Rate-Distortion Optimized Reference Picture Management for High Efficiency Video Coding

Houqiang Li, *Member, IEEE,* Bin Li, and Jizheng Xu, *Senior Member, IEEE*

***Abstract*—Motion compensation with multiple reference pictures has been widely used during the development of the emerging High Efficiency Video Coding (HEVC) standard, which greatly helps to improve the coding efficiency. Usually, a heuristic strategy is exploited to use the nearest reconstructed pictures as references. However, such a strategy may not be efficient on all occasions, especially when different content characteristics and coding settings are considered. In this paper, we investigate how to manage reference pictures so as to achieve better rate-distortion performance under the memory constraint of the decoded picture buffer at the decoder. We formulate the reference picture management as an optimization problem and approximate its optimal solution. Moreover, we explore how to adjust quality for each picture according to the reference structure to further improve coding efficiency. For some coding cases, where a complicated encoder optimization is unaffordable, we also develop fast algorithms to get the most benefit from reference picture selection. Among them, one strategy has been adopted by the HEVC software and common test conditions to generate the anchor. Experimental results show that the proposed full search algorithm and fast search algorithms achieve significant bitrate reduction.**

***Index Terms*—High Efficiency Video Coding (HEVC), rate-distortion (RD) optimization, reference picture management, reference picture selection.**

I. Introduction

CODING efficiency improvements of generations of video standards have been benefiting from more and more sophisticated motion-compensation design, which is the most important part in a coding system to remove redundancy within video signals. From MPEG-1, H.261 to the latest video coding standard, H.264/MPEG-4 advanced video coding (AVC) [1] and the emerging video coding standard, High Efficiency Video Coding (HEVC) [2]–[4], the precision of motion vectors has been increasing from integer pixel to quarter pixel. The representation of motion vectors has been evolving from a fixed 8×8 block to variously sized blocks. All these advances make the motion prediction more accurate.

Manuscript received April 16, 2012; revised July 20, 2012; accepted August 21, 2012. Date of publication October 5, 2012; date of current version January 8, 2013. The work was supported in part by the NSFC under Grant 61272316 and the 973 Program under Grant 2013CB329004. This paper was recommended by Associate Editor J. Ridge.

H. Li and B. Li are with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, Anhui 230027, China (e-mail: lihq@ustc.edu.cn; yhlibin@mail.ustc.edu.cn).

J. Xu is with Microsoft Research Asia, Beijing 100080, China (e-mail: jzxu@microsoft.com).

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/TCSVT.2012.2223038

Another remarkable advance of motion compensation is the use of more reference pictures. Instead of using only one block in the previous picture as the prediction, bidirectional prediction uses the average of one block in a previous picture and the other in a latter picture. It is shown in [5] that the prediction quality can be significantly improved by using bidirectional motion compensation. Later, researchers found that further improvement of coding efficiency can be achieved by using multiple reference pictures [5], [6]. Basically, the effectiveness of using multiple reference pictures comes from several aspects.

1) *Multiple hypotheses of video signals:* Camera noises are introduced when video signals are captured. Those noises are random signals and cannot be predicted from other pictures. Thus, the motion-compensated residues may contain those noises, which affect the coding efficiency. By combining multiple reference pictures, we may cancel out the influence of the noises of different pictures. From this aspect, the more reference pictures are used, the more coding gain can be achieved, as long as the same content exists in those reference pictures. In [7], the authors showed that coding efficiency improvement could still be observed even when they increased the number of reference pictures to 50.
2) *Occlusion:* Occlusion may make a block unable to find its corresponding block in a previous picture. However, since objects are moving, the same content may be present in other previous pictures. Thus, increasing the number of reference pictures can also increase the probability for a block to find its match. However, once a block finds its correspondence, a further increase of reference pictures may not help anymore from this aspect.
3) *Quality of the reference pictures:* The quality of each reconstructed reference picture may vary, especially when bitrates change significantly. In such a case, more reference pictures will increase the chance of finding a correspondence with a higher quality, and will also lead to coding efficiency improvement.
4) *New reference structures:* The design of multiple reference pictures enables some new reference structures. Fig. 1 shows an example of the hierarchical-B structure. In such a coding structure, each picture, except picture f_n and f_{n+4}, will have reference pictures in both directions, i.e., forward and backward. Such a reference structure can provide more efficient coding

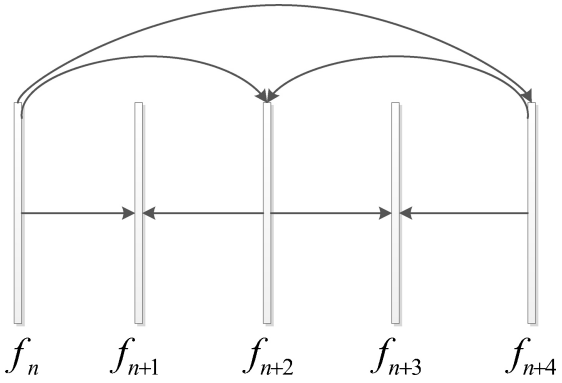


Fig. 1. Example of the hierarchical coding structure.

performance [8]. To enable hierarchical-B structures, multiple reference pictures maintained in the decoded picture buffer (DPB) are necessary. For example, at least three reference pictures are needed to support the structure shown in Fig. 1.

In recognition of the effectiveness of the multiple reference pictures design, both H.264/AVC and HEVC enable a flexible reference picture-management mechanism, and multiple reference pictures can be stored in the DPB. In H.264/AVC, up to 16 reference pictures can be stored in the DPB and can be used simultaneously when coding a picture. It also supports the concept of long-term reference to let the DPB keep a reference picture for a long period of time. During the development of the HEVC standard, all default testing configurations [9], except for all intra coding, use multiple reference pictures.

Since the introduction of multiple reference pictures into the standards, there have been lots of related researches. To address the complexity increase resulting from motion estimation with multiple reference pictures, many schemes have been developed to speed up motion estimation, such as in [10] and [11]. Instead of providing higher coding performance, these schemes, however, are proposed to preserve the coding efficiency with simplified motion estimation and mode decision procedures when there are multiple reference pictures. Another research direction is encoder optimization for coding efficiency, which is also our focus in this paper. Some researchers focus on the selection of long-term reference picture. In [12], a high-quality picture is used as the long-term reference picture for a relatively long period of time on the condition that the network switches from high bandwidth to low, which is helpful to improve the coding efficiency. The performance can also be improved by periodically inserting high-quality long-term reference pictures [13]. In [14], a long-term reference picture is selected by a simulated annealing method to further improve the performance. Liu *et al.* [15] analyzed the motion-compensated prediction in dual frame motion compensation, and proposed an adaptive algorithm to decide whether or not one picture should be coded as long-term reference picture with higher quality by finding some critical points. They also develop the associated bit allocation mechanism. To sum up, although the schemes mentioned above have shown coding performance improvement, they only deal with the simplest case of reference picture selection, i.e., IPPP coding structure and dual reference pictures. It is not straightforward to extend those schemes to handle general cases.

In this paper, we investigate how to maintain multiple reference pictures in DPB to achieve better coding efficiency for various coding settings. Generally, with multiple reference pictures enabled, the nearest several pictures will be used to predict the current one. Such a straightforward strategy is simple yet effective given the assumption that the correlation between two pictures is stronger when their distance is smaller. However, such an assumption is generally true only in a statistical way. For a particular sequence or a particular picture, it may not be true any more. For example, in [16], it is shown that similar pictures may be scattered in the whole video. Of course, buffering as many reference pictures as possible may be a remedy. However, in practice, buffering too many reference pictures may lead to several problems. One is that the decoder needs a large on-chip memory, which is very costly and may consume too much power. Another is that the encoder complexity will also increase. Thus, a practical and reasonable constraint should be placed on the size of DPB. Under such a constraint, how to manage the reference pictures effectively for a particular picture is consequently not so easy, not to mention the optimal DPB management for the whole sequence. In this paper, we formulate the reference picture management under the DPB size constraint for general cases as an optimization problem, and derive its optimal solution. We also present the corresponding quality-adjustment scheme to further improve the performance given the above optimal reference structure. However, to get the optimal solution is too complicated for some applications. In order to address the complexity issue, based on the model we use to derive the optimal solution, we further develop fast algorithms and even derive a fixed pattern for reference picture management. It has been shown that with the proposed full search algorithm, up to 40% and on average 9% bit-saving can be achieved. The fixed reference picture pattern has been adopted by the HEVC reference software, HM (HEVC test model) [17] and used in the common test settings in HEVC, and in the H.264/AVC reference software JM-18.3 (joint model) [18]. Early work has been published in [19] and this paper provides a more comprehensive solution including newly developed quality-adjustment algorithms and fast algorithms for the IBBB coding structure.

The proposed full search algorithm can be applied where a high compression ratio is highly desired, while complexity and delay are not so critical, e.g., offline video compression. Furthermore, for applications where complexity and delay are important, such as video conferencing, the proposed fast search algorithms can be helpful to improve the performance.

The rest of this paper is organized as follows. In Section II, the reference picture management used in HEVC is briefly introduced. In Section III, the problem of reference picture management is formulated and the optimal solution is provided. The related quality-adjustment algorithm will be introduced in that section too. In Section IV, fast algorithms for the IBBB coding structure are discussed in detail. Section V shows some experimental results and Section VI concludes this paper.

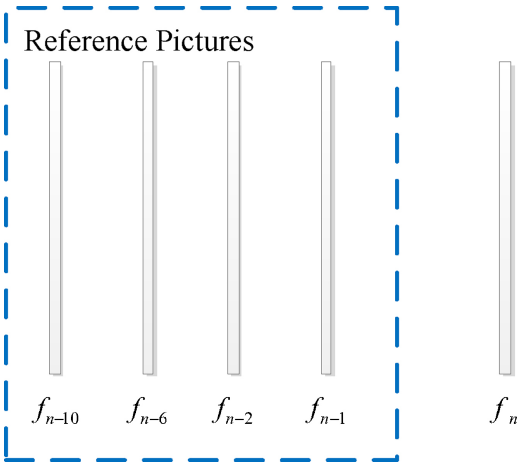


Fig. 2. Example of RPS with four reference pictures.

