Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Distributed video coding with adaptive selection of hash functions*

Xin-hao CHEN[1,2], Lu YU[†‡1,2]

([1]Institute of Information and Communication Engineering, Zhejiang University, Hangzhou 310027, China)

([2]Zhejiang Provincial Key Laboratory of Information Network Technology, Hangzhou 310027, China)

[†]E-mail: yul@zju.edu.cn

Received June 16, 2010; Revision accepted Nov. 19, 2010; Crosschecked Mar. 31, 2011

Abstract: We address the compression efficiency of feedback-free and hash-check distributed video coding, which generates and transmits a hash code of a source information sequence. The hash code helps the decoder perform a motion search. A hash collision is a special case in which the hash codes of wrongly reconstructed information sequences occasionally match the hash code of the source information sequence. This deteriorates the quality of the decoded image greatly. In this paper, the statistics of hash collision are analyzed to help the codec select the optimal trade-off between the probability of hash collision and the length of the hash code, according to the principle of rate-distortion optimization. Furthermore, two novel algorithms are proposed: (1) the nonzero prefix of coefficients (NPC), which indicates the count of nonzero coefficients of each block for the second algorithm, and also saves 8.4% bitrate independently; (2) the adaptive selection of hash functions (AHF), which is based on the NPC and saves a further 2%–6% bitrate on average. The detailed optimization of the parameters of AHF is also presented.

Key words: Hash, Collision, Distributed video coding, Wyner-Ziv
doi:10.1631/jzus.C1000198 **Document code:** A **CLC number:** TN919.8

1 Introduction

Distributed video coding (DVC) is a new video coding architecture, based on distributed source coding theory, which studies the independent encoding and joint decoding of two correlated information sequences (Slepian and Wolf, 1973; Wyner, 1975; Wyner and Ziv, 1976; Girod *et al.*, 2005; Guillemot *et al.*, 2007; Yang *et al.*, 2007; Pereira *et al.*, 2008; Dufaux *et al.*, 2009). This paper addresses how to improve the compression efficiency of feedback-free and hash-check DVC, one of the most important challenges in this area (Pereira *et al.*, 2008).

According to the existence of feedback channel from the decoder to encoder, the practical architectures of DVC can be classified as either feedback-based systems (Aaron *et al.*, 2004; Varodayan *et al.*, 2005; Ascenso and Pereira, 2007; Guo *et al.*, 2007; Hua and Chen, 2008) or feedback-free systems (Puri and Ramchandran, 2003a; 2003b; Puri *et al.*, 2007; Asif and Soraghan, 2008). The feedback-based systems yield rate control signals at the decoder, and then make use of the feedback channel to transmit these signals to the encoder side. Since the side information, i.e., the temporal predictors of the source information, is available only at the decoder, feedback-based systems can achieve more accurate rate control and higher coding efficiency than the same-level feedback-free systems. On the other hand, feedback-free systems allocate bits for each coding unit at the encoder independently, and avoid the requirement of feedback channel and the complex procedures of synchronization and interaction between the encoder and decoder. Thus, feedback-free systems benefit from a more flexible architecture and wider application scenarios.

‡ Corresponding author

* Project supported by the National Basic Research Program (973) of China (No. 2009CB320903) and the Program for New Century Excellent Talents in University, China

Hash code is another common technique adopted in DVC, and plays an important role in motion search at the decoder. In the feedback-free systems of Puri and Ramchandran (2003a; 2003b) and Puri *et al.* (2007), a cyclic redundancy check (CRC) code of each coding unit is transmitted to help the decoder check whether the reconstructed information R matches the source information X. If the two hash codes of X and R match, R will be treated as a correct result; otherwise, a new version of R will be generated by the coset code decoding procedure, given the next temporal predictor in the region of motion search. The hash matching procedure is then repeated. This solution of motion search is named 'hash-check' in this paper. An alternative solution of motion search, named 'hash-search', is presented in the feedback-based systems of Aaron *et al.* (2004) and Ascenso and Pereira (2007). In this solution, a subset of the source information is treated as hash bits; for example, a group of high frequency coefficients of a source block in transform-domain is transmitted to help the decoder 'search' a temporal predictor with maximum quality, compared to the source information. The decoder then tries to decode the remaining bits of this coding unit received, through an interactive procedure with the encoder.

In the hash-check solution, there is a special case named 'hash collision'. In this case, the hash codes of wrongly reconstructed information sequences occasionally match the hash codes of source information sequence. The decoder will treat the wrongly decoded information as correct, and break the motion search procedure mistakenly. As a result, such collisions will deteriorate the quality of the decoded image greatly. In general, a long hash code/function has a lower probability of hash collision than a short hash code/function, given the same input binary sequences. However, a long code will increase the bitrate. Thus, there is a trade-off between the collision probability and the length of the hash code. In our analysis in this paper we try to find the optimal trade-off and provide a better compression efficiency than both long and short hash codes (Fig. 1).

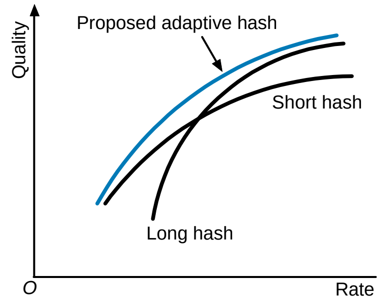

Fig. 1 The trade-off between the hash collision probability and the hash code length, and the aim of this study

The major contributions of this paper include: (1) The statistics of hash collision are analyzed to help the encoder select the optimal trade-off between the hash collision probability and the length of hash code, according to the principle of rate-distortion optimization. (2) Two novel algorithms are proposed: the nonzero prefix of coefficients (NPC) and the adaptive selection of hash functions (AHF). The NPC algorithm provides the necessary information required by the AHF algorithm, and also saves 8.4% bitrate independently; the AHF algorithm saves a further 2%–6% bitrate by exploiting the statistics of hash collisions.

