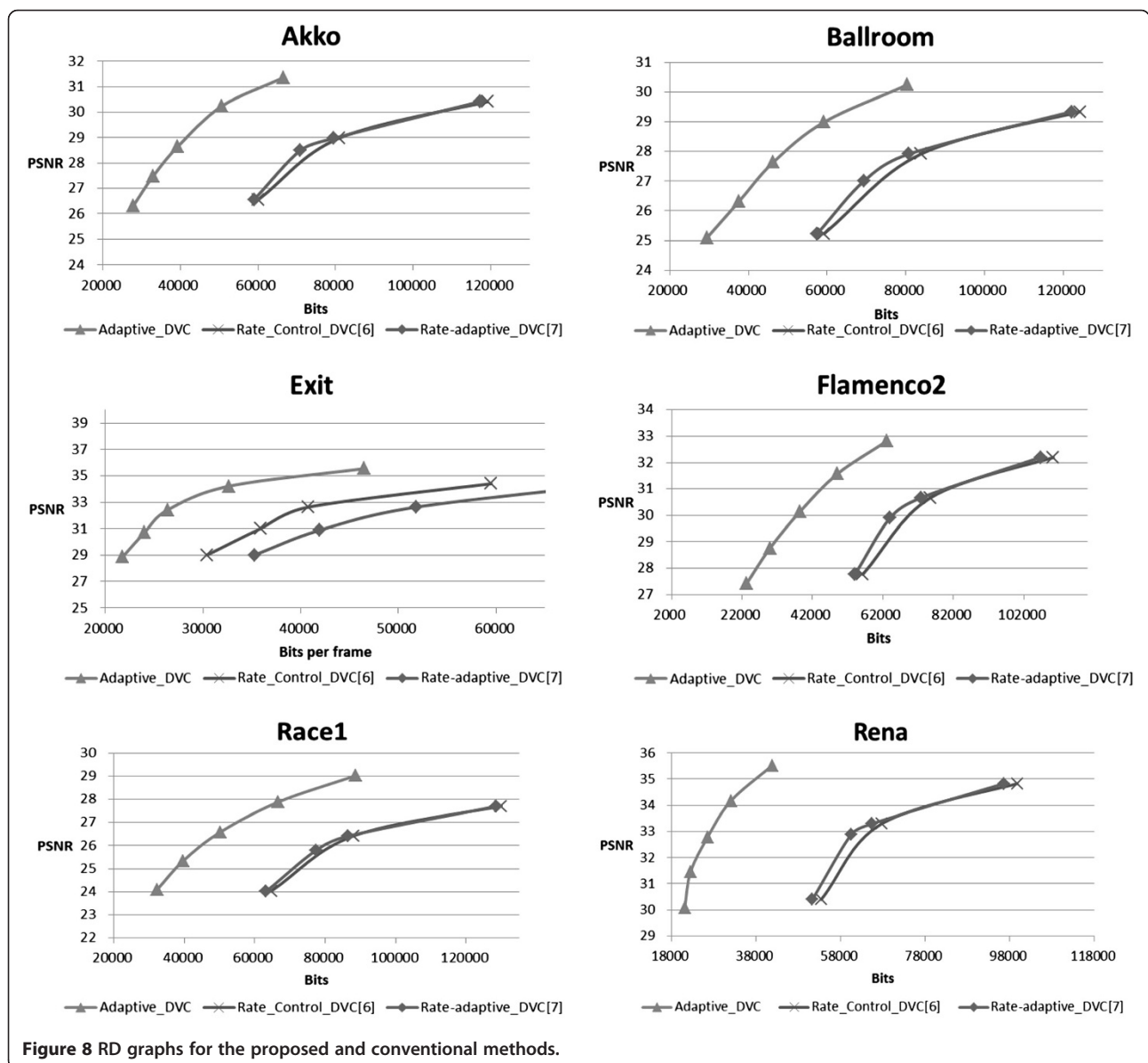


Figure 8 RD graphs for the proposed and conventional methods.



decided according to a target rate. The LDPCA length is set to 6,336, and the target rate is 700/6,336 in the test. Because the proposed method determines the optimal slice length, all slices have almost the same error rate. Therefore, the computational complexity of the proposed decoder can be significantly reduced because there are no failures in correcting the SI with a channel decoder.