II. Reference Picture Management Supported in HEVC

Both HEVC and H.264/AVC support flexible reference picture-management mechanism although they operate such tools in different ways.

Generally speaking, reference picture-management mechanism can be divided into two separate parts. The first part provides the flexibility to refine the default reconstructed reference picture lists, named reference picture list reordering or reference picture list modification [20]. It uses a command to let a particular picture be used at a particular position in the reference picture list. The second part provides the flexibility to decide which picture to be removed when DPB is full, as there is usually a constraint at the decoder side on how many pictures could be stored in DPB. For the first part, HEVC and H.264/AVC work in the similar way; but for the second part, they have different mechanisms. H.264/AVC uses sliding window and memory management control operation (MMCO) [20] to remove unused pictures from DPB. Command is signalled to mark pictures that are unused for reference in MMCO. However, during the development of HEVC, it has been recognized that such an MMCO mechanism may lead to incorrect DPB status when temporal scalability is supported and some high temporal level pictures are discarded [21]. When MMCO command is not present due to the intentionally dropping high temporal level pictures, incorrect DPB status will be reached. To provide more robustness, a new reference picture memory-management mechanism named reference picture set (RPS) [22] is proposed to support similar functionality. Instead of signalling command to operate DPB, status of current DPB is signalled in RPS, and the status of current DPB does not depend on any other pervious status. Every picture, which is marked as "used for reference," will be sent with the picture order count (POC) difference (the POC difference between the reference picture and the current picture) at the slice header. Once receiving RPS, the pictures not in RPS will be marked as "unused for reference" and could be removed from DPB when needed [22]. Thus, flexible reference picture memory management strategy could be achieved in a more robust way with RPS.

Fig. 2 shows an example of RPS with 4 reference pictures. Picture f_{n-1}, f_{n-2}, f_{n-6}, and f_{n-10} are in the reference picture set of picture f_n. To save bits used to represent RPS, those pictures in the RPS are usually signalled as POC difference. The reference picture set of the next picture, i.e., f_{n+1}, can only include f_n and the pictures in the reference picture set of f_n. Otherwise, the bitstream is illegal. For instance, if the RPS of f_{n+1} also contains four pictures, there are in total only five different possible reference picture sets.

More generally, we assume that the pictures are encoded sequentially, i.e., when encoding picture f_i, the pictures from f_0 to f_{i-1} are all possible to be used as reference pictures for the current picture f_i, where i is the index according to the encoding order. $s(i)$ is the reference picture set of f_i. $|s(i)|$ is the size of $s(i)$, i.e., the number of reference pictures in $s(i)$. As a legal bitstream, the following two constraints must be obeyed:

$$s(i) \subseteq s(i-1) \cup \{f_{i-1}\} \tag{1}$$

$$|s(i)| \leq r. \tag{2}$$

In (2), r is the maximum number of reference pictures that could be used for one picture.

Although the RPS mechanism has been introduced into HEVC just since HM-5.0 [17], however, as long as the above constraints are satisfied, the algorithms presented in this paper can be applied to all versions of HEVC softwares. In this paper, we base on HM-3.0 to develop corresponding algorithms. A simple reference picture-management mechanism, which in principle can provide similar functions as RPS, is implemented on HM-3.0. All the following investigation is based on the modified HM-3.0 software. Since both HEVC and H.264/AVC support flexible reference structure, the proposed full search algorithm and fast search algorithms can be easily applied to other versions of HM (HEVC) or JM (H.264/AVC). Actually, one reference picture-management strategy presented in this paper was adopted by both HEVC and H.264/AVC reference softwares. Both show the similar coding performance improvement [23], [24].

III. Rate-Distortion (RD) Optimized Reference Picture Management

As introduced in Section II, the encoder has flexibility to choose the reference picture set for the current picture and order the reference pictures in the reference picture list. We consider how to obtain the best coding efficiency by selecting different reference pictures. Usually, the optimization target for the video coding is

$$\{Para\}_{opt} = \arg\min_{\{Para\}} D \quad s.t. R \leq R_c \tag{3}$$

where $\{Para\}$ is the coding parameter set, i.e., reference pictures selection, quantization parameter (QP) setting, motion vector, residue, and other related parameters; D and R are the distortion and bitrate for the sequence; and R_c is the bitrate constraint. The optimization problem in (3) is a constraint opti-

mization problem, which can be converted to an unconstrained optimization problem [25] as shown in

$$\{Para\}_{opt} = \arg\min_{\{Para\}} (D + \lambda R) \tag{4}$$

where λ is the Lagrange multiplier. It indicates the slope in the RD curve, which is usually determined by the experiments or by the QP value [25]. In this paper, we consider how to choose reference picture set for each picture to minimize the total cost of the whole sequence. We also investigate how to adjust each picture's quality to improve the performance according to a given reference structure. The reference picture-management algorithm described in this section will be referred to as full search algorithm in this paper.

A. Reference Picture Management as an Optimization Problem

The status of DPB may significantly impact the coding efficiency. If there is not any constraint on the buffer size at the decoder, i.e., the decoder can buffer all reconstructed pictures, then encoding one picture can use all previous encoded pictures and we do not need to manage DPB. But, as mentioned in Section I, in practice a decoder always has the constraint on the DPB size. Moreover, the size may be restricted to just several pictures due to the high cost of on-chip memory.

Without loss of generality, we assume that the total number of pictures to be encoded is $n + 1$, i.e., from f_0 to f_n; encoding/decoding is performed from f_0 to f_n in sequence. At this stage, we also assume that the QP setting for each picture is fixed since we are more interested in how to select the reference picture for the time being. However, how to adjust QP setting will be discussed in the next subsection. Because there is no reference picture initially, the first picture must be coded as an intra picture. To achieve the best performance by selecting the proper reference picture set for each inter picture is equivalent to optimizing the following equation:

$$\begin{aligned} &\{s(1), s(2), \ldots, s(n)\}_{opt} \\ &= \arg\min_{\{s(1),s(2),\ldots,s(n)\}} D + \lambda R \\ &= \arg\min_{\{s(1),s(2),\ldots,s(n)\}} \sum_{i=1}^{n} (D_i + \lambda R_i) \\ &\quad \text{s.t. } s(i) \subseteq s(i-1) \cup \{f_{i-1}\} \text{ and } |s(i)| \le r. \end{aligned} \tag{5}$$

For the first $r + 1$ pictures, all the previously encoded pictures can be stored in DPB and can be used as references. But from picture $r + 2$, the encoder needs to discard one picture to meet the constraint of the total number of reference pictures. For encoding every picture (except the first $r + 1$ pictures), we need to choose r pictures from the existing $r + 1$ pictures to be used as reference pictures. Actually, we can choose less than r pictures; however, considering using more reference pictures will not have negative effects to the coding efficiency, especially when CABAC is used as the entropy coder, we will not consider the case that less than r pictures are used. In this way, the total searching space will be $(P_{r+1}^r)^{n-r} \sim O(r^n)$, where P stands for the permutation, taking into consideration the order of the pictures listed in the reference picture list. Clearly, it is impractical to find the optimal solution by enumerating each possible selection for the whole sequence. Therefore, we need to simplify the problem to make it tractable. As we know, coding of the reference picture index in HEVC employs CABAC, which can adapt to the order of reference pictures in the reference picture lists. No matter how the reference pictures are placed, entropy coder can always allocate the most efficient bin string to the most useful reference picture. Thus, changing of the order of reference pictures in reference picture list will not make much difference to the distortion and the bitrate. It has been shown in [19] that changing of the order of reference pictures only has about 0.4% impact on the BD rate [26]. As the order of pictures listed is not an important factor to the coding efficiency, we can ignore the order in the reference picture list, and the permutation (P_{r+1}^r) could be simplified to combination (C_{r+1}^r). Nevertheless, the total amount of calculation is still $O(r^n)$, which is computationally prohibitive.

To further simplify the problem, we examine how a reference picture can influence the coding efficiency. For a certain picture, the impact to the following pictures depends on its content and its reconstructed quality. Here, the former factor is determined by the picture itself, and the latter factor can be mainly controlled by the quantization step size applied to that picture. The reconstructed quality of the current picture only depends on the distribution of the residue and the quantization step size. In general, when the variance of the residue is large and the quantization step size is small, the quantization errors tend to be approximately the same, regardless of the distribution of the residue [27]. Of course for some extreme cases, such as the case where there is no residue, the quality of the reference pictures also has impact on the reconstructed quality; a better reference will result in a better reconstructed quality. To simplify the discussion, we just ignore those extreme cases, especially when the quantization step size is not large. Actually, as shown in Fig. 8 in Section IV, the impact on the reconstructed quality is very small when changing the reference pictures. To simplify the problem, we ignore such a small difference here when evaluating the reference picture selection. That is to say, with this simplification, the coding efficiency of one picture only depends on how it chooses the reference pictures. In other words, how other pictures choose their references does not have much influence on the coding efficiency of the current picture. Assuming $J_i = D_i + \lambda R_i$, then J_i is a function of $s(i)$. Problem (5) can be converted to the following one:

$$\begin{aligned} &\{s(1), s(2), \ldots, s(n)\}_{opt} \\ &= \arg\min_{\{s(1),s(2),\ldots,s(n)\}} \sum_{i=1}^{n} (J_i(s(i))) \\ &\quad \text{s.t. } s(i) \subseteq s(i-1) \cup \{f_{i-1}\} \text{ and } |s(i)| \le r. \end{aligned} \tag{6}$$