2 Feedback-free and hash-check distributed video coding architectures

We adopt the PRISM framework, which is characterized by a feedback-free architecture with a hash-check motion search at the decoder (Puri and Ramchandran, 2003a; 2003b; Puri *et al.*, 2007). A diagram of our implementation is shown in Fig. 2.

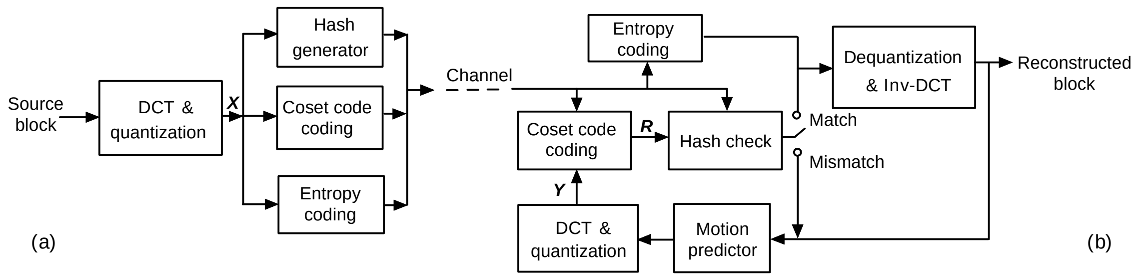

Fig. 2 Diagram of codec: (a) encoder; (b) decoder

The source 8×8 pixel block is transformed by the discrete cosine transform (DCT) and quantized by H.263+ quantization parameters (QP) (Cote *et al.*, 1998) to yield the source information, which is denoted as a matrix of random variables: $\boldsymbol{X}=(X_{ij})$, $0\le i<8$, $0\le j<8$, and X_{00} denotes the DC coefficient. The coset indices of X_{ij} are computed as

$$\overline{X_{ij}} = X_{ij} - \left\lfloor \frac{X_{ij}}{C_{ij}} \right\rfloor \times C_{ij}, \tag{1}$$

where $\overline{X_{ij}}$ are the coset indices, and $\boldsymbol{C}=(C_{ij})$ is a parameter matrix, which is related to the probability of correct decoding. $\boldsymbol{C}$ is stored in both the encoder and decoder. Besides, the coefficients X_{ij} with $C_{ij}<4$ are coded by the entropy coding in the entropy coding module of the practical codec. More analysis of coset code and $\boldsymbol{C}$ can be found in Mukherjee (2009).

All bits of coefficients X_{ij} that have $C_{ij}\ge 4$ are arranged as a binary sequence as the input of a hash function, and then a hash code of $\boldsymbol{X}$ is computed for each block. The cyclic redundancy check (CRC) is adopted as the hash function (Peterson and Brown, 1961; Koopman and Chakravarty, 2004).

Thereafter, the bitstream is composed of the coset indices, the hash codes, and the entropy coded bits of $\boldsymbol{X}$.

At the decoder, a temporal predictor of $\boldsymbol{X}$ is generated from a reference frame by the motion predictor module and DCT & quantization module (Fig. 2). The predictor is denoted as $\boldsymbol{Y}=(Y_{ij})$; the initial motion vector is zero. The noises between $\boldsymbol{X}$ and $\boldsymbol{Y}$ are denoted as $\boldsymbol{N}=(N_{ij})$, $N_{ij}=X_{ij}-Y_{ij}$. In the coset code decoding module, given the side information $\boldsymbol{Y}$, the coset indices of $\boldsymbol{X}$ are decoded to yield a reconstructed matrix $\boldsymbol{R}=(R_{ij})$ as

$$R_{ij} = \arg\min_{r_{ij}\in\Lambda^{\overline{X_{ij}}}} (|r_{ij} - Y_{ij}|), \tag{2}$$

where $\Lambda^{\overline{X_{ij}}}$ is the coset whose elements all share the same coset index $\overline{X_{ij}}$. Obviously, $R_{ij}=X_{ij}$ if and only if the noise $|N_{ij}|<C_{ij}/2$. Then the hash code of $\boldsymbol{R}$ is computed and compared to the hash code of $\boldsymbol{X}$. If the two hash codes match, then $\boldsymbol{R}$ is treated as the correct reconstructed information of $\boldsymbol{X}$ and the motion search stops. Otherwise, the next motion vector in the motion search region is used to yield a new version of $\boldsymbol{Y}$, and then the procedure of motion search and hash check is repeated.

The motion vectors in the region of the motion search are used to generate $\boldsymbol{Y}$ one-by-one if the correct $\boldsymbol{R}$ is not yet found. In the case where all motion vectors are tested but there is no proper $\boldsymbol{Y}$ to yield the correct $\boldsymbol{R}$, the error concealment will be applied.

3 Statistics of hash collision

Puri *et al.* (2007) proposed a 16-bit CRC as a hash function for all blocks. However, a detailed analysis was not presented in the literature. In this section, the statistics of hash collision are analyzed to determine a reasonable and efficient length of hash code for each block.

Hash collision is the case when different inputs of the hash function share the same output value, i.e., the hash code of a wrong $\boldsymbol{R}$ matches the hash code of $\boldsymbol{X}$ accidentally, and the decoder will take the wrong $\boldsymbol{R}$ as the right one. Obviously, the collisions will cause random errors that the decoder cannot detect, and thus the quality of the decoded image is deteriorated.

We assume that: (1) the collision probability increases as the effective binary length of input sequence increases, given the same-level noises and a specified hash function; (2) the longer is the hash code, the lower is the collision probability, given the same length of input sequence with the same-level noises.

A statistical experiment is proposed to verify these assumptions: it estimates the conditional probability of hash collisions, which is denoted as $P(\text{Collision}|\text{QP, Hash, Cnt})$, where QP is the quantization parameter, Hash denotes a certain hash function, and Cnt is the random variable of the count of nonzero coefficients of $\boldsymbol{X}$ (we assume that three coefficients are nonzero: X_{00}, X_{01}, X_{10}, i.e., Cnt≥3. The reason is presented in the next section).

