Fast Bi-Directional Prediction Selection in H.264/MPEG-4 AVC Temporal Scalable Video Coding

Hung-Chih Lin, Hsueh-Ming Hang, *Fellow, IEEE*, and Wen-Hsiao Peng

***Abstract*—In this paper, we propose a fast algorithm that efficiently selects the temporal prediction type for the dyadic hierarchical-B prediction structure in the H.264/MPEG-4 temporal scalable video coding (SVC). We make use of the strong correlations in prediction type inheritance to eliminate the superfluous computations for the bi-directional (BI) prediction in the finer partitions, $16 \times 8/8 \times 16/8 \times 8$, by referring to the best temporal prediction type of 16×16. In addition, we carefully examine the relationship in motion bit-rate costs and distortions between the BI and the uni-directional temporal prediction types. As a result, we construct a set of adaptive thresholds to remove the unnecessary BI calculations. Moreover, for the block partitions smaller than 8×8, either the forward prediction (FW) or the backward prediction (BW) is skipped based upon the information of their 8×8 partitions. Hence, the proposed schemes can efficiently reduce the extensive computational burden in calculating the BI prediction. As compared to the JSVM 9.11 software, our method saves the encoding time from 48% to 67% for a large variety of test videos over a wide range of coding bit-rates and has only a minor coding performance loss.**

***Index Terms*—Bi-directional prediction, bi-directionally predictive frame, encoder optimization, hierarchical prediction structure, H.264/MPEG-4 AVC scalable video coding, temporal scalability.**

I. Introduction

FOR the need of delivering digital video over heterogeneous network and interacting with devices of different capacities, the scalable bit-stream approach is developed to meet the demands of such applications [1]. A desirable video compression scheme should offer both scalability feature as well as high coding efficiency. The Joint Video Team (JVT) thus recently, based on the latest international video coding standard H.264/MPEG-4 AVC [2], [3], standardized the scalable video coding (SVC) extension [4], [5]. JVT also developed a Joint Scalable Video Model [6] (referred hereafter as JSVM) as an example of the SVC encoder that can simultaneously support temporal, spatial, and quality combined scalability within a single bit-stream. The SVC encoder implements the temporal scalability by using the *hierarchical prediction structure* [7], [8]. Moreover, the hierarchical prediction structure not only provides the temporal scalability, it also significantly outperforms the conventional IPPP/IBBP coding structures in the rate-distortion (RD) performance.

However, this improved coding efficiency pays a penalty of huge amount of increased computations. They mainly come from the two nested exhaustive motion vector (MV) searches in the hierarchical-B frames at the temporal enhancement layers. There are three temporal prediction types in each inter prediction mode. They are the two uni-directional predictions, forward prediction (FW) and backward prediction (BW), and the time-consuming bi-directional prediction (BI). To show the good encoding efficiency, JSVM [6] first runs all the motion estimation procedures associated with three prediction types. The best prediction type of each inter mode is selected by competing their RD costs. After that, the best mode partition is obtained by fully searching for the one with the minimum RD cost. As a result, the massive computations at the temporal enhancement layers are mostly due to the searching process for two sets of encoding tool parameters: one is the selection of mode partitions, and the other, which is our focus in this study, is the temporal prediction types.

The concept of the multiple reference frames included in H.264/MPEG-4 AVC [9] is potentially able to improve the prediction accuracy. Apparently, its complexity is linearly proportional to the number of used reference frames. Hence, a number of prior studies [10]–[17] have tried to reduce these extra computations. The diversity of the four MVs, obtained from the Inter8 $\times$ 8 partition mode, determines whether the reference frames prior to the nearest one should be included in the candidate set [10]. In [11], the MV decimal value (fractional-pel) is used to estimate which reference frame the moving object locates. For example, a half-pel MV implies the object locates more exactly two frames ahead. Therefore, for a given MV, we can select the closest reference frame as the candidate. Furthermore, the motion continuity is studied to conjecture the initial MV search location. The initial MV can be estimated by the weighted sum [12] or the median [13] of the MVs of the nearby reference blocks. Then, the search range is significantly narrowed down. To provide a more accurate initial guess, the approach proposed in [14] combines the block MVs from several previous references. On the other hand, in [15]–[17] the selection of multiple reference frame is early chosen by a set

Manuscript received August 25, 2010; revised January 20, 2011 and April 28, 2011; accepted May 02, 2011. Date of publication May 19, 2011; date of current version November 18, 2011. This work was supported in part by the NSC, Taiwan under Grants 98-2622-8-009-011 and 98-2219-E-009-076. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhou Wang.

H.-C. Lin is with MediaTek, Inc., Hsinchu 300, Taiwan (e-mail: huchlin@gmail.com).

H.-M. Hang is with the Department of Electronics Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan (e-mail: hmhang@mail.nctu.edu.tw).

W.-H. Peng is with the Department of Computer Science, National Chiao-Tung University, Hsinchu 300, Taiwan (e-mail: pawn@mail.si2lab.org).

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

Digital Object Identifier 10.1109/TIP.2011.2156802

of termination conditions, such as all-zero block detection, the power of prediction error, and the optimal block partition in the (first) previous frame. However, Huang *et al.* [16] empirically show that the coding gain provided by the multiple reference frames highly depends on the video contents, not on the number of searched references, which is also theoretically justified in [17]. Thus, the multiple reference frames tool does not usually have noticeable improvement (say, more than 1 dB) in the RD measure, but it requires a huge amount of computations.

On the other hand, a fairly large body of literature has been proposed on the complexity reduction of the H.264/MPEG-4 AVC coder, based on the estimated RD cost as thresholds and/or the known mode selection. In [18], the candidate modes and the RD cost thresholds are given by the temporally and spatially neighboring area. Furthermore, the transformed residuals and the corresponding coding bits have a highly linear correlation [19]. Based on the non-zero quantized transform coefficients, the proposed schemes [19], [20] construct an accurate RD estimator to avoid calculating the true RD values during the mode decision process. Furthermore, a modified Langrangian cost function [21], composed of the distortion, the motions, and the required header, eliminates entirely the entropy coding in the decision process. Another popular approach in fast mode selection is the so-called early termination [22]–[28]. For example, the sophisticated mode search can be eased by constructing hierarchically multiple termination criteria covering large to small block partitions [22]. In [23] and [24], they theoretically study the sufficient conditions in detecting all-zero blocks to reject small partition modes. In addition, several researches use the spatio-temporal motion characteristics to prioritize the candidate modes, such as moving trajectory [25], [26] and spatial motion homogeneity [27], [28]. All the above schemes are applicable to the low-delay (IPPP/IBP/IBBP) coding structures, but few focus on the superior hierarchical prediction in the SVC temporal scalability. Moreover, the AVC-based fast algorithms could not be well extended to SVC, because the correlations between the current frame and its references are often not sufficiently strong and reliable when they are calculated at the low temporal layers. In [29], the characteristics of low/high-motion areas at low temporal layers are employed to select the block mode at high temporal layers. Lee *et al.* [30] make use of the statistical hypothesis testing to conditionally skip the partitions smaller than 16×16. However, only the encoding parameter in the mode partition is considered in constructing their fast algorithms.

Up to now, very few researchers pay attention to the selection of temporal prediction types (FW, BW, and BI). Although these three temporal prediction types can provide highly efficient compression, they cost more than tripling the motion search calculation in the IPPP coding structure. Therefore, this paper aims to design a fast temporal prediction selection algorithm for the dyadic hierarchical-B prediction structure in SVC. To achieve this goal, we statistically analyze the correlations of temporal prediction types in large partitions and show that the BI type has limited coding benefits in small partitions [31]. The correlations of motion bit-rates among three temporal prediction types are examined and they are formulated by a simple linear model. Additionally, the relationships among the distortions are also investigated [32] and the prediction error in the uni-directional temporal predictions tends to be a jointly Laplacian distribution, verified by the goodness-of-fit tests. Hence, based on these observations, we propose a novel scheme that avoids unnecessarily massive BI evaluations through the temporal prediction inheritance and the adaptive thresholds in the hierarchical-B prediction structure of SVC. On the average, our approaches can provide up to 67% overall encoder time saving over JSVM 9.11 [6], which is equivalent to three times faster in the encoding process. Some initial ideas of this paper were presented separately [31], [32], but many new issues arise due to merging two separate techniques together. Thus, after solving these new issues, the revised and integrated scheme is first-time described in this report.

The rest of this paper is organized as follows. Section II contains a brief review of the hierarchical prediction structure and its decision process of temporal prediction type in the JSVM encoder [6]. Its dramatically encoding complexity is also revealed, as compared to the IPPP coding structure. Section III summaries our observations on the correlations among three temporal prediction types. Based on these analyses, Section IV presents our fast bi-directional prediction selection algorithm. In Section V, our proposed scheme is compared with JSVM 9.11 [6] and the state-of-the-art algorithms [29], [30] in terms of complexity reduction and RD performance. Lastly, Section VI ends with a summary of our work.

II. Hierarchical Prediction Structure in SVC Temporal Scalability

To have a better understanding of our coding algorithms, this section explains the basic concepts of hierarchical prediction structure and it briefly reviews the decision process on the temporal prediction type in the SVC temporal enhancement layers. In addition, we also examine its complexity based on the empirical data. Some degree of familiarity with H.264/MPEG-4 AVC is assumed herein. The reader is referred to the overview papers [3], [5] for details of AVC and its scalable extension.

A. Basic Concepts

Currently, the temporal scalability in JSVM [6] is realized by using dyadic hierarchical prediction [7], [8]. A set of successive images is grouped into the so-called *Group of Pictures* (GOP), of which the size is typically power of two. A temporal scalable bit-stream is composed of one temporal base layer and one or more temporal enhancement layers. For example, if the GOP size is 2^n, then its structure consists of the temporal base layer T_0 and n temporal enhancement layers, denoted as $T_1, T_2, \ldots, T_n$. The last frame of each GOP is an *anchor frame*. The anchor frames form the temporal base layer T_0; they are coded as either I- or P-frame. The remaining frames, all located at the temporal enhancement layers, are coded as the hierarchical-B frames. Moreover, the BI operation is restricted to take the weighted sum of one preceding and one succeeding reference frames for prediction. Such a coding mechanism is referred to as the *hierarchical-B prediction structure*. Fig. 1 demonstrates an example of hierarchical-B prediction structure with $\mathrm{GOP\ size} = 8$, where the notation $\mathcal{P}_\ell$ in frame type denotes the frame is $\mathcal{P}$-type ($\mathcal{P} \in \{\mathrm{I}, \mathrm{P}, \mathrm{B}\}$) and located at temporal layer ℓ. Moreover, a B-frame takes a past frame and/or a future frame as its reference(s), both of which are equally distant from this B-frame.

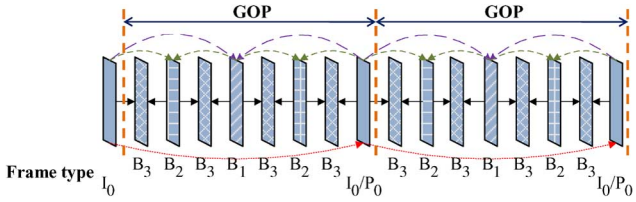


Fig. 1. Hierarchical-B prediction structure (GOP size = 8).

Temporal scalability made by the dyadic hierarchical-B prediction provides a high compression quality. In comparison to the commonly adopted IBBP and IPPP coding structures, the Y-PSNR can be averagely improved by at least 1.0 dB and 2.0 dB, respectively. Moreover, in this structure, experiments show that each reference list containing only one reference frame is sufficient. Empirically, the maximum coding efficiency occurs when the GOP size is between 8 and 32, as reported in [5].

B. Temporal Prediction Type Selection

To ensure the high compression performance, the SVC encoder has to choose the most suitable block partition (mode) that leads to the optimal tradeoff between distortion $\mathcal{D}$ and bit-rate R by the Lagrange multiplier method [33]–[35]. Typically, two major nested processes are fully checked to obtain the optimal coding parameters: one is the block mode and the other is the temporal prediction type.

If a specific block mode $\mathcal{M}$ with $M \times N$ size (in pixel) is applied to an MB, there are $16 \times 16/M \times N$ sub-blocks inside this MB. Each sub-block needs to find its best temporal prediction type $\mathcal{T}^*$. This is done typically by performing the rate-constrained motion estimation that minimizes the RD cost $J_{\mathcal{T}}$:

$$\begin{aligned}\mathcal{T}^* &= \arg \min_{\mathcal{T} \in \mathcal{T}^{\mathcal{U}} \cup \{\mathrm{BI}\}} \{J_{\mathcal{T}}\} \\ &= \arg \min_{\mathcal{T} \in \mathcal{T}^{\mathcal{U}} \cup \{\mathrm{BI}\}} \{\mathcal{D}_{\mathcal{T}}(\mathbf{mv}) + \lambda_{\mathrm{MOTION}} \\ &\qquad \times R_{\mathcal{T}}(\mathbf{mv} - \mathbf{mv}_p)\} \end{aligned} \quad (1)$$

where

$\mathcal{T}^{\mathcal{U}}$	denotes the set of the uni-directional temporal predictions {FW, BW},
$\lambda_{\mathrm{MOTION}}$	is the Lagrange multiplier for the rate-constrained motion estimation (a commonly used formulation is $\lambda_{\mathrm{MOTION}} = \sqrt{0.85 \times 2^{(Qp-12)/3}}$ [33]–[35]),
$\mathcal{D}_{\mathcal{T}}(\mathbf{mv})$	is the pixel distortion, usually, the sum of absolute difference (SAD) using two MVs given by

$$\begin{aligned}\mathcal{D}_{\mathcal{T}}(\mathbf{mv}) = \sum_{\mathbf{x} \in \mathbf{N}} |f(\mathbf{x}) - (1-\eta) f'_{\mathrm{FW}}(\mathbf{x} - \mathbf{mx}_0) \\ - \eta f'_{\mathrm{BW}}(\mathbf{x} - \mathbf{mx}_1)| \end{aligned} \quad (2)$$

$\mathbf{N} = (M, N)$	is the sub-block (pixel set) specified by mode $\mathcal{M}$,
$\mathbf{mv}_p$	is the *predictive motion vector* (pmv) generated by an MV predictor, (Typically, it is the median of the neighboring MVs.)
$R_{\mathcal{T}}(\mathbf{mv} - \mathbf{mv}_p)$	denotes the number of bits representing the difference between $\mathbf{mv}_p$ and the motion vector ($\mathbf{mv} = \mathbf{mv}_0$ and/or $\mathbf{mv}_1$),
$f(\mathbf{x})$	is the current frame pixel value,
$f'_{\mathrm{FW}}(\mathbf{x})$ and $f'_{\mathrm{BW}}(\mathbf{x})$	are the pixel values of the forwardly and the backwardly reconstructed frames, respectively.