As shown in Fig. 3, each point, labeled as $s(i, j)$, stands for a state of coding one certain picture f_i using the jth possible reference picture selection. Each state of f_i, i.e., $s(i, *)$, connects to each state of the following picture, i.e., $s(i + 1, *)$. As the pictures are encoded according to the predefined order, there is no other arrows except the ones from $s(i, *)$ to $s(i + 1, *)$. The arrow between each connected state pairs is the cost of connecting them. To find the optimal reference picture set of

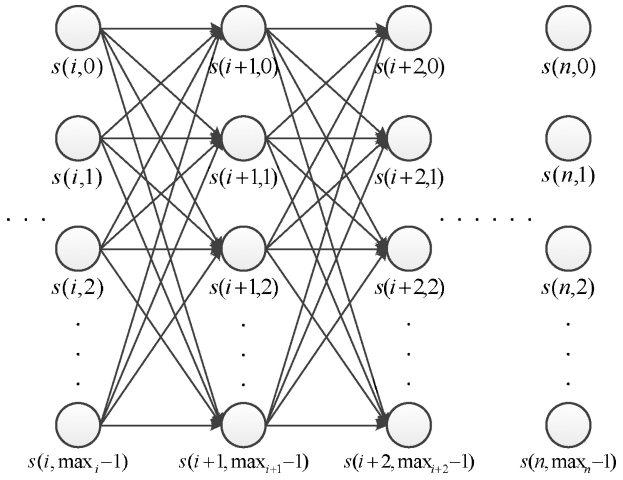


Fig. 3. Finding optimal reference structure.

each picture is equivalent to finding the reference structure with the smallest cost. During the optimization, the encoder can buffer as many reference pictures as it is able to. The only limit is that final reference structure must satisfy the constraint expressed in (1) and (2). When encoding one picture f_i, the encoder can choose r pictures used as reference pictures from the reconstructed ones it has already buffered. The ideal case is that the encoder can buffer all the pictures before f_i. Let max_i denote the number of possible different $s(i)$ when encoding picture f_i (without considering the order of reference pictures in it). Clearly, $max_i = C_i^r$. To solve (6), we need to calculate the value of each $J_i(s(i))$. The complexity is max_i when $i > r$. Since different $J_i(s(i))$ are independent of each other, the total complexity is about $\sum_{i=r+1}^{n}(max_i) \sim O(n^r)$. Usually, r is a small number. Thus, compared to problem (5), problem (6) is less complex and therefore tractable now. When encoding picture i, let $s(i, j)$ denote the jth ($j = 0, 1, 2, \ldots, max_i - 1$) possible set of reference pictures. Then there is a cost J, associated with it. To let the solution meet the constraints of (1) and (2), we define the corresponding cost function as

$$C_{s(i,j)\to s(i+1,k)} = \begin{cases} Cost_{i+1,k}, & \text{if } s(i+1,k) \subseteq s(i,j) \cup \{f_i\} \\ & \text{and } |s(i+1,k)| < r \\ \infty, & \text{otherwise} \end{cases} \tag{7}$$

where $C_{s(i,j)\to s(i+1,k)}$ is the RD cost when connecting the states $s(i, j)$ and $s(i + 1, k)$, and $Cost_{i+1,k}$ is the cost of encoding f_{i+1} using the kth reference picture selection, i.e., $Cost_{i+1,k} = J_{i+1}(s(i + 1, k))$. According to the simplification discussed above, $C_{s(i,j)\to s(i+1,k)}$ only relates to $s(i, j)$ and $s(i + 1, k)$, which is a first-order Markov problem. Then, the optimization problem in (5) can be converted to

$$\{s(1), s(2), \ldots, s(n)\}_{opt} = \arg \min_{\{s(1),s(2),\ldots,s(n)\}} \sum_{i=1}^{n} C_{s(i-1,*)\to s(i,*)}. \tag{8}$$

The optimization problem in (8) can be solved rapidly by the well-known Viterbi algorithm [28].

B. Quality Adjustment for Each Picture

During the reference picture selection mentioned above, some pictures may be used as reference more than the others. Thus, it is straightforward to allocate more bits to such pictures so as to improve the reconstructed pictures' quality, considering that they might have more influence on the coding efficiency of the whole sequence. In this subsection, we investigate how to adjust each picture's RD tradeoff to improve the coding efficiency for the whole sequence, given that the reference structure has been determined by the algorithm described in the previous subsection. Here, the optimization problem is similar to (4), the objective of which is also to minimize the total cost J

$$J = D + \lambda R = \sum_{i=0}^{n} D_i + \lambda \left(\sum_{i=0}^{n} R_i \right). \tag{9}$$

It should be pointed out that we fix the quantization setting for each picture in (4), while, to minimize (9), we fix the reference structure and try to adjust the bitrate for each picture.

Let D_i be the distortion of f_i. As we know, D_i is related to all the reconstructed quality of the previous pictures (they may be used as the reference picture for f_i) and the bitrate of f_i, i.e., R_i. Thus, we have

$$D_i = \mathbb{D}_i(D_{i-1}, D_{i-2}, \ldots, D_0, R_i) \tag{10}$$

where $\mathbb{D}_i(\cdot)$ is a function to determine D_i.

For the first D_0, the optimal D_0 only relates to R_0. Then, we can also have

$$\begin{aligned} D_1 &= \mathbb{D}_1(D_0, R_1) \\ &= \mathbb{D}_1(\mathbb{D}_0(R_0), R_1) \\ &= \mathbf{D}_1(R_0, R_1). \end{aligned} \tag{11}$$

More generally, we can deduce that

$$D_i = \mathbf{D}_i(R_0, R_1, \ldots, R_i) \tag{12}$$

where $\mathbf{D}_i(\cdot)$ is another function to determine D_i, whose variables are $R_0, R_1, \ldots, R_i$.

In (12), the optimal D_i only relates to the bitrate of f_i and those of all the previous pictures. Thus, (9) can be rewritten as

$$J = \sum_{i=0}^{n} \mathbf{D}_i(R_0, R_1, \ldots, R_i) + \lambda \left(\sum_{i=0}^{n} R_i \right). \tag{13}$$

To minimize J in (13), it should be guaranteed that

$$\frac{\partial J}{\partial R_i} = \sum_{j=i}^{n} \frac{\partial \mathbf{D}_j}{\partial R_i} + \lambda = 0 \quad for\ any \quad 0 \le i \le n. \tag{14}$$

Let $\hat{D}_i = \sum_{j=i}^{n} \mathbf{D}_j$, and we can deduce that

$$\frac{\partial \hat{D}_i}{\partial R_i} = \sum_{j=i}^{n} \frac{\partial \mathbf{D}_j}{\partial R_i}. \tag{15}$$

$\hat{D}_i$ can also be expressed as Taylor's expansion at $D_i = D_t$, if high-order terms are ignored

$$\hat{D}_i \approx \hat{D}_i|_{D_i=D_t} + \frac{\partial \hat{D}_i}{\partial D_i}(D_i - D_t) = c + \frac{\partial \hat{D}_i}{\partial D_i}(D_i - D_t) \tag{16}$$

where c is the value of $\hat{D}_i$ when $D_i = D_t$. The high-order terms are ignored to simplify the optimization problem, which is reasonable when the distortion change is small.

By (14) and (15), the optimization problem of minimizing J in (13) is equivalent to

$$\frac{\partial \hat{D}_i}{\partial R_i} + \lambda = 0 \quad \text{for any} \quad 0 \le i \le n \tag{17}$$

which is equivalent to minimizing

$$\hat{J}_i = \hat{D}_i + \lambda R_i \quad \text{for any} \quad 0 \le i \le n. \tag{18}$$

From (15), it can be derived that

$$\frac{\partial \hat{D}_i}{\partial R_i}\frac{\partial R_i}{\partial D_i} = \sum_{j=i}^{n} \frac{\partial \mathbf{D}_j}{\partial R_i}\frac{\partial R_i}{\partial D_i} \tag{19}$$

$$\frac{\partial \hat{D}_i}{\partial D_i} = \sum_{j=i}^{n} \frac{\partial \mathbf{D}_j}{\partial D_i} = 1 + \sum_{j=i+1}^{n} \frac{\partial \mathbf{D}_j}{\partial D_i} \triangleq 1 + \alpha_i \triangleq \omega_i. \tag{20}$$

Taking (16) into consideration, to minimize (18) is equivalent to minimizing

$$\begin{aligned} \hat{J}_i = &\quad \hat{D}_i + \lambda R_i \\ \approx &\quad c + \omega_i (D_i - D_t) + \lambda R_i \quad \text{for any} \quad 0 \le i \le n \end{aligned} \tag{21}$$

where ω_i can be calculated by (20) (how to measure ω_i in practice will be explained in details in Section V-A). c and $\omega_i D_t$ can be ignored in the minimization operation because they are constant during the optimization. Thus, minimizing (21) is equivalent to minimizing

$$\hat{J}_i' = D_i + \frac{\lambda}{\omega_i} R_i \quad \text{for any} \quad 0 \le i \le n \tag{22}$$

where λ is the Lagrange multiplier for the whole sequence. How to determine λ is also described in detail in Section V-A.

In (22), different pictures use different ω_i. The encoder will spend more bitrate on the pictures with smaller $\frac{\lambda}{\omega_i}$, and vice versa. Thus, different bitrates will be allocated to different pictures with different values of ω_i as expressed in (20). More important pictures will have larger ω_i; therefore, more bitrates will be allocated to them.

Note that quality adjustment may lead to the change of the quantization parameter for each picture, which may make the reference structure not optimal any more. In such a case, we can fix the quantization setting and apply the reference picture selection optimization procedure again on each picture. Change of reference structure may also lead to a different quality adjustment. Fortunately, both reference structure adjustment and quality adjustment can improve the coding efficiency. Thus, we can apply these two steps iteratively to obtain better coding performance. In the experimental result section, we will show and discuss how iterations may influence the coding performance.