Two of the reasons for choosing QP as one of the statistical conditions are: (1) the energies of $\boldsymbol{X}$ and $\boldsymbol{N}$ depend on QP, given the same video sequence; (2) the theoretical probabilities of correct decoding and the parameter matrix $\boldsymbol{C}$ vary under different QPs according to the rate-distortion trade-offs (Sullivan and

Wiegand, 1998; Mukherjee, 2009). In this study, the range of QP is 1–30.

Five different CRC functions (Table 1) are selected as the hash functions. The detailed computation of CRC code for a certain binary sequence can be found in Peterson and Brown (1961) and Koopman and Chakravarty (2004).

Table 1 Hash functions

Name	Size (bit)	Polynomial
crc4	4	x^4+x+1
crc6	6	x^6+x+1
crc8	8	$x^8+x^5+x^4+1$
crc12	12	$x^{12}+x^{11}+x^3+x^2+x+1$
crc16	16	$x^{16}+x^{12}+x^5+1$

Three of the reasons for choosing the random variable Cnt are: (1) Cnt represents approximately the effective length of input sequence of hash function, if the decoder has been indicated the positions of zero coefficients, given a certain QP; (2) it is easy to compute Cnt for the encoder, unlike any variables depending on the side information $\boldsymbol{Y}$ which the encoder computes with difficulty; (3) the information of Cnt can be transmitted with low cost, as proposed in the next section.

Thereafter, the conditional probability of collisions P(Collision|QP, Hash, Cnt) can be estimated. In the off-line statistical experiment, 14 CIF (common intermediate format) and 19 QCIF (quarter common intermediate format) sequences are chosen, and the size of the group of pictures (GOP) is 4. In this study, the range of QP is 1–30. The result of QP=2 is presented in Fig. 3 and Table 2 as an example. The discrete distribution of Cnt is also presented in Table 2. For brevity, the statistical results of the remaining QPs are not listed.

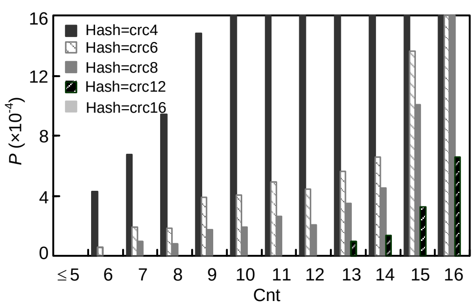

Fig. 3 Conditional probability of collision: *P*(Collision|QP=2, Hash, Cnt)

The part with a probability>0.0016 is not depicted here

Table 2 Distribution of cnt and conditional probability of collision, given QP=2

i	P(Cnt=i) (%)	P(Collision|QP=2, Hash, Cnt=i) (%)				
		crc4	crc6	crc8	crc12	crc16
≤5	6.2	0	0	0	0	0
6	3.6	0.04	0.01	0	0	0
7	3.8	0.07	0.02	0.01	0	0
8	4.1	0.09	0.02	0.01	0	0
9	4.6	0.15	0.04	0.02	0	0
10	5.5	0.17	0.04	0.02	0	0
11	7.4	0.23	0.05	0.03	0	0
12	10.5	0.23	0.04	0.02	0	0
13	10.5	0.24	0.06	0.04	0.01	0
14	10.9	0.29	0.07	0.05	0.01	0
15	10.1	0.50	0.14	0.10	0.03	0
16	7.9	0.80	0.22	0.18	0.07	0

The experimental result confirms the assumptions above. The result shown in Table 2 is a typical example: (1) given QP=2 and a certain hash function, i.e., crc4 or crc6, the conditional probabilities of collision increase as Cnt increases; (2) given QP=2 and a certain value of Cnt, the conditional probabilities of collision reduce as the length of CRC increases.

The result also shows a redundancy of hash code if the hash function is crc16, as in Puri *et al.* (2007). In Table 2, P(Collision|QP=2, Hash=crc4, Cnt≤5)=0, i.e., for 6.2% blocks that have Cnt≤5 no collision occurs when crc4 is adopted. Thus, the adoption of crc16 will cause 12-bit waste for each of these blocks. Similarly, for the 45.7% blocks that have Cnt≤12, P(Collision|QP=2, Hash=crc12, Cnt≤12)=0, and thus the adoption of crc16 will cause 4-bit waste for each of these blocks. Obviously, the compression efficiency can be improved.

In the next section, more sophisticated algorithms are proposed to exploit the statistics of collision, based on the principle of rate-distortion optimization.

4 The proposed algorithms

Two novel algorithms are proposed to exploit the statistics of hash collision: (1) the nonzero prefix of coefficients (NPC), and (2) the adaptive selection of hash functions (AHF). The NPC is the basis of the AHF: it indicates the value of Cnt for the decoder, and provides a reduction of bitrate with no loss of image quality independently. The AHF provides a further improvement of compression efficiency. The optimization of the parameters θ_{QP}, which is used in the AHF, is also proposed.

4.1 Nonzero prefix of coefficients (NPC)

Because of the statistical analysis of hash collision, the information of Cnt must be transmitted to help the decoder select the same hash function for each block as the encoder side. The NPC provides this function with a reduction of bitrate by exploiting the spatial redundancy of $\boldsymbol{X}$.

It is well known that there is a spatial redundancy of $\boldsymbol{X}$: a large number of the coefficients are quantized to zero, while the nonzero coefficients are likely to appear around the DC coefficient X_{00}, for the natural content video. We present the probabilities P_{ij} that each coefficient X_{ij} equals zero to show this redundancy (Table 3), under the test conditions: container_cif, 220 frames, QP=15, GOP=8. In traditional hybrid video coding architectures like H.263, this spatial redundancy is exploited by the run-length variable length code (VLC) (Cote *et al.*, 1998; Yu and Wang, 2010). However, the compression efficiency of the previous architecture of DVC suffers from this redundancy. In PRISM (Puri and Ramchandran, 2002), the refinement quantization bits of each codeword (e.g., each DCT coefficient) are generated and transmitted to the decoder, whether the codeword is zero or not.