### 3.4. Generation of coded map

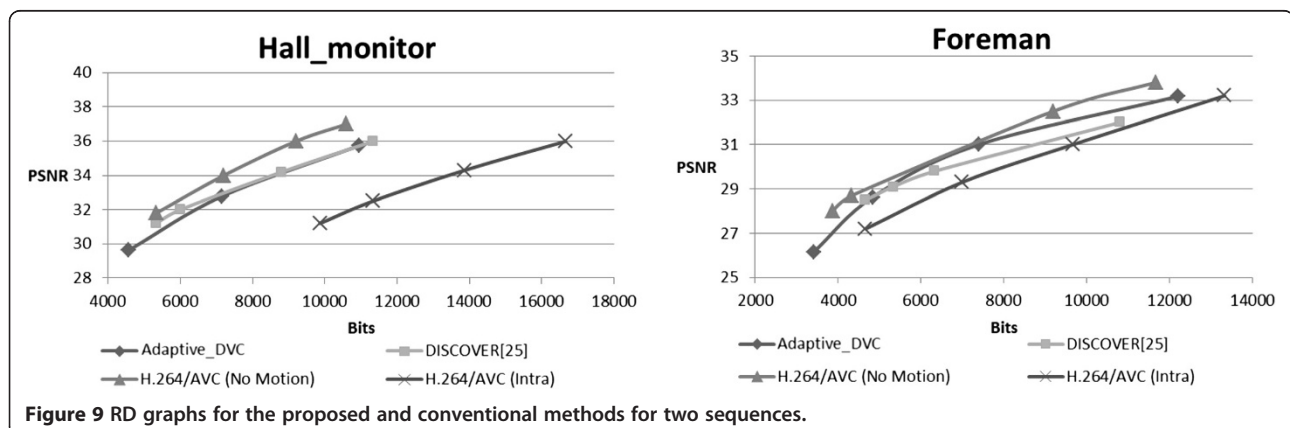
The performance of a channel coder is maximized when the crossover probabilities are set to the actual SI error rate. The channel coder corrects corrupted bits in proportion to the crossover probability. Therefore, it is important to know the correct SI error rate. However, because the channel decoder cannot determine any uncorrected bits in the SI, all variable nodes are set to the same (incorrect) crossover probability in conventional algorithms, even though several bits are corrupted, but others are not, as shown in Figure 7. Therefore, the performance of the LDPCA coder decreases due to incorrect belief propagation.

To alleviate these problems, the proposed method estimates accurate per-block error rates and generates a coded map. The coded map is a group of flags, with each flag representing a block. Each flag indicates whether its block has an error rate greater than a threshold ( $T_{MAP} = 10$ ). Note that the threshold influences video quality and rates. When the threshold is set to a small value, video quality improves and rates increase. The threshold can be adjusted to control the rate and quality. The coded map received from the encoder is used to set each variable node to the correct crossover probability in the proposed decoder. Because the crossover probabilities of the variable nodes associated with non-coded blocks are set to zero and those of other variable nodes are set to the target error rate ( $=700/6,336$ ), the error-free variable nodes will not be altered, and the others will be corrected with the predetermined probability.

## 4. Experimental results

For the performance evaluation of the proposed algorithm, the RD performance of the proposed and conventional algorithms was compared. Six test sequences ('Akko,' 'Ballroom,' 'Exit,' 'Flamenco2,' 'Race1,' and 'Rena') were used, with a format and size of 4:0:0 YUV and 640 × 480, respectively. Key frames were coded using JM 17.0, and five quantization parameters (QPs) (33, 37, 41, 45, and 49) were used. All the sequences consist of 100 frames, with 50 frames of these coded as key frames. The SI was reconstructed based on an adaptive search range [19] and an LDPCA channel coder with a matrix length of 6,336 [5]. Slice lengths, MVs, and the coded map were coded using Exp-Golomb. The two quantization parameters, the height and width of the isosceles TSQ, were set to ((7, 8), (6, 7), (5, 6), (4, 5), (3, 4)), respectively, for the five QPs of the key frames. For example, when the QP for the key frames is 33, the height and width for the two quantization parameters are 7 and 8. The target error rate ( $T_{Slice}$ ) for the slice length was set to 700, and  $T_{MAP}$  for the coded map was set to 10.

