Fast Context-Adaptive Mode Decision Algorithm for Scalable Video Coding With Combined Coarse-Grain Quality Scalability (CGS) and Temporal Scalability

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

***Abstract*—To speed up the H.264/MPEG scalable video coding (SVC) encoder, we propose a layer-adaptive intra/inter mode decision algorithm and a motion search scheme for the hierarchical B-frames in SVC with combined coarse-grain quality scalability (CGS) and temporal scalability. To reduce computation but maintain the same level of coding efficiency, we examine the rate-distortion (R-D) performance contributed by different coding modes at the enhancement layers (EL) and the mode conditional probabilities at different temporal layers. For the intra prediction on inter frames, we can reduce the number of Intra4×4/Intra8×8 prediction modes by 50% or more, based on the reference/base layer intra prediction directions. For the EL inter prediction, the look-up tables containing inter prediction candidate modes are designed to use the macroblock (MB) coding mode dependence and the reference/base layer quantization parameters (Qp). In addition, to avoid checking all motion estimation (ME) reference frames, the base layer (BL) reference frame index is selectively reused. And according to the EL MB partition, the BL motion vector can be used as the initial search point for the EL ME. Compared with Joint Scalable Video Model 9.11, our proposed algorithm provides a 20× speedup on encoding the EL and an 85% time saving on the entire encoding process with negligible loss in coding efficiency. Moreover, compared with other fast mode decision algorithms, our scheme can demonstrate a 7–41% complexity reduction on the overall encoding process.**

***Index Terms*—Coarse-grain quality scalability, encoder optimization, fast mode decision, scalable video coding (SVC).**

I. Introduction

IN RESPONSE to the increasing demand for scalability features in many applications, the Joint Video Team has recently, based upon H.264/advanced video coding (AVC) [1], standardized a scalable video coding standard (referred hereafter to as SVC) [2], [3] that furnishes spatial, temporal, signal-to-noise ratio (SNR) and their combined scalabilities within a fully scalable bit stream. By employing *multilayer coding* along with *hierarchical temporal prediction* [4], [5], the SVC encodes a video sequence into an inter-dependent set of scalable layers, allowing a variety of viewing devices to perform discretionary layer extraction and partial decoding according to their playback capability, processing power, and/or network quality. As a scalable extension to H.264/AVC, the SVC inherits all the coding tools of H.264/AVC and additionally it incorporates an *adaptive inter-layer prediction* mechanism for reducing the coding efficiency loss relative to the single-layer coding. A superior coding efficiency is achieved with little increase in decoding complexity by means of the so-called *single-loop decoding*. These key features distinguish the SVC from the scalable systems in the prior video coding standards.

Although the decoding complexity was well studied and amended during the design phase of the SVC, its encoding complexity has rarely been addressed. An SVC encoder, the operations of which are non-normative, can be quite flexible in its implementation, as long as its bit streams conform to the specifications. The current Joint Scalable Video Model (JSVM) v.9 [6] uses a *bottom-up encoding process* that adopts the *exhaustive mode search* for coder parameter selection. The exhaustive search strategy, though providing a good rate-distortion (R-D) performance, spends a large amount of computations on evaluating all possible coding options and it turns out that most of these options have little benefit in increasing coding efficiency. For example, in a typical encoding experiment with the combined temporal and coarse-grain quality scalability (CGS), it takes about 10–40 min of central processing unit (CPU) time (see the test conditions in Section V), depending on the number of enhancement layers (EL), to encode a two-second common intermediate format (CIF) video clip. A further study reveals that a large percentage of computations come from encoding EL; more specifically, a CGS EL requires approximately three times the computations of its base layer (BL) due to the extra motion search for inter-layer motion estimation and residual prediction. A fast encoding algorithm is thus desirable and advisable for reducing the EL computational complexity without sacrificing the R-D performance.

An effective way to reduce the encoding complexity is to restrict the number of candidate modes. There exists a large

Manuscript received April 8, 2009; revised September 16, 2009. First version published January 29, 2010; current version published May 5, 2010. This work was supported in part by the National Science Council, Taiwan, under Grants NSC 96-2221-E-009-063, NSC 95-2221-E-009-146, and NSC 95-2221-E-009-071. This paper was recommended by Associate Editor V. Bottreau.

H.-C. Lin and H.-M. Hang are with the Department of Electronics Engineering, National Chiao-Tung University (NCTU), Hsinchu 30010, Taiwan (e-mail: huchlin@gmail.com; hmhang@mail.nctu.edu.tw).

W.-H. Peng is with the Department of Computer Science, National Chiao-Tung University, Hsinchu 30010, 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/TCSVT.2010.2045832

body of literature devoted to the studies on mode reduction for H.264/AVC. For example, Tsai *et al.* [7] design a set of gradient filters to extract the edge direction, which decides the intra prediction mode to avoid testing all possible directions. They further improve the mode detection accuracy in texture areas by computing the intensity difference at both subblock and pixel levels [8]. Another example of using macroblock (MB) features to predict mode sets can be found in [9]. They first classify MBs into three categories according to their inter, intra, and motion features, and then for each category a risk-minimized candidate mode set is designed by using the Bayesian rules. Similarly, Zeng *et al.* [10] pick up the mode set for each MB based on its motion activity. There are some other mode reduction approaches that exploit the spatial and temporal correlation between MB modes. Their processes usually predict the most probable MB mode by observing the coding mode of its nearby MBs [11] or of its co-located MB in the previous frame [12]. Similar concepts are adopted to develop early termination conditions in the mode decision process. For example, a skip decision scheme is designed based on the conditions of evaluating various inter/intra modes [13]. These type of techniques have often been generalized to a hierarchical decision process with multiple termination criteria in [12], [14], and [15]. All these methods are equally applicable to the intra-layer mode reduction in SVC.

Thus far, little research has been devoted to the study of the SVC fast mode decision. Most of published articles use the *inter-layer correlation* to confine the mode search at the EL. Li *et al.* [16], [18], for example, observe that owing to the Lagrange R-D optimization process, the inter MB motion partition at EL tends to be the same as or smaller than that of its corresponding BL MB. This observation is used in conjunction with the BL mode decision to design a fast mode search for the EL. In [17], the complexity reduction is made a step further, by considering both the spatial homogeneity of the mode distribution and its consistency across temporal layers. In [19], Ren *et al.* notice a high correlation exists in spatially neighboring MBs. Thus, they develop an intra-layer fast algorithm without considering the inter-layer relationship. For each coding layer, their method collects the local area's best partition with R-D costs to progressively perform the mode search for each MB until an early termination condition is satisfied. Some other previous work has been associated with the intra MB mode reduction. Yang *et al.* [20] show that the *inter-layer intra prediction* can effectively replace Intra16×16 and Intra8×8 modes. On top of that, Xiong [21] makes an additional simplification by restricting the Intra4×4 prediction to three options only: vertical, horizontal, and DC modes. Through the effective use of the *inter and/or intra-layer correlation* between coding modes, an average computing time saving of 40–60% (in comparison with JSVM 9.11 [6]) has been reported at the cost of 1–4% bit-rate increase for typical test sequences.

However, in determining the reduced candidate mode set for EL, most existing approaches have not yet considered the following issues, leading to a loss of R-D performance and/or a waste of computational power.

1) *The effect of layer settings on the mode distribution at EL.* In our previous studies [22], [23], we noticed that the quality of BL affects the reliability on the candidate mode prediction, and that an EL, when coded at a much higher bit-rate than its BL, may have a completely different behavior in mode selection. The candidate mode set must therefore be adaptively adjusted for different layer settings. The need for this adjustment becomes most obvious in the multilayer coding scenarios, where the quantization parameters (Qp) values and the inter-layer dependence change on a layer-to-layer basis.
2) *The correlation between the motion parameters of BL and EL.* As also shown in our previous studies [22], [23], an EL (inter) MB usually has the same *reference frame index* and *prediction direction* as its co-located MB at BL, especially when both are coded with the same MB partition. In this regard, the exhaustive motion search (adopted by most previous researchers) may not be needed for reaching the target R-D performance.

Based on the above observations, we propose in this paper a fast context-adaptive mode decision algorithm and a reduced-complexity motion search strategy for SVC with combined CGS and temporal scalability. Our scheme distinguishes from the other approaches in two significant ways: 1) the candidate mode set for each EL MB is chosen according to both local and global contexts—including the coding mode adopted by its co-located MB at BL, the Qp assigned to BL and EL, as well as its temporal layer index; and 2) the search for motion parameters, for a particular candidate mode, is conducted only when the BL motion information is not reusable. That is, the exhaustive motion search is performed only when the BL motion information is judged unreliable for that EL. Compared with JSVM 9.11 [6], our method shows an overall time reduction of 65–85% with a minor bit-rate increase of less than 1%. The computational complexity for coding the EL alone is reduced to 10% of that of the JSVM implementation. Compared with the state-of-the-art fast algorithms, [16], [18], [19], an up to 41% improvement can be achieved solely by the use of *inter-layer correlation*; further improvement is expected when the *intra-layer correlation* is also incorporated.

The rest of this paper is organized as follows. Section II contains a brief review of the JSVM coder control and it discusses the coding complexity analysis of BL and EL. Section III analyzes the correlation between the mode distributions of BL and EL. Section IV describes our context-adaptive mode decision algorithm, and also presents our motion search strategy. Section V compares the proposed schemes with JSVM and the other state-of-the-art algorithms in terms of complexity reduction and R-D performance. Lastly, this paper is concluded with a summary of our observations and future work.