Table 3 Probability of zero coefficient X_{ij}*

i	Probability of X_{ij} (%)							
	j=0	1	2	3	4	5	6	7
0	0.0	63.3	69.3	72.9	76.3	81.7	89.9	98.3
1	53.4	67.4	74.0	78.8	83.3	88.7	95.5	99.6
2	59.6	69.6	78.0	83.3	86.5	92.1	97.7	99.8
3	59.2	71.7	79.6	85.8	90.6	94.7	98.6	99.9
4	65.1	75.7	85.4	90.7	94.3	97.9	99.6	99.9
5	73.3	85.1	92.0	95.7	98.0	99.2	99.9	99.9
6	85.3	94.6	98.8	99.5	99.8	99.9	99.9	99.9
7	98.0	99.4	99.9	99.9	99.9	99.9	99.9	99.9

* container_cif, 220 frames, QP=15, GOP=8

We propose a simple and efficient algorithm, namely, the nonzero prefix of coefficients (NPC), to exploit this redundancy for DVC. The encoder transmits a 1-bit syntax element for each coefficient that is coded by coset code before its coset index in bitstream, except the three coefficients X_{00}, X_{01}, and X_{10} due to their statistics. This 1-bit nonzero prefix indicates whether the current coefficient is nonzero or not: '1' for a nonzero coefficient, and '0' for zero. Thereafter, the existence of coset index of each coefficient depends on this nonzero prefix in bitstream: for zero coefficients, the coset indices are no longer transmitted.

Given a coefficient X_{ij} and its parameter of coset code C_{ij}, for zero coefficients, there is only the 1-bit nonzero prefix in bitstream; on the other hand, 1-bit nonzero prefix and $\log_2 C_{ij}$ bits coset index exist in bitstream for each nonzero coefficient. The average number of bits transmitted for each coefficient X_{ij} is: $P_{ij}+(1+\log_2 C_{ij})(1-P_{ij})$. Obviously, the sufficient condition for further compression can be written as

$$\log_2 C_{ij} > P_{ij} + (1+\log_2 C_{ij})(1-P_{ij}). \tag{3}$$

Since $C_{ij}\geq 4$ in practice, the sufficient condition for further compression can be rewritten as $P_{ij}\geq 0.5$. From the statistical result in Table 3, this condition is satisfied for most natural video sequences and most ranges of bitrate.

The algorithm of NPC at the decoder is described as follows. After receiving the bitstream, the decoder will first read the nonzero prefix of each coefficient, except for X_{00}, X_{01}, and X_{10}. If the nonzero prefix is '1', the decoder will read the following $\log_2 C_{ij}$ bits information of coset index; otherwise, this coefficient is set to zero. The decoder then repeats this procedure to decode all of the prefix and coset indices of this block.

A further check of the reconstructed information $\boldsymbol{R}$ is used to refine the results of the hash check at the decoder. Given a temporary result of $\boldsymbol{R}$ during the motion search at the decoder, if the nonzero prefixes of $\boldsymbol{R}$ are different from the corresponding nonzero prefixes received from channel, then this candidate of $\boldsymbol{R}$ is wrong even if the hash codes match. Thus, the motion search procedure avoids an incorrect breaking. In this regard, the nonzero prefix can be treated as an additional hash code for each block. The NPC also improves the error concealment for the blocks decoded unsuccessfully. It first copies the co-located information to yield $\boldsymbol{R}$ as the original system, and then determines which coefficients of $\boldsymbol{R}$ will be set to zero according to the nonzero prefix of each coefficient of $\boldsymbol{X}$.

Finally, the statistical condition Cnt is calculated from the nonzero prefixes of each block at the decoder too. Since X_{00}, X_{01}, and X_{10} are processed as nonzero coefficients in the NPC, the result of Cnt will add 3.

4.2 Adaptive selection of hash functions (AHF)

Since it is difficult to calculate the distortion caused by hash collision theoretically, we define a probability threshold θ_{QP} to represent the maximum tolerable probability of hash collision under each QP. The threshold simplifies the algorithm as well as the optimization procedure. Thereafter, the AHF is designed on the principle that the codec chooses the shortest hash function for each block, if the conditional probability of hash collision remains smaller than such a probability threshold θ_{QP}:

$$\begin{gathered}\min\{\text{Hash}\}\\ \text{s.t. } P(\text{Collision}|\text{QP, Hash, Cnt}) < \theta_{\mathrm{QP}}.\end{gathered} \tag{4}$$

The detailed optimization of θ_{QP} is presented in the next subsection.

Given θ_{QP} and the conditional probability of collision P(Collision|QP, Hash, Cnt), the threshold of Cnt, $T_{\mathrm{crc}X}$, is defined as the minimum value of Cnt given that P(Collision|QP, Hash=crc$X$, Cnt)>θ_{QP}, where crc$X$ represents one of the four hash functions: crc4, crc6, crc8, and crc12 (Fig. 4). These Cnt thresholds $T_{\mathrm{crc}X}$ control the codec directly, and are stored as a table of Cnt thresholds at both the encoder and decoder. Table 4 shows a part of this table. Given a specified QP, the codec will obtain the four Cnt thresholds directly from this table.