IV. Fast Algorithms for the IBBB Coding Structure

The algorithm mentioned in Section III needs multipass encoding, which is not suitable for the applications that require low delay and low complexity. In this section, we develop the fast algorithms for a specific coding structure, the IBBB coding structure, which is the default low delay coding structure in the HM. In the IBBB coding structure, the first picture is coded as an intra picture followed by lots of B pictures. Different from normal B pictures, B pictures in the IBBB structure cannot use future pictures as references; instead, they can use two or more previous pictures as references. Each B picture has identical reference picture lists. Thus, the bitstream for one picture can be generated without waiting for later pictures, which greatly reduces the coding latency. It should be noticed that our basic assumption is that the pictures are encoded according to the display order such that if one picture as a reference picture is not of much benefit to the encoding of the current picture, then it is not of much benefit to the encoding of the following pictures. It should be admitted that this assumption is no longer true when periodical scene change exits and it cannot be applied to the hierarchical-B coding structures.

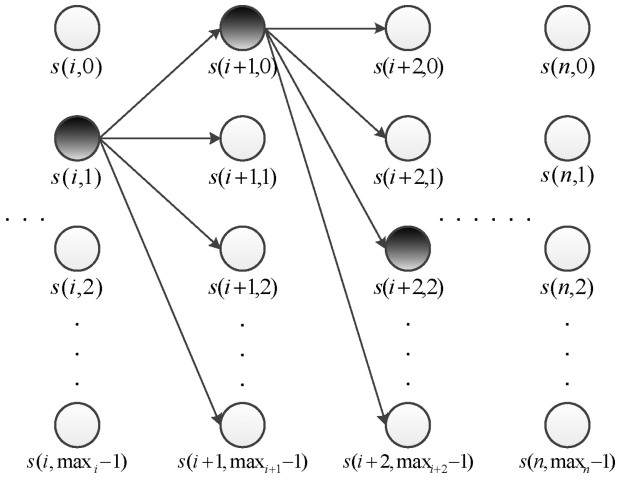


Fig. 4. Finding optimal reference structure under low delay constraint.

A. Suboptimal Solution With a Greedy Strategy

In Fig. 3 and (8), the optimization is for the whole sequence, which requires the encoder to know the information of every picture. However, in the IBBB coding structure, the encoder cannot take advantage of any information of the subsequent pictures. In such a case, the encoder needs to make decision for the picture immediately. We illustrate such a decision procedure in Fig. 4, in which the states filled with black are the best status of the current picture. When f_{i+1} is encoded, all the encoding parameters of f_i have already been decided. Thus, we cannot maintain multiple paths in Fig. 4. Instead, we need to determine one path in Fig. 4 for each picture, which is a greedy strategy to approximately solve problem (8), that is

$$\begin{aligned} &\{s(1), s(2), \ldots, s(n)\}_{opt} \\ = &\arg \min_{\{s(1),s(2),\ldots,s(n)\}} \sum_{i=1}^{n} C_{s(i-1,*)\to s(i,*)} \\ \approx &\bigcup_{i=1}^{n} \arg\min_{\{s(i)\}} C_{s(i-1,*)\to s(i,*)} \\ = &\bigcup_{i=1}^{n} \arg\min_{\{s(i)\}} Cost_i. \end{aligned} \tag{23}$$

It should be explained that the reason we can reach the last step of (23) is that, for low delay encoding, the reference pictures of f_{i+1} can only be chosen from the reference pictures of f_i and f_i itself. Thus, the first condition in (7) is always satisfied. To solve the optimization problem in (23), we will have at most $r+1$ (r is the number of the reference pictures that can be used, also the number of the reference pictures that can be buffered) possible choices to discard one reference picture in the DPB for each picture, except for the first $r+1$ pictures. According to the RD cost of discarding one reference picture among r reference pictures of the previous picture and the reconstructed previous picture, an encoder can test $r+1$ times and get the best solution. Thus, the complexity of greedy strategy is roughly $r+1$ times of the default HEVC IBBB coding structure. In most cases, f_i and f_{i+1} have a strong correlation, therefore, f_i is most likely to be selected as one of the reference pictures of f_{i+1}. Taking this into consideration, we can reduce the encoding complexity to r times for each picture approximately. Thus, we call this method $r\times$ complexity algorithm in this paper.

B. $2\times$ and $1\times$ Complexity Algorithm

For the $r\times$ complexity algorithm proposed above, we need to encode each picture r times to evaluate the cost of discarding a specific reference picture, which is still computation consuming. To further reduce the encoding complexity, we also develop two simpler algorithms, which are called $2\times$ complexity algorithm and $1\times$ complexity algorithm, respectively. As their names indicate, their encoding complexities are roughly two times and one times as much as the default HEVC encoding, respectively.

The optimization target here is the same as that in (23). The only difference is that the $Cost$ in (23) is the actual RD cost, whereas the $Cost'$ in (24) is the estimated RD cost

$$\begin{aligned}&\{s(1), s(2), \ldots, s(n)\}_{opt}\\ \approx\ &\bigcup_{i=1}^{n} \arg\min_{\{s(i)\}} Cost_i\\ \approx\ &\bigcup_{i=1}^{n} \arg\min_{\{s(i)\}} Cost'_i.\end{aligned} \tag{24}$$

To estimate $Cost'$ in (24), we assume that when encoding an inter coded picture f_i, m different pictures could be used as reference pictures. $\beta_{i,j}$ is the percentage of the blocks in f_i using the jth picture as the reference picture. (We apply both unidirectional prediction and bidirectional prediction. If one block uses bidirectional prediction from two different reference pictures, it can be roughly considered that each reference picture contributes half to the region.) Then, we have

$$\sum_{j=0}^{m-1} \beta_{i,j} \le 1. \tag{25}$$

The sum of all the $\beta_{i,j}$ may be smaller than one because some blocks may be coded as intra blocks. When the number of reference pictures decreases from m to $n = m-1$, we have m different ways to discard one reference picture. Generally, discarding one reference picture will cause the increase of the cost, which means the degradation of the coding performance. Suppose that the cost of using m reference pictures is $Cost_{i,base}$, and $Cost'_{i,j}$ is the estimated cost of using $n = m-1$ pictures as reference picture by discarding the jth picture, we have

$$Cost'_{i,j} = Cost_{i,base} + \Delta Cost_{i,j} \tag{26}$$

where $\Delta Cost_j$ is the cost increase by discarding the jth reference picture. When one picture is used as reference by more blocks, its contribution to the coding efficiency is expected to be more important. Therefore, $\Delta Cost_j$ should be larger. To reflect such a trend, we use a linear model to roughly estimate $\Delta Cost_{i,j}$, that is

$$\Delta Cost_{i,j} \approx \kappa \beta_{i,j} \tag{27}$$

where κ is a constant greater than 0. In (27), if $\beta_{i,j}$ is 0, the $\Delta Cost_{i,j}$ is also 0. $\beta_{i,j} = 0$ means that no block chooses the jth picture as reference. In such a case, discarding of the jth reference picture certainly will not cause any performance loss. Under these assumptions, the optimization problem in (24) can be approximated by

$$\begin{aligned}&\{s(1), s(2), \ldots, s(n)\}_{opt}\\ \approx\ &\bigcup_{i=1}^{n} \arg\min_{\{s(i)\}} Cost'_i\\ \approx\ &\bigcup_{i=1}^{n} \arg\min_{\{s(i)\}} (Cost_{i,base} + \Delta Cost_{i,j})\\ \approx\ &\bigcup_{i=1}^{n} \arg\min_{\{\beta_{i,j}\}} (Cost_{i,base} + \kappa \beta_{i,j})\\ =\ &\bigcup_{i=1}^{n} \arg\min_{\{\beta_{i,j}\}} \beta_{i,j}.\end{aligned} \tag{28}$$

In (28), the optimization target is changed to find the smallest $\beta_{i,j}$ for each picture f_i. It is to say that, before encoding f_i using r reference pictures, we can encode it using $r+1$ reference pictures, including r reference pictures of f_{i-1} and f_{i-1} itself. Then, we can find the smallest $\beta_{i,j}$ and discard that picture. Thus, one additional encoding pass is introduced for each picture, which costs two times of computation as much as the default HEVC encoding. Therefore, we call this method $2\times$ complexity algorithm.

Furthermore, we can reduce the complexity by estimating $\beta_{i,j}$ from the encoding result of f_{i-1} due to their strong correlation. When f_{i-1} is encoded (only r reference pictures are used instead of $r+1$ pictures), we can get $\beta_{i-1,j}$ and use them to approximate $\beta_{i,j}$. In addition, we always select f_{i-1} as one reference picture for f_i due to the strong temporal correlation. The picture with the smallest $\beta_{i-1,j}$ will be discarded when encoding f_i, therefore, there is no additional computation needed. The complexity is similar to that of the default HEVC encoding, and we call this method $1\times$ complexity algorithm.

C. Coding Structure for QP Fluctuation Case

In the default IBBB coding setting of HEVC reference software, four pictures constitute a group; within each group, each picture is assigned a different QP, as shown in Fig. 5. A smaller QP is assigned to those pictures numbered as $4n$, which means they will cost more bits and have better qualities. Pictures labeled by odd numbers use a larger QP. They will cost fewer bits. During the development of HEVC, it was believed that with such a QP setting, the decoded video shows

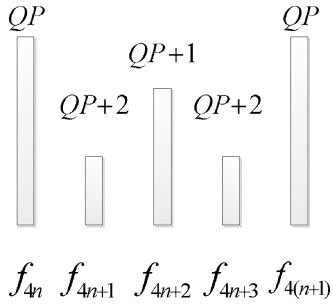


Fig. 5. QP setting for the default IBBB coding structure.

a better quality compared to the video with a constant QP for every picture. Thus, such a QP setting has been used in the default configuration in HEVC.