II. Coding Tools and Coder Control

To have a better understanding of our coding algorithms, this section explains the basic concepts of SVC and its coder control. Some degree of familiarity with H.264/AVC is

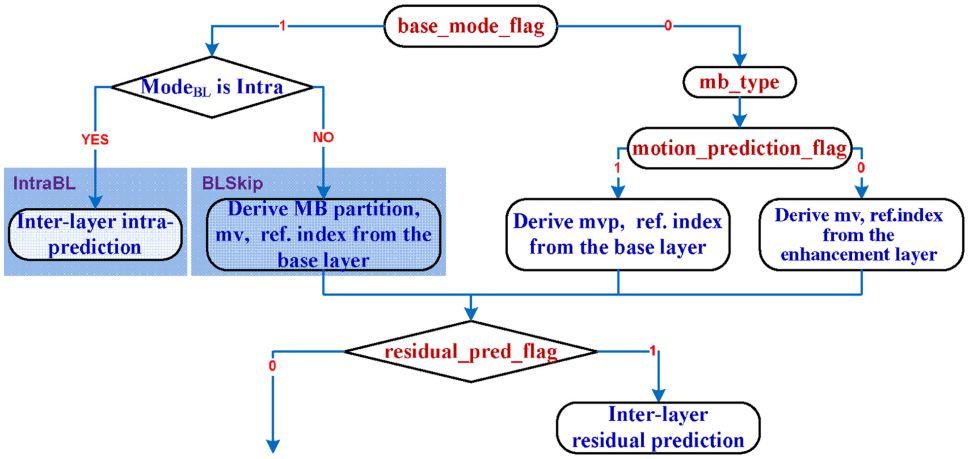

Fig. 1. Syntax elements and their combinations for the inter-layer prediction in the CGS [2].

assumed herein. The reader is referred to the overview paper [3] for details of H.264/AVC and its scalable extension.

A. Coding Tools

In order to support the spatial, temporal, and fidelity (SNR) scalabilities, the SVC encodes a video sequence into a layer-dependent set of scalable layers. Along the temporal axis, a group of pictures (GOP) is decomposed into a temporal base layer T_0 and one or more temporal enhancement layers $\{T_k|k > 0\}$ in a nested, hierarchical fashion. Frames belonging to a lower temporal layer T_l are coded independently of the higher temporal layers $\{T_h|h > l\}$ For the applications that require lower temporal frame rates, only the frames that constitute the needed lower layers are decoded. In principle, the temporal frame rate (temporal prediction structure) does not have to be dyadic. The prediction structure can be modified as needed and can vary over time to support irregular, non-dyadic scalability. In this paper, however, we consider only the dyadic temporal scalability case so that we can use the current release of JSVM software.

In the spatial dimension, the SVC adopts the conventional approach of image pyramid to represent a source video sequence at various spatial resolutions. The spatial encoding process begins with a multiresolution decomposition of the original high-resolution sequence. The lowest-resolution sequence is coded by H.264/AVC as the BL, and each higher resolution sequence is coded sequentially as a spatial EL. A specified spatial resolution image is reconstructed at the decoder when all its designated layers are received. A similar philosophy is carried over to facilitate the quality (SNR) scalability. In this scalability mode, the BLs and the EL have identical spatial resolutions, but different quantization step sizes.

To achieve the high coding efficiency goal, the SVC has an adaptive inter-layer prediction mechanism [24], which allows the decoded information of the reference/base layer to be reused in the following three different ways.

1) *Inter-Layer motion Prediction:* To avoid repeatedly sending the same motion parameters in the cases when the EL cannot benefit from motion refinement, a flag (*base_mode_flag*) can be sent for each non-skipped MB to indicate whether its motion parameters (MB mode, reference frame indices, and motion vectors) are to be inferred from the reference/base layer. In the other cases when it is more efficient to change the MB mode but leave most of the other parameters unchanged, another flag (*motion_prediction_flag*) can be additionally sent for a reference picture list to signal whether the reference frame index and motion vector (MV) are predicted from the reference/base layer.
2) *Inter-Layer Texture Prediction:* To provide a better prediction for the EL samples, especially for the fast-motion sequences, the reconstructed samples of the reference/base layer can be used as an alternative prediction source. However, the texture prediction is available only when the co-located MB is an intra-coded MB with *constrained intra prediction*, because the *single-loop* structure prohibits the reference/base layer to conduct motion compensation after it being coded.
3) *Inter-Layer Residual Prediction:* To enhance the coding efficiency of inter-coded MB within the framework of *single-loop decoding*, the residual prediction, which subtracts the residual signal of the reference/base layer from that of the EL, can be adaptively activated by the *residual_prediction_flag*.

Despite certain restrictions, these inter-layer prediction tools can be combined together to form a number of coding modes for each EL MB. Fig. 1 shows all possible combinations of the *base_mode_flag*, *motion_prediction_flag*, and *residual_prediction_flag*, as well as their associated coding modes.

B. Coder Control

The task of the coder control is to choose, for each MB, the most efficient coding mode in the R-D sense. Similar to the Joint Model of H.264/AVC, JSVM also adopts a Lagrangian-based coder control. The best mode is decided by minimizing a Lagrangian cost function $J_{MODE}(m) = D_{MODE}(m) + \lambda_{MODE} R_{MODE}(m)$ that weights the distortion $D_{MODE}(m)$ of an MB against the bit usage $R_{MODE}(m)$ using a Lagrangian

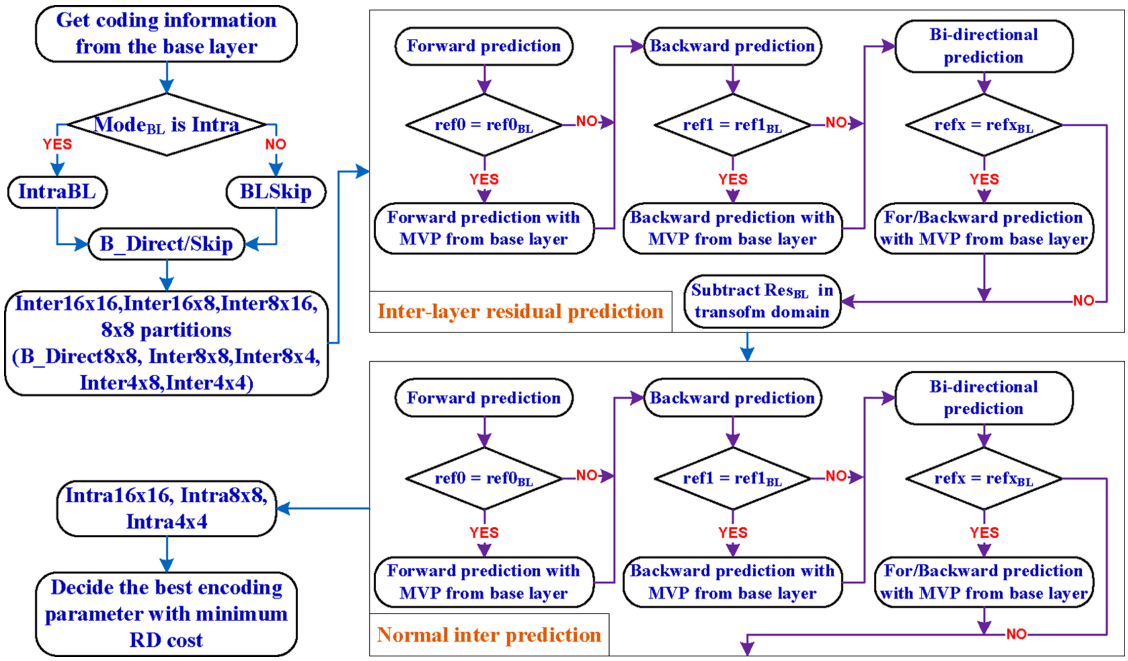

Fig. 2. Flowchart of mode decision at EL for hierarchical B-frames in JSVM 9.11 [6].

multiplier λ_{MODE}. The function value is determined by the mode index m. To achieve the best R-D performance, JSVM actually encodes each MB using all possible coding modes and picks up the best one.

In the mode selection process, the evaluation of inter modes demands much more computations than that of intra modes. Thus, we first try to simplify the inter mode selection, which includes the MV, $\mathbf{d}(\mathbf{x})$, selection. In this case, the Lagrangian cost function consists of two terms: 1) a distortion term due to the displaced frame difference $D_{DFD}(\mathbf{d}(\mathbf{x}))$; and 2) a bit-rate term $R_{MOTION}(\mathbf{d}(\mathbf{x}))$ used for representing the MV difference. Therefore, the MV search criterion becomes $J_{MOTION}(\mathbf{d}(\mathbf{x})) = D_{DFD}(\mathbf{d}(\mathbf{x})) + \lambda_{MOTION} R_{MOTION}(\mathbf{d}(\mathbf{x}))$. It makes a trade-off between bit-rate and distortion in choosing $\mathbf{d}(\mathbf{x})$ A bias, when properly added, can help to improve the MV regularity and the accuracy in estimating true motion. However, it also complicates the motion search process. Furthermore, there are many possible MV predictors allowed in the SVC standard (such as inter-frame prediction and inter-layer motion prediction). All together, numerous options are possible and the computational complexity is extremely high.

The MV search criterion must be carefully adjusted in the case of inter-layer residual prediction. In this case, an encoder should minimize the residual difference signal rather than the residual signal in evaluating each MV candidate. For this purpose, a preprocessing step is conducted by JSVM. In the spatial scalability, the decoded residual signal $\epsilon(\mathbf{x}, t)$ is upsampled as $\epsilon_U(\mathbf{x}, t)$ and then it is subtracted from the source image $I(\mathbf{x}, t)$. Finally, an optimal MV is estimated based on the EL reference frame $I_E(\mathbf{x}, t^-)$ and the preprocessed source image $I_E(\mathbf{x}, t) - \epsilon_U(\mathbf{x}, t)$. That is, a modified cost function now becomes the search criterion, namely

$$\breve{J}_{MOTION}(\mathbf{d}(\mathbf{x})) = \breve{D}_{DFD}(\mathbf{d}(\mathbf{x})) + \lambda_{MOTION}\, \breve{R}_{MOTION}(\mathbf{d}(\mathbf{x}))$$

where $\breve{D}_{DFD}(\mathbf{d}(\mathbf{x})) = I(\mathbf{x}, t) - \epsilon_U(\mathbf{x}, t) - I_E(\mathbf{x} - \mathbf{d}(\mathbf{x}), t^-)$. Similarly, the inter-layer residual prediction in CGS has to be evaluated in the transform domain.