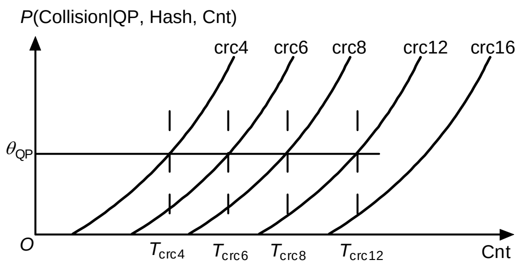

Fig. 4 Calculation of Cnt thresholds, given θ_{QP} and statistics of collisions

Table 4 The Cnt thresholds table

QP	T_{crc4}	T_{crc6}	T_{crc8}	T_{crc12}
1	6	7	7	9
		…		
9	5	5	6	12
		…		
17	16	18	18	99
		…		
26	16	99	99	99
		…		
30	99	99	99	99

Finally, during the practical encoding and decoding procedures, the codec will select a proper hash function for each block by searching the Cnt thresholds table, given QP and Cnt (Fig. 5). Obviously, the collision probability remains smaller than θ_{QP} until the longest hash function crc16 is adopted.

```
if (Cnt<Tcrc4) {
   Hash=crc4; }
else if (Cnt<Tcrc6) {
   Hash=crc6; }
else if (Cnt<Tcrc8) {
   Hash=crc8; }
else if (Cnt<Tcrc12) {
   Hash=crc12; }
else {
   Hash=crc16; }
```

Fig. 5 Adaptive selection of hash functions

4.3 Optimization of θ_{QP}

The parameter θ_{QP} for each QP is calculated to obtain the optimal trade-off between the hash length and the collision probability, according to the principle of rate-distortion-optimization (RDO) (Sullivan and Wiegand, 1998):

$$\theta_{\mathrm{QP}} = \underset{\theta\in[0,\,1]}{\arg\min}(D_{\mathrm{QP}}(\theta) + \lambda_{\mathrm{QP}} R_{\mathrm{QP}}(\theta)), \tag{5}$$

where the variable θ is a real number and takes a value in the range from 0 to 1, D_{QP} and R_{QP} are the distortion function and the rate function of θ respectively, under a certain QP, and λ_{QP} is the Lagrange multiplier of RDO under a certain QP (Sullivan and Wiegand, 1998).

In the off-line optimization procedure of θ_{QP}, the Cnt thresholds are treated as functions of θ: $T_{\mathrm{crc4}}(\theta)$, $T_{\mathrm{crc6}}(\theta)$, $T_{\mathrm{crc8}}(\theta)$, and $T_{\mathrm{crc12}}(\theta)$, respectively. An example of three specified operating points of θ and the corresponding values of $T_{crc4}(\theta)$ is illustrated in Fig. 6. Given different values of θ, the corresponding values of four Cnt thresholds are computed and used to control the codec (Fig. 6).

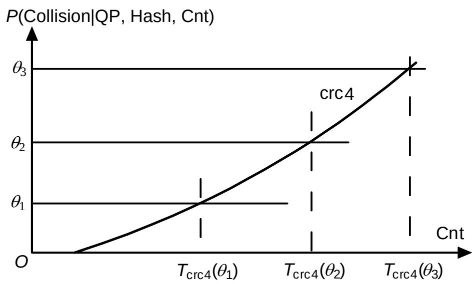

Fig. 6 Calculation of $T_{crc4}(\theta)$, given QP and Hash=crc4

Through the encoding and decoding of video test sequences, the pairs of $D_{QP}(\theta)$ and $R_{QP}(\theta)$ can be computed, given each operating point of θ. If enough operating points of θ between 0 and 1 are tested exhaustively, then the optimal θ_{QP} in Eq (5) can be located.

If we plot each pair of $D_{QP}(\theta)$ and $R_{QP}(\theta)$ as one rate-distortion point, then the optimal rate-distortion performance can be plotted as a convex hull of the set of the operating points of θ (Fig. 7). Note that the *y*-axis in Fig. 7 represents the peak signal-to-noise ratio (PSNR) of the decoded image, and thus the distortion of the decoded image reduces as the PSNR increases. The experimental conditions are: GOP=4, 20 operating points of θ are tested, and the range of QP is 1–30; the tested video sequence has 1772 frames and is combined from six CIF sequences: foreman_cif, coastguard_cif, container_cif, hall_monitor_cif, mother_daughter_cif, and news_cif.

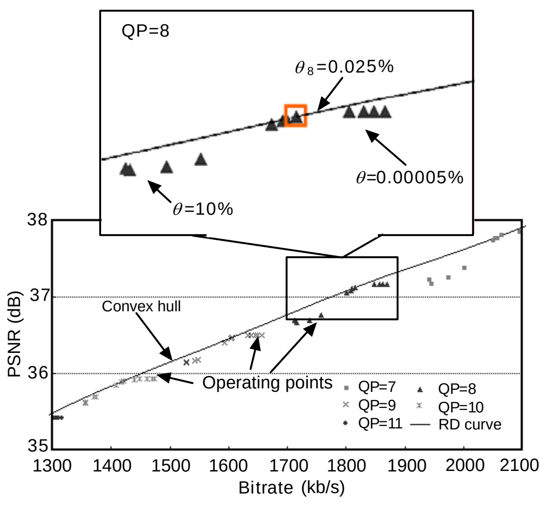

Fig. 7 The optimization of θ_{QP}

5 Experimental results

Three experiments were designed and carried out to evaluate the proposed algorithms. We implemented a feedback-free and hash-check DVC system (Puri and Ramchandran, 2003a; 2003b; Puri *et al.*, 2007) with zero motion skip (Hua and Chen, 2008) as the original system. The key frames were encoded by H.263+ intra mode (except Fig. 9, whose key frames were encoded by H.264/AVC I slice), and the H.263 quantizer was adopted in distributed coded frames (Cote *et al.*, 1998).

5.1 Total improvement in compression efficiency

The proposed algorithms NPC and AHF were implemented on the original codec, and then the total performance of the two algorithms was evaluated. Comparisons between the rate-distortion (RD) curves of the proposed algorithms and those of the original systems are presented in Figs. 8–11. These results show that the proposed algorithms provided a 0–0.5 dB gain on salesman_qcif, and a 0.5–1.0 dB gain on carphone_qcif, compared with the original system.