The JSVM encoder [6] splits the selecting process of temporal prediction types into two stages, consisting of two uni-directional predictions (FW and BW) and one bi-directional prediction (BI).

- First stage (FW and BW): The motion-compensated prediction attempts to find the motions $\mathbf{mv}_{\mathrm{FW}}$ and $\mathbf{mv}_{\mathrm{BW}}$ by minimizing the costs J_{FW} and J_{BW}, separately. In computing (2), we first set $\eta = 0$ and thus $\mathbf{mv}_0 = \mathbf{mv}_{\mathrm{FW}}$ for finding the FW MV; and then set $\eta = 1$ and thus $\mathbf{mv}_1 = \mathbf{mv}_{\mathrm{BW}}$ in the case of BW.
- Second stage (BI): As mentioned in [36], the motion vectors found by FW and BW are sub-optimal for BI. Therefore, the JSVM encoder [6] adopts an iterative approach to find the optimal MV pair for BI, denoted as $\mathbf{mv}_{\mathrm{BI}\leftarrow\mathrm{FW}}$ and $\mathbf{mv}_{\mathrm{BI}\leftarrow\mathrm{BW}}$, by taking $\mathbf{mv}_{\mathrm{FW}}$ and $\mathbf{mv}_{\mathrm{BW}}$ as the initial search points. For example, the iterative process first locally refines $\mathbf{mv}_{\mathrm{FW}}$ with $\mathbf{mv}_{\mathrm{BW}}$ fixed. Then, $\mathbf{mv}_{\mathrm{BW}}$ is refined with $\mathbf{mv}_{\mathrm{FW}}$ fixed in the next iteration. The detailed analysis of this iterative approach is referred to [36]. During each iteration, the JSVM encoder performs the local refinement by an exhaustive search with a much smaller search range than that in $\mathcal{T}^{\mathcal{U}}$. Furthermore, the iterative process is terminated if the J_{BI} in the current iteration is not better than that in the previous one. On the average, it results in about 2.3 BI iterations per MB (or sub-block). Note that the distortion in (2) computes the difference between the current MB and the average of two reference MBs by setting $\eta = 1/2$, $\mathbf{mv}_0 = \mathbf{mv}_{\mathrm{BI}\leftarrow\mathrm{FW}}$, and $\mathbf{mv}_1 = \mathbf{mv}_{\mathrm{BI}\leftarrow\mathrm{BW}}$.

Finally, three sets of distortions $\mathcal{D}_{\mathcal{T}}$ and motion bit-rate cost $\mathcal{R}_{\mathcal{T}} \triangleq \lambda_{\mathrm{MOTION}} \times R_{\mathcal{T}}$ are collected and compared to produce the best RD costs $J_{\mathcal{T}}$ and $\mathcal{T}^*$. For example, there are two sub-blocks in the block mode 16×8. Each of them has its best temporal prediction type and, say, one is FW and the other is BI. Then, each of them performs the reconstruction process to produce the RD performance of this 16×8 block mode. Next, we pick up the 8×8 mode and there are thus 4 sub-blocks. The MV search process is repeated for each sub-block and at the end we obtain the RD cost of this 8×8 mode. We try all possible modes and finally, we compare all these candidate modes and select the least RD cost block mode $\mathcal{M}^*$.

TABLE I
COMPLEXITY RATIO COMPARED TO IPPP (FOR A GOP)

Test Sequence		Prediction Structure			
		Hierarchical-B		Hierarchical-B with $\mathcal{T}^{\mathcal{U}}$ only	
		GOP = 8	GOP = 16	GOP = 8	GOP = 16
CIF	FOOTBALL	3.43	3.68	1.81	1.92
	FOREMAN	3.36	3.60	1.50	1.59
	MOBILE	3.23	3.44	1.35	1.45
4CIF	CITY	3.23	3.45	1.43	1.51
	CREW	3.25	3.47	1.52	1.61
	SOCCER	3.35	3.59	1.55	1.65
HD (720p)	MOBCAL	3.12	3.32	1.34	1.40
	SHIELDS	3.19	3.39	1.45	1.52
	STOCKHOLM	3.20	3.40	1.39	1.46
AVG.		**3.3**	**3.5**	**1.5**	**1.6**

Average CPU time using $Qp = 40, 35, 30, 25$; JSVM 9.11 [6] with the encoding parameters in Table VII.

Platform: Athlon 3800+, 64-bit, dual-core processors, 2.0 GB RAM with Windows XP

C. Complexity Analysis

As discussed earlier, by allowing three temporal prediction types (FW, BW, and BI), the dyadic hierarchical-B prediction offers a high compression efficiency. However, its accompanying penalty is the very large amount of computation. Our statistics, collected from 9 sequences with four selected Qp values, show very intensive computations in two prediction structures; one is the hierarchical-B prediction structure (FW, BW and BI), and the other is $\mathcal{T}^{\mathcal{U}}$ only. As listed in Table I, the complexity increases as the GOP size goes up. The increased computation is related to the percentage of frames (inside a GOP) that use multiple temporal prediction types. For example, there are $((2^n - 1)/2^n) \times 100\%$ of frames (in percentage) also use BW (and BI) for GOP $= 2^n$. Therefore, a hierarchical prediction structure using a large GOP size results in more computations.

On the average, as compared to the IPPP coding structure, the hierarchical-B prediction structure introduces additional 240% computations and the uni-prediction structure ($\mathcal{T}^{\mathcal{U}}$ only) needs only about extra 55% encoding time. More specifically, the complexity ratio of $\mathcal{T}^{\mathcal{U}}$ and the BI is as follows:

$$(\mathrm{FW} + \mathrm{BW}) : \mathrm{BI} \cong \begin{cases} 1 : 1.26, & Qp = 40 \\ 1 : 1.23, & Qp = 35 \\ 1 : 1.21, & Qp = 30 \\ 1 : 1.19, & Qp = 25. \end{cases}$$

Although the complexity of the uni-prediction structure ($\mathcal{T}^{\mathcal{U}}$ only) is much lower than that of hierarchical-B prediction, the $\mathcal{T}^{\mathcal{U}}$ only structure has a coding penalty of about 0.2 dB PSNR drop and 5% bit-rate increase. Thus, the BI type prediction is helpful in improving the RD performance but it consumes more than half of the encoding time. In this study we propose efficient methods to eliminate the unnecessarily computational load in calculating BI and to maintain a similar level of coded picture quality at the same time.

III. STATISTICAL CHARACTERISTICS OF TEMPORAL PREDICTION AT TEMPORAL ENHANCEMENT LAYERS

In this section, we investigate the statistical correlations of three temporal prediction types (FW, BW, and BI) at the SVC temporal enhancement layers. In order to study these correlations, our strategy is to collect the coding information by applying the exhaustive search on uni- and bi-directional predictions before the optimal partition size is determined. In Section III-A, we examine the prediction type distributions and their inheritances in the hierarchical blocks from large partitions to small ones. Then, Section III-B analyzes the relative coding efficiency contributed by the BI. In terms of the RD costs and the motion bit-rate costs, the last subsection explores their correlations between $\mathcal{T}^{\mathcal{U}}$ and BI. These statistical analyses are conducted based on encoding one temporal base layer T_0 with four temporal enhancement layers $T_1 \sim T_4$; that is, the GOP size is 16. To evaluate the Qp impact, two Qp values, 30 and 40, are tested in experiments. The training set contains nine MPEG test videos

$$\begin{aligned} &\mathrm{QCIF} : \mathrm{BUS},\ \mathrm{COASTGUARD},\ \mathrm{STEFAN}; \\ &\mathrm{CIF} : \mathrm{FOOTBAL},\ \mathrm{FOREMAN},\ \mathrm{MOBILE}; \\ &4\mathrm{CIF} : \mathrm{CITY},\ \mathrm{CREW},\ \mathrm{SOCCER}. \end{aligned}$$

Due to limited space, we mainly show the statistical data of the CIF videos and the mean values ("MEAN"), which represent the average behavior of these nine videos. In addition, the B-Skip mode calculates the predicted motion vectors to estimate its cost without actually performing the motion estimation operation. The estimated cost is used to compete with the other inter modes in making decision (on the optimal block size). Therefore, the B-Skip statistics are excluded from the following analysis.

A. Inheritance of Temporal Prediction Types

In this subsection, we collect the probability distributions of various temporal prediction types.

1) Prediction Type Distributions: We first gather the probability distributions of temporal prediction types used in several distinct block partitions at various temporal enhancement layers and at different Qp values. In any temporal layer of the hierarchical-B prediction structure, the temporally forward and backward reference frames have equal distance away from the current frame. If the objects move at a constant speed, the current MB can find its shifted version in either the forward reference or the backward reference. It implies that the selections of FW and BW are nearly equally likely, as shown in Fig. 2 for, particularly, the MOBILE sequence.

The selection of BI is highly dependent on the video content and the Qp values, especially when the block partitions are larger than 8×8. Also, BI is selected more often at the high bit-rates (small Qp) because the encoder has sufficient bits to reduce the reconstruction distortion. In the MOBILE sequence, the BI probability of the 16×16 partition reaches about 80% at the low temporal enhancement layers, but its probability decreases to 20% or less at the high temporal layers. In Fig. 2(c), BI is generally favored at large partitions (16×16 or 16×8) because BI offers more accurate motion compensation at a small MV bit-rate overhead. Clearly, distant reference frames

TABLE II
CONDITIONAL PROBABILITIES OF $p_{(16\times8|16\times16)\notin\mathrm{BI}}$, AND $p_{(8\times8|16\times16)\notin\mathrm{BI}}$

Test Sequence	Qp	$p_{(16\times8\|16\times16)\notin\mathrm{BI}}$				$p_{(8\times16\|16\times16)\notin\mathrm{BI}}$				$p_{(8\times8\|16\times16)\notin\mathrm{BI}}$			
		T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4
FOOTBALL	40	0.94	0.95	0.96	0.97	0.94	0.95	0.96	0.97	0.96	0.97	0.98	0.98
FOREMAN		0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.99	0.99	0.99	0.99
MOBILE		0.93	0.96	0.97	0.98	0.93	0.95	0.97	0.97	0.97	0.98	0.99	0.99
MEAN		**0.96**	**0.97**	**0.98**	**0.98**	**0.95**	**0.96**	**0.97**	**0.98**	**0.97**	**0.98**	**0.99**	**0.99**
FOOTBALL	30	0.85	0.89	0.90	0.93	0.84	0.88	0.90	0.92	0.92	0.93	0.94	0.96
FOREMAN		0.94	0.95	0.97	0.97	0.93	0.95	0.96	0.97	0.97	0.98	0.98	0.99
MOBILE		0.82	0.86	0.88	0.91	0.80	0.83	0.88	0.89	0.85	0.90	0.92	0.94
MEAN		**0.88**	**0.91**	**0.93**	**0.96**	**0.87**	**0.90**	**0.92**	**0.94**	**0.91**	**0.94**	**0.95**	**0.97**

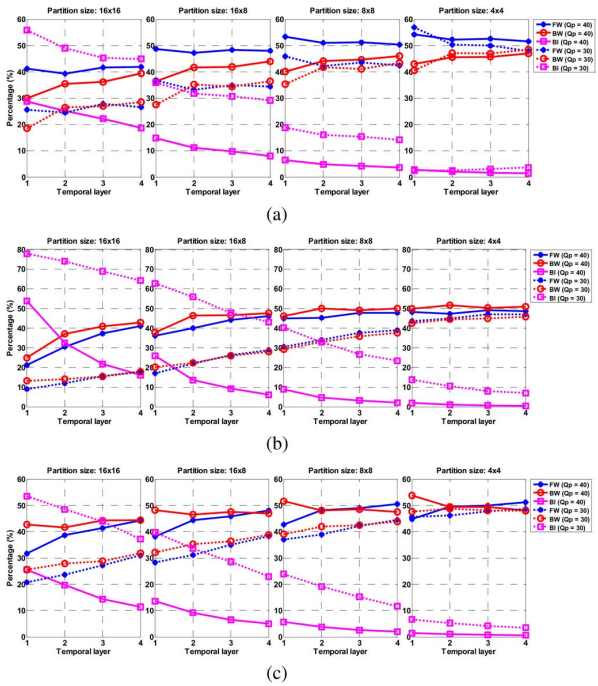


Fig. 2. Distribution of temporal prediction types (FW, BW, and BI). (a) FOOTBALL; (b) MOBILE; (c) MEAN.

reduce the motion compensation effectiveness. Therefore, BI percentage goes down drastically at higher temporal layers. On the other hand, BI needs to transmit more MVs, twice as many as those of the FW/BW. If the reduced distortion provided by BI cannot compensate for the increased motion bit-rate cost, the BI type is not chosen. This is particularly true in the case of small blocks (4×4, 8×8). Therefore, at the same temporal enhancement layer, larger partitions prefer BI, especially in the complex-textured sequence MOBILE.