C. Complexity

The motion search for inter modes is the major source of encoding complexity and deserves further analysis. Fig. 2 describes the mode decision process of JSVM, with an elaboration on the motion search procedure. It starts with the inter-layer residual prediction. For each admissible MB partition, the search for its motion parameters begins with a series of motion estimation (ME) processes that use $\breve{J}_{MOTION}$ as the search criterion. Thus, an optimal combination of MVs, reference picture indices, and prediction modes is first found in the inter-layer residual prediction. In the second part of the procedure, all the ME processes are repeated with a replaced search criterion. Now, J_{MOTION} is used in place of $\breve{J}_{MOTION}$ to signal that now the prediction does not use the residual signal. In both parts, the inter-layer motion prediction is checked for improvements. Thus, the MV of the co-located MB in the reference/base layer is always examined. In summary, the motion search involves four types of ME processes. Each of them is dedicated to an MV search with a specific use of motion vector prediction and residual prediction.

1) ME_R: ME dedicated to the MV search with residual prediction.
2) ME_M: ME dedicated to the MV search with motion prediction.
3) ME_{R+M}: ME dedicated to the MV search with both residual and motion predictions.
4) ME_O: ME without residual and motion predictions.

The single-layer coding only perform ME_O, but the SVC can do all four types of ME processes, which explains the prolonged latency needed for SVC encoding. Based on this observation we expect that the complexity ratio of an EL

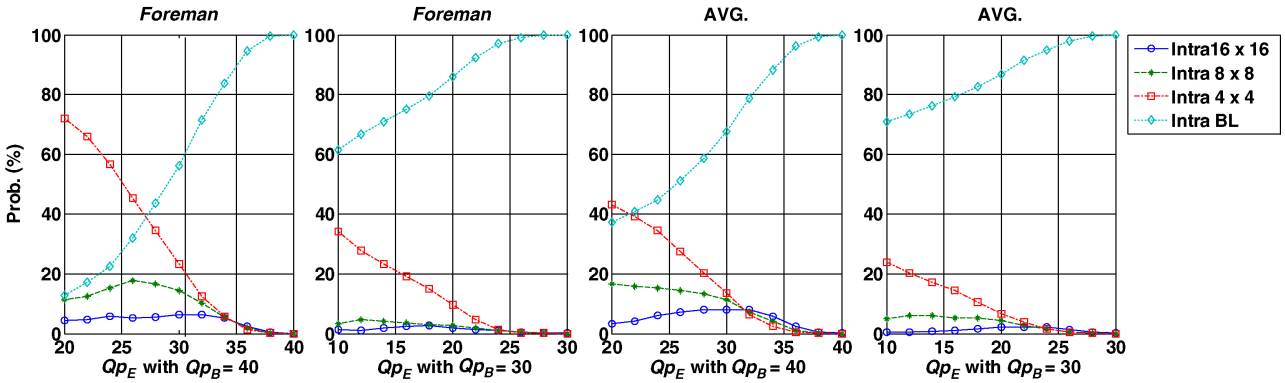

Fig. 3. Distribution of intra prediction types at EL (CGS configuration).

(CGS) to its base layer is 4 to 1. However, our experiments show that the actual CPU time ratio is about 3.2 to 1. This is due to several simplifications made on the coder control scheme in JSVM. For instance, the ME_M and ME_{R+M} processes are not always turned on; they are activated only when the *inter-layer motion prediction* is applicable. Similarly, ME_R and ME_{R+M} are turned on only when the residual signal of the reference/base layer is non-zero. These simplifications help in reducing the computational complexity, but there is still plenty of room for further reduction.

III. Correlations Between Base and Enhancement Layers

In this section, we are going to investigate the relationship between the BL coding modes and the EL coding modes, with a focus on the CGS configuration. We like to know from the statistical analysis that: (1) which intra/inter modes are the EL dominating modes; (2) how these modes are distributed when the BL mode is given; and (3) which coding modes are most critical to the EL R-D performance. In addition, we examine the statistics of the reference frame selection and the inter-layer residual predictor efficiency. Our codec contains one BL and one CGS EL and is tested on six video sequences: *Akiyo* [quarter common intermediate format (QCIF)], *Stefan* (QCIF), *Foreman* (CIF), *Mobile* (CIF), *City* (4CIF), and *Crew* (4CIF). The notations Qp_B and Qp_E denote the quantization parameters of BL and EL, respectively, and Avg. shows the averaged behavior of all six test sequences.

A. Distributions of Intra Prediction Mode in CGS

Our first study aims at exploring the effect of Qp value on the correlation of intra prediction types/modes between coding layers. In Fig. 3, the distribution of the EL intra modes is displayed as a function of Qp_B and Qp_E. We can see that the distribution is highly dependent on the quality of BL and EL. When the BL is coded with good quality (using a small Qp_B), most of the intra MBs are coded in the IntraBL type, whose predictor comes from the BL intra-coded MB. However, when the EL quality gradually improves, the intra predictor is switched from BL to EL. Particularly, the Intra4 $\times$ 4 percentage increases more noticeably than the other two types, Intra8 $\times$ 8 and Intra16 $\times$ 16; together with the IntraBL, it makes up 80% or more of the intra prediction types at EL. Its percentage can be higher than 90%, especially in the complex-texture sequences such as *Mobile* and *Stefan*. Our results agree with the findings reported in [21]. In addition, the Intra16 $\times$ 16 is preferred for smooth areas, but its presence is usually less than 10% at the CGS EL because it must compete with the IntraBL mode, which is chosen more often in the smooth areas due to less overhead. As the BL quality improves, the Intra8 $\times$ 8 and the Intra16 $\times$ 16 do not seem to offer benefit in coding efficiency.

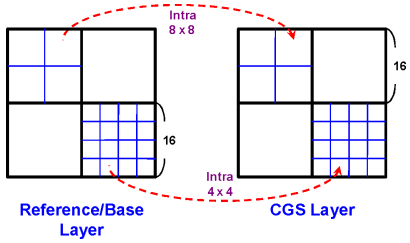

Fig. 4. One-to-one block address mapping of CGS.

In addition to the intra prediction type, we compare the nine prediction directions in intra coding when both layers are coded by either Intra4 $\times$ 4 or Intra8 $\times$ 8. Specifically, an EL coding block is said to have a *similar* prediction mode to its counterpart at BL if the best prediction comes from the same or neighboring directions, or if it uses the DC mode. For instance, if the coding block at BL selects the Vertical mode and the one at EL picks up either Vertical, Vertical Right, Vertical Left, or DC predictions, these two blocks are called *similar* in prediction direction. The similarity check requires locating the BL counterpart of a coding block. As shown by Fig. 4, this process can be implemented by a one-to-one block address mapping in the CGS configuration.

Fig. 5 shows the probability of BL and EL having *similar* intra prediction modes for fixed Qp_B and a set of Qp_E values ranging from $(Qp_E - 20)$ to Qp_B. From this data we can conclude that the intra prediction modes between BL and EL are strongly correlated and, on the average, 75% or higher block pairs adopt *similar* prediction modes. Moreover, this correlation becomes even stronger when Qp_E is closer to Qp_B and this tendency does not seem to be affected by the BL quality and the test sequence.

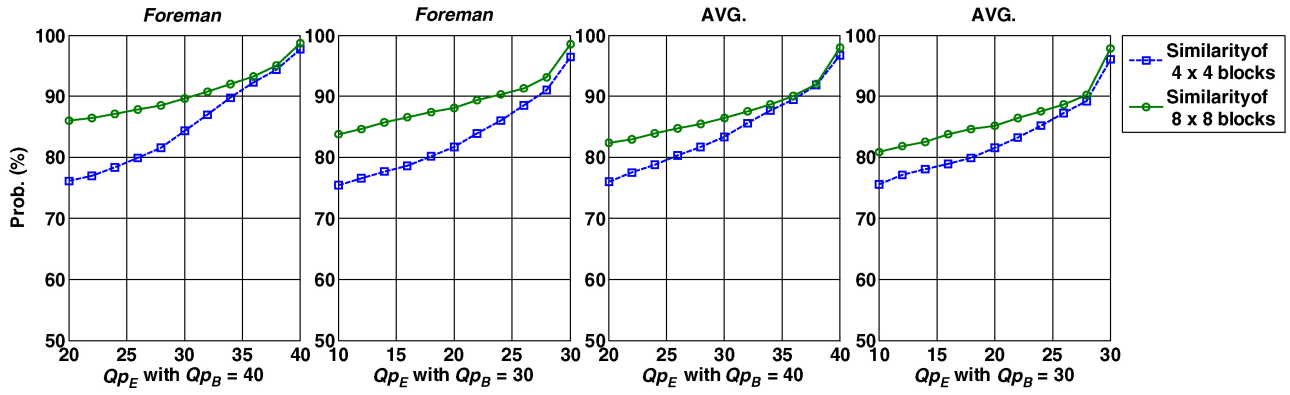

Fig. 5. Similarity probability profiles in CGS with poor-quality BL ($Qp_B = 40$) and high-quality BL ($Qp_B = 30$).