In such cases, the reference picture selection and the related DPB updating algorithm become very interesting. For example, if two reference pictures are used, f_{4n+3} has at least two choices. One is to use f_{4n+2} and f_{4n+1} as reference pictures; the other is to use f_{4n+2} and f_{4n}. Considering the strong temporal correlation, the nearest picture f_{4n+2} should be used as reference picture for f_{4n+3}. One more picture can be chosen as reference picture from either f_{4n+1} or f_{4n}. One observation is that, from the aspect of temporal correlation, f_{4n+1} is weaker than f_{4n+2}; while from the aspect of quality, f_{4n+1} is worse than f_{4n}. To make full use of the reference pictures, keeping one picture with high temporal correlation is enough. It usually works well to use some pictures with high quality, therefore the using of f_{4n} as reference is expected to be better.

More generally, for the encoding structure with QP fluctuating, from the aspect of keeping strong temporal correlation and providing more high-quality reference pictures, it may improve the coding efficiency by using one nearest reference picture and other high-quality pictures as reference pictures. Let the setting of $m+n$ denote using m nearest reconstructed pictures and n high-quality reconstructed pictures as reference. The effectiveness of the setting of $1+X$ (where X is equal to $r-1$) can also be derived by finding a low cost path in Fig. 3. We can find a path for $r+0$ setting used as anchor, and find more alternative paths for the cases from $(r-1)+1$ to $0+r$. We compare the cost ratios (relative to the anchor) among all the possible settings, and it can be seen from Table I that the average cost of $1+X$ setting is the smallest for the cases of using two, three, and four reference pictures. Thus, for practical applications where the number of reference pictures is not very large and a fixed pattern is needed, the setting of $1+X$ provides the best performance. The BD-rate results of different kinds of reference methods are shown in [23] and [29]. For four reference pictures cases, the setting of $2+2$ and $1+3$ provides about 2.2% and 3.4% bits savings, respectively. It should be noticed that the RD cost saving is relatively small compared with the bitrate savings because the bitrate saving is calculated in PSNR-bitrate domain, and the small difference of RD cost will be enlarged after the log calculation when computing PSNR. This simple $1+X$ reference picture setting for the default HEVC low delay test configuration is proposed in [23] and has already been adopted and integrated into the HEVC reference software from HM-4.0 [17]. Such a setting has also been adopted into the latest H.264/AVC reference software since JM18.3 [18] to improve the coding efficiency.

TABLE I
AVERAGE COST RATIO OF DIFFERENT REFERENCE PICTURE SETTING PATTERN

	2 + 0	**1+1**	0 + 2		
Two reference pictures	1.000	**0.995**	1.079		
	3 + 0	2 + 1	**1+2**	0 + 3	
Three reference pictures	1.000	0.997	**0.994**	1.075	
	4 + 0	3 + 1	2 + 2	**1+3**	0 + 4
Four reference pictures	1.000	0.999	0.997	**0.996**	1.059

V. EXPERIMENTAL RESULTS

Extensive experiments have been conducted to verify the effectiveness of the proposed algorithms. The algorithms are implemented based on the modified HM-3.0 [17], as described in Section II. Class B (1080p, five sequences: *Kimono*, *ParkScene, Cactus, BasketballDrive*, and *BQTerrace*), Class C (WVGA, four sequences: *BasketballDrill, BQMall, PartyScene*, and *RaceHorsesC*), and Class D (WQVGA, four sequences: *BasketballPass, BQSquare, BlowingBubbles*, and *RaceHorses*) sequences are tested. For each sequence, the first 64 pictures (65 pictures for the hierarchical-B coding structure) are used in the experiments. Besides these HEVC test sequences, two additional sequences with 33 pictures (only the first 32 pictures are tested for the IBBB coding structure. We use 33 pictures because we will also test the hierarchical-B coding structure, and 33 pictures can guarantee that the last GOP is complete) are generated to test some special cases. One sequence is called *Scene_Change* (in WQVGA format), which is used to test the case when scene change exits. The sequence is generated from two Class D sequences, *BQSquare* and *BasketballPass*. In the sequence, picture 0 (count from 0) to 8, and 17 to 24 are from the pictures of *BQSquare*. Pictures 9–16, and 25–32 are from *BasketballPass*. The other sequence is called *Occlusion* (in a 800×600 format), which is used to test the case when occlusion exists. This sequence is generated from *ParkScene* and *BasketballPass*. *ParkScene* is background while *BasketballPass* is the foreground. The foreground moves here and there, occluding different regions of the background at different times. Fig. 6 shows several pictures of that sequence.

The results are measured in terms of combined BD rate [26] [the average PSNR is calculated by (29) [30]], where negative number means bitrate savings (coding performance gain). It should be noticed that, as we do not adjust the distortion weight of different color components, using (29) as the measurement and only using $PSNR_Y$ as the measurement lead to similar BD-rate results.

$$PSNR_{avg} = \frac{6 * PSNR_Y + PSNR_U + PSNR_V}{8}. \quad (29)$$

Although there are other video quality-assessment approaches [31], PSNR is a still valid quality measurement for the same video content with the codec type unchanged [32].

Fig. 6. First, 17th, and 27th pictures of the *Occlusion* sequence.

Considering that the PSNR-based video quality measurement is widely used during the development of HEVC [9], for simplicity, we also applies the PSNR based measurement in this research.

The experimental results of the proposed full search reference picture selection are shown in Section V-A, and the results of the fast algorithms designed for low delay cases are provided in Section V-B.

A. Experimental Results on Adaptive Reference Picture Selection

We use both the IBBB and hierarchical-B coding structures to test the algorithm presented in Section III. The experiments can be divided into two steps: 1) reference picture selection algorithm (RefSel step), and 2) quality adjustment for each picture (QuaAdj step). For the hierarchical-B structure, we rearrange the pictures according to the encoding order, so that we can guarantee that the encoding is performed from the first picture to the last sequentially. For QuaAdj step, to adjust the quality for each picture, we calculate the cost of each picture by (21), where λ is the sequence level Lagrange multiplier, i.e., the slope of the RD curve. We can estimate the slope using several RD pairs on the RD curve. The value of ω_i in (21) is calculated according to (20). We give a disturbance at picture f_i (i.e., let f_i become better or worse by using a little different QP), which leads to ∂D_i; we then measure the total cost change of all the following pictures, which is $\partial \mathbf{J}_j$. Since we do not use a rate control scheme, the disturbance at picture f_i influences both the distortion and the rate for the following pictures. According to the mapping between distortion and bitrate, we can compensate the bitrate change by using the total cost, that is

$$\omega_i = 1 + \alpha_i = 1 + \sum_{j=i+1}^{n} \frac{\partial \mathbf{D}_j}{\partial D_i} \approx 1 + \sum_{j=i+1}^{n} \frac{\partial \mathbf{J}_j}{\partial D_i} \qquad (30)$$

where $J_j = D_j + \lambda R_j$ is the cost for picture j. In the quality-adjustment step, as the bitrate of each picture may be changed during the RD optimization, the original QP setting may not be efficient. Thus, after determining ω_i of each picture, the QP may also be changed to obtain better performance. We choose the best QP among nine QPs at picture level according to the total RD cost of each picture. As we use picture-level multiple-QP optimization, it is necessary to know where the gain comes from. Thus, we also test the case that only picture-level multiple-QP optimization based on the RefSel step (called Multi-QP Only) is used to show the efficiency of the proposed quality-adjustment algorithm. Thus, there are in total three operations in our solution: 1) reference picture selection; 2) weight calculation; and 3) multiple-QP optimization. And the results of RefSel includes: 1) the results of QulAdj contains 1) + 2) + 3), and 1) + 3) are included in the results of Multi-QP only.

TABLE II
ENCODING PARAMETERS USED IN EXPERIMENTS

	IBBB	Hierarchical-B
GOP size	4	8
Encoded pictures	64 or 32	65 or 33
Number of reference pictures	2, 3, 4	6
Reference pictures	Every previous coded picture can be used as reference	
Intra pictures	Only the first one	
Anchor QP for intra pictures	22, 27, 32, 37	
Anchor QP setting	hierarchical QP setting as default HEVC common test conditions	
RDO	Enabled	
Entropy coder	CABAC	
ALF, SAO	Enabled	

The main encoding parameters are listed in Table II. All the other encoding parameters are the same as the default high efficiency test configuration in HEVC [9]. The RPS setting used in the anchor applies the default reference structure in HM-3.0 [17]. When the number of existing reference pictures reaches the predetermined maximum value, the picture with the smallest POC will be marked as "unused for reference." Only six reference pictures in the hierarchical-B case is tested, as 5 is the smallest number of reference pictures to fulfill the hierarchical-B coding structure in HM-3.0. Fewer number of reference pictures will lead to a significant performance loss; and we use six reference pictures to provide flexibility to find a more efficient referencing structure. For the IBBB coding structure, we provide the experimental results with two, three, and four reference pictures being used.