In summary, BI benefits the 16×16 and $16\times8/8\times16$ partitions at the lower temporal enhancement layers, but for the small partitions from 8×8 down to 4×4, BI is seldom selected. This observation does not seem to be strongly affected by the video contents and the coding bit-rates.

2) Elimination of BI for Large Partitions: Our second observation focuses on the use of BI in the $16\times8/8\times16/8\times8$ partitions. As discussed in Section III-A-1, the BI probability in these partitions is much smaller than that in the 16×16 block size. That is, for instance, $\mathbf{Pr}\{16\times8\in\mathrm{BI}\}$ is less than $\mathbf{Pr}\{16\times16\in\mathrm{BI}\}$, which implies $\mathbf{Pr}\{16\times8\notin\mathrm{BI}\}>\mathbf{Pr}\{16\times16\notin\mathrm{BI}\}$. In order to find out whether or not these two groups $\{16\times16\notin\mathrm{BI}\}$ and $\{16\times8\notin\mathrm{BI}\}$ overlap with each other, we consider three conditional probabilities defined below:

$$p_{(16\times8|16\times16)\notin\mathrm{BI}}\triangleq\mathbf{Pr}\{16\times8\notin\mathrm{BI}|16\times16\notin\mathrm{BI}\},$$
$$p_{(8\times16|16\times16)\notin\mathrm{BI}}\triangleq\mathbf{Pr}\{8\times16\notin\mathrm{BI}|16\times16\notin\mathrm{BI}\},\ \text{and}$$
$$p_{(8\times8|16\times16)\notin\mathrm{BI}}\triangleq\mathbf{Pr}\{8\times8\notin\mathrm{BI}|16\times16\notin\mathrm{BI}\}.$$

In Table II, these three conditional probabilities are higher than 80% in all cases and are about 95% on average. Moreover, their total probability is very close to one at higher temporal enhancement layers. This strong correlation indicates that the uni-direction prediction types are inheritable from the 16×16 partition to $16\times8/8\times16/8\times8$ partitions. Thus, the information of the prior evaluation on 16×16 BI can provide a very reliable estimate to the use of BI for the $16\times8/8\times16/8\times8$ partitions at both low and high bit-rates. We can quite confidently eliminate the use of BI in those partitions.

3) Consistency of FW and BW in Small Partitions: We now look into the block partitions smaller than 8×8. We find that we only need to evaluate the set $\mathcal{T}^{\mathcal{U}}$; also, the temporal prediction types of the 8×8 partition are strongly correlated to those of the 4×4 partition. As discussed in Section III-A-1, the probabilities of using BI for the 8×8 and the smaller partitions are often less than 20% and 10%, respectively. We collect the following two conditional probabilities of using FW and BW types. One is defined by $p_{(4\times4|8\times8)\in\mathrm{FW}}\triangleq\mathbf{Pr}\{4\times4\in\mathrm{FW}|8\times8\in\mathrm{FW}\}$, which is equivalent to $1-\mathbf{Pr}\{4\times4\in\mathrm{BI}|8\times8\in\mathrm{FW}\}-\mathbf{Pr}\{4\times4\in\mathrm{BW}|8\times8\in\mathrm{FW}\}$.

Typically, the $\mathbf{Pr}\{4\times4\in\mathrm{BI}|8\times8\in\mathrm{FW}\}$ term is less than 2% in our collected data. The probability $p_{(4\times4|8\times8)\in\mathrm{FW}}$ can thus be approximated by $\mathbf{Pr}\{4\times4\notin\mathrm{BW}|8\times8\in\mathrm{FW}\}$. Similarly defined $p_{(4\times4|8\times8)\in\mathrm{BW}}$ is close to $\mathbf{Pr}\{4\times4\notin\mathrm{FW}|8\times8\in\mathrm{BW}\}$. Experiments show that the approximated values of $p_{(4\times4|8\times8)\in\mathrm{FW}}$ and $p_{(4\times4|8\times8)\in\mathrm{BW}}$ are fairly close to data in Table III. Moreover, these two conditional probabilities slightly increase at higher temporal enhancement layers, except for the MOBILE sequence with $Qp=30$, of which the correlations are rather similar for all temporal enhancement layers. Averagely, the consistency in selecting the same prediction direction can be up to 90%. Thus, the 8×8 prediction mode information serves as a good reference to its smaller partitions.

B. Rate-Distortion Contribution by BI

In this subsection, we address the relative RD improvement offered by BI in different block modes. As investigated before, the hierarchical-B prediction structure takes the advantage of

TABLE III
CONDITIONAL PROBABILITIES OF $p_{(4\times4|8\times8)\in\mathrm{FW}}$ AND $p_{(4\times4|8\times8)\in\mathrm{BW}}$

Test Sequence	Qp	$p_{(4\times4\|8\times8)\in\mathrm{FW}}$				$p_{(4\times4\|8\times8)\in\mathrm{BW}}$			
		T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4
FOOTBALL	40	0.86	0.88	0.90	0.92	0.83	0.86	0.88	0.90
FOREMAN		0.94	0.95	0.96	0.97	0.91	0.93	0.95	0.96
MOBILE		0.85	0.87	0.89	0.90	0.85	0.88	0.89	0.90
MEAN		**0.89**	**0.91**	**0.93**	**0.94**	**0.91**	**0.92**	**0.93**	**0.94**
FOOTBALL	30	0.83	0.85	0.87	0.89	0.74	0.83	0.86	0.89
FOREMAN		0.89	0.91	0.93	0.93	0.89	0.91	0.92	0.92
MOBILE		0.88	0.88	0.87	0.87	0.88	0.88	0.87	0.87
MEAN		**0.87**	**0.89**	**0.90**	**0.92**	**0.88**	**0.89**	**0.90**	**0.92**

TABLE IV
AVERAGE $\gamma_{N\times N}$ FOR 16×16, 8×8, AND 4×4 BLOCKS IN EACH TEMPORAL ENHANCEMENT LAYER (IN PERCENTAGE)

Test Sequence	Qp	$\gamma_{16\times16}$				$\gamma_{8\times8}$				$\gamma_{4\times4}$			
		T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4
FOOTBALL	40	39.7	35.5	30.2	25.9	4.7	3.6	2.8	2.3	1.3	1.0	0.7	0.7
FOREMAN		15.8	12.7	11.0	9.4	1.5	1.3	1.4	1.8	0.6	0.6	0.8	1.4
MOBILE		57.5	45.6	40.4	30.2	6.1	3.9	3.7	3.0	0.5	0.4	0.3	0.3
MEAN		**28.6**	**25.6**	**22.5**	**19.4**	**3.9**	**3.4**	**2.5**	**2.6**	**0.6**	**0.6**	**0.4**	**0.6**
FOOTBALL	30	67.7	62.4	61.3	63.8	21.8	17.1	16.4	14.9	1.2	1.0	1.2	1.7
FOREMAN		40.7	39.2	40.7	44.6	7.1	6.5	5.6	5.7	0.5	0.9	1.2	1.4
MOBILE		79.9	85.2	88.6	87.3	40.0	39.3	39.5	37.7	7.3	6.9	6.5	6.8
MEAN		**55.4**	**56.4**	**58.0**	**58.2**	**21.5**	**20.3**	**18.7**	**16.7**	**2.8**	**2.8**	**2.6**	**2.7**

using three temporal prediction types to improve the coding efficiency. According to the rate-distortion theory, a temporal prediction type with smaller RD cost provides better coding efficiency. We adopt the RD cost function defined by JSVM [6] (which came from essentially the rate distortion theory) and collect all J_{FW}, J_{BW}, and J_{BI} for three squared-shape partitions, 16×16, 8×8, and 4×4. For each sub-block $\mathcal{S}_i$, we define the *relative RD improvement* $\omega_{i|\mathcal{T}^*}$, offered by the best temporal prediction type $\mathcal{T}^*$, as follows:

$$\omega_{i|\mathcal{T}^*}=\left(\min_{\mathcal{T}\in(\mathcal{T}^{\mathcal{U}}\cup\{\mathrm{BI}\})\setminus\mathcal{T}^*}\{J_{\mathcal{T}}\}\right)-J_{\mathcal{T}^*},\quad \mathcal{S}_i\in\mathcal{T}^*. \tag{3}$$

The overall relative RD improvement $\mathcal{W}$ is the sum of $\omega_{i|\mathcal{T}^*}$ of all sub-blocks; that is, $\mathcal{W}\triangleq\mathcal{W}_{\mathrm{FW}}+\mathcal{W}_{\mathrm{BW}}+\mathcal{W}_{\mathrm{BI}}$, where $\mathcal{W}_{\mathcal{T}}=\sum_{S_i\in\mathcal{T}}\omega_{i|\mathcal{T}}$. Furthermore, in order to quantitatively determine the coding efficiency of using BI in the squared $N\times N$ blocks, we define a BI performance index by

$$\gamma_{N\times N}\triangleq\frac{\mathcal{W}_{\mathrm{BI},N\times N}}{\mathcal{W}_{N\times N}}=\frac{\mathcal{W}_{\mathrm{BI},N\times N}}{\mathcal{W}_{\mathrm{FW},N\times N}+\mathcal{W}_{\mathrm{BW},N\times N}+\mathcal{W}_{\mathrm{BI},N\times N}} \tag{4}$$

In other words, the index $\gamma_{N\times N}$, ranging from 0 to 1, indicates the percentage of the relative RD improvement contributed by BI for the totality of an $N\times N$ size block. Moreover, a large $\gamma_{N\times N}$ value shows that the BI has a significantly relative improvement in $J_{\mathcal{T}}$ and that the BI should not be skipped.

Fig. 3 depicts the performance index value for each hierarchical-B frame. As shown, the relative improvement offered by BI at high bit-rates is superior to that at low bit-rates because the encoder has more bits to reduce the distortion. This superiority at different bit-rates is particularly noticeable in the large 16×16 partition. On the other hand, the BI type usually furnishes less than 20% improvement on $\gamma_{8\times8}$ and $\gamma_{4\times4}$, even at high coding rates.

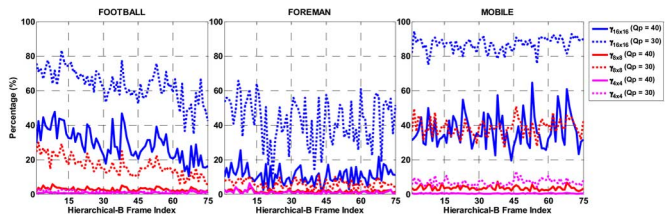


Fig. 3. Measure index $\gamma_{N\times N}$ for individual hierarchical-B frame.

Table IV shows the average benefits offered by the BI type at various temporal enhancement layers. As illustrated, the performance index values decrease as the partition becomes finer. The $\gamma_{16\times16}$ has relatively average values of 24% and 57% for $Qp=40$ and $Qp=30$, respectively; that is, BI plays an important role in improving coding efficiency for large partitions. For the 8×8 partition, the effect of BI plunges to 11% on average, which says that the set $\mathcal{T}^{\mathcal{U}}$ is sufficient to provide a good compression. Furthermore, when the partition is getting finer down to 4×4, the contribution of BI can be ignored because $\gamma_{4\times4}$ is less than 3% typically. However, some test videos such as MOBILE need BI can achieve better RD performance for both 16×16 and 8×8 partitions, because its $\gamma_{16\times16}$ and $\gamma_{8\times8}$ values reach 88.6% and 40.0%, respectively. In conclusion, the BI prediction type offers little coding gain for the block partitions smaller than 8×8.

C. Rate-Distortion Relationships Between Uni-Directional Predictions $\mathcal{T}^{\mathcal{U}}$ and biDirectional Prediction

In this subsection, we are interested in the relationships between $\mathcal{T}^{\mathcal{U}}$ and BI in motion bit-rate cost and residual distortion. We collect the following information in our experiments: (a) the motion vector (MV) difference, (b) the motion bit-rate cost $\mathcal{R}_{\mathcal{T}}$, and (c) the distortion $\mathcal{D}_{\mathcal{T}}$ for the three temporal prediction types.

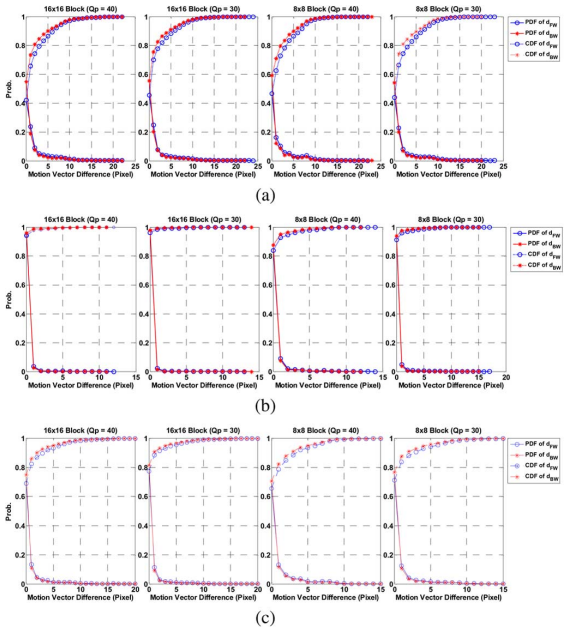


Fig. 4. PDFs and CDFs of the vector difference $d_{\mathcal{T}}$. (a) FOOTBALL; (b) MOBILE; (c) MEAN.