B. Distributions of Inter Prediction Mode in CGS

Next, we investigate the correlation of the motion partition between BL and EL, under different Qp values and prediction distances. To this aim, we collect the conditional probability of partition modes at different temporal enhancement layers with $Qp_B = 40$ and Qp_E varying from 20 to 40. This conditional probability is defined by

$$\mathbf{Pr}\left\{\text{Mode}_{\text{EL}}(Qp_E, T_k) = j | \text{Mode}_{\text{BL}}(Qp_B = 40, T_k) = i\right\}$$

where $\text{Mode}_{\text{BL}}(Qp_B = 40, T_k)$ denotes the best mode selected by the BL with Qp_B at temporal layer k; $\text{Mode}_{\text{EL}}(Qp_E, T_k)$ is the optimal mode at the EL with Qp_E at temporal layer k; $i \in$ {B_Direct/Skip, Inter16 × 16, Inter16 × 8, Inter8 × 16, Inter8 × 8}; and $i \in$ {B_Direct/Skip, Inter16 × 16, Inter16 × 8, Inter8 × 16, Inter8 × 8, Intra, BLSkip}. The collected statistics are given in Fig. 6(a)–(e). In addition, in Fig. 6(f), a different conditional probability is defined as

$$\mathbf{Pr}\left\{\text{SubMode}_{\text{EL}}(Qp_E, T_k) = m | \text{Mode}_{\text{EL}}(Qp_E, T_k) = \text{Inter8} \times 8\right\}$$

where $m \in$ {B_Direct8 × 8, Inter8 × 8, Inter8 × 4, Inter4 × 8, Inter4 × 4}. This conditional probability presents the distribution of the finer partitions including 8 × 8 and those smaller than 8 × 8.

From Fig. 6, several important observations can be made as follows.

1) More than 50% of MB pairs choose the same motion partition for both BL and EL, namely,

   $\mathbf{Pr}\left\{\text{Mode}_{\text{EL}}(Qp_E, T_k) = \text{BLSkip} | \text{Mode}_{\text{BL}}(Qp_B = 40, T_k) = i\right\} +$

   $\mathbf{Pr}\left\{\text{Mode}_{\text{EL}}(Qp_E, T_k) = i | \text{Mode}_{\text{BL}}(Qp_B = 40, T_k) = i\right\} > 0.5.$

   Among them, the EL MB can be coded in either BLSkip mode or the other inter modes, which may or may not use inter-layer motion prediction. The BL_Skip mode is chosen most often especially at higher temporal enhancement layers. The second and the third most probable modes are B_Direct/Skip and Inter16 × 16, respectively. This observation is slightly different from those in [16]–[18], which suggest the EL candidate mode generally does not have partition size larger than its co-located BL MB mode. Interestingly, if the BL MB chooses Inter8 × 8 mode, the choice for the EL MB is also likely (>70%) to be the same. These results seem to be independent of the Qp difference, $(Qp_B - Qp_E)$.
2) When a BL MB is coded in B_Direct/Skip mode, its co-located EL MB is often coded in either B_Direct/Skip or Inter16 × 16.
3) If a BL MB is coded with the 8 × 16 (or 16 × 8) partition, it is unlikely that its EL counterpart will choose the 16 × 8 (or 8 × 16) partition.
4) The probability for an EL MB to be coded in BLSkip mode is greater than 0.5 at the two highest temporal layers, T_{N-1} and T_N.
5) The probability for an EL sub-MB having a sub-partition finer than 8 × 8 is usually less than 0.2. Even though the *Mobile* and *Stefan* have more MBs coded with finer partitions, on the average, 70% of sub-MBs still select the B_Direct8 × 8 and Inter8 × 8 as their sub-partition modes.
6) $\mathbf{Pr}\{\text{SubMode}_{\text{EL}}(Qp_E, T_k) = \text{Inter4} \times 4 | \text{Mode}_{\text{EL}}(Qp_E, T_k) = \text{Inter8} \times 8\} < 0.05$: Our experimental data reveal that when an EL MB is further partitioned into sub-partitions smaller than 8 × 8, the conditional probability of Inter4 × 4 is typically less than 0.05, whereas it can increase to 0.1 for the sequences *Mobile* and *Stefan*.

Fig. 6(a)–(e) also shows that the most probable mode in the hierarchical B-frames is the BLSkip mode. This is a direct consequence of the Lagrangian R-D optimization process, which looks for a balanced compromise between distortion and coding rate. To achieve a better quality, an EL MB may search for new MVs with the same-size partition or additional MVs offered by finer partitions. However, these two alternatives may require extra coding bits. Statistically, using the lower layer information as much possible seems to be a good policy for the mode decision at EL, especially in the CGS configuration because it has the benefits of reducing the number of candidate modes. This is most obvious when the BL is coded with good quality using a small Qp_B. In such a case, the conditional probability

$$\mathbf{Pr}\left\{\text{Mode}_{\text{EL}}(Qp_E, T_k) = i | \text{Mode}_{\text{BL}}(Qp_B = 30, T_k) = i\right\} +$$

$$\mathbf{Pr}\left\{\text{Mode}_{\text{EL}}(Qp_E, T_k) = \text{BLSkip} | \text{Mode}_{\text{BL}}(Qp_B = 30, T_k) = i\right\}$$

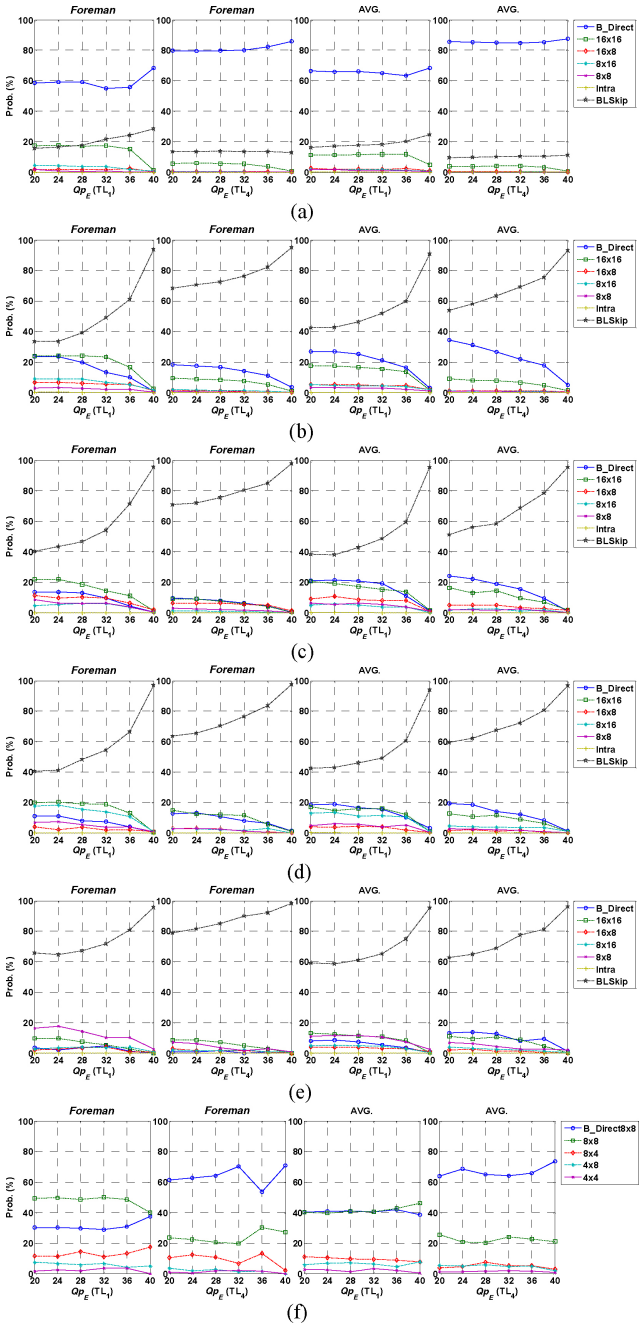

Fig. 6. Mode conditional probability distribution in CGS for Qp_B = 40, Qp_E between 20 and 40, and GOP size = 16. (a) Conditional probability with Mode_{BL} = B_Direct/Skip. (b) Conditional probability with Mode_{BL} = Inter16 × 16. (c) Conditional probability with Mode_{BL} = Inter16 × 8. (d) Conditional probability with Mode_{BL} = Inter8 × 16. (e) Conditional probability with Mode_{BL} = Inter16 × 8Mode_{BL} = Inter8 × 8. (f) Distribution of sub-partition at EL.

can go higher than 0.9, making it possible to skip more coding modes with different partition size from that at BL. Furthermore, the inter-layer relation represented by $\mathbf{Pr}\left\{\text{Mode}_{\text{EL}}(Qp_E, T_k) = i | \text{Mode}_{\text{BL}}(Qp_B, T_k) = i\right\}$ becomes stronger as the index of temporal layer increases. We thus divide the mode conditional probabilities into four regions along two dimensions, the temporal layer and the quantization parameter Qp_{n-1} of the reference layer, as illustrated by Fig. 7. High conditional probabilities appear at small Qp_{n-1} and

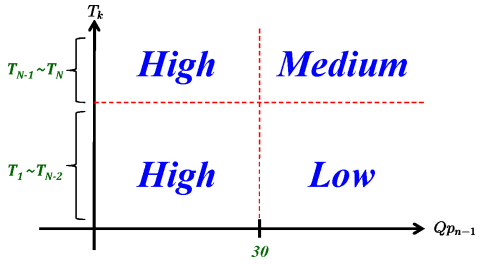

Fig. 7. Four regions representing different degrees of mode correlations between coding layers.

higher temporal layers. In our scheme, T_{N-1} and T_N refer to the highest two temporal enhancement layers in a GOP. For a small GOP size, such as 4, it is possible that all the temporal enhancement layers belong to the $T_{N-1} \sim T_N$ category.

In summary, the BL coding information can be a good reference for predicting the EL coding mode in the CGS configuration. Generally, which coding mode would be the best for a BL MB depends highly on the image texture. However, the conditional probabilities of the EL modes do not vary drastically with video content. In other words, the inter-layer mode correlation is nearly content-independent in the sense that when conditioned by the BL modes, the distribution of the EL modes has a weak dependence on video content. Therefore, in Section IV we will use these observations to design our fast EL mode decision algorithm.