The detailed results are shown in Tables III–VI. It can be seen from those tables that for the hierarchical-B coding structure, the proposed full search reference picture selection algorithm (RefSel step) does not bring as much gain as the IBBB coding structure. The most gain comes from the sequence of *Scene_Change* (27.8%) while the average gain

TABLE III
DETAILED BD RATE OF THE IBBB CODING STRUCTURE WITH TWO REFERENCE PICTURES

	RefSel	QuaAdj	Multi-QP Only
Kimono	−0.2%	−4.5%	−3.2%
ParkScene	−2.5%	−7.3%	−3.9%
Cactus	−4.0%	−10.3%	−6.2%
BasketballDrive	−1.2%	−5.4%	−3.5%
BQTerrace	−9.2%	−19.3%	−10.1%
BasketballDrill	−9.9%	−24.4%	−13.4%
BQMall	−3.7%	−11.8%	−6.0%
PartyScene	−6.5%	−13.5%	−8.9%
RaceHorsesC	−1.5%	−4.5%	−2.8%
BasketballPass	−0.9%	−10.1%	−3.1%
BQSquare	−15.7%	−23.4%	−17.4%
BlowingBubbles	−2.4%	−5.6%	−3.8%
RaceHorses	−1.1%	−3.5%	−2.5%
Scene_Change	−32.1%	−40.3%	−32.7%
Occlusion	−18.7%	−24.7%	−23.6%
Average	**−7.3%**	**−13.9%**	**−9.4%**

TABLE IV
DETAILED BD RATE OF THE IBBB CODING STRUCTURE WITH THREE REFERENCE PICTURES

	RefSel	QuaAdj	Multi-QP Only
Kimono	−0.2%	−4.4%	−3.1%
ParkScene	−2.0%	−7.2%	−3.5%
Cactus	−4.2%	−10.5%	−6.4%
BasketballDrive	−0.7%	−5.2%	−3.1%
BQTerrace	−9.7%	−20.4%	−10.8%
BasketballDrill	−9.4%	−22.8%	−13.1%
BQMall	−5.0%	−13.1%	−7.2%
PartyScene	−6.5%	−13.3%	−8.9%
RaceHorsesC	−2.4%	−5.0%	−3.6%
BasketballPass	−1.0%	−9.7%	−3.7%
BQSquare	−16.5%	−25.0%	−18.1%
BlowingBubbles	−2.7%	−5.9%	−4.2%
RaceHorses	−2.0%	−4.2%	−3.1%
Scene_Change	−34.3%	−40.8%	−34.6%
Occlusion	−16.7%	−22.4%	−22.4%
Average	**−7.6%**	**−14.0%**	**−9.7%**

TABLE V
DETAILED BD RATE OF THE IBBB CODING STRUCTURE WITH FOUR REFERENCE PICTURES

	RefSel	QuaAdj	Multi-QP Only
Kimono	−0.2%	−4.4%	−3.1%
ParkScene	−1.1%	−6.6%	−2.8%
Cactus	−3.9%	−10.6%	−6.2%
BasketballDrive	−0.7%	−4.9%	−3.0%
BQTerrace	−7.3%	−20.6%	−8.3%
BasketballDrill	−9.5%	−23.2%	−12.9%
BQMall	−5.5%	−13.7%	−7.8%
PartyScene	−4.9%	−13.2%	−7.3%
RaceHorsesC	−2.2%	−4.9%	−3.5%
BasketballPass	−0.9%	−9.8%	−3.1%
BQSquare	−12.6%	−21.3%	−14.5%
BlowingBubbles	−2.4%	−5.7%	−3.9%
RaceHorses	−1.3%	−3.2%	−2.6%
Scene_Change	−34.4%	−40.2%	−34.8%
Occlusion	−13.0%	−20.1%	−17.7%
Average	**−6.7%**	**−13.5%**	**−8.8%**

TABLE VI
DETAILED BD RATE OF THE HIERARCHICAL-B CODING STRUCTURE WITH SIX REFERENCE PICTURES

	RefSel	QuaAdj	Multi-QP Only
Kimono	−0.3%	−2.4%	−1.6%
ParkScene	−0.1%	−6.7%	−2.7%
Cactus	−2.3%	−7.2%	−5.1%
BasketballDrive	1.0%	−2.4%	−0.9%
BQTerrace	−2.3%	−10.1%	−4.3%
BasketballDrill	−3.2%	−6.2%	−7.8%
BQMall	−1.9%	−8.8%	−4.5%
PartyScene	−1.0%	−8.6%	−4.7%
RaceHorsesC	−0.8%	−2.2%	−2.0%
BasketballPass	−0.1%	−7.3%	−3.3%
BQSquare	−2.0%	−6.9%	−4.1%
BlowingBubbles	−0.7%	−3.8%	−3.3%
RaceHorses	−0.7%	−1.9%	−1.9%
Scene_Change	−27.8%	−32.7%	−28.7%
Occlusion	−3.8%	−2.5%	−5.0%
Average	**–3.1%**	**–7.3%**	**–5.3%**

for the other sequences is 1.3%. That is to say that the default reference structure for the hierarchical-B coding structure used by HEVC is efficient enough for most of sequences. However, for some sequences, e.g., those with scene changes, significant coding gain can be obtained if the proposed full search reference picture selection algorithm is used. For example, in the *Scene_Change* sequence, as described above, there are two scenes interleaving in the whole sequence, i.e., $A_0B_0A_1B_1$, where A and B are two different scenes, and A_0, B_0, A_1, and B_1 contain several pictures. With the proposed full search reference picture selection algorithm, one or more pictures in A_0 may be still kept in DPB when encoding the pictures in B_0 although they may not help. But when encoding the pictures in A_1, those pictures kept from A_0 are helpful to improve the coding efficiency. It has similar benefits to keep one or more pictures in B_0 for the future use when encoding the pictures in B_1. Thus, a significant coding gain is achieved in the *Scene_Change* sequence.

For the IBBB coding structure, nearly 14% average bit savings can be obtained. Comparing the results of QuaAdj step with Multi-QP Only optimization method, we can clearly see that the quality adjustment brings much larger gain than the picture-level multiple-QP optimization. The RD curves for some sequences coded with the IBBB coding structure and four reference pictures are illustrated in Fig. 7. From those curves, we can see that the proposed full search algorithm has significant improvement on the coding efficiency. For each encoding point, RefSel step and the anchor have similar PSNR results, but RefSel step costs fewer bits. It can be observed that RefSel step improves the coding efficiency by saving bits rather than improving PSNRs.

Per-picture PSNR results for one sequence are also provided in Fig. 8. The PSNR results for the 20th picture to the 40th picture are shown in the figure. The anchor and RefSel step have similar per-picture PSNR results, while RefSel step costs fewer bits (smaller bitrate). Both the results for RefSel step and the per-picture PSNRs show that the assumption in Section III-A (i.e., changing reference structure does not influence the

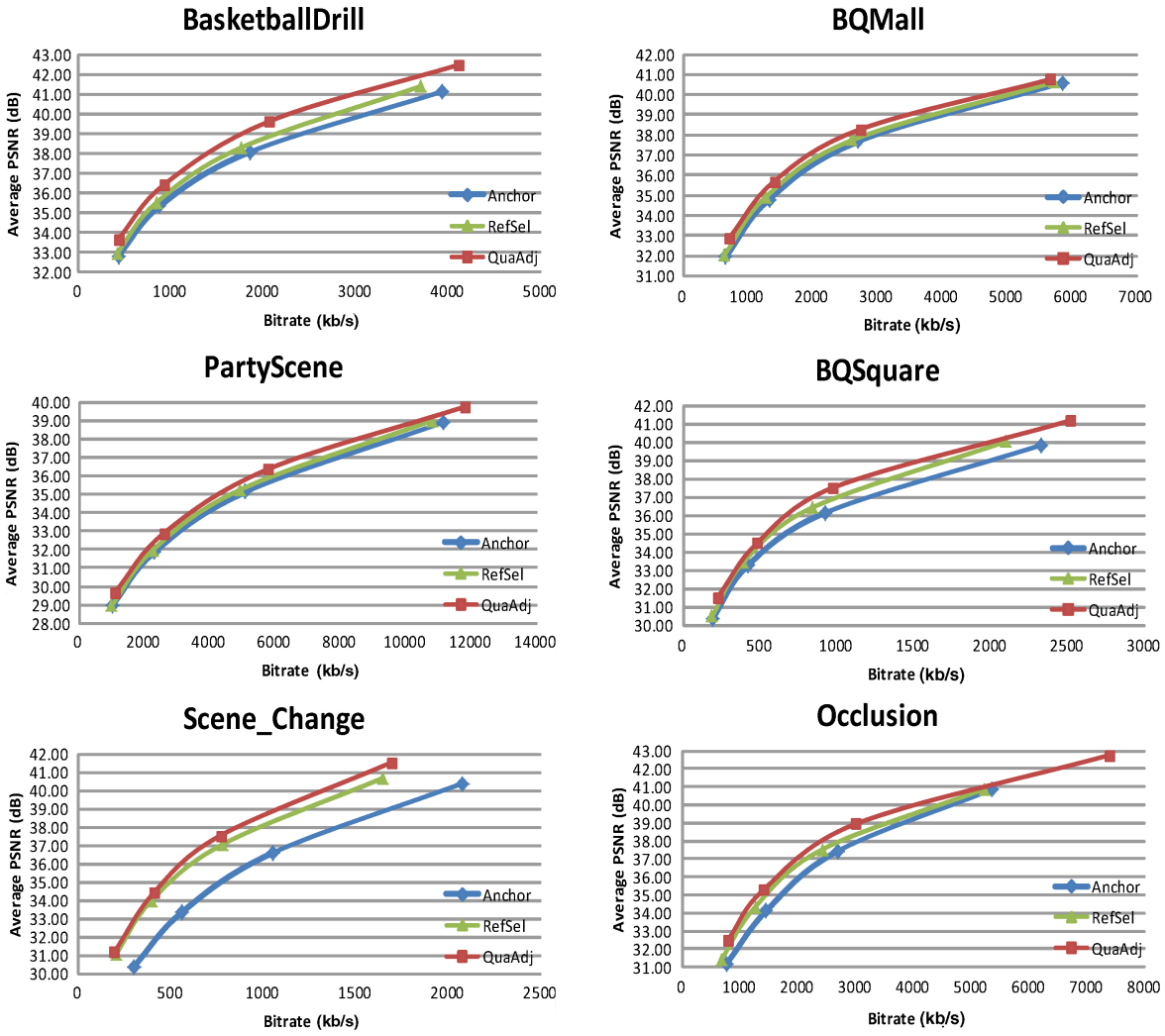


Fig. 7. R-D curves of the IBBB coding structure with four reference pictures.

reconstructed qualities much) is reasonable. From Fig. 8 we can also see that the proposed full search algorithm could improve the PSNR for almost all the pictures. Especially when the reference picture selection and the quality-adjustment algorithms are used together, we can improve the quality of the decoded video up to 1dB for each picture without increasing the bitrate.