The corresponding RD curves of the systems in the literature (Puri and Ramchandran, 2003a; Hua and Chen, 2008; Asif and Soraghan, 2008; Do *et al.*, 2009) are also presented in Figs. 8–11. The results of the reference systems were obtained directly from the literature. Compared to the feedback-free system (Asif and Soraghan, 2008) in Fig. 8, the proposed algorithms (GOP=4) provide about a 1–3 dB gain. Compared to the systems of Do *et al.* (2009) and DISCOVER (Artigas *et al.*, 2007) in Fig. 9, the proposed algorithms (GOP=2) provide a better compression performance at a bitrate range of less than 150 kb/s. Compared to the feedback-based system of Hua and Chen (2008) in Fig. 10, the proposed algorithms (GOP=8) provide about a 0–0.5 dB gain over the low bitrate range, although a quality loss up to 1.2 dB between the feedback-free system and the same-level feedback-based system of DVC was reported by Brites and Pereira (2007). Compared to the system of Puri and Ramchandran (2003a) in Fig. 11, the proposed algorithms (only the first frame is the key frame) provide about a 0.75 dB gain, although the results from the literature do not take into account the unsuccessfully decoded blocks and the collisions during the calculating of PSNR.

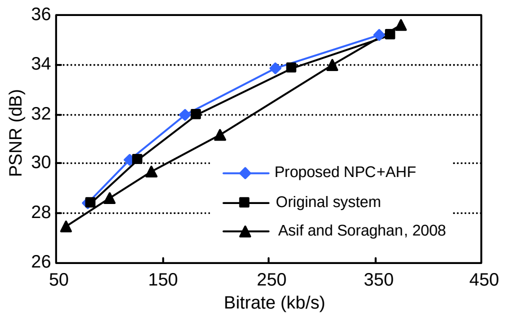

Fig. 8 Rate-distortion curve for salesman_qcif at 30 Hz

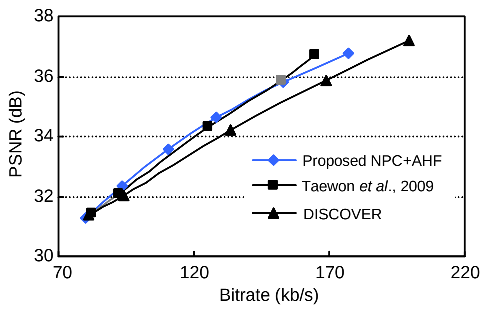

Fig. 9 Rate-distortion curve for hall_qcif at 15 Hz

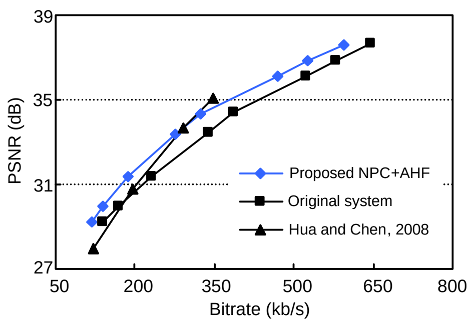

Fig. 10 Rate-distortion curve for carphone_qcif at 30 Hz

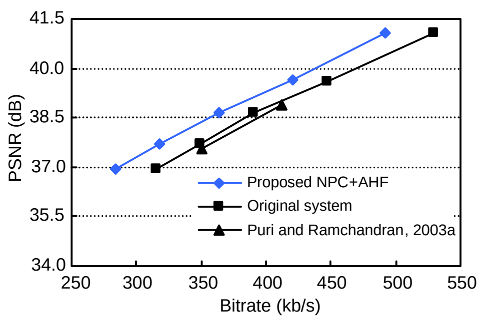

Fig. 11 Rate-distortion curve for carphone_qcif at 15 Hz

5.2 Performance evaluation of NPC

The performance of the NPC was examined. The result in Table 5 shows that the NPC saves 8.4% of total bits on average independently. In the table, ΔBitrate denotes the change of total bitrate, and $\Delta\text{cosetIdxBits}=[(B_{NPC}+B_{RCI})-B_{CI}]/B_{CI}$, where B_{NPC} is the number of bits of nonzero prefixes, B_{RCI} is the number of bits of remaining coset indices, and B_{CI} is the number of bits of coset indices in the original system.

Table 5 Performance of nonzero prefix of coefficients (NPC)*

Sequence	ΔBitrate (%)	ΔcosetIdxBits (%)	ΔPSNR (dB)
foreman_cif	−9.6	−34.8	0
foreman_qcif	−9.2	−30.1	0
football_cif	−8.6	−36.2	0
football_qcif	−9.2	−32.7	0
bus_qcif	−9.5	−28.4	0.0015
coastguard_cif	−9.6	−29.7	0.0005
carphone_qcif	−8.0	−29.3	0.00025
glasgow_qcif	−7.7	−30.3	0
silent_qcif	−5.6	−33.9	0
tempete_cif	−7.5	−23.9	0.00025
Average	−8.4	−30.9	0.00025

* The frame rate was 30 frames/s, GOP=8, and only the luminance component was processed; the hash function adopted was crc16; four QPs (14, 20, 25, 30) were chosen

In Table 5, the calculation of ΔPSNR is independent of the calculation of ΔBitrate and ΔcosetIdxBits: $\Delta\text{PSNR}=D_{NPC}-D_{ORI}$, where D_{NPC} is the distortion result with NPC, and D_{ORI} is the distortion result without NPC. ΔPSNR shows that NPC provides an almost similar quality of the decoded image. The slight improvement of quality (e.g., 0.00025 dB) is from the further check of $\boldsymbol{R}$ with the nonzero prefixes of $\boldsymbol{X}$, which will detect some hash collisions for the decoder, as described in Section 4.1. If the decoder does not adopt this approach of further checking, then the quality of the decoded image remains unchanged (e.g., ΔPSNR=0).