Because neither the encoder nor the decoder has the original or the SI, it is not easy to predict accurate SI error rates on either the encoder or the decoder side with minimum computational complexity. To predict SI error rates with minimum computational load on the encoder side, linear interpolation is widely used. However, the peak signal-to-noise ratio (PSNR) between the SI on the decoder side and the PSI obtained by linear interpolation was approximately 27.99 dB for the test sequences. Conventional DVC that generates PSI by linear interpolation does not work for fast-moving sequences. However, the proposed method generates a PSI identical to the SI on the decoder side. Therefore, the proposed method works well regardless of the sequence. Furthermore, the additional computational load is negligible; for example, it takes 8 ms to generate the PSI using a 64-bit Intel Core i5 with a 2.53-GHz CPU.



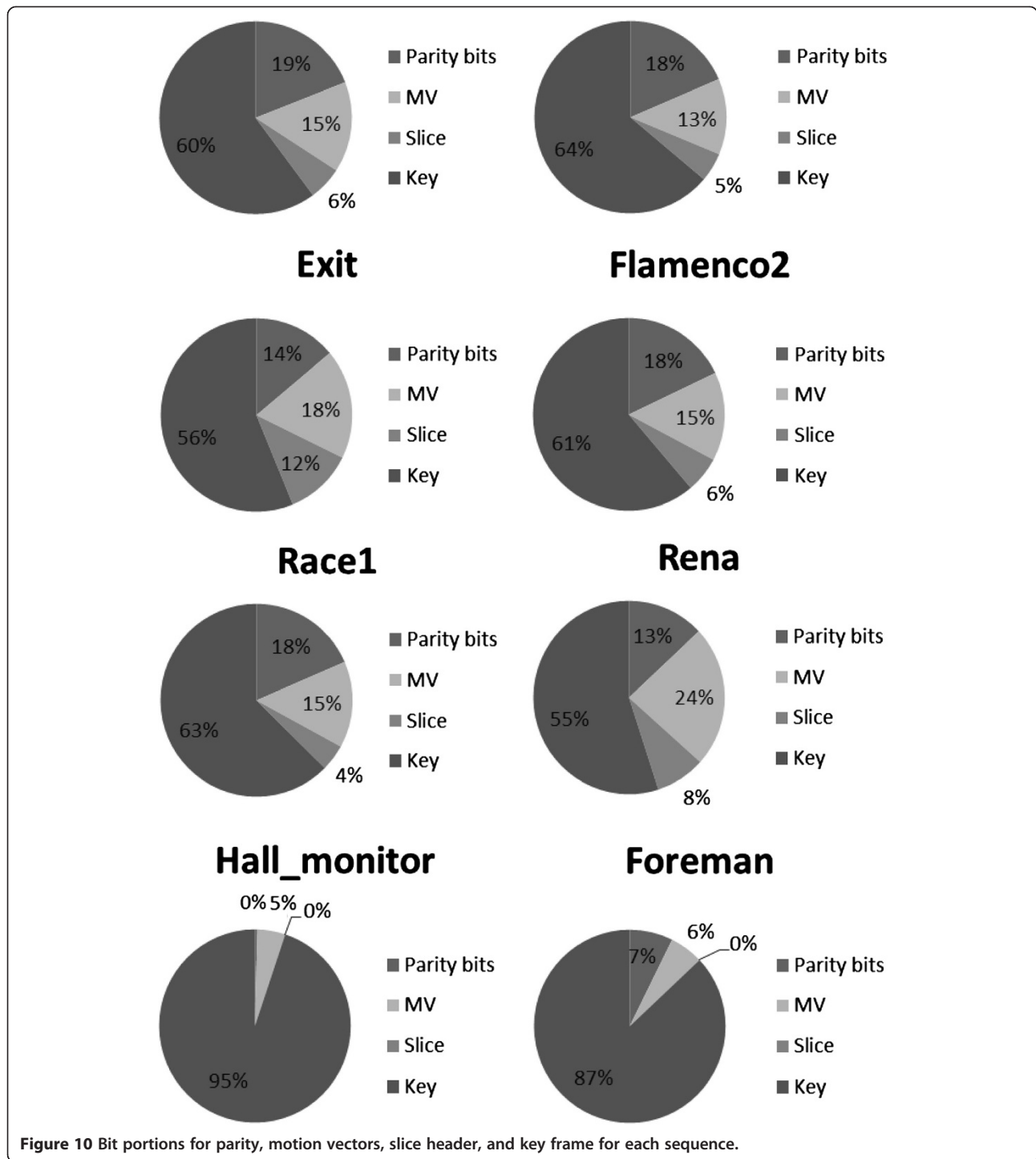


Figure 8 shows RD graphs for the conventional and proposed methods [6,7]. Varodayan et al. proposed a rate-adaptive DVC with a feedback channel [6], and Brites and Pereira proposed an adaptive DVC by estimating error rates using linear interpolation [7]. Note that the additional bits for motion vectors, slice header, indication bits, and so on are included for the proposed

algorithm. Figure 8 shows that the proposed algorithm is better than the conventional methods in terms of rate distortion. The proposed method works well for sequences that include a large concentration of errors in certain regions because the proposed algorithm disperses concentrated errors and sets accurate crossover probabilities for the variable nodes. In addition, the proposed

method gives good coding gains both for slow-moving and fast-moving sequences. The proposed DVC encoder knows all the information about the SI and can provide correct crossover probabilities for any sequence.