C. Temporal Reference Frames Between Coding Layers

As described before, the ME operation in the hierarchical B-frames needs to find the best match among three types of temporal predictions, namely, forward, backward, and bi-directional predictions. The motion search finds the best MV in all reference frames for each of these temporal predictions. Moreover, the EL should additionally perform the ME_{R}, ME_{M}, and $\text{ME}_{\text{R+M}}$ modes calculation and selection. Then, based on the R-D cost, the encoder finally chooses the best temporal prediction type and its associated MVs for a specified inter coding mode. The current JSVM 9.11 [6] adopts the exhaustive motion search at the EL, leading to enormously high complexity.

To reduce the EL computational load but to maintain good temporal prediction performance, we examine the temporal prediction reference frame selection between BL and EL. The experiments are performed with reference frame number = 3 and GOP size = 16. As shown in Fig. 8, 80% or higher EL MBs choose the same reference frames as their BL counterparts. Moreover, the percentage increases as the BL quality improves. In other words, the reference frames selected at BL can be reliably reused for EL MBs particularly when the BL is coded with good quality.

D. Inter-Layer Residual Prediction in Transform/Pixel Domain

The inter-layer residual prediction is designed to reduce the inter-layer redundancy in residual signals between two layers. Starting from version 8.10 of the JSVM software, the inter-layer residual prediction has been converted from the pixel

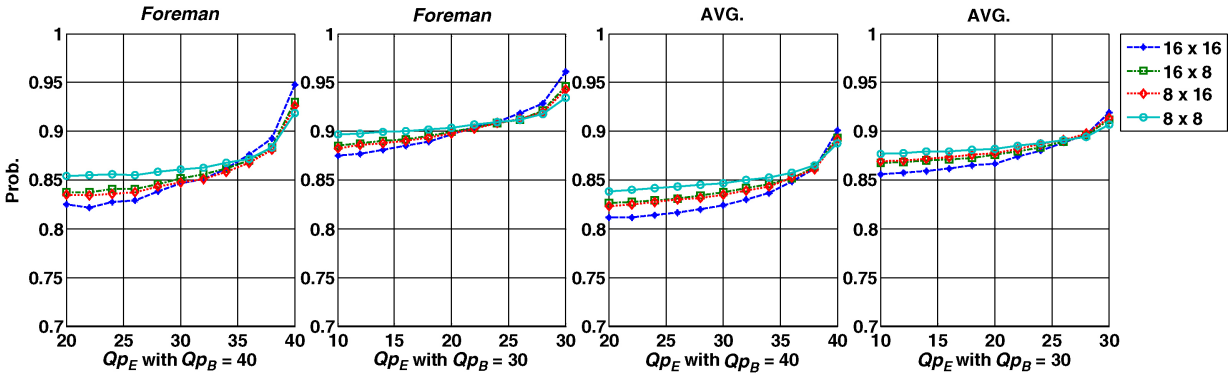

Fig. 8. Consistency in selecting reference frames between BL and EL.

TABLE I
ENCODING PROCEDURES ON THE HIERARCHICAL B-FRAMES AT EL

	JSVM 9.11 [6]	JSVM 8.0 [25]
Step 1	The reconstructed transform coefficients from the base layer: $\mathbf{T}(\epsilon(\mathbf{x}, t))$ The predicted signal: original frame $I(\mathbf{x}, t)$ The reference signal: reference frame $I_E(\mathbf{x}, t^-)$	The reconstructed residual signal (pixel domain) from the base layer: $\epsilon(\mathbf{x}, t)$ The predicted signal: $I(\mathbf{x}, t) - \epsilon(\mathbf{x}, t)$ The reference signal: reference frame $I_E(\mathbf{x}, t^-)$
Step 2	Motion estimation to find the best match $I_E(\mathbf{x}+\mathbf{d}(\mathbf{x}), t^-)$ with the associated MV $\mathbf{d}(\mathbf{x})$	
Step 3	Determine the residual signal: $\epsilon_E(\mathbf{x}, t) = I(\mathbf{x}, t) - I_E(\mathbf{x}+\mathbf{d}(\mathbf{x}), t^-)$	Determine the residual signal: $\epsilon_E(\mathbf{x}, t) = [I(\mathbf{x}, t) - \epsilon(\mathbf{x}, t)] - I_E(\mathbf{x}+\mathbf{d}(\mathbf{x}), t^-)$
Step 4	Integer transform: $T_t = \mathbf{T}(\epsilon_E(\mathbf{x}, t)) - \mathbf{T}(\epsilon(\mathbf{x}, t))$	Integer transform: $T_P = \mathbf{T}(\epsilon_E(\mathbf{x}, t))$
⋮	⋮	

domain to the transform domain in the CGS configuration. As for the spatial scalability, the inter-layer residual prediction has to be operated in the pixel domain. Hence, there are now two different mechanisms in performing the inter-layer residual prediction depending on the configuration.

Extensive experiments have been conducted to analyze the coding improvement provided by the inter-layer residual predictions, performing in pixel domain (JSVM 8.0 [25]) and in transform domain (JSVM 9.11 [6]) respectively, on the hierarchical B-frames at the CGS EL. For CGS applications, the experimental results show that the inter-layer residual prediction in transform domain does not offer a significant coding improvement on the hierarchal B-frames, especially when $(Qp_B - Qp_E)$ is greater than six. The reason could be that the temporal correlations between the hierarchical B-frames at the CGS EL are much stronger than the correlations between coding layers. Moreover, when $(Qp_B - Qp_E)$ is greater than six, the reconstructed residual signal at the BL is noise-like signal for the CGS EL. Similar results can be also observed in the pixel-domain inter-layer residual prediction.

Furthermore, the average coding improvement that uses pixel-domain inter-layer residual prediction is slightly greater than that of adopting transform-domain scheme. The minor coding improvement due to inter-layer residual prediction in the transform domain may be attributed to the encoding procedure in the JSVM software, described in Table I. At the EL, the JSVM 8.0 encoding procedure for inter-layer residual prediction [25] finds the best block match $I_E(\mathbf{x} - \mathbf{d}(\mathbf{x}), t^-)$ from the reference frame $I_E(\mathbf{x}, t^-)$ for the predicted signal $I(\mathbf{x}, t) - \epsilon(\mathbf{x}, t)$. On the other hand, JSVM 9.11 [6] performs the motion search between $I(\mathbf{x}, t)$ and $I_E(\mathbf{x}, t^-)$, and then subtracts $\mathbf{T}(\epsilon(\mathbf{x}, t))$ in determining the R-D cost. Note that $\mathbf{T}(\cdot)$ indicates the integer transform operation. Thus, in JSVM 9.11 [6], the selected MV at EL is optimized only for the difference between the current MB and its reference MB without considering $\mathbf{T}(\epsilon(\mathbf{x}, t))$.

Although, in the spatial scalability, the coding efficiency of the test sequence CREW with GOP size 16 is very close to that of single-layer coding [3], there is no significant coding gain provided by the inter-layer residual prediction for CGS, especially with a low-quality BL. In conclusion, the inter-layer residual prediction, whether in transform domain or pixel domain, can be neglected in encoding the hierarchical B-frames in CGS configuration. That is, the penalty of coding loss can be neglected even if the inter-layer residual prediction is disabled, particularly, when the visual quality of BL is poor.

IV. PROPOSED MODE DECISION ALGORITHM

Based on the observations presented in Section III, we develop a fast context-adaptive mode decision algorithm and a motion search scheme for the hierarchical B-frames in the SVC with combined CGS and temporal scalability. The proposed algorithm is designed based on the mode conditional distributions and we also carefully make the trade-off between computational complexity and R-D performance at EL. Moreover, we skip the coding modes that are not used often and that have a little contribution to the R-D performance. Our algorithms are described by the flowcharts in Figs. 9–11. The BL is encoded with the exhaustive search (or a fast search

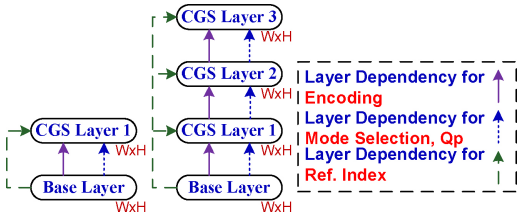

Fig. 9. Inter-layer dependence settings in our study: Two-layer case and four-layer case.

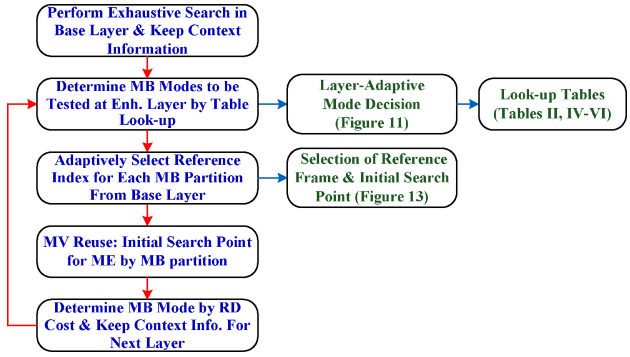

Fig. 10. Flowchart of the proposed algorithm.

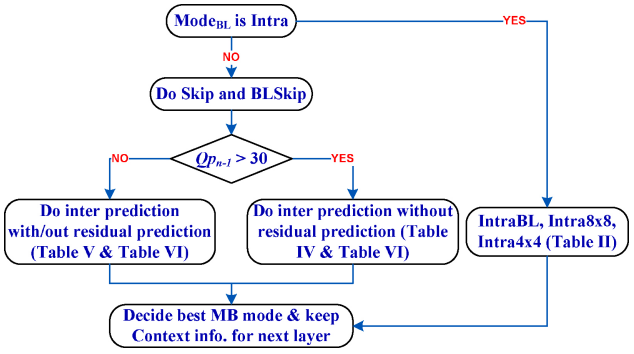

Fig. 11. Layer-adaptive mode decision.

scheme having a nearly full search performance) and all the motion information for each possible combination is stored for future use. The intra and inter coding candidate modes to be checked at EL are defined by Table II and Tables IV–VI sorted by the Qp values and the BL coding modes. Furthermore, the algorithm in Fig. 13 determines the reference frames for motion search and, depending on the partition sizes; the initial point is also adaptively generated by the MVs at BL or the MV predictor at EL. These procedures are described in the following subsections.