The statistical observations and theoretical analyses on the experimental results are reported below.

1) Motion Vector Difference: In order to find the correlations of two cost terms $\mathcal{R}_{\mathcal{T}}$ and $\mathcal{D}_{\mathcal{T}}$ between these temporal prediction types, we examine the MV differences after the MVs are refined by the BI search process. We look at two square block partitions, 16×16 and 8×8. On the JSVM 9.11 platform [6], we search for the best MVs of different prediction types for a specified block partition $N \times N$. Their notations are as follows:

$$\begin{aligned}
&\mathbf{mv}_{\mathrm{FW},N\times N} : \text{the FW MV of } N \times N \text{ blocks}\\
&\mathbf{mv}_{\mathrm{BW},N\times N} : \text{the BW MV of } N \times N \text{ blocks}\\
&\mathbf{mv}_{\mathrm{BI}\leftarrow\mathrm{FW},N\times N} : \text{the forward MV refined}\\
&\qquad \text{by BI for } N \times N \text{ blocks}\\
&\mathbf{mv}_{\mathrm{BI}\leftarrow\mathrm{BW},N\times N} : \text{the backward MV refined}\\
&\qquad \text{by BI for } N \times N \text{ blocks.}
\end{aligned}$$

As described earlier, the BI search process takes and as its initial search points for motion estimation. The Euclidean distance is used to measure the MV difference and

$$d_{\mathcal{T},N\times N} = \|\mathbf{mv}_{\mathcal{T},N\times N} - \mathbf{mv}_{\mathrm{BI}\leftarrow\mathcal{T},N\times N}\|, \quad \mathcal{T} \in \mathcal{T}^{\mathcal{U}}. \tag{5}$$

We statistically gather the 16×16 and 8×8 blocks that choose BI as their best temporal prediction type for generating the probability distributions (PDF) and cumulative probabilities (CDF) of d_{FW} and d_{BW}, as shown in Fig. 4. As shown, the PDFs of the MV difference are strongly clustered around the starting search points. Particularly, the one-pixel probability $\mathbf{Pr}\{d_{\mathcal{T}} < 1 \text{ (pixel)}\}$ is close to 90% for the MOBILE. That is, most of the MVs after locally refined by BI are still very close to $\mathbf{mv}_{\mathrm{FW}}$ and $\mathbf{mv}_{\mathrm{BW}}$. Typically, the MV differences are within three pixels; the CDFs of MV differences less than three pixels usually reach 80% or more. Our experiments show that

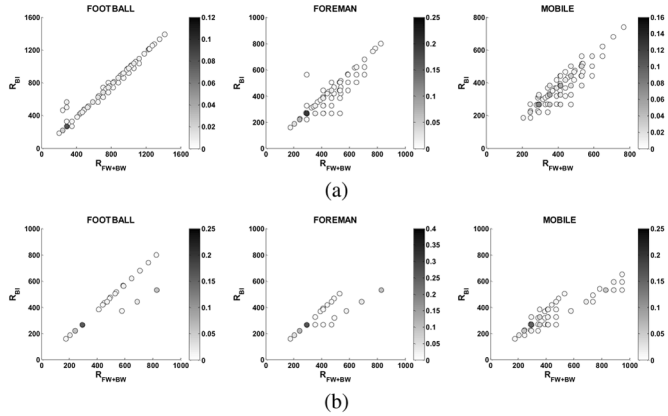


Fig. 5. Distributions of motion bit-rate cost. (a) 16×16 partition size with $Qp = 40$; (b) 8×8 partition size with $Qp = 40$.

different block partitions have similar probability distributions. The similarity in the two motion pairs $(\mathbf{mv}_{\mathrm{FW}}, \mathbf{mv}_{\mathrm{BI}\leftarrow\mathrm{FW}})$ and $(\mathbf{mv}_{\mathrm{BW}}, \mathbf{mv}_{\mathrm{BI}\leftarrow\mathrm{BW}})$ is the foundation of the following analysis.

2) Motion Bit-Rate Cost: Our second study tries to identify the correlation of the motion bit-rate costs $\mathcal{R}_{\mathcal{T}}$ between $\mathcal{T}^{\mathcal{U}}$ and BI. As discussed in Section III-C-1, the MVs, $\mathbf{mv}_{\mathrm{BI}\leftarrow\mathrm{FW}}$ and $\mathbf{mv}_{\mathrm{BI}\leftarrow\mathrm{BW}}$, produced by BI are often close to $\mathbf{mv}_{\mathrm{FW}}$ and $\mathbf{mv}_{\mathrm{BW}}$, respectively. In addition, the motion bit-rate cost is proportional to the MV length. Based on those two observations, we anticipate that there exists a strong correlation in motion bit-rate cost between $\mathcal{T}^{\mathcal{U}}$ and BI. More specifically, since the BI operation needs two MVs to fetch two reference blocks for prediction, we collect the motion bit-rate costs of three temporal prediction types, $\mathcal{R}_{\mathrm{FW}}$, $\mathcal{R}_{\mathrm{BW}}$, and $\mathcal{R}_{\mathrm{BI}} \triangleq (\mathcal{R}_{\mathrm{BI}\leftarrow\mathrm{FW}} + \mathcal{R}_{\mathrm{BI}\leftarrow\mathrm{BW}})$ to find out the relationship between $\mathcal{R}_{\mathrm{BI}}$ and the combined cost of $\mathcal{T}^{\mathcal{U}}$, denoted as $\mathcal{R}_{\mathrm{FW+BW}} \triangleq (\mathcal{R}_{\mathrm{FW}} + \mathcal{R}_{\mathrm{BW}})$.

As depicted in Fig. 5, the distributions of $\mathcal{R}_{\mathrm{FW+BW}}$ and $\mathcal{R}_{\mathrm{BI}}$ are noticeably concentrated along a straight line; this high correlation is foreseen because $\mathbf{mv}_{\mathrm{FW}} \approx \mathbf{mv}_{\mathrm{BI}\leftarrow\mathrm{FW}}$ and $\mathbf{mv}_{\mathrm{BW}} \approx \mathbf{mv}_{\mathrm{BI}\leftarrow\mathrm{BW}}$, as discussed earlier. This implies that $\mathcal{R}_{\mathrm{FW}} \approx \mathcal{R}_{\mathrm{BI}\leftarrow\mathrm{FW}}$ and $\mathcal{R}_{\mathrm{BW}} \approx \mathcal{R}_{\mathrm{BI}\leftarrow\mathrm{BW}}$. Therefore, we make use of the first-order regression to represent the motion bit-rate cost $\mathcal{R}_{\mathrm{BI}}$ by $\mathcal{R}_{\mathrm{FW+BW}}$. Here, the motion bit-rate cost $\mathcal{R}_{\mathrm{BI}}$ is modeled as an affine function of $\mathcal{R}_{\mathrm{FW+BW}}$. This regression for $\mathcal{R}_{\mathrm{BI}}$ based on $\mathcal{R}_{\mathrm{FW+BW}}$ is thus formulated as

$$\widehat{\mathcal{R}}_{\mathrm{BI}} = \alpha + \beta \cdot \mathcal{R}_{\mathrm{FW+BW}} \tag{6}$$

where α and β are the regression parameters. Furthermore, according to the collected data, the term α in the regression model is usually nearly zero because $\mathcal{R}_{\mathrm{FW}} \approx 0$ when $\|\mathbf{mv}_{\mathrm{FW}}\| \approx 0$ and $\mathcal{R}_{\mathrm{BW}} \approx 0$ when $\|\mathbf{mv}_{\mathrm{BW}}\| \approx 0$. The first-order model is thus simplified to a linear function,

$$\widehat{\mathcal{R}}_{\mathrm{BI}} = \beta \cdot \mathcal{R}_{\mathrm{FW+BW}}. \tag{7}$$

Applying the least squares technique, we can thus determine the optimal β value by

$$\beta^* = \frac{\mathbf{E}(\mathcal{R}_{\mathrm{BI}}\mathcal{R}_{\mathrm{FW+BW}})}{\mathbf{E}\left(\mathcal{R}^2_{\mathrm{FW+BW}}\right)} \tag{8}$$

TABLE V
OPTIMAL β^* VALUES FOR THE LINEAR REGRESSION MODEL

Test Sequence	Qp	16x16				8x8			
		T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4
FOOTBALL	40	0.97	0.97	0.95	0.95	0.96	0.95	0.94	0.92
FOREMAN		0.95	0.93	0.91	0.87	0.93	0.90	0.89	0.86
MOBILE		0.95	0.94	0.92	0.90	0.93	0.91	0.88	0.87
MEAN		**0.92**							
FOOTBALL	30	0.99	0.98	0.98	0.96	0.98	0.94	0.96	0.94
FOREMAN		0.97	0.96	0.93	0.90	0.95	0.93	0.90	0.86
MOBILE		0.96	0.95	0.94	0.92	0.96	0.93	0.92	0.90
MEAN		**0.94**							

As tabulated in Table V, the optimal β^* for 16 × 16 and 8 × 8 block partitions is around 0.93 and it slightly decreases at the higher temporal enhancement layers. Different block partitions have similar slope values. This linear model, $\widehat{\mathcal{R}}_{\rm BI} = \overline{\beta^*} \cdot \mathcal{R}_{\rm FW+BW}$, is a rather good approximation to the real $\mathcal{R}_{\rm BI}$ because the percentage error $\mathbf{E}(|(\widehat{\mathcal{R}}_{\rm BI} - \mathcal{R}_{\rm BI})/\mathcal{R}_{\rm BI}|)$ is 11.3%. Here, $\overline{\beta^*}$ in this linear regression model is 0.93, which is the average value of MEAN.

3) Distortion Relationship: Our last study addresses on the correlations in the distortions of different prediction types. We obtain an upper bound of $\mathcal{D}_{\rm BI}$ and will construct an approximated distribution of $\mathcal{D}_{\rm BI}$ (denoted as $\widetilde{\mathcal{D}}_{\rm BI}$).

For a given $M \times N$ partition mode, the SAD metric used to evaluate its distortions of $\mathcal{T}^{\mathcal{U}}$ is defined by (2), unfolded as

$$\begin{aligned}\mathcal{D}_{\mathcal{T}}(\mathbf{mv}_{\mathcal{T}}) &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} |f(x,y) - f'_{\mathcal{T}}\,(x - mv^x_{\mathcal{T}}, y - mv^y_{\mathcal{T}})| \\ &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} |e_{\mathcal{T}}(x,y)|\,, \quad \mathcal{T} \in \mathcal{T}^{\mathcal{U}}. \end{aligned} \tag{9}$$

The distortion is calculated by two nested summations over $x = 0 \sim M-1$ and $y = 0 \sim N-1$, f is the current block, $f'_{\mathcal{T}}$ is the reference block in the $\mathcal{T}$ direction, and $(mv^x_{\mathcal{T}}, mv^y_{\mathcal{T}})$ are the components of $\mathbf{mv}_{\mathcal{T}}$. Furthermore, each prediction error $e_{\mathcal{T}}(x,y)$ denotes its corresponding location in the block difference, $f(x,y) - f'_{\mathcal{T}}(x - mv^x_{\mathcal{T}}, y - mv^y_{\mathcal{T}})$. In the BI case with equal weighted prediction, its distortion value is defined by

$$\mathcal{D}_{\rm BI} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \left| f(x,y) - \frac{1}{2}\sum_{\mathcal{T}\in\mathcal{T}^{\mathcal{U}}} f'_{\mathcal{T}}\,(x - mv^x_{{\rm B}\leftarrow\mathcal{T}}, y - mv^y_{{\rm BI}\leftarrow\mathcal{T}}) \right|. \tag{10}$$

Because the MVs finally adopted by BI are close to those produced by $\mathcal{T}^{\mathcal{U}}$, the value of $\mathcal{D}_{\rm BI}$ may be approximated by $\widetilde{\mathcal{D}}_{\rm BI}$, namely

$$\mathcal{D}_{\rm BI} \approx \widetilde{\mathcal{D}}_{\rm BI} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \left| f(x,y) - \frac{1}{2}\sum_{\mathcal{T}\in\mathcal{T}^{\mathcal{U}}} f'_{\mathcal{T}}\,(x - mv^x_{\mathcal{T}}, y - mv^y_{\mathcal{T}}) \right|. \tag{11}$$

As shown below, the average of $(\mathcal{D}_{\rm FW} + \mathcal{D}_{\rm BW})$ can be derived as an upper bound of $\widetilde{\mathcal{D}}_{\rm BI}$ by using the well-known triangle inequality.

$$\begin{aligned}\widetilde{\mathcal{D}}_{\rm BI} &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \left| f(x,y) - \frac{1}{2}\sum_{\mathcal{T}\in\mathcal{T}^{\mathcal{U}}} f'_{\mathcal{T}}\,(x - mv^x_{\mathcal{T}}, y - mv^y_{\mathcal{T}}) \right| \\ &= \frac{1}{2}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} |[f(x,y) - f'_{\rm FW}\,(x - mv^x_{\rm FW}, y - mv^y_{\rm FW})] \\ &\qquad + [f(x,y) - f'_{\rm BW}\,(x - mv^x_{\rm BW}, y - mv^y_{\rm BW})]| \\ &= \frac{1}{2}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} |e_{\rm FW}(x,y) + e_{\rm BW}(x,y)| \\ &\le \frac{1}{2}\sum_{\mathcal{T}\in\mathcal{T}^{\mathcal{U}}} \left(\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} |e_{\mathcal{T}}(x,y)| \right) \\ &= \frac{1}{2}(\mathcal{D}_{\rm FW} + \mathcal{D}_{\rm BW}) \triangleq \mathcal{D}_{\rm FW+BW} \end{aligned} \tag{12}$$

In addition to derive an upper bound of $\widetilde{\mathcal{D}}_{\rm BI}$, we verify that the probability distribution of $\widetilde{\mathcal{D}}_{\rm BI}$ is a Gamma distribution as described in the Appendix. The statistical tests indicate that the pair of $(e_{\rm FW}, e_{\rm BW})$ tends to be bivariate-Laplacian distributed. This distribution model is used to construct a set of adaptive thresholds in Section IV.