We conducted a comparative study in RD performance with the distributed coding for video services (DISCOVER) codec [22]. Figure 9 shows the RD performance of the proposed, DISCOVER, and H.264/AVC methods for two sequences. Note that only executable software of DISCOVER is available, and it can work for only quarter common intermediate format (QCIF) size sequences. We used 'Hall\_monitor' and 'Foreman' sequences of QCIF size for our evaluation. As shown in the figure, the RD performance of the proposed method is comparable to that of the DISCOVER for the slow sequence. However, the proposed method yields higher performance than the DISCOVER for 'Foreman' sequence. In addition, the RD performance of the proposed method is similar to that of H.264/AVC that uses zero motion vectors.

Figure 10 shows bit portions of the parity bits, motion vectors, slice header, and key frames for the test sequences. In the figure, the portion for 'MV' covers the amount of bits for reference indication bits and motion vectors. The portion for 'Slice' is for slice length and coded block indication bits. As shown in the figure, the portion for the key frames is relatively large and that for the parity bits is quite small. The portions for the motion vector vary depending on sequences.

To evaluate the computational load, the relative complexity ratio is defined as:

$$C_R = \frac{T_C - T_P}{T_C}, \quad (3)$$

where  $T_C$  and  $T_P$  are the running times of the conventional and the proposed algorithms, respectively. The computation time of the proposed encoder was increased by approximately 57.493% for computing slice partitions, the coded map, and an  $8 \times 8$  DCT; however, the proposed encoder can run in real time. On the other hand, the complexity of the decoder was significantly reduced. In fact, the reduction in complexity for the LDPCA decoder itself is as much as 99.03%, but the complexity reduction of the whole DVC decoder is approximately 24.50% using Equation 3, compared with that of DISCOVER. The proposed DVC decoder does not require iterative feedback, and the number of iterations in the LDPCA itself is also significantly reduced by setting accurate crossover probabilities in all the variable nodes of the LDPCA decoder. The complexity of the proposed DVC encoder increases around 57.5%, compared with that of DISCOVER, because the proposed encoder needs to generate PSI, coded map, and so on.

However, the proposed DVC encoder can be implemented in real time even with the additional computational load.

## 5. Conclusions

In this paper, a new adaptive distributed video coder is proposed based on received MVs from the DVC decoder side. In the proposed encoder, the PSI is reconstructed using reference key frames and MVs without motion estimation. The correct number and positions of SI errors can be computed on the encoder side using the PSI because the PSI is the same as the SI in the decoder. Based on this error information, slice partitions can be determined by maintaining a fixed per-slice error rate and by generating a coded map and transmitting it to the decoder side. The proposed decoder can then set accurate crossover probabilities for the variable nodes according to the received coded map. The proposed method improves the performance of a channel coder by providing accurate crossover probabilities. It also leads to reduce coding delays in the decoder by eliminating iterative feedbacks. However, a back channel is still needed for the proposed method. In the future, the authors will focus on a new framework to remove the back channel.

### Competing interests

The authors declare that they have no competing interests.

### Acknowledgements

This work was partly supported by the IT R&D program of MKE/KEIT (10039199, A Study on Core Technologies of Perceptual Quality based Scalable 3D Video Codecs) and by The Ministry of Knowledge Economy (MKE), Korea, under the Information Technology Research Center (ITRC) support program (NIPA-2013-H0301-13-1011) supervised by the National IT Industry Promotion Agency (NIPA).