A. Layer-Adaptive Mode Decision for Hierarchical B-Frames

1) *Intra Mode Selection:* Due to the strong correlation between the coding layers in the Intra4 × 4 and Intra8 × 8 modes, the EL MB skips the less probable prediction modes by checking their reference/base layer coding states. As suggested by our statistical analysis, 75% or more Intra4 × 4/8 × 8 coding blocks choose prediction modes *similar* to their counterparts at the reference/base layer. This strong correlation is used to design Table II for the layer-adaptive intra mode selection.

TABLE II
LOOK-UP TABLE FOR LAYER-ADAPTIVE INTRA MODE SELECTION

Candidate Modes		Prediction Modes of Reference Layers: Intra 4 x 4/Intra 8 x 8								
		0	1	2	3	4	5	6	7	8
Intra 4 x 4/Intra 8 x 8	0 (V)	●		○			○		○	
	1 (H)		●	○				○		○
	2 (DC)	○	○	●	○	○	○	○	○	○
	3 (DDL)				●				○	○
	4 (DDR)					●	○	○		
	5 (VR)					○	●			
	6 (HD)					○		●		
	7 (VL)				○				●	
	8 (HU)				○					●

The candidate mode set is {●} if the previous two layers use the same mode; otherwise, the candidate mode set should include {●, ○}.

TABLE III
CODING TYPE AGREEMENT BETWEEN BL AND EL IN HIERARCHICAL B-FRAMES

Coding Type $Qp_B = 40$	Average Probability at EL, Qp_B = 20–40	
	Foreman	Avg.
$\text{Mode}_{BL} = \text{Mode}_{EL}$ = Intra	>0.99	>0.95
$\text{Mode}_{BL} = \text{Mode}_{EL}$ = Inter	>0.99	>0.99

TABLE IV
CANDIDATE MODES OF INTER PREDICTION FOR $Qp_{n-1} > 30$

Temporal Layer	$T_1 \sim T_{N-2}/T_{N-1} \sim T_N$				
	B_Direct/Skip	16 × 16	16 × 8	8 × 16	8 × 8
16 × 16	○/○	○/○	○/○	○/○	○/○
16 × 8	○/×	○/×	○/○	×/×	×/×
8 × 16	○/×	○/×	×/×	○/○	×/×
8 × 8	×/×	×/×	×/×	×/×	○/○

As shown, each MB at EL is tested with four or fewer intra prediction modes (a column in this table). These candidate modes possess the same or similar prediction directions in the reference/base layer. If a BL MB is encoded in one of the following three modes, DC, Vertical, and Horizontal, the diagonal direction predictions can also be omitted for further complexity reduction. Similarly, only one prediction mode is retained if both of the previous two layers choose the identical mode.

Furthermore, we tabulate the probability of EL Inter/Intra coding types at different Qp_E values in Table III. Obviously, the EL MB most likely has the same coding type as the BL counterpart. Thus, Fig. 11 suggests that an EL MB only needs to evaluate the intra/inter modes when the BL MB (Mode_{BL}) is also intra/inter-coded.

2) *Inter Mode Selection:* To achieve greater savings of coding time while minimizing the coding efficiency loss, the layer-adaptive inter candidate mode sets in Tables IV–VI are designed by examining the inter-layer correlation. We consider both the mode conditional distribution, as shown in Fig. 6, and the R-D performance, shown in Fig. 12.

Based on the statistical data, the less effective MB modes that do not contribute much to the coding efficiency are

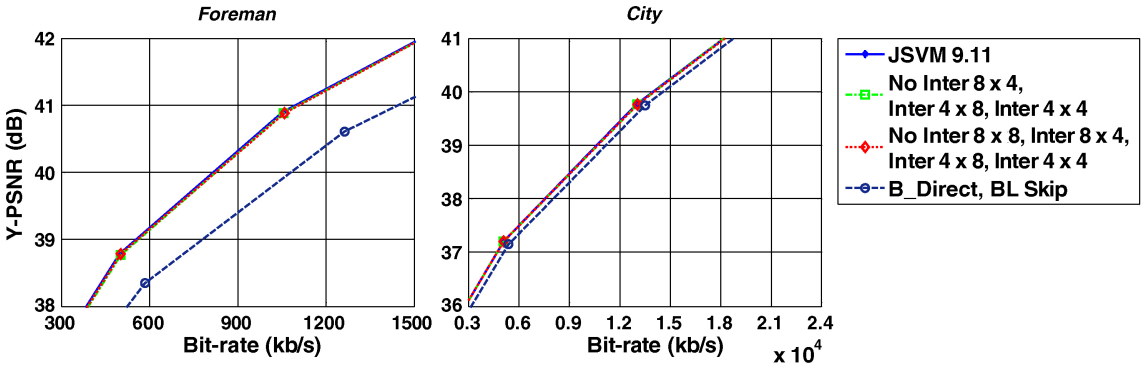

Fig. 12. Comparison of R-D performance of JSVM 9.11 [6] at EL.

TABLE V
CANDIDATE MODES OF INTER PREDICTION FOR $Qp_{n-1} \leq 30$

Temporal Layer	$T_1 \sim T_N$				
	B_Direct/Skip	16 × 16	16 × 8	8 × 16	8 × 8
16 × 16		○			
16 × 8			○		
8 × 16				○	
8 × 8					○

TABLE VI
CANDIDATE MODES OF SUB-MB OF INTER PREDICTION FOR ALL Qp VALUES

Temporal Layer	$T_1 \sim T_N$
B_Direct8 × 8	○
8 × 8	○
8 × 4	
4 × 8	
4 × 4	

neglected. As mentioned earlier, the best mode (size) at EL is likely equal to or larger than that at the reference/base layer. Although the EL distortion could be reduced by using a finer partition, the refinement for a partition mode may introduce the overhead in coding bits for encoding the new partition mode, MVs, and reference frame indices. Fig. 12 shows that the EL coding performance loss is negligible when small-size partitions are disabled, especially for the Inter8 × 8 mode and those smaller than the 8 × 8 size. Therefore, the single-layer advanced video standard [26] supports only four types of block sizes ranging from 16 × 16 down to 8 × 8. In the literature [26], it reports that smaller partitions of the H.264/AVC standard are seldom used, especially in coding high resolution videos. Thus, the coding modes smaller than 8 × 8 can be skipped at EL coding for complexity reduction, as confirmed by the experimental data.

In Tables IV and V, the proposed algorithm evaluates the Inter8 × 8 mode only when Mode_{BL} is coded as the Inter8 × 8 mode. In contrast, for larger partitions, the same and larger partition sizes, the B_Direct/Skip and Inter16 × 16 coding modes may be included in the candidate mode set (the rows in these tables) because they prevent significant coding losses. The sub-block modes are restricted to the B_Direct8 × 8 and Inter8 × 8 modes because the conditional probabilities of the finer partition modes, Inter8 × 4, Inter4 × 8, and Inter4 × 4, are usually less than 0.2. The inter modes with sizes smaller than 8 × 8 require a high computational complexity, but they provide a very limited coding improvement. Also, statistically they are seldom used at EL. We thus skip these three coding modes for the EL MBs, tabulated in Table VI.

In addition, we always check the B_Direct/Skip and BLSkip modes at EL. These two modes provide a significant R-D improvement but only introduce a slight computational load due to their derived MVs (without motion search). Moreover, when the reference/base layer is coded using a quantization parameter ranging from 31 to 51, the inter-layer residual prediction in the transform domain is skipped, as suggested by the analysis in Section III-D. Similar results can be found in [22]. Moreover, as suggested by the statistical analysis, when an MB at BL is coded with the 16×8 (8×16) partition size, its counterpart at the EL will not be evaluated with the partition of 8 × 16 (16 × 8).

Based on our collected data, we always check the Inter16 × 16 mode with only one exception when the BL MB is coded with high quality and Mode_{BL} = B_Direct/Skip. For a BL MB coded with the 8 × 8 partition, its EL counterpart is not tested with any modes (other than 16 × 16) having a partition size larger than 8 × 8. If the BL MB has a coding mode size larger than 8 × 8, such as 16 × 8, then the same size mode and larger size modes should be checked although the finer partition modes are skipped.

On the other hand, when a BL MB is coded with good quality, our algorithm is quite different. The candidate mode set now includes all the modes with the inter-layer residual prediction, and when an MB at the reference/base layer is coded with Inter16 × 16, Inter16 × 8, Inter8 × 16, or Inter8 × 8, only the mode with the same partition is checked. Although Fig. 6 indicates that such a design may not be optimal in terms of the mode distribution, the experimental results in Fig. 12 show that replacing the 8 × 8 partition with larger partitions has negligible impact on the coding efficiency, especially when the EL is coded with high quality.

B. Layer-Adaptive Reference Frame and Motion Reuse

Similar to the mode selection, the layer-adaptive motion search described by Fig. 13 is designed to avoid evaluating

four types of motion searches (ME_R, ME_M, ME_{R+M}, and ME_O) at EL. We check only a selected subset of the reference frames and make use of the BL motion information. Our scheme is composed of two sequential steps: first, select the reference frame candidate indices ref_{EL}, and second, determine the MV starting point according to the information of ref_{EL} and ref_{BL} (defined later).