IV. PROPOSED FAST TEMPORAL PREDICTION SELECTION ALGORITHM

In this section, we develop a fast temporal prediction type selection algorithm for the dyadic hierarchical-B prediction structure based on the observations in Section III. We derive a set of adaptive thresholds that efficiently eliminate unnecessary BI evaluations in Section IV-A. Then, combined with the adaptive thresholds, our proposed schemes are elaborated in Section IV-B.

A. Adaptive Thresholds

The highly correlated motion bit-rate costs and distortions between $\mathcal{T}^{\mathcal{U}}$ and BI are used to develop a set of thresholds for block partitions from 16 × 16 to 8 × 8. According to the information obtained from the FW and BW processes, we can build a BI motion bit-rate cost estimator and a BI distortion estimator for a specified $M \times N$ block.

As analyzed earlier, the motion bit-rate costs of $\mathcal{R}_{\rm FW+BW}$ and $\mathcal{R}_{\rm BI}$ are related by a linear regression model; that is, an estimated motion bit-rate cost of BI, $\widehat{\mathcal{R}}_{\rm BI}$, is modeled as $\widehat{\mathcal{R}}_{\rm BI} = \beta \cdot \mathcal{R}_{\rm FW+BW}$. Moreover, among different block sizes, the optimal slope β^* does not change much, which ranges from 0.86 to 0.97. Hence, the mean value of β^* of all block partitions (sizes) is adequate for all cases in estimating the motion bit-rate cost $\widehat{\mathcal{R}}_{\rm BI}$.

Next, we try to estimate from the probability model of $\widetilde{\mathcal{D}}_{\rm BI}$ and the percentage of the exception case of $(\mathcal{D}_{\rm BI} > \mathcal{D}_{\rm FW+BW})$. Occasionally, $\mathcal{D}_{\rm FW+BW}$ is not an upper bound of $\mathcal{D}_{\rm BI}$ if the chosen BI is inferior in terms of distortion. (Note that the MV chosen by BI is judged by its better combined RD value, not by the distortion value only.) From our collected data, the exception $(\mathcal{D}_{\rm BI} > \mathcal{D}_{\rm FW+BW})$ is 1% ∼ 5% on the average for different block partitions.

TABLE VI
k VALUES AND ξ VALUES

Block Mode	16x16	16x8 & 8x16	8x8
k	256	128	64
$\xi_{M\times N}=\frac{\widehat{\mathcal{D}}_{\mathrm{BI},M\times N}}{\mathcal{D}_{\mathrm{FW+BW},M\times N}}$	0.90	0.85	0.80

As a consequence of the preceding discussions, the mean value of $\Gamma(k,\theta)$[1] is sufficient to represent $\mathcal{D}_{\mathrm{BI}}$, namely, the estimated distortion is

$$\widehat{D}_{\mathrm{BI}}=k\theta. \tag{13}$$

As discussed earlier, $\mathcal{D}_{\mathrm{BI}}\approx\widetilde{\mathcal{D}}_{\mathrm{BI}}$ (Normally, $\mathcal{D}_{\mathrm{BI}}<\widetilde{\mathcal{D}}_{\mathrm{BI}}$ indicates that the MV refinement in BI is effective.) and $\mathbf{Pr}\{\Gamma(k,\theta)=\theta\Gamma(k,1)>\mathcal{D}_{\mathrm{FW+BW}}\}=1\%\sim5\%$ for different block sizes. The probability $\mathbf{Pr}\{\Gamma(k,\theta)=\theta\Gamma(k,1)>\widehat{\mathcal{D}}_{\mathrm{BI}}\}$ can be calculated for a fixed k. Therefore, we can determine the relationship between $\mathcal{D}_{\mathrm{FW+BW}}$ and $\widehat{\mathcal{D}}_{\mathrm{BI}}$ without any knowledge of θ, as listed in Table VI. For example, the probability $\mathbf{Pr}\{\mathcal{D}_{\mathrm{BI},16\times16}>\mathcal{D}_{\mathrm{FW+BW},16\times16}\}$ is 3.6% from our experimental data.

$$\begin{aligned}0.036&=\mathbf{Pr}\{\mathcal{D}_{\mathrm{BI},16\times16}>\mathcal{D}_{\mathrm{FW+BW}16\times16}\}\\&\approx\mathbf{Pr}\{\widehat{\mathcal{D}}_{\mathrm{BI},16\times16}>\mathcal{D}_{\mathrm{FW+BW}16\times16}\}\\&=\mathbf{Pr}\{\Gamma(k,\theta)>\mathcal{D}_{\mathrm{FW+BW},16\times16}\}\\&=\mathbf{Pr}\{\theta\Gamma(k,1)>\mathcal{D}_{\mathrm{FW+BW},16\times16}\}.\end{aligned} \tag{14}$$

From the above, we can derive $(\mathcal{D}_{\mathrm{FW+BW}})_{16\times16}=\theta\cdot\mathrm{CDF}^{-1}_{\Gamma(k,1)}(0.964)$ where $\mathrm{CDF}^{-1}_{\Gamma(k,1)}$ denotes the inverse CDF of $\Gamma(k,1)$. Furthermore, it yields

$$\begin{aligned}\frac{\widehat{\mathcal{D}}_{\mathrm{BI},16\times16}}{\mathcal{D}_{\mathrm{FW+BW},16\times16}}&=\frac{k}{\mathrm{CDF}^{-1}_{\Gamma(k,1)}(0.964)}\overset{k:=256}{=}\frac{256}{285.5196}\\&=0.90.\end{aligned} \tag{15}$$

Finally, using the ξ definition in Table VI, an estimate of $\mathcal{D}_{\mathrm{BI}}$ (for the $M\times N$ partition mode) is represented by

$$\widehat{\mathcal{D}}_{\mathrm{BI},M\times N}=\xi_{M\times N}\cdot\mathcal{D}_{\mathrm{FW+BW},M\times N}. \tag{16}$$

In addition, the percentage error $\mathbf{E}(|(\widehat{\mathcal{D}}_{\mathrm{BI}}-\mathcal{D}_{\mathrm{BI}})/\mathcal{D}_{\mathrm{BI}}|)$ is around 12.5%.

B. Algorithm Overview

The flowchart in Fig. 6 depicts our proposed algorithm on eliminating the futile BI calculations. It mainly consists of two early termination criterions. First, a part of ineffective BI is skipped by the strong consistency in the temporal prediction types of large partitions. Then, the remaining unnecessary BI can be further detected by making use of the adaptive thresholds. The proposed approaches are split into three major stages, described below.

Stage 1: Conditionally Exhaustive Temporal Prediction Type Search for Inter16 $\times$ 16. The Inter16 $\times$ 16 partition mode conditionally checks all FW, BW, and BI types to identify its best temporal prediction type to be used in the Stage 2.

Step 1.1: Exhaustive Search on Uni-directional Predictions. For the current MB, the set $\mathcal{T}^{\mathcal{U}}$ is evaluated in order to collect their distortions ($\mathcal{D}_{\mathrm{FW}}$ and $\mathcal{D}_{\mathrm{BW}}$) and motion bit-rate costs ($\mathcal{R}_{\mathrm{FW}}$ and $\mathcal{R}_{\mathrm{BW}}$) that will be used in Step 1.3. If the MB is located in the two highest temporal enhancement layers (T_{n-1} and T_n), denoted as HTEL, go to Step 1.3; otherwise, go to Step 1.2.

Step 1.2: BI Evaluation at Lower Temporal Enhancement Layers. At the temporal enhancement layers $T_1\sim T_{n-2}$, the BI is always tested. Go to Stage 2.

Step 1.3: Conditional BI Execution at Higher Temporal Enhancement Layers. Using the information obtained from Step 1.1, the threshold $\delta_{16\times16}$ can be obtained by

$$\begin{aligned}\delta_{16\times16}&=\widehat{\mathcal{D}}_{\mathrm{BI},16\times16}+\widehat{\mathcal{R}}_{\mathrm{BI},16\times16}\\&=\xi_{16\times16}\cdot\mathcal{D}_{\mathrm{FW+BW},16\times16}+\overline{\beta^*}\cdot\mathcal{R}_{\mathrm{FW+BW},16\times16}\\&=0.9\times\left(\frac{\mathcal{D}_{\mathrm{FW},16\times16}+\mathcal{D}_{\mathrm{BW},16\times16}}{2}\right)\\&\quad+0.93\times(\mathcal{R}_{\mathrm{FW},16\times16}+\mathcal{R}_{\mathrm{BW},16\times16}).\end{aligned} \tag{17}$$

If these two conditions $\delta_{16\times16}<J_{\mathrm{FW},16\times16}$ and $\delta_{16\times16}<J_{\mathrm{BW},16\times16}$ are satisfied, the BI process is performed. Otherwise, BI is judged ineffective and thus skipped. Go to Stage 2.

Stage 2: Early Termination on BI for Large Partitions. Before entering the Stage 2, the best temporal prediction type $\mathcal{T}^*_{16\times16}$ is determined in Stage 1. Two steps of this stage predict whether or not the BI type in $16\times8/8\times16/8\times8$ partitions have an inferior RD performance and thus can be excluded from testing. Section IV-B-1 details the early termination procedure.

Step 2.1: Exhaustive Uni-directional Prediction Type Search for Partitions $16\times8/8\times16/8\times8$. To gather the distortions ($\mathcal{D}_{\mathrm{FW}}$ and $\mathcal{D}_{\mathrm{BW}}$) and motion bit-rate costs ($\mathcal{R}_{\mathrm{FW}}$ and $\mathcal{R}_{\mathrm{BW}}$) of partitions $16\times8/8\times16/8\times8$, the set $\mathcal{T}^{\mathcal{U}}$ is calculated.

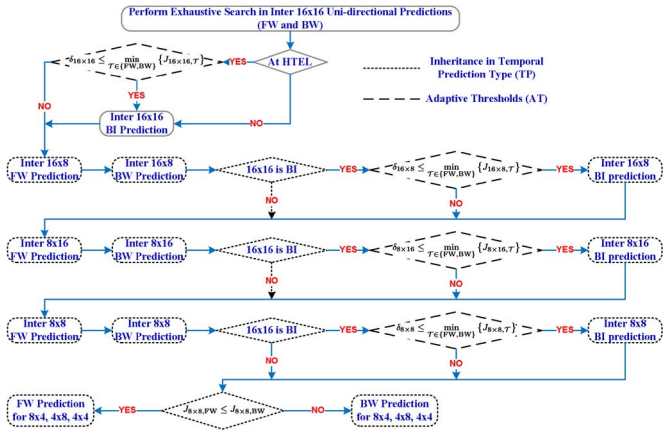


Fig. 6. Selection algorithm for temporal prediction types.

[1]The notation $\Gamma(k,\theta)$ represents the Gamma distribution, where k is the shape parameter and θ is the scale parameter.

Step 2.2: Pre-decided BI Elimination for Partitions $16\times8/8\times16/8\times8$. If the pre-determined information $\mathcal{T}^*_{16\times16}$ is not BI, go to Stage 3. Otherwise, continue.

Step 2.3: Provisory BI Expulsion by Adaptive Thresholds. Similar to Step 1.3, the adaptive thresholds $\delta_{M\times N}$ are obtained by

$$\begin{aligned}\delta_{M\times N} &= \widehat{\mathcal{D}}_{\mathrm{BI},M\times N} + \widehat{\mathcal{R}}_{\mathrm{BI},M\times N}\\ &= \xi_{M\times N}\cdot\mathcal{D}_{\mathrm{FW+BW},M\times N}\\ &\quad + \overline{\beta^*}\cdot\mathcal{R}_{\mathrm{FW+BW},M\times N},\end{aligned}\tag{18}$$

where $M\times N\in\{16\times8, 8\times16, 8\times8\}$. The BI type is calculated if the specifications $\delta_{M\times N} < J_{\mathrm{FW},M\times N}$ and $\delta_{M\times N} < J_{\mathrm{BW},M\times N}$ are both true. Otherwise, the BI type is discarded.

Stage 3: Adaptive Prediction Type Selection for Small Partitions. Due to the strong correlation between the prediction type of 8×8 and those of the smaller partitions, we can skip the less probable prediction types by checking the 8×8 prediction type. The Section IV-B-2 elaborates our implementation.