Table 6 shows the overhead of nonzero prefix (e.g., percentage of the number of bits) compared to the whole bitstream of DVC frames. This overhead depends on the parameter matrix of coset code and the video's content. Since the coefficients with C_{ij}<4 are coded by the entropy coding, these coefficients have no nonzero prefix in bitstream. The blocks coded by skip mode also have no nonzero prefix in bitstream. The experimental results in Table 6 show that the overhead is 13.4% on average. As a result, in future work we may design a sophisticated approach to compress these nonzero prefixes.

Table 6 Overhead of nonzero prefixes

Sequence	Percentage of the number of bits (%)			
	QP=14	QP=20	QP=25	QP=30
foreman_cif	14.0	13.6	11.5	9.9
foreman_qcif	15.8	16.6	15.4	15.0
football_cif	11.8	9.9	8.0	7.3
football_qcif	10.1	10.8	10.1	10.3
bus_qcif	12.7	15.5	15.7	17.4
coastguard_cif	16.1	16.8	15.5	15.8
carphone_qcif	15.3	15.7	13.8	12.7
glasgow_qcif	11.9	13.5	12.7	12.7
silent_qcif	14.3	12.9	10.1	7.8
tempete_cif	15.3	17.0	16.3	17.1

5.3 Rate-distortion curves of AHF

Comparisons between the RD curves of the proposed AHF and the fixed length solutions for 4-, 6-, 8-, 12-, and 16-bit CRC functions are presented in Figs. 12 and 13. The results show that the proposed algorithm outperforms all of the fixed length solutions. Compared with the 16-bit CRC solution, AHF saves 2% bitrate (Fig. 12) and 6% bitrate (Fig. 13) on average. This average difference of bitrate between the two RD curves was calculated according to Bjontegaard (2001). In the experiment, GOP=4, both luminance and chrominance components were processed, and the NPC was adopted. The combination sequence with 1772 frames mentioned in Section 4 was used as the test sequence. Since the curve of the 4-bit CRC is almost the same as that of our proposed algorithm at low bitrates, it is not shown in Fig. 13.

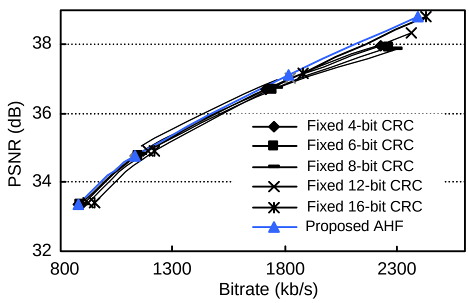

Fig. 12 Rate-distortion curves for the combination sequence at 30 Hz (QP: 6, 8, 12, and 16)

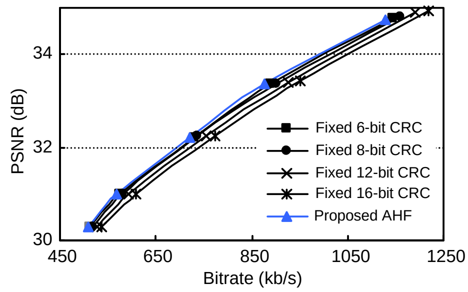

Fig. 13 Rate-distortion curves for the combination sequence at 30 Hz (QP: 12, 16, 20, 26, and 30)

To verify the generality of AHF, Figs. 14 and 15 present the simulation results with different video sequences, which are not selected from the training set (e.g., the combination sequences).

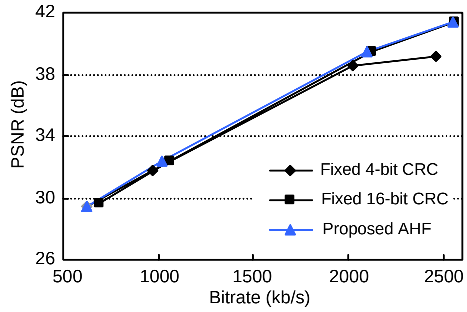

Fig. 14 Rate-distortion curves for bus_qcif at 30 Hz

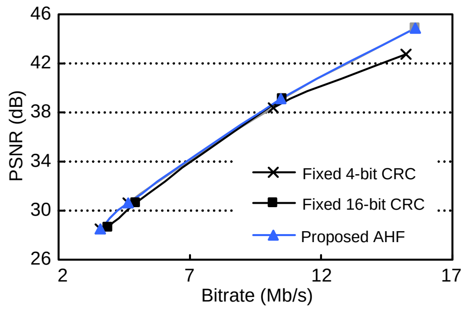

Fig. 15 Rate-distortion curves for mobile_cif at 30 Hz

6 Conclusions

In this paper, the count of nonzero coefficients of each block is treated as a random variable to represent the effective length of input of hash functions. The relationship between the probability of hash collisions and the count of nonzero coefficients is then analyzed. Based on the statistics of hash collisions, two novel algorithms are proposed to achieve the optimal trade-off between the hash collision probability and the length of hash code, according to the principle of rate-distortion optimization. Experimental results confirmed the theoretical analysis and showed that the nonzero prefix of coefficients (NPC) algorithm saves 8.4% bitrate on average while the adaptive selection of hash functions (AHF) algorithm saves a further 2%–6% bitrate on average.

References

Aaron, A., Rane, S., Girod, B., 2004. Wyner-Ziv Video Coding with Hash-Based Motion Compensation at the Receiver. Int. Conf. on Image Processing, **5**:3097-3100. [doi:10.1109/ICIP.2004.1421768]

Artigas, X., Ascenso, J., Dalai, M., Klomp, S., Kubasov, D., Ouaret, M., 2007. The DISCOVER Codec: Architecture, Techniques and Evaluation. Picture Coding Symp, p.1-4.