1) *Step 1: Selection of Reference Frame Candidate Indices:* ref_{EL}

For each BL MB, the exhaustive search picks up the optimal coding mode $Mode_{BL}$ together with its own set of reference frame indices, ref_{BL}. For example, assuming that $Mode_{BL}$ is Inter16 $\times$ 8, two 16 $\times$ 8 blocks in an MB may have different reference frame indices. One is forwardly predicted with frame index r_0 and the other is backwardly predicted with index r_1. Then, for each 16 $\times$ 8 block, its ref_{BL} contains a single reference index (r_0 or r_1). Moreover, normally the encoding process only stores ref_{BL} for inter-layer prediction. Now, our speed-up scheme uses also the intermediate data in the encoding process although this data is not the BL final selection. Thus, we have to additionally retain the reference frame indices in the other sub-optimal inter coding modes, which may be reused by the EL MBs and are denoted as kept_ref_{BL}. For instance, the best Inter16 $\times$ 16 is a bi-directional prediction with reference indices r'_0 and r'_1. Then, contains r'_0 and r'_1, even if $Mode_{BL}$ is Inter16 $\times$ 8. This is because the Inter16 $\times$ 16 may become a candidate mode at EL. For the convenience in notation, r'_0 and r'_1 in this example are denoted as r_0 and r_1 in Fig. 13.

As discussed before, the best temporal prediction of each (sub-)MB at EL is highly correlated with that of its BL counterpart. This high correlation suggests that the set of ref_{BL} or kept_ref_{BL} is sufficient to be the EL candidate set. The EL reference frame candidate set, ref_{EL}, may take either ref_{BL} or kept_ref_{BL} depending on whether or not the evaluated EL mode is the same as $Mode_{BL}$. That is, ref_{EL} in Fig. 13(a) can be either ref_{BL} or kept_ref_{BL}, except when an EL MB is checked with the Inter16 $\times$ 16 mode and the BL is of low quality (i.e., $Qp >$ 30). For example, if is the Inter16 $\times$ 8 mode, the EL candidate mode set is specified by Tables V and VI. Thus, if an EL MB is evaluated using the Inter16 $\times$ 8 mode, ref_{EL} takes as its reference frame indices. Otherwise, kept_ref_{BL} becomes the reference frame index set, ref_{EL}. To ensure a good interframe prediction performance, the EL should perform the forward prediction with index r_0, the backward prediction with index r_0, and the bi-direction prediction with both indices r_0 and r_1 if ref_{EL} contains r_0 and r_1.

An exception is that when the BL is coded at low bit-rate and the BL MB chooses the partition size of 16 $\times$ 16, the probability for two coding layers to select identical temporal prediction mode can be lower than 50%. Thus, the reference frames and the associated motion information of this BL Inter16 $\times$ 16 mode may not be reusable for the EL MB. In this case, the exhaustive search on the reference frames is thus performed for the Inter16 $\times$ 16 prediction mode. In Fig. 13(a) the Qp threshold (=30) is found empirically from extensive experiments as a trade-off between the loss in coding efficiency and the gain in complexity reduction.

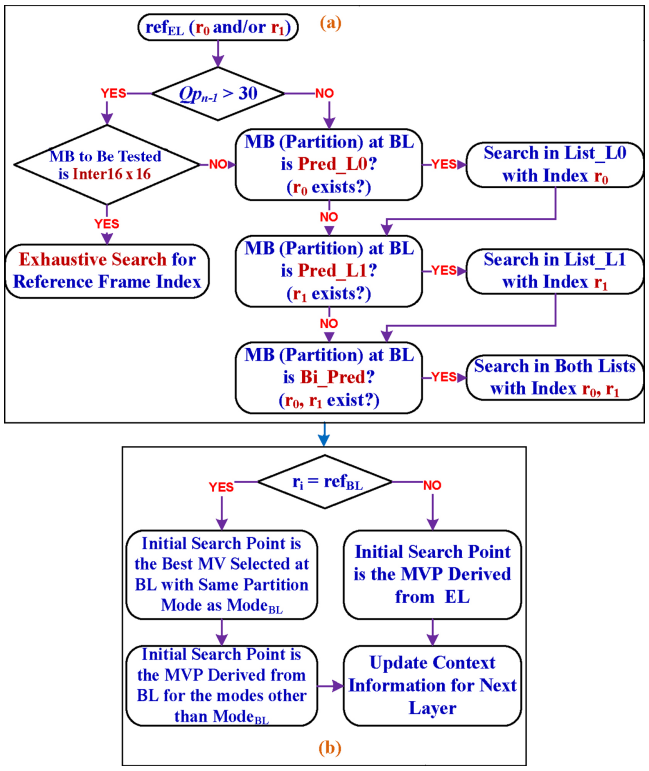

Fig. 13. Layer-adaptive selection in reference frame index and initial search point for hierarchical B-frames. (a) Selection on Reference frame. (b) Selection of Initial Search Point.

Depending on the choice of reference frame indices ref_{EL}, different types of motion searches are executed.

1) If the reference frame index set ref_{EL} (ref_{EL} or kept_ref_{BL}) equals ref_{EL}, the EL motion estimation operation does ME_R and ME_O. Additionally, the inter-layer prediction performs ME with the MV predictor derived from the BL (ME_{R+M} and ME_M) to determine the value of *motion_prediction_flag*. Although four types of motion searches are executed in this case, the complexity of ME_R and ME_O can probably be decreased without executing the bi-directional prediction if the reference frame index set includes only one of r_0 and r_1.
2) Otherwise, the EL motion estimation operation evaluates ME_R and ME_O only; that is, both ME_{R+M} and ME_M are skipped to reduce computation. Similarly, the complexity of ME_R and ME_O can be greatly reduced if the reference frame index set does not contain both r_0 and r_1.

2) *Step 2: Determination of Initial Search Point:* After narrowing down the reference frame candidates, we also consider reducing the number of search points in ME. As discussed earlier, the BLSkip mode is the most probable mode when the partition size of $Mode_{BL}$ is smaller than 16 $\times$ 16. It means that the MVs selected from the BL are reliable and reusable when the EL checks the same mode (as in $Mode_{BL}$). Moreover, in our previous work [22], we found that the MVs of BL and EL are largely correlated. We also reported that the BL MV would provide a better prediction when the MB partition

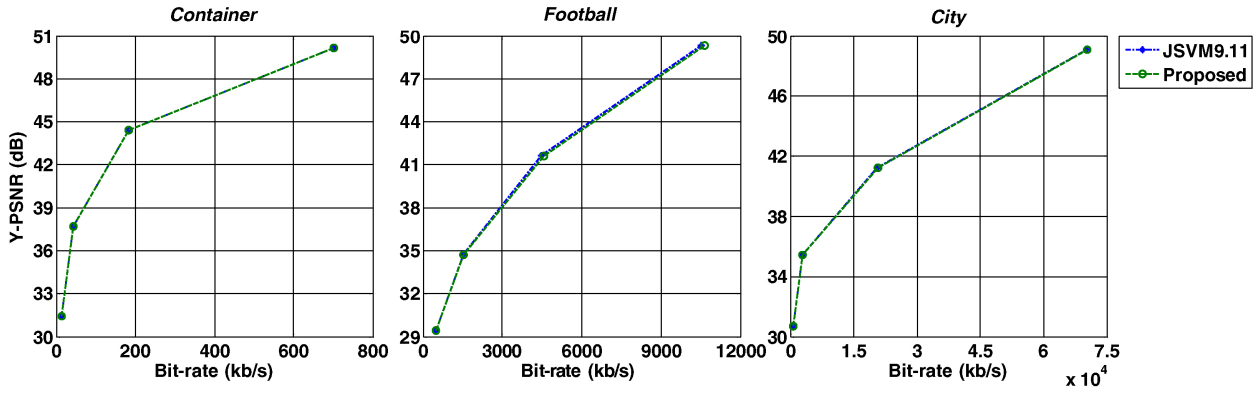

Fig. 14. R-D curves of JSVM9.11 [6] and our approaches.

size is greater than 8×8. More specifically, we compare the initial search points derived by using the BL MVs and the EL MV predictor. We examine the MV difference between the initial search point and the final MV. The statistics show that the MV difference using the BL-derived initial search point is, on average, one pixel less than that using the EL MV predictor. Thus, in Fig. 13(b) our scheme suggests that the MV search starting point should be the one determined by BL when ref_{EL} is equal to ref_{BL}. Otherwise, the EL MV predictor provides the MV search starting point. Consequently, except for the inter modes smaller than 8×8, the MVs of the other BL coding modes should be stored for possibly being used as the EL initial search points.

V. Experimental Results and Discussions

A. Test Conditions

For performance assessment, we have implemented our proposed algorithms in JSVM 9.11 [6] and have tested 15 typical video sequences in three resolutions (QCIF/CIF/4CIF formats), covering a broad range of visual characteristics. Our proposed schemes focus on the complexity reduction at EL. The dyadic hierarchical prediction structure is enabled for the temporal scalability and the CGS EL are created on top of BL for quality scalability. Our experiments include several combinations of coarse-grain quality values and temporal scalabilities using the inter-layer coding structures specified in Fig. 9, which depicts the two-layer and four-layer cases. The detailed encoder parameters are given in Table VII.

In all simulations, we follow the common practice in setting up the Qp values. According to [27], the accumulated bit-rate of EL and BL together should be within three times of the BL bit-rate, so that the inter-layer prediction is effective. Also, as a rule of thumb, a unit increase in Qp value corresponds approximately to a coding rate reduction of 12.5% [28]. The above two rules imply that the Qp difference between two successive coding layers should be less than 10. In addition, the nominal Qp_B value in JSVM 9.11 [6] is from 28 to 40. Therefore, we set Qp_B value to either 30 or 40 in our experiments.