In Fig. 9, we show an example of the referencing structure founded by the proposed full search reference structure optimization method. The figure shows that every picture takes the adjacent previous picture as reference picture because of the strong temporal correlation. It also shows that the first picture (picture 0) is used as a reference picture for the whole sequence. The reason is that all the sequence of BasketballDrill is in the same scene, which is a basketball court. The first picture is encoded as an intra-picture at a relatively high quality, thereby all the other pictures can obtain better prediction if they keep the first picture (a high-quality picture) as a reference picture. Pictures 10 and 20 are the other two pictures that are also stored and used for a long period of time. Obviously, the proposed full search referencing structure optimization algorithm can find a solution according to the characteristics of a sequence.

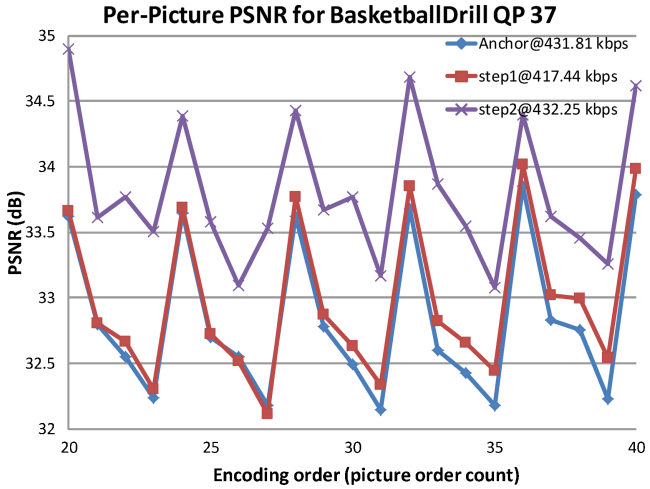


Fig. 8. Per-picture PSNR result for *BaskeballDrill*, IBBB, four reference pictures.

Fig. 10 shows an example of QP setting found by the quality-adjustment step for the first 32 pictures in the *BasketballPass* sequence. Four reference pictures are used and a flat QP setting is set as the initial state. It shows that the result generated by the quality-adjustment step may not lead

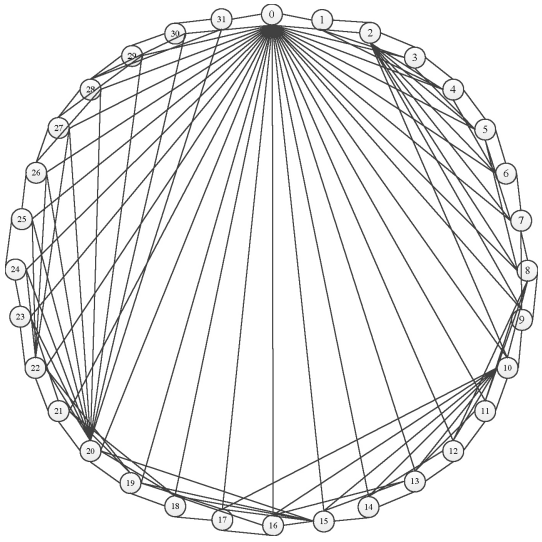


Fig. 9. Referencing structure of first 32 pictures in *BaskeballDrill*, IBBB, four reference pictures, QP 22.

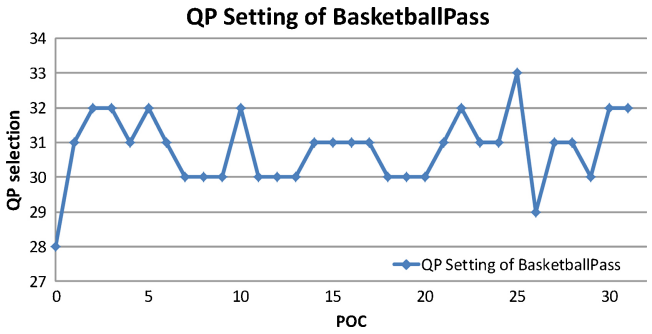


Fig. 10. QP pattern of *BasketballPass* sequence with four reference pictures.

TABLE VII
AVERAGE RESULT OF ITERATIVE ENCODING

Iteration 1		Iteration 2		Iteration 3	
RefSel	QuaAdj	RefSel	QuaAdj	RefSel	QuaAdj
–3.9%	−10.6%	−10.6%	−10.8%	−10.8%	−11.1%

to a regular QP fluctuation pattern. For other sequences, the QP fluctuation pattern may be very different from Fig. 10 as it is also highly related to the characteristics of the sequence.

We also test how interactive using of RefSel step and QuaAdj step influences the coding performance. As iterative encoding is time consuming, we only perform the iterative encoding for the IBBB coding structure with 2 reference pictures. Only the first 32 pictures in Class B, C, and D sequences are encoded in this test. The average results of these sequences with iterative encoding is shown in Table VII. From Table VII we can see that, although iterative encoding can achieve a little better results, the results of only two steps applied are already good enough. The performance difference between two-step encoding and iterative encoding is rather small. Thus, applying steps 1 and 2 once can achieve most of the bit savings.

TABLE VIII
BD RATE OF TWO REFERENCE PICTURES WITH FLAT QP SETTING

	1×	2×	r×	1+X	Full Search
Cactus	0.0%	−0.7%	−2.7%	−1.3%	−2.9%
BQTerrace	0.0%	−5.4%	−6.7%	−5.2%	−7.8%
RaceHorsesC	0.0%	−0.3%	−0.8%	0.1%	−1.1%
Scene_Change	0.0%	−2.0%	−2.7%	−1.2%	−21.7%
Occlusion	0.0%	−6.2%	−9.7%	−5.6%	−9.7%

TABLE IX
BD RATE OF FOUR REFERENCE PICTURES WITH FLUCTUATED QP SETTING

	1×	2×	r×	1+X	full search
Cactus	−1.1%	−1.6%	−3.4%	−3.7%	−3.9%
BQTerrace	−2.3%	−5.2%	−6.7%	−7.2%	−7.3%
RaceHorsesC	−0.3%	−0.6%	−2.0%	−2.1%	−2.2%
Scene_Change	−1.3%	−1.1%	−16.8%	−33.0%	−34.4%
Occlusion	−4.6%	−8.1%	−13.1%	−8.8%	−13.0%

TABLE X
AVERAGE BD RATE WHEN DIFFERENT REFERENCE PICTURE MANAGEMENT ALGORITHMS ARE USED

	1×	2×	r×	1+X	Full Search
Flat QP, 2 ref	0.0%	−2.2%	−3.6%	−2.0%	−5.0%
Flat QP, 3 ref	−1.1%	−2.4%	−4.5%	−4.0%	−5.1%
Flat QP, 4 ref	−1.3%	−2.2%	−4.5%	−3.7%	−4.6%
Fluctuated QP, 2 ref	−0.2%	−3.1%	−4.8%	−3.7%	−7.3%
Fluctuated QP, 3 ref	−1.7%	−3.3%	−5.3%	−6.9%	−7.6%
Fluctuated QP, 4 ref	−1.8%	−2.8%	−5.5%	−6.2%	−6.7%

B. Experimental Results on Fast Algorithms for Low Delay Cases

We test both the flat QP case (i.e., the QP of all the B pictures are the same, which is equal to the QP of I picture plus one) and the fluctuated QP case (as shown in Fig. 5) under three different conditions with two, three, and four reference pictures being used. The approaches of full search (which is the algorithm introduced in Section III-A and tested in Section V-A), r× complexity, 2× complexity, 1× complexity and simple $1 + X$ setting are all tested. Some of the detailed results are shown in Tables VIII and IX, and the summary results are shown in Table X.

It can be seen from these tables that the behaviors of these algorithms on different sequences are consistent. The proposed full search algorithm always performs best on average. For some special cases, certain fast algorithms may also work as well as the full search algorithm. For example, when the flat QP is used with two reference pictures, for the *Occlusion* sequence, r× complexity algorithm works as well as the full search algorithm but the $1 + X$ approach cannot. When the fluctuated QP is used with four reference pictures, for the *Scene_Change* sequence, $1 + X$ approach works as well as the full search algorithm but the r× complexity algorithm cannot. Generally speaking, the proposed full search algorithm works well in the majority of sequences and

provides the best average results as shown in Table X, while the average performance of some fast algorithms, especially $r\times$ complexity algorithm and $1 + X$ approach, is also very good but may not be efficiency-consistent for some sequences (e.g., *Scene_Change* in Table VIII).

Another observation is that $1\times$ complexity algorithm does not help to improve the performance when only two reference pictures are used, especially for the flat QP cases. It is not the design mistake of $1\times$ complexity algorithm that leads to such results. For the case where two reference pictures are used, when encoding f_i, f_{i-1} together with another picture [denoting as f_j $(j < i-1)$] are used as reference pictures. To encode f_{i+1}, the encoder has to decide which picture (f_{i-1} or f_j) is to be kept, and together with f_i to be used as reference pictures. Considering there exists strong temporal correlation between neighboring pictures, f_i prefers to referencing the nearer picture f_{i-1}, especially in the circumstances that f_j and f_{i-1} have similar quality. Thus, f_{i-1} together with f_i tends to be used as the reference pictures for f_{i+1}, which is the same referencing structure as the anchor. The situation changes a little when the fluctuated QP is used, because in such case f_j may have better quality than f_{i-1}, such that $1\times$ complexity algorithm with the fluctuated QP may achieve a little bit savings.

C. Experimental Results Analysis and Discussion

From the experimental results, we can draw the following conclusions.