Received: 14 March 2012 Accepted: 11 April 2013

Published: 25 April 2013

### References

1. D Slepian, J Wolf, Noiseless coding of correlated information sources. *IEEE Trans. on Information Theory* **19**(4), 471–480 (1973)
2. A Wyner, J Ziv, The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. on Information Theory* **22**(1), 1–10 (1976)
3. J Micallef, JR Farrugia, C Debono, Low-density parity-check codes for asymmetric distributed source coding, in *Conference on ICITIS 2010* (Beijing, 17–19 Dec 2010), pp. 985–988
4. Q Linbo, H Xiaohai, L Rui, D Xiewei, Application of punctured turbo codes in distributed video coding, in *Conference on ICIG 2007* (Sichuan, 22–24 Aug 2007), pp. 241–245
5. A Aaron, R Zhang, B Girod, Wyner-Ziv coding of motion video, in *Conference on Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems, and Computers 2002, vol. 1* (Pacific Grove, 3–6 Nov 2002), pp. 240–244
6. D Varodayan, A Aaron, B Girod, Rate-adaptive codes for distributed source coding. *EURASIP Signal Processing Journal, Special Section on Distributed Source Coding* **86**(11), 3123–3130 (2006)
7. C Brites, F Pereira, Encoder rate control for transform domain Wyner-Ziv video coding, in *Conference on ICIP 2007, vol. 2* (San Antonio, 16 Sep–19 Oct 2007), pp. 5–8
8. F Zhai, IJ Fair, Techniques for early stopping and error detection in turbo decoding. *Trans. on IEEE Communications* **51**, 1617–1623 (2003)
9. WJ Chien, LJ Karam, GP Abovsleman, Rate-distortion based selective decoding for pixel-domain distributed video coding, in *Conference on ICIP 2008* (San Diego, 12–15 Oct 2008), pp. 1132–1135

10. J Skorupa, J Slowack, S Mys, P Lambert, R Van de Walle, C Grecos, Stopping criterions for turbo coding in a Wyner-Ziv video codec, in *Conference on PCS 2009* (Chicago, 6–8 May 2009), pp. 1–4
11. JL Martinez, C Holder, GE Fernandez, H Kalva, F Quiles, DVC using a half-feedback based approach, in *Conference and Expo on Multimedia* (Hannover, 23 Jun–26 Apr 2008), pp. 1125–1128
12. B Du, H Shen, Encoder rate control for pixel-domain distributed video coding without feedback channel, in *Conference on MUE 2009* (Qingdao, 4–6 Jun 2009), pp. 9–13
13. K Min, S Park, D Sim, Distributed video coding based on adaptive slice size using received motion vectors, in *28th Picture Coding Symposium* (Nagoya, 8–10 Dec 2010), pp. 262–265
14. K Min, S Park, D Sim, Distributed video coding based on adaptive crossover probability using received motion vectors, in *CEWIT 2010* (Incheon, 27–29 Sep 2010)
15. J-S Kim, J-G Kim, K-D Seo, A selective block encoding scheme based on motion information feedback in distributed video coding. *IEICE Trans. on Comm.* **E94.B**(3), 860–862 (2011)
16. W Jia, W Xiaolin, Y Songyu, S Jun, New results on multiple descriptions in the Wyner-Ziv setting. *IEEE Trans. on Information Theory* **55**(4), 1710–1708 (2009)
17. R Liu, Z Yue, C Chen, Side information generation based on hierarchical motion estimation in distributed video coding. *Journal of Aeronautics* **22**(2), 167–173 (2009)
18. Y Shuiming, M Ouaret, F Dufaux, T Ebrahimi, Improved side information generation with iterative decoding and frame interpolation for distributed video coding, in *Conference on ICIP 2008* (San Diego, 12–15 Oct 2008), pp. 2228–2231
19. KY Min, SN Park, JH Nam, DG Sim, SH Kim, Distributed video coding based on adaptive block quantization using received motion vectors. *KICS Journals* **35**(2), 172–181 (2010)
20. G Huchet, W Demin, Distributed video coding without channel codes, in *Symposium on IEEE BMSB 2010* (Shanghai, 24–26 Mar 2010), pp. 1–5
21. KY Min, SN Park, DG Sim, Side information generation using adaptive search range for distributed video coding, in *Conference on PacRim 2009* (Victoria, 23–26 Aug 2009), pp. 854–857
22. X Artigas, J Ascenso, M Dalai, S Klomp, D Kubasov, M Ouaret, The discover codec, architecture, techniques and evaluation, in *Conference on PCS 2007* (Lisbon, 7–9 Nov 2007), pp. 6–9

doi:10.1186/1687-5281-2013-22

Cite this article as: Min and Sim: Adaptive distributed video coding with motion vectors through a back channel. *EURASIP Journal on Image and Video Processing* 2013 **2013**:22.

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---