1) Early Termination on BI for Large Partitions: Based on our observations discussed earlier, the prediction type information of Inter16 $\times$ 16 partition mode is useful in skipping the BI type for large block partitions. More precisely, the conditional probabilities, $p_{(16\times8|16\times16)\notin\mathrm{BI}}$, $p_{(8\times16|16\times16)\notin\mathrm{BI}}$, and $p_{(8\times8|16\times16)\notin\mathrm{BI}}$, can suggest whether the unnecessary computations of BI of $16\times8/8\times16/8\times8$ partitions can be avoided if the 16×16 partition is of the BI type. Thus, in Step 2.2, the saved computations in BI for large partitions depend on the BI selection rate for the 16×16 partition. Furthermore, for the case that $\mathcal{T}^*_{16\times16}$ is BI, the remaining superfluous BI can be detected by the $\delta_{M\times N}$ thresholds in Step 2.3.

2) Adaptive Prediction Type Selection for Small Partitions: As suggested by our previous analysis for small partitions, we notice that (a) 10% blocks or less are coded with BI, (b) BI contributes less in improving compression efficiency as compared to $\mathcal{T}^{\mathcal{U}}$, and (c) small partitions often have the same prediction types as that of their inherited 8×8 parent block. These three observations help us in developing an adaptive prediction selection algorithm for small partitions.

The 8×8 or smaller blocks are seldom coded with BI. As discussed earlier, the prediction type of a smaller partition can be reliably estimated by its 8×8 parent partition. Thus, each smaller partition refined from an 8×8 partition only needs to check one prediction type. Furthermore, the candidate can be well predicted by comparing the J_{FW} and J_{BW} of its associated 8×8 partition, even if its BI is not calculated in Stage 2. Therefore, in Stage 3, reduction in computation for small partitions can be achieved by skipping either FW or BW.

V. Experimental Results and Discussions

A. Test Conditions

For performance assessment, we have implemented the proposed algorithms in JSVM 9.11 [6] and have tested 19 typical video sequences in four resolutions (QCIF/CIF/4CIF/HD formats), covering a broad range of visual characteristics. Our proposed schemes focus on the complexity reduction at the temporal enhancement layers in the dyadic hierarchical-B prediction structure. The detailed encoder parameters are given in Table VII and the other parameters are the default values set by the reference software JSVM 9.11 [6].

TABLE VII
Testing Conditions

Sequence	QCIF	CARPHONE (CP), CONTAINER (CTN), MOTHERDAUTHER (MD), SUZIE (SZ)
	CIF	AKIYO (AK), BUS (BU), FOOTBALL (FO), FOREMAN (FM), MOBILE (MO), STEFAN (SF)
	4CIF	CITY (CT), CREW (CR), HARBOUR (HA), ICE (IC), SOCCER (SC)
	HD (720p)	BIGSHIPS (BS), MOBCAL (MC), SHIELDS (SH), STOCKHOLM (ST)
$\boldsymbol{Qp_i}$	Eight Qp values from 20 to 48, denoted by $Qp_1\sim Qp_8$ where $Qp_1=20$ and $Qp_{i+1}=Qp_i+4$ for $i=1\sim7$	
Encoder Configuration	Motion search range: ±32 pixels with ¼-pel accuracy RDO: Enabled; GOP size: 8 and 16 Entropy coding: CABAC Number of reference frame in each reference list: 1 Frame encoded: 1 Intra followed by 128 Inter frames Default setting in JSVM 9.11[6]: SearchMode = 4 (apply fast motion estimation) BiPredIter = 4 (maximum iterations in BI prediction) IterSearchRange = 8 (search range of each iteration in BI prediction) Platform: Athlon 3800+, 64-bit, dual-core processors, 2.0GB RAM with Windows XP	

B. Performance Measures

To show the change in RD performance, we adopt the Bjontegaard metric [37], [38], which needs four RD points to measure the averaged Y-PSNR [BDP (dB)] and bit-rate differences [BDR (%)] between the two RD curves produced by JSVM 9.11 [6] and by our schemes, respectively. Hence, we separate eight Qp values into two sets, denoted by Qp^{I} and Qp^{II}, to measure the average RD performance for a wide range of bit-rates. These two Qp sets are $Qp^{\mathrm{I}}=\{Qp_1,Qp_2,Qp_3,Qp_4\}=\{20,24,28,32\}$ and $Qp^{\mathrm{II}}=\{Qp_5,Qp_6,Qp_7,Qp_8\}=\{36,40,44,48\}$

To measure the average speedup performance at these eight RD test points, we define time saving (TS) for the whole encoding process and the complexity reduction on the hierarchical-B frame process ($\mathrm{TS}_{\mathrm{HBvsIP}}$) only.

1) The overall time saving is defined as

$$\begin{aligned}\mathrm{TS} &= \frac{1}{\langle\langle Qp^{\mathrm{I}}\rangle\rangle+\langle\langle Qp^{\mathrm{II}}\rangle\rangle}\\ &\quad\times\sum_{Qp\in\{Qp^{\mathrm{I}},Qp^{\mathrm{II}}\}}\frac{\mathrm{T}_{\mathrm{JSVM9.11}}(Qp)-\mathrm{T}_{\mathrm{Proposed}}(Qp)}{\mathrm{T}_{\mathrm{JSVM9.11}}(Qp)}\times100\%,\end{aligned}\tag{19}$$

where $\mathrm{T}_{\mathrm{JSVM9.11}}(Qp)$ and $\mathrm{T}_{\mathrm{Proposed}}(Qp)$ denotes the encoding time of JSVM 9.11 [6] and that of our schemes with quantization parameter Qp, respectively. The notation $\langle\langle E\rangle\rangle$ is the number of elements in the set E.

2) In this $\mathrm{TS}_{\mathrm{HBvsIP}}$ measure, the denominator $\mathrm{T}_{\mathrm{JSVM9.11}}-\mathrm{T}_{\mathrm{JSVM9.11\ with\ IPPP}}$ is the additional computing time due to the use of hierarchical-B frames, compared to the

TABLE VIII
TIME SAVING CONTRIBUTED BY TP AND AT (AVERAGED FROM Qp_1 TO Qp_8)

Test Sequence	GOP = 8				GOP = 16			
	TP		AT		TP		AT	
	TS (%)	$\mathrm{TS}_{\mathrm{HBvsIP}}$ (%)	TS (%)	$\mathrm{TS}_{\mathrm{HBvsIP}}$ (%)	TS (%)	$\mathrm{TS}_{\mathrm{HBvsIP}}$ (%)	TS (%)	$\mathrm{TS}_{\mathrm{HBvsIP}}$ (%)
CP	50.1	27.8	26.2	62.4	50.4	29.4	26.6	62.7
CTN	51.0	25.2	17.7	74.2	51.4	26.7	19.0	72.9
MD	52.0	24.3	22.7	67.0	52.4	25.7	24.1	65.9
SZ	49.1	29.8	25.4	63.7	49.5	31.3	26.1	63.7
AK	54.1	21.4	26.2	61.9	55.2	21.9	27.1	61.7
BU	45.9	34.7	22.7	67.8	46.2	36.0	22.9	68.3
FO	44.0	37.6	23.5	66.7	44.2	38.9	23.6	67.4
FM	49.4	29.3	26.7	61.8	50.0	30.5	27.2	62.2
MO	44.6	35.9	21.6	69.0	45.0	36.9	22.0	69.2
SF	46.4	33.7	23.1	67.1	46.9	34.9	23.3	67.7
CT	49.1	28.6	22.4	67.4	49.6	29.9	23.1	67.3
CR	48.3	29.7	25.6	62.7	48.7	31.2	26.0	63.2
HA	45.8	34.0	23.2	66.6	46.3	35.0	23.5	67.0
IC	52.9	23.6	26.3	62.1	53.5	24.8	26.8	62.3
SC	48.7	30.4	23.2	66.9	49.2	31.6	23.6	67.1
BS	50.4	26.6	23.1	66.3	51.0	27.7	23.7	66.4
MC	49.6	27.0	19.7	71.1	50.1	28.3	20.7	70.4
SH	49.5	27.7	20.8	69.7	49.8	29.3	21.3	69.8
ST	49.6	27.5	19.4	71.7	50.1	28.6	19.9	71.8
AVG.	**49.0**	**29.2**	**23.1**	**66.6**	**49.4**	**30.5**	**23.7**	**66.7**

low-delay IPPP coding structure. Similarly, the numerator is the additional computing time of using our fast algorithms, shown in (20) at the bottom of the page, where $\mathrm{T}_{\mathrm{JSVM9.11\ with\ IPPP}}(Qp)$ represents the encoding time of JSVM 9.11 [6] with the IPPP coding structure and quantization parameter Qp. Thus, $\mathrm{TS}_{\mathrm{HBvsIP}}$ represents how much additional encoding time our approach needs (compared to the IPPP coding structure).

C. Performance Comparison With JSVM

Table VIII and Table IX present the time savings of the proposed schemes in comparison with JSVM 9.11. Listed in Table VIII are the improvements contributed by the inheritance of temporal prediction type (TP) and the adaptive thresholds to eliminate the superfluous BI computation (AT), respectively. The results are obtained by comparing the running time of the encoder with the following configurations:

Setting #1: JSVM 9.11 versus JSVM 9.11+TP: The TP setting makes use of the information produced by the 16×16 partition size to skip the BI type in the $16 \times 8/8 \times 16/8 \times 8$ partitions. For the block sizes smaller than 8×8, they only evaluate one of the uni-directional predictions, depending on the encoding information of 8×8.

Setting #2: JSVM 9.11 versus JSVM 9.11+AT: The AT setting uses the adaptive thresholds to conditionally select the BI type within the candidate set in the block partitions from 16×6 to 8×8 after performing FW and BW.

It can be seen that enabling the TP alone can averagely reduce the overall running time by 50%, equivalent to a speedup of about $2\times$, whereas the AT offers only a moderate time saving of $17\% \sim 27\%$. Because the TP setting considers the temporal prediction selection in all block modes, it provides more complexity reduction, as compared to the AT setting. Interestingly, the results are similar regardless of the GOP size.

To see their combined effects, Table IX provides the time savings relative to the exhaustive search, with both TP and AT enabled. The results given in these two tables correspond to two GOP sizes: 8 and 16. As can be seen, when the TP is coupled with the AT, an average saving of 62% for the overall encoding time is achieved. In other words, we can observe an approximated $3\times$ speedup. The improvement is achieved with a minor change in both bit-rate and Y-PSNR, as confirmed by the BDP/BDR values in the tables and the RD curves in Fig. 7. As discussed before, the BI examination in the encoding time is about $1.22/(1 + 1.22) = 55\%$. That is, the improvement in is generally limited to 55% when the BI computations are all skipped. However, our method can go beyond this limit, because in our algorithm the small partitions keep only one temporal prediction from $\mathcal{T}^{\mathcal{U}}$ for evaluation. Furthermore, the $\mathrm{TS}_{\mathrm{HBvsIP}}$ values demonstrate that the additional computation required by the hierarchical-B prediction structure can be averagely reduced by 89%. (The average $\mathrm{TS}_{\mathrm{HBvsIP}}$ is around 11%.)

The overall time savings in Table IX are not the sum of the results from Table VIII. In our approach, we set two succes-

$$\mathrm{TS}_{\mathrm{HBvsIP}} = \frac{1}{\langle\langle Qp^{\mathrm{I}}\rangle\rangle + \langle\langle Qp^{\mathrm{II}}\rangle\rangle} \sum_{Qp \in \{Qp^{\mathrm{I}}, Qp^{\mathrm{II}}\}} \frac{\mathrm{T}_{\mathrm{Proposed}}(Qp) - \mathrm{T}_{\mathrm{JSVM9.11\ with\ IPPP}}(Qp)}{\mathrm{T}_{\mathrm{JSVM9.11}}(Qp) - \mathrm{T}_{\mathrm{JSVM9.11\ with\ IPPP}}(Qp)} \times 100\% \quad (20)$$

TABLE IX
PERFORMANCE COMPARISONS WITH JSVM 9.11 [6]

Test Sequence	GOP size = 8				GOP size = 16			
	BDP (dB)	BDR (%)	TS (%)	TS_{HBvsIP} (%)	BDP (dB)	BDR (%)	TS (%)	TS_{HBvsIP} (%)
CP	-0.12	2.66	63.7	8.3	-0.12	2.85	64.2	10.1
CTN	0.00	0.11	60.2	11.7	-0.01	0.28	60.8	13.3
MD	-0.05	1.06	62.5	8.9	-0.07	1.48	63.3	10.4
SZ	-0.12	2.96	63.0	10.0	-0.14	3.39	63.5	11.8
AK	-0.03	0.82	64.6	6.1	-0.03	1.03	65.3	7.54
BU	-0.15	2.84	60.9	13.3	-0.17	3.27	61.1	15.4
FO	-0.17	3.15	59.4	15.8	-0.19	3.70	59.9	17.3
FM	-0.10	2.60	63.5	9.2	-0.12	3.20	64.1	10.9
MO	-0.09	1.80	61.0	12.1	-0.11	2.19	61.5	13.7
SF	-0.16	3.20	61.3	12.5	-0.17	3.51	61.7	14.3
CT	-0.02	0.64	61.3	10.7	-0.02	0.73	62.0	12.2
CR	-0.08	2.73	61.9	10.0	-0.09	3.04	62.5	11.7
HA	-0.04	1.20	61.7	11.1	-0.05	1.39	62.3	12.7
IC	-0.07	2.38	63.8	7.9	-0.08	2.64	64.4	9.5
SC	-0.04	1.28	61.3	12.3	-0.06	1.65	61.9	13.9
BS	-0.01	0.47	62.2	9.4	-0.01	0.60	63.0	10.8
MC	-0.01	0.41	60.5	10.8	-0.12	2.85	64.2	10.1
SH	0.00	0.47	60.3	11.9	-0.01	0.28	60.8	13.3
ST	-0.01	0.35	60.0	12.3	-0.07	1.48	63.3	10.4
AVG.	**-0.07**	**1.64**	**61.7**	**10.8**	**-0.08**	**2.04**	**62.5**	**11.8**