1) The proposed full search reference picture management algorithm brings about 7% bit savings for the IBBB coding structure and 3% bit savings for the hierarchical-B coding structure. The proposed full search approach is suitable for predefined coding order.
2) For the IBBB coding structure, some fast algorithms can also bring somewhat bit savings. But for the average result, the proposed full search algorithm always performs best.
3) For the IBBB coding structure, when the fluctuated QP is used, $1 + X$ reference structure works well for most of the sequences.
4) For the hierarchical-B coding structure, the default referencing structure is efficient enough. But for some special sequences, such as sequence including scene change, the proposed full search algorithm can significantly improve the coding performance.
5) The proposed quality-adjustment algorithm and the related bit allocation bring additional 6% bit savings for the IBBB coding structure and 4% for the hierarchical-B coding structure, which indicates that the proposed quality-adjustment algorithm works well with various coding structures.
6) The totally bit savings for the proposed full search reference picture selection algorithm and quality-adjustment algorithm is nearly 14% for the IBBB coding structure and more than 7% for the hierarchical-B coding structure, which could be treated as significant improvement for HEVC.

VI. Conclusion and Future Work

In this paper, we modeled the reference picture management as an optimization problem and approximated its optimal solution. We also investigated the associated quality adjustment according to the reference structure. With the method introduced in this paper, up to 40% and average 9% bit savings can be achieved. For low delay applications, we further developed fast algorithms to reduce the delay and complexity, which also showed significant improvements on coding efficiency. The proposed algorithms can be applied to not only HEVC, but also other video coding schemes where the flexible referencing structure is supported.

The main objective of this paper was to improve coding efficiency by reference picture management. There are still various other related topics to further investigate. For example, when rate control is taken into consideration, the number of bits for each picture will be another constraint. The other case is that a good reference structure can provide a way to reduce the distortion of the decoded video due to transmission errors. How to improve the performance while considering other constraints is interesting and practical.

In our iterative solution, the reference management step and the quality-adjustment step are performed iteratively. Considering the complexity, we did not optimize the problem within one step in this paper. However, a joint optimization may reduce the risk of reaching a locally optimal result. How to obtain the optimal solution efficiently is still needed to be investigated.

Another interesting topic is to investigate how different encoding orders influence the coding efficiency. In this paper, the encoding order is assumed to be predetermined. However, changing coding order may bring additional gains. How to optimize the encoding order under some constraints, such as a given maximum latency, is still worth further investigating. More gains may be achieved if encoding order, reference frame management, and quality adjustment can jointly be optimized.

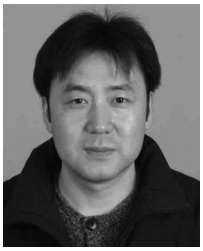
Acknowledgment

The authors would like to thank the anonymous reviewers for their thoughtful comments and suggestions that improved the quality of this paper. They are also grateful to Dr. F. Wu for related discussions.

References

[1] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[2] *Joint Call for Proposals on Video Compression Technology*, document VCEG-AM91, ITU-T Q6/16 Visual Coding and ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, Kyoto, Japan, Jan. 2010.
[3] G. Sullivan and J.-R. Ohm, "Recent developments in standardization of High Efficiency Video Coding (HEVC)," in *Proc. 33th SPIE Appl. Digit. Image*, vol. 7798. Aug. 2010, pp. 77980V–77980V-7.
[4] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1648–1667, Dec. 2012.
[5] B. Girod, "Efficiency analysis of multihypothesis motion-compensated prediction for video coding," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 173–183, Feb. 2000.

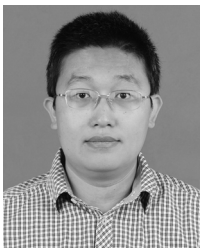
[6] T. Wiegand and B. Girod, *Multi-Frame Motion-Compensated Prediction for Video Transmission* (Kluwer International Series in Engineering and Computer Science). Dordrecht, The Netherlands: Kluwer Academic, 2001.
[7] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 9, no. 2, pp. 70–84, Feb. 1999.
[8] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical b pictures and MCTF," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2006, pp. 1929–1932.
[9] F. Bossen, *Common Test Conditions and Software Reference Configurations*, document JCTVC-E700, Geneva, Switzerland, Mar. 2011.
[10] Y. Su and M.-T. Sun, "Fast multiple reference frame motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 16, no. 3, pp. 447–452, Mar. 2006.
[11] T.-Y. Kuo and H.-J. Lu, "Efficient reference frame selector for H.264," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 18, no. 3, pp. 400–405, Mar. 2008.
[12] V. Chellappa, P. Cosman, and G. Voelker, "Dual frame motion compensation for a rate switching network," in *Proc. 37th Asilomar Conf. Signals Syst. Comput.*, vol. 2. Nov. 2003, pp. 1539–1543.
[13] V. Chellappa, P. Cosman, and G. Voelker, "Dual frame motion compensation with uneven quality assignment," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 18, no. 2, pp. 249–256, Feb. 2008.
[14] M. Tiwari and P. Cosman, "Selection of long-term reference frames in dual-frame video coding using simulated annealing," *IEEE Signal Process. Lett.*, vol. 15, no. 1, pp. 249–252, 2008.
[15] D. Liu, D. Zhao, X. Ji, and W. Gao, "Dual frame motion compensation with optimal long-term reference frame selection and bit allocation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 3, pp. 325–339, Mar. 2010.
[16] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa, "Video textures," in *Proc. 27th Annu. Conf. Comput. Graph. Interactive Tech. (SIGGRAPH)*, 2000, pp. 489–498.
[17] HM. *HEVC Test Model* [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/
[18] JM. *H.264/AVC Reference Software* [Online]. Available: http://iphome.hhi.de/suehring/tml/download/
[19] B. Li, J. Xu, H. Li, and F. Wu, "Optimized reference frame selection for video coding by cloud," in *Proc. 13th IEEE Int. Workshop MMSP*, Oct. 2011, pp. 1–5.
[20] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec.H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1, edition 5.0, Mar. 2010.
[21] R. Sjöberg and J. Samuelsson, *Absolute Signaling of Reference Pictures*, document JCTVC-F493, Turin, Italy, Jul. 2011.
[22] R. Sjöberg, R. Flynn, Y. Chen, Y.-K. Wang, T. K. Tan, and W. K. Wan, *JCT-VC AHG Report: Reference Picture Buffering and List Construction (AHG21)*, document JCTVC-G021, Geneva, Switzerland, Nov. 2011.
[23] B. Li, J. Xu, F. Wu, and H. Li, *Encoding Optimization to Improve Coding Efficiency for Low Delay Cases*, document JCTVC-F701, Turin, Italy, Jul. 2011.
[24] B. Li, G. J. Sullivan, and J. Xu, "Compression performance of High Efficiency Video Coding (HEVC) working draft 4," in *Proc. IEEE ISCAS*, May 2012, pp. 886–889.
[25] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.
[26] G. Bjontegaard, *Calculation of Average PSNR Differences Between Rd-Curves*, document VCEG-M33, Austin, TX, Apr. 2001.
[27] R. Gray and D. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2325–2383, Oct. 1998.
[28] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.
[29] L. ChongSoon and N. S. M. Thet, *Reference Lists for Low Delay Settings*, document JCTVC-F433, Turin, Italy, Jul. 2011.
[30] G. Sullivan and J.-R. Ohm, *Meeting Report of the Fourth Meeting of the Joint Collaborative Team on Video Coding (JCT-VC)*, document JCTVC-D500, Daegu, Korea, Jan. 2011.
[31] K. Seshadrinathan, R. Soundararajan, A. Bovik, and L. Cormack, "Study of subjective and objective quality assessment of video," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1427–1441, Jun. 2010.
[32] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron. Lett.*, vol. 44, no. 19, pp. 800–801, 2008.



Houqiang Li (M'10) received the B.S., M.Eng., and Ph.D. degrees in electronic engineering from the University of Science and Technology of China (USTC), Hefei, Anhui, China, in 1992, 1997, and 2000, respectively.

He is currently a Professor with the Department of Electronic Engineering and Information Science, USTC. He has authored or co-authored over 90 papers. His current research interests include video coding and communication, multimedia search, and image and video analysis.

Dr. Li is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and in the Editorial Board of the *Journal of Multimedia*. He has served as the Program Co-Chair and the Track/Session Chair for over ten international conferences. He was the recipient of the Best Paper Award for the International Conference on Mobile and Ubiquitous Multimedia from ACM in 2011, and a Senior Author of the Best Student Paper of the 5th International Mobile Multimedia Communications Conference in 2009.



Bin Li received the B.S. degree in electronic engineering from the University of Science and Technology of China (USTC), Hefei, Anhui, China, in 2008. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering and Information Science, USTC.

He has almost 20 technical proposals that have been adopted by High Efficiency Video Coding standards. His current research interests include video coding, processing, and communication.

Mr. Li has been an active contributor to ISO/MPEG and ITU-T video coding standards. He was the recipient of the Best Paper Award for the International Conference on Mobile and Ubiquitous Multimedia from ACM in 2011.

Jizheng Xu (M'07–SM'10) received the B.S. and M.S. degrees in computer science from the University of Science and Technology of China (USTC), Hefei, Anhui, China, and the Ph.D. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China.

In 2003, he joined Microsoft Research Asia (MSRA), Beijing, China, initially as an Assistant Researcher, where he is currently a Lead Researcher. He has authored or co-authored over 80 papers presented in conferences and refereed journals. He holds over 20 granted or pending U.S. patents in the areas of image and video coding. His current research interests include image and video representation, media compression, and communication.

Dr. Xu has been an active contributor to ISO/MPEG and ITU-T video coding standards. He has over 20 technical proposals being adopted by H.264/AVC, H.264/AVC scalable extension, and High Efficiency Video Coding standards. He was the Chair and Co-Chair of the ad hoc group of exploration on wavelet video coding of MPEG from January 2005 to April 2006, the ad hoc group on screen content coding, the ad hoc group on parsing robustness, and the ad hoc group on lossless coding of JCT-VC in 2010 and 2011. He has co-organized and co-chaired special sessions on scalable video coding, directional transform, and high-quality video coding at various conferences.