TABLE VII
Testing Conditions

Testing sequences	QCIF	*Carphone* (CP), *Coastguard* (CG), *Container* (CTN), *Motherdaughter* (MD), *Suzie* (SZ)
	CIF	*Akiyo* (AK), *Bus* (BU), *Football* (FB), *Mobile* (MB), *Stefan* (SF)
	4CIF	*City* (CT), *Crew* (CR), *Harbour* (HB), *Ice* (IC), *Soccer* (SC)
Encoder configurations and platform	Software: JSVM_9_11 [6] M.E. search range: ±32pixels with 1/4-pel accuracy RDO: enabled GOP size: 8 Entropy coding mode: CABAC Macroblock adaptive inter-layer prediction: enabled Machine: Athlon 3800+, 64-bit, dual-core processors, 2.0 GB RAM with Windows XP	

B. Performance Measures

To measure the speedup performance, we define "time saving (TS)" for the whole encoding process and "EL time saving (TS_{E})" for coding the EL only.

1) The *overall time saving* TS is defined as $\text{TS} = (T_{\text{JSVM9.11}} - T_{\text{Proposed}})/T_{\text{JSVM9.11}} \times 100\%$, where $T_{\text{JSVM9.11}}$ and T_{Proposed} denote the encoding time of JSVM 9.11 [6] and that of our schemes, respectively.
2) The *EL time saving* TS_{E} is defined as $\text{TS}_{\text{E}} = (T_{\text{JSVM9.11}} - T_{\text{Proposed}})/(T_{\text{JSVM9.11}} - T_{\text{BL}}) \times 100\%$, where T_{BL} is the BL encoding time.

To show the change in R-D performance, we adopt the Bjontegaard metric [29] to measure the averaged Y-PSNR [BDP (dB)] and bit-rate differences [BDR (%)] between the R-D curves produced by JSVM 9.11 [6] and by our schemes, respectively. Because the computation of BDP and BDR requires at least four R-D points on each curve, these figures are provided only in the comparison of the four-layer case. For the two-layer case, we simply compare the Y-PSNR [ΔPSNR (dB)] and bit-rate [ΔBitrate (%)] differences at EL by the following formulae. In either case, we use ΔFS (%) to indicate the percentage of the total file size increase.

1) The *PSNR difference* is defined as $\Delta\text{PSNR} = \text{PSNR}_{\text{Proposed}} - \text{PSNR}_{\text{JSVM9.11}}$, where $\text{PSNR}_{\text{JSVM9.11}}$ and $\text{PSNR}_{\text{Proposed}}$ are the Y-PSNR values

TABLE VIII
AVERAGE TIME SAVING OF MD AND MR/RF

Sequences	Reference Frame = 1				Reference Frame = 3			
	MD		MR/RF		MD		MR/RF	
	TS (%)	TS_E (%)	TS (%)	TS_E (%)	TS (%)	TS_E (%)	TS (%)	TS_E (%)
CP	78.7	87.0	17.1	18.9	79.0	88.9	45.9	51.6
CTN	80.7	88.6	21.1	23.2	80.7	90.3	46.4	51.9
SZ	79.5	87.7	19.9	22.0	79.9	89.8	48.6	54.6
AK	81.1	89.0	33.1	36.3	81.0	90.5	52.3	58.5
FT	77.7	87.2	28.2	31.6	78.7	89.7	57.5	65.5
SF	79.3	87.6	24.3	26.9	79.7	89.9	58.3	65.6
CT	81.2	89.3	28.8	31.7	78.8	88.6	63.5	71.3
CR	80.0	88.3	31.4	34.6	71.4	80.1	63.6	71.4
SC	80.2	88.4	27.3	30.1	74.5	83.7	63.2	71.0
Avg.	79.8	88.1	25.7	28.4	78.2	87.9	55.5	62.4

$(Qp_B,\ Qp_{E1},\ Qp_{E2},\ Qp_{E3})$ = (40, 30, 20, 10) and (30, 20, 10, 0).

TABLE IX
Qp SETTING OF $(Qp_B,\ Qp_{E1},\ Qp_{E2},\ Qp_{E3})$ = (40, 30, 20, 10)

Sequences	Reference Frame = 1					Reference Frame = 3				
	BDP (dB)	BDR (%)	ΔFS (%)	TS (%)	TS_E (%)	BDP (dB)	BDR (%)	ΔFS (%)	TS (%)	TS_E (%)
CP	−0.07	1.67	1.07	81.8	90.6	−0.09	1.94	1.25	82.5	92.9
CG	−0.08	1.69	1.20	81.4	90.0	−0.08	1.64	1.15	82.4	92.8
CTN	−0.01	0.12	0.05	83.3	91.7	−0.01	0.11	0.05	83.6	93.6
MD	−0.07	1.40	0.93	83.4	91.8	−0.07	1.50	1.10	83.7	93.7
SZ	−0.08	1.89	1.23	82.5	91.2	−0.08	1.94	1.20	83.2	93.6
AK	−0.02	0.61	0.33	84.1	92.3	−0.03	0.68	0.39	84.2	94.0
BU	−0.08	1.43	0.85	81.2	90.3	−0.08	1.41	0.87	82.6	93.5
FB	−0.12	1.92	1.34	80.6	90.5	−0.13	2.06	1.42	82.3	93.6
MB	−0.05	0.83	0.62	81.9	90.2	−0.07	1.25	0.86	82.4	92.6
SF	−0.04	0.75	0.44	81.8	90.4	−0.05	1.03	0.61	82.7	93.3
CT	−0.01	0.36	0.19	83.3	91.7	−0.02	0.40	0.22	83.9	94.3
CR	−0.02	0.50	0.14	82.2	90.7	−0.02	0.49	0.14	83.1	93.2
HB	−0.01	0.27	0.15	82.3	90.7	−0.01	0.30	0.17	83.1	93.2
IC	−0.03	0.95	0.33	82.7	91.0	−0.03	0.88	0.31	83.4	93.5
SC	−0.03	0.81	0.39	82.5	91.0	−0.04	0.85	0.41	83.5	93.8
Avg.	−0.05	1.01	0.62	82.3	90.9	−0.05	1.10	0.68	83.1	93.4

TABLE X
Qp SETTING OF $(Qp_B,\ Qp_{E1},\ Qp_{E2},\ Qp_{E3})$ = (40, 30, 20, 10)

Sequences	Reference Frame = 1					Reference Frame = 3				
	BDP (dB)	BDR (%)	ΔFS (%)	TS (%)	TS_E (%)	BDP (dB)	BDR (%)	ΔFS (%)	TS (%)	TS_E (%)
CP	−0.01	0.15	0.06	82.3	90.9	−0.01	0.19	0.06	83.3	93.6
CG	−0.01	0.10	0.05	82.0	90.5	−0.01	0.06	0.02	83.3	93.7
CTN	0.00	0.01	0.00	84.6	92.7	0.00	0.01	0.00	84.8	94.8
MD	−0.01	0.20	0.04	84.2	92.4	−0.01	0.16	0.02	84.6	94.5
SZ	−0.01	0.16	0.07	83.2	91.7	−0.01	0.14	0.06	84.0	94.4
AK	−0.01	0.15	0.03	85.0	93.3	−0.01	0.16	0.04	85.2	95.2
BU	−0.03	0.35	0.09	81.9	91.0	−0.03	0.41	0.09	83.4	94.5
FB	−0.02	0.18	0.07	81.3	91.4	−0.01	0.16	0.05	83.1	94.7
MB	−0.02	0.28	0.17	82.0	90.1	−0.04	0.43	0.20	82.9	93.0
SF	−0.01	0.15	0.06	82.4	91.0	−0.02	0.24	0.08	83.5	94.0
CT	−0.01	0.13	−0.01	84.3	92.7	−0.01	0.12	−0.06	84.9	95.3
CR	0.00	0.05	−0.02	83.5	92.1	0.00	0.06	−0.02	84.4	94.7
HB	0.00	0.03	−0.01	83.3	91.6	0.00	0.04	−0.01	84.2	94.3
IC	−0.01	0.08	0.02	83.8	92.2	−0.01	0.09	0.02	84.6	94.8
SC	−0.01	0.11	−0.04	83.5	92.0	−0.01	0.18	−0.02	84.4	95.0
Avg.	−0.01	0.15	0.04	83.2	91.7	−0.01	0.16	0.04	84.0	94.4

obtained by using JSVM 9.11 [6] and our schemes, respectively.

2) The *bit-rate increase* is defined as $\Delta\text{Bitrate} = (\text{Bitrate}_{\text{Proposed}} - \text{Bitrate}_{\text{JSVM9.11}})/\text{Bitrate}_{\text{JSVM9.11}} \times 100\%$, where $\text{Bitrate}_{\text{JSVM9.11}}$ and $\text{Bitrate}_{\text{Proposed}}$ correspond to the bit-rate of JSVM 9.11 [6] and that of our schemes, respectively.

C. Simulation Results

Tables VIII–X present the time savings of the proposed schemes in comparison with JSVM 9.11. Listed in Table VIII are the improvements contributed by the mode decision (MD) and the motion information reuse with pre-selected reference frame (MR/RF), separately. The results are obtained by comparing the running time of the encoder with the following configurations:

MD setting: JSVM 9.11 versus JSVM 9.11 + MD

MR/RF setting: JSVM 9.11 versus JSVM 9.11 + MR/RF.

It can be seen that enabling the MD mechanism alone can reduce the overall running time by 79% (equivalent to a speedup of about 5×), and it gives a higher improvement (up to 90%) in coding EL. The results are consistent regardless of the number of reference frames. By comparison, the MR/RF offers only a moderate time saving of 25–55% depending on the number of reference frames in use. More reference frames lead to higher improvement. This is because MR/RF checks at most two frames in the worse case when both forward and backward prediction directions are active and it often needs to check only one frame.

To see their combined effects, Tables IX and X provide the time savings relative to the exhaustive search, with both MD and MR/RF enabled. The results given in these two tables correspond to two different Qp settings: (Qp_B, Qp_{E1}, Qp_