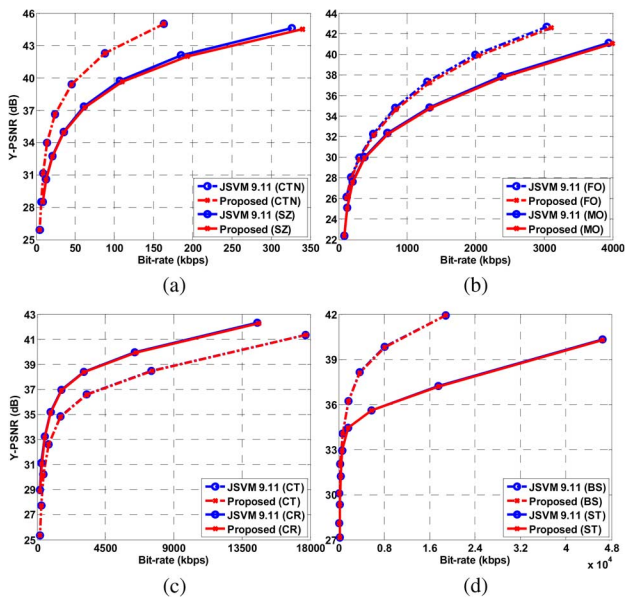


Fig. 7. Comparisons in RD curve ($\mathrm{GOP} = 8$). (a) QCIF; (b) CIF; (c) 4CIF; (d) HD (720p).

sive criterions to conditionally eliminate the BI computation, as illustrated in Fig. 6; one is TP, and the other is AT. However, their contributions are overlapped for some MBs; that is, if these MBs satisfy the first TP criterion, the AT criterion is not active in our design flowchart. Hence, with the average time saving of 50% by the TP, the AT criterion can additionally provide about 12% improvement. The additional improvement comes from the cases when the 16×16 partition is not BI, and the AT condition is satisfied. For example, in Fig. 2(a), about 70% of MBs do not select BI at T_1 (with $Qp = 40$) when 16×16 partition is examined. In this case, our algorithm skips these 70% of BI calculations. In total, there are 85% of 16×8 partitions do not prefer BI in our collected data. Therefore, only the remaining 15% of 16×8 partition blocks are further checked by the AT criterion for further complexity reduction.

Moreover, in Table X, the overall time saving decreases as the Qp value becomes small. The complexity reduction goes down from 67% to 48% as Qp decreases. This is due to the combined RD cost $J_{\mathcal{T}} = \mathcal{D}_{\mathcal{T}} + \lambda_{\mathrm{MOTION}} \times R_{\mathcal{T}}$ that affects the selection of temporal prediction type. Such an optimization principle tends to cut down the motion bit-rate term $\mathcal{R}_{\mathcal{T}}$ when Qp is large. On the other hand, because of the abundant bit budget, this optimization process spends more bits to reduce the distortion term $\mathcal{D}_{\mathcal{T}}$ at a small Qp. Thus, BI is used more often for small Qp values because BI is effective in reducing the distortion. Hence, fewer BI blocks can be skipped by our approach.

D. Performance Comparison With Other Fast Algorithms

In addition to the exhaustive search, we also compare our approaches with two well recognized fast algorithms, Li's method [29] and Lee's method [30]; both save the computing time based on mode reduction. Therefore, our approach is very different from theirs. Nevertheless, for comparison purpose, the same encoding configuration and test videos (Table VII) are used in the experiment (on JSVM 9.11).

As shown in Table XI, the Li's method [29] can averagely achieve about 55% time reduction and has a Y-PSNR loss of 0.11 dB and 2.8% bit-rate increase. The Lee's method [30] has the best time saving at about 65%, but it has also the highest coding quality loss, -0.15 dB Y-PSNR and $+3.6\%$ bits. Our scheme has roughly the least RD loss, -0.08 dB Y-PSNR and $+1.8\%$ bits, and its time saving is about 62%. If all are measured in ratio, our time saving is better than [29] by about 10% with a slightly better RD quality. And our time saving is slightly worse than [30] by 5% but both the PSNR and the bit rate losses are 50% better. However, we like to point out that our approach, focusing on the temporal prediction type reduction, is very different from the fast mode decision schemes in [29] and [30]. In

TABLE X
AVERAGE TIME SAVING FOR TWO GOP SIZES, 8 AND 16, WITH VARIOUS Qp VALUES

Test Sequence	Qp_1	Qp_2	Qp_3	Qp_4	Qp_5	Qp_6	Qp_7	Qp_8
CP	60.6	62.1	63.5	64.5	65.6	65.3	65.2	65.0
CTN	53.9	56.6	59.1	60.8	62.2	63.5	63.9	64.4
MD	59.2	61.0	62.4	63.3	63.8	64.2	64.4	65.0
SZ	59.1	60.9	63.0	64.3	64.9	64.8	64.6	64.7
AK	60.9	62.4	64.9	65.9	66.3	66.4	66.6	66.5
BU	55.7	57.8	59.5	60.9	61.9	63.0	64.2	64.9
FO	56.5	57.7	58.6	59.3	60.0	60.8	61.9	62.3
FM	60.0	62.2	63.7	64.6	64.9	65.0	65.0	64.9
MO	52.6	54.4	57.2	60.2	63.4	66.1	67.7	68.5
SF	56.3	57.6	59.3	61.1	62.9	64.1	64.9	65.5
CT	52.2	56.1	60.3	63.4	64.8	65.3	65.6	65.7
CR	57.0	59.8	61.6	62.8	63.4	63.9	64.4	64.9
HA	54.5	57.0	59.7	62.4	64.3	65.4	66.1	66.5
IC	60.2	61.7	63.5	65.0	65.3	65.6	65.8	65.8
SC	55.2	58.2	60.9	62.4	63.1	63.7	64.4	65.1
BS	52.8	57.6	61.8	64.3	65.5	66.1	66.4	66.2
MC	50.0	54.6	58.8	62.0	63.9	65.4	66.2	66.6
SH	49.9	55.9	59.5	61.8	63.4	64.4	65.0	65.5
ST	48.2	53.0	59.1	61.3	63.3	64.8	65.9	66.6
AVG.	**55.5**	**58.2**	**60.9**	**62.6**	**63.8**	**64.6**	**65.2**	**65.5**

TABLE XI
PERFORMANCE COMPARISONS (AVERAGE BDP, BDR, AND TS)

	Li's method [29]			Lee's method [30]			Proposed		
	BDP (dB)	BDR (%)	TS (%)	BDP (dB)	BDR (%)	TS (%)	BDP (dB)	BDR (%)	TS (%)
GOP = 8	-0.08	2.01	52.5	-0.14	3.20	64.7	-0.07	1.64	61.7
GOP = 16	-0.15	3.70	58.0	-0.17	4.03	66.1	-0.08	2.04	62.5

other words, we are not aiming at the same target; in contrast, our scheme may be combined with these schemes to achieve further complexity reduction. Simulations also indicate that our scheme is not sensitive to the video variation. Thus, instead of mode selection, reducing candidates in temporal prediction can be a promising approach for decreasing complexity in the hierarchical-B prediction structure.

VI. CONCLUSIONS

In this paper, we propose an effective temporal prediction type selection algorithm for the dyadic hierarchical-B prediction structure in SVC [4], [5], in which the unnecessary BI calculations are skipped for large block partitions and only one of the uni-directional temporal predictions is calculated for small partitions. The techniques used are (a) conditional elimination of BI for large partitions, (b) adaptive thresholds produced by the information obtained in the FW and BW processes, and (c) adaptive selection of FW and BW for small partitions. We first perform the uni-directional temporal predictions, FW and BW. Then, we make use of the strong correlations in the inheritance of temporal predictions and also construct a set of adaptive thresholds, both of which decide the BI operation is to be performed or not. To construct a reliable threshold, we examine the correlations of motion bit-rate costs and the distortions between the uni-directional temporal predictions and the BI type. Also, our statistical analysis shows that BI in small partitions does not contribute much in improving compression efficiency. These findings in the temporal enhancement layers are intelligently used for accelerating the encoding process.

On the average, our scheme can demonstrate up to 65.3% complexity reduction for the entire encoding process with minor changes in coding efficiency, as compared to JSVM 9.11 [6]. Also, the extra computations in the hierarchical-B prediction can be reduced by up to 93.9%. Hence, our approach can achieve a similar coding performance of JSVM 9.11 [6] but with much lower computational complexity. Moreover, our fast algorithm only reduces the complexity in temporal prediction types without any *a priori* mode reduction and can be applied to all enhancement coding layers.

APPENDIX

In Section III-C-3, we have derived the upper bound of $\widetilde{\mathcal{D}}_{\mathrm{BI}}$, as shown in (A.1). In addition, we further study its probability distribution in this appendix.

$$\widetilde{\mathcal{D}}_{\mathrm{BI}} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left| f(x,y) - \frac{1}{2}\sum_{\mathcal{T}\in\mathcal{T}^{\mathcal{U}}} f'_{\mathcal{T}}\left(x - mv_{\mathcal{T}}^{x}, y - mv_{\mathcal{T}}^{y}\right)\right| = \frac{1}{2}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left|e_{\mathrm{FW}}(x,y) + e_{\mathrm{BW}}(x,y)\right| \tag{A.1}$$

Ideally the FW and BW operations can find the shifted version of the current block if there are no quantization error and noise in the reference frames. As a result, most correlations between frames can be removed by the inter prediction, except for

the prediction error term, composed of the quantization error and noise. Typically, the prediction error $e_{\mathcal{I}}$ from FW and BW is nearly Laplacian distributed, as reported in [39]. Hence, we assume that the $e_{\rm FW}$'s inside a block have the i.i.d. Laplacian distribution, and so do the $e_{\rm BW}$'s. Also, we assume the data in one block have the same statistical parameters such as mean and variance.

Next, for a specific location (x, y), we like to show that the $e_{\rm FW}(\mathrm{x}, \mathrm{y})$ and $e_{\rm BW}(\mathrm{x}, \mathrm{y})$ pair is jointly Laplacian. Note that although two random variables $\mathcal{X}$ and $\mathcal{Y}$ are marginally Laplacian-distributed, it does not imply the $(\mathcal{X}, \mathcal{Y})$ pair is jointly Laplacian. We adopt a popular goodness-of-fit test to examine the distribution of $(e_{\rm FW}, e_{\rm BW})$. It is the Pearson's χ^2-test

Chi-square test: The χ^2-test divides the data range into K mutually exclusive and exhaustive intervals (events), denoted by $\mathcal{A}_1 \sim \mathcal{A}_K$. The χ^2-test statistic is defined as [40]

$$Q = \sum_{i=1}^{K} \frac{(O_i - mp_i)^2}{mp_i} \tag{A.2}$$

where m is the total number of data samples in a block, O_i is the observed frequency (number of samples) of the event $\mathcal{A}_i$, and mp_i is the expected value of the event $\mathcal{A}_i$ (p_i is the model probability of event $\mathcal{A}_i$). Essentially, the χ^2-test statistic shows the difference between the empirical frequencies and the model-derived mean values.

These two tests measure the similarity between the collected observations and a chosen model distribution. We pick up the following two bivariate distributions to match our collected data.

Bivariate Gaussian distribution: The commonly used bivariate Gaussian distribution is defined as

$$\mathcal{G}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{2\pi|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right). \tag{A.3}$$

where two random variables $\mathcal{X}$ and $\mathcal{Y}$ form the vector $\mathbf{x}$, $\boldsymbol{\mu}$ is the expected value of $\mathbf{x}$, the covariance matrix $\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{\mathcal{X}}^2 & \rho\sigma_{\mathcal{X}}\sigma_{\mathcal{Y}} \\ \rho\sigma_{\mathcal{X}}\sigma_{\mathcal{Y}} & \sigma_{\mathcal{Y}}^2 \end{bmatrix}$, and ρ is the correlation between $\mathcal{X}$ and $\mathcal{Y}$.

Bivariate Laplacian distribution: The bivariate Laplacian distribution has heavier tails than the bivariate Gaussian distribution and its PDF is defined by [41]

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\pi|\boldsymbol{\Sigma}|^{1/2}} K_0\left(\sqrt{2(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}\right) \tag{A.4}$$

where $K_0(\cdot)$ is the modified Bessel function of the second kind.

In the data fitting process, we decide two parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ by adopting the approach of method of moments [42], [43]. Again, the distribution parameters of each block are calculated individually because they may vary from block to block. After the parameters of these two bivariate PDF are determined, we evaluate how well they match the empirical data of pair $(e_{\rm FW}, e_{\rm BW})$.