Ascenso, J., Pereira, F., 2007. Adaptive Hash-Based Side Information Exploitation for Efficient Wyner-Ziv Video Coding. IEEE Int. Conf. on Image Processing, p.29-32. [doi:10.1109/ICIP.2007.4379238]

Asif, M., Soraghan, J.J., 2008. Wyner Ziv Codec Design for Surveillance System Using Adaptive Puncturing Rate. 3rd Int. Symp. on Communications, Control and Signal Processing, p.1454-1459. [doi:10.1109/ISCCSP.2008.4537456]

Bjontegaard, G., 2001. Calculation of Average PSNR Differences Between RD-Curves. VCEG 13th Meeting. VCEG-M33.

Brites, C., Pereira, F., 2007. Encoder Rate Control for Transform Domain Wyner-Ziv Video Coding. IEEE Int. Conf. on Image Processing, p.5-8. [doi:10.1109/ICIP.2007.4379078]

Cote, G., Erol, B., Gallant, M., Kossentini, F., 1998. H.263+: video coding at low bit rates. *IEEE Trans. Circ. Syst. Video Technol.*, **8**(7):849-866. [doi:10.1109/76.735381]

Do, T., Shim, H.J., Jeon, B., 2009. Motion Linearity Based Skip Decision for Wyner-Ziv Coding. 2nd IEEE Int. Conf. on Computer Science and Information Technology, p.410-413. [doi:10.1109/ICCSIT.2009.5234792]

Dufaux, F., Gao, W., Tubaro, S., Vetro, A., 2009. Distributed video coding: trends and perspectives. *EURASIP J. Image Video Process.*, **2009**:508167. [doi:10.1155/2009/508167]

Girod, B., Aaron, A.M., Rane, S., Rebollo-Monedero, D., 2005. Distributed video coding. *Proc. IEEE*, **93**(1):71-83. [doi:10.1109/JPROC.2004.839619]

Guillemot, C., Pereira, F., Torres, L., Ebrahimi, T., Leonardi, R., Ostermann, J., 2007. Distributed monoview and multiview video coding. *IEEE Signal Process. Mag.*, **24**(5):67-76. [doi:10.1109/MSP.2007.904808]

Guo, M., Lu, Y., Wu, F., Li, S.P., Gao, W., 2007. Distributed Video Coding with Spatial Correlation Exploited Only at the Decoder in Circuits and Systems. IEEE Int. Symp. on Circuits and Systems, p.41-44. [doi:10.1109/ISCAS.2007.378177]

Hua, G., Chen, C.W., 2008. Distributed Video Coding with Zero Motion Skip and Efficient DCT Coefficient Encoding. IEEE Int. Conf. on Multimedia and Expo, p.777-780. [doi:10.1109/ICME.2008.4607550]

Koopman, P., Chakravarty, T., 2004. Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks. Int. Conf. on Dependable Systems and Networks, p.145-154. [doi:10.1109/DSN.2004.1311885]

Mukherjee, D., 2009. Parameter selection for Wyner & Ziv coding of Laplacian sources with additive Laplacian or Gaussian innovation. *IEEE Trans. Signal Process.*, **57**(8):3208-3225. [doi:10.1109/TSP.2009.2018617]

Pereira, F., Torres, L., Guillemot, C., Ebrahimi, T., Leonardi, R., Klomp, S., 2008. Distributed video coding: selecting the most promising application scenarios. *Signal Process. Image Commun.*, **23**(5):339-352. [doi:10.1016/j.image.2008.04.002]

Peterson, W.W., Brown, D.T., 1961. Cyclic codes for error detection. *Proc. IRE*, **49**(1):228-235. [doi:10.1109/JRPROC.1961.287814]

Puri, R., Ramchandran, K., 2002. PRISM: a New Robust Video Coding Architecture Based on Distributed Compression Principles. Proc. Annual Allerton Conf. on Communication Control and Computing, **40**:586-595.

Puri, R., Ramchandran, K., 2003a. PRISM: a 'Reversed' Multimedia Coding Paradigm. Int. Conf. on Image Processing, **1**:617-620. [doi:10.1109/ICIP.2003.1247037]

Puri, R., Ramchandran, K., 2003b. PRISM: an Uplink-Friendly Multimedia Coding Paradigm. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, **4**:856-859. [doi:10.1109/ICASSP.2003.1202778]

Puri, R., Majumdar, A., Ramchandran, K., 2007. PRISM: a video coding paradigm with motion estimation at the decoder. *IEEE Trans. Image Process.*, **16**(10):2436-2448. [doi:10.1109/TIP.2007.904949]

Slepian, D., Wolf, J., 1973. Noiseless coding of correlated information sources. *IEEE Trans. Inform. Theory*, **19**(4):471-480. [doi:10.1109/TIT.1973.1055037]

Sullivan, G.J., Wiegand, T., 1998. Rate-distortion optimization for video compression. *IEEE Signal Process. Mag.*, **15**:74-90. [doi:10.1109/79.733497]

Varodayan, D., Aaron, A., Girod, B., 2005. Rate-Adaptive Distributed Source Coding Using Low-Density Parity-Check Codes. Conf. Record of the 39th Asilomar Conf. on Signals, Systems and Computers, p.1203-1207.

Wyner, A., 1975. On source coding with side information at the decoder. *IEEE Trans. Inform. Theory*, **21**(3):294-300. [doi:10.1109/TIT.1975.1055374]

Wyner, A., Ziv, J., 1976. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. Inform. Theory*, **22**(1):1-10. [doi:10.1109/TIT.1976.1055508]

Yang, S.T., Zhao, M.J., Qiu, P.L., 2007. On Wyner-Ziv problem for general sources with average distortion criterion. *J. Zhejiang Univ.-Sci. A*, **8**(8):1263-1270. [doi:10.1631/jzus.2007.A1263]

Yu, L., Wang, J., 2010. Review of the current and future technologies for video compression. *J. Zhejiang Univ.-Sci. C (Comput. & Electron.)*, **11**(1):1-13. [doi:10.1631/jzus.C0910684]