We examine this goodness-of-fit test in two distinct block sizes, 16 × 16 and 8 × 8. The empirical data are evaluated against these two selected model distributions. In Table A-I,

TABLE A-I
AVERAGE χ^2 TEST-STATISTIC VALUE FOR TEMPORAL ENHANCEMENT LAYER

Test Sequence	Qp	$\mathcal{Q}_{\mathcal{L},16\times16}$		$\mathcal{Q}_{\mathcal{G},16\times16}$		$\mathcal{Q}_{\mathcal{L},8\times8}$		$\mathcal{Q}_{\mathcal{G},8\times8}$	
		T_1	T_4	T_1	T_4	T_1	T_4	T_1	T_4
FOOTBALL	40	33.1	23.8	45.5	39.9	25.8	20.9	42.5	37.2
FOREMAN		29.1	22.7	65.6	53.7	24.9	23.1	61.6	56.3
MOBILE		30.4	21.2	93.3	67.0	26.4	20.9	87.9	68.2
FOOTBALL	30	31.5	23.1	43.6	45.1	22.1	17.2	42.4	38.8
FOREMAN		26.8	18.1	60.8	40.3	20.4	16.1	54.1	36.6
MOBILE		30.7	20.8	105.7	75.8	25.7	19.2	97.7	72.7
AVG.		**25.9**		**61.4**		**21.9**		**58.0**	

the reference model $\mathcal{L}$ is better than the other model $\mathcal{G}$ in terms of the χ^2-test statistic $\mathcal{Q}$. The $\mathcal{Q}$ value of $\mathcal{L}$ usually varies from 16 to 33, but the $\mathcal{Q}$ value of $\mathcal{Q}$ is about 60 on average. We thus conclude that the collected data $(e_{\rm FW}, e_{\rm BW})$ are closer to the bivariate Laplacian distribution. We also use another goodness-of-fit test, Kolmogorov-Smirnov test (KS-test), and the conclusion is similar. Additional test data and discussions can be found in [44].

After we use the jointly Laplacian distribution to model $(e_{\rm FW}, e_{\rm BW})$, we can derive the distribution of $(1/2)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}|e_{\rm FW}(x,y) + e_{\rm BW}(x,y)|$; namely, the distribution of $\widetilde{\mathcal{D}}_{\rm BI}$. According to the property of Laplacian distribution [41], a linear combination $a\mathcal{X} + b\mathcal{Y}$ is one-dimensional Laplacian distribution if $\mathcal{X}$ and $\mathcal{Y}$ are jointly Laplacian. Hence, the term

$$e_{\rm FW+BW}(x,y) \triangleq e_{\rm FW}(x,y) + e_{\rm BW}(x,y) \tag{A.5}$$

is also Laplacian distributed. Moreover, from the probability theory, the absolute value of a Laplacian distribution is exponentially distributed and the sum of i.i.d. exponential distributions forms a gamma distribution, as shown below [45], [46].

$$\frac{1}{2}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left|\underbrace{e_{\rm FW}(x,y) + e_{\rm BW}(x,y)}_{\text{jointly Laplacian}}\right| = \overbrace{\frac{1}{2}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\underbrace{\left|\underbrace{e_{\rm FW+BW}(x,y)}_{\text{Laplacian}}\right|}_{\text{Exponential}}}^{\text{Gamma}} \tag{A.6}$$

Hence, $\widetilde{\mathcal{D}}_{\rm BI}$ should have a Gamma distribution $\Gamma(k, \theta)$, where $k = M \times N$ is the shape parameter and θ is the scale parameter.

REFERENCES

[1] J.-R. Ohm, "Advances in scalable video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 42–56, Jan. 2005.

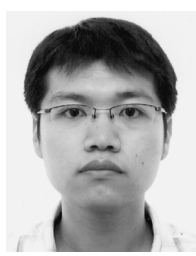
[2] Advanced Video Coding for Generic Audiovisual Services ITU-T H.264 (03/2010), Mar. 2010.

[3] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[4] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz, and M. Wien, Joint Draft ITU-T Rec. H.264 | ISO/IEC 14496-10/Amd. 3 Scalable Video Coding *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-X201*, Jun. 2007.

[5] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[6] J. Reichel, H. Schwarz, and M. Wien, Joint Scalable Video Model JSVM-12 Text *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-Y202*, Oct. 2007.
[7] H. Schwarz, D. Marpe, and T. Wiegand, Hierarchical B Pictures *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-P014*, Jul. 2005.
[8] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," in *IEEE Int. Conf. Multimedia Expo*, 2006, pp. 1929–1932.
[9] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 70–84, Jan. 1999.
[10] T.-Y. Kuo and H.-J. Lu, "Efficient reference frame selector for H.264," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 400–405, Mar. 2008.
[11] A. Chang, O. C. Au, and Y. M. Yeung, "A novel approach to fast multi-frame selection for H.264 video coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2003, pp. III-413–III-416.
[12] Y. Su and M.-T. Sun, "Fast multiple reference frame motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 447–452, 2006.
[13] S.-E. Lim, J.-K. Han, and J.-G. Kim, "An efficient scheme for motion estimation using multireference frames in H.264/AVC," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 457–466, Mar. 2006.
[14] M.-J. Chen, G.-L. Li, Y.-Y. Chiang, and C.-T. Hsu, "Fast multi-frame estimation algorithms by motion vector composition for the MPEG-4/AVC/H.264 standard," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 478–487, Mar. 2006.
[15] L. Shen, Z. Liu, Z. Zhang, and G. Wang, "An adaptive and fast multiframe selection algorithm for H.264 video coding," *IEEE Signal Process. Lett.*, vol. 14, no. 11, pp. 836–839, Nov, 2007.
[16] Huang *et al.*, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 507–522, Apr. 2006.
[17] Z. Liu *et al.*, "Motion feature and hadamard coefficients-based fast multiple reference frame motion estimation for H.264," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 5, pp. 620–632, May 2008.
[18] S.-H. Ri, Y. Vatis, and J. Ostermann, "Fast inter-mode decision in an H.264/AVC encoder using mode and lagrangian cost correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 302–306, Feb. 2009.
[19] Y.-K. Tu, J.-F. Yang, and M.-T. Sun, "Efficient rate-distortion estimation for H.264/AVC coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 600–611, May 2006.
[20] K.-Y. Liao, J.-F. Yang, and M.-T. Sun, "Rate-distortion cost estimation for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 1, pp. 38–49, Jan. 2010.
[21] M. Paul, M. R. Frater, and J. F. Arnold, "An efficient mode selection prior to the actual encoding for H.264/AVC encoder," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 581–588, Apr. 2009.
[22] A. C.-W. Yu, G. R. Martin, and H. Park, "Fast inter-mode selection in the H.264/AVC standard using a hierarchical decision process," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 2, pp. 186–195, Feb. 2008.
[23] H. Wang, S. Kwong, and C.-W. Kok, "An efficient mode decision algorithm for H.264/AVC encoding optimization," *IEEE Trans. Multimedia*, vol. 9, no. 4, pp. 882–888, Apr. 2007.
[24] Y.-M. Lee and Y. Lin, "Zero-block mode decision algorithm for H.264/AVC," *IEEE Trans. Image Process.*, vol. 18, no. 3, pp. 524–533, Mar. 2009.
[25] B.-G. Kim, "Novel inter-mode decision algorithm based on macroblock (MB) tracking for the P-slice in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 2, pp. 273–279, 2008.
[26] H. Zeng, C. Cai, and K.-K. Ma, "Fast mode decision for H.264/AVC based on macroblock motion activity," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 4, pp. 1–10, Apr. 2009.
[27] L. Shen, Z. Liu, Z. Zhang, and X. Shi, "Fast inter mode decision using spatial property of motion field," *IEEE Trans. Multimedia*, vol. 10, no. 6, pp. 1208–1214, 2008.
[28] Z. Liu, L. Shen, and Z. Zhang, "An efficient intermode decision algorithm based on motion homogeneity for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 1, pp. 128–132, Jan. 2009.
[29] H. Li, Z.-G. Li, and C. Wen, "Fast mode decision algorithm for inter-frame coding in fully scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 7, pp. 889–895, Jul. 2006.
[30] B. Lee *et al.*, "An efficient block mode decision for temporal scalability in scalable video coding," in *Proc. SPIE*, 2008, vol. 6822, pp. 68220L-1–68220L-9.
[31] H.-C. Lin, H.-M. Hang, and W.-H. Peng, "Fast temporal prediction selection for H.264/AVC scalable video coding," in *Proc. IEEE Int. Conf. Image Process.*, 2009, pp. 3425–3428.
[32] H.-C. Lin and H.-M. Hang, "Fast algorithm on selecting bi-directional prediction type in H.264/AVC scalable video coding," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2010, pp. 113–116.
[33] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, 1998.
[34] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," in *Proc. IEEE Int. Conf. Image Process.*, 2001, vol. 3, pp. 542–545.
[35] T. Wiegand, H. Schwarz, A. Joch, and F. Kossentini, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–702, 2003.
[36] M. Flierl, T. Wiegand, and B. Girod, "A locally optimal design algorithm for block-based multi-hypothesis motion-compensated prediction," in *Proc. Data Compression Conf.*, 1998, pp. 239–248.
[37] G. Bjontegaard, Calculation of Average PSNR Differences Between RD-curves *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. VCEG-M33*, Apr. 2001.
[38] S. Pateux and J. Jung, An Excel Add-in for Computing Bjontegaard metric and Its Evolution *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. VCEG-AE07*, Jan. 2007.
[39] M. Narroschke, "Extending H.264/AVC by an Adaptive Coding of the Prediction Error," in *Proc. Picture Coding Symposium*, 2006.
[40] R. V. Hogg, J. W. McKean, and A. T. Craig, *Introduction to Mathematical Statistics*, 6th ed. Englewood Cliffs, NJ: Prentice Hall, 2004.
[41] S. Kotz, T. J. Kozubowski, and K. Podgorski, *The Laplace Distribution and Generalizations*. New York: Birkhauser, 2001.
[42] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993, vol. I.
[43] T. Sltoft, T. Kim, and T.-W. Lee, "On the multivariate laplace distribution," *IEEE Signal Lett.*, vol. 13, no. 5, pp. 300–303, 2006.
[44] H.-C. Lin, "Fast encoding algorithm design for H.264/MPEG-4 AVC scalable video coding standard," Ph.D. dissertation, NCTU, Taiwan, Jun. 2010.
[45] Laplace Distribution [Online]. Available: http://en.wikipedia.org/wiki/Laplace_distribution
[46] Exponential Distribution [Online]. Available: http://en.wikipedia.org/wiki/Exponential_distribution



Hung-Chih Lin was born in Nantou, Taiwan, in 1982. He received the B.S. degree in electrical control engineering in 2004, and the M.S. and Ph.D. degree in electronics engineering in 2005 and 2010, all from National Chiao-Tung University (NCTU), Hsinchu, Taiwan.

He currently serves as a senior engineer in MediaTek, Inc., Hsinchu, Taiwan. His work mainly focuses on software video encoder for real-time applications, specifically, encoder optimization using fast motion search and ARM-related SIMD operations. His research interests include numerical matrix computations, digital image/video processing, and scalable video compression. In particular, he specializes in fast algorithm designs and implementations. He has explored on the encoder optimization of the MPEG video coding standards since his M.S. program.

For outstanding accomplishments in these techniques, Dr. Lin received Excellent Achievement Award of Ph.D. Dissertation from Department of Electrical Engineering, NCTU in 2010.

Hsueh-Ming Hang (F'02) received the B.S. and M.S. degrees in control engineering and electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1978 and 1980, respectively, and Ph.D. in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1984.

From 1984 to 1991, he was a Technical Staff Member with AT&T Bell Laboratories, Holmdel, NJ, and then he joined the Electronics Engineering Department of National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1991. From 2006 to 2009, he took a leave from NCTU and was appointed as Dean of the EECS College at National Taipei University of Technology, Taipei, Taiwan. He is currently a Distinguished Professor of the EE Dept and an Associate Dean of the ECE College, NCTU. He holds 13 patents in the U.S., China, and Japan, and has published over 170 technical papers related to image compression, signal processing. Since 1984, he has been actively involved in the international MPEG standards. His current research interests include multimedia compression, image/signal processing algorithms and architectures, and multimedia communication systems.

Dr. Hang was an associate editor (AE) of the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP, 1992–1994), the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (1997–1999), and currently an AE of the IEEE TIP again. He is co-editor and contributor of the *Handbook of Visual Communications* published by Academic Press. He is a recipient of the IEEE Third Millennium Medal and is a Fellow of IEEE and IET and a member of Sigma Xi.

Wen-Hsiao Peng was born in HsinChu, Taiwan in 1975. He received his B.S., M.S., and Ph.D. degrees in Electronics Engineering from National Chiao Tung University (NCTU), Taiwan in 1997, 1999, and 2005, respectively.

From 2000–2001, he was with the Intel Microprocessor Research Laboratory, Santa Clara, U.S., where he developed the first real-time MPEG-4 fine granularity scalability codec and demonstrated its application in a 3-D, peer-to-peer video conferencing. In 2005, he joined the Department of Computer Science, NCTU, where he is currently an Assistant Professor. Since 2003, he has actively participated in the International Organization for Standardization (ISO) Moving Picture Expert Group (MPEG) digital video coding standardization process and contributed to the development of MPEG-4 Part 10 AVC Amd.3 scalable video coding standard. He has published more than 30 technical papers in the field of video and signal processing. His current research interests include high-efficiency video coding, scalable video coding, video codec optimization, and platform-based architecture design for video compression.

Dr. Peng currently serves as a Technical Committee Member for "Visual Signal Processing and Communications" and "Multimedia Systems and Application" tracks for the IEEE Circuits and Systems Society. He organized two special sessions on High-Efficiency Video Coding in ICME 2010 and APSIPA ASC 2010, and served as a Technical Program Co-Chair for VCIP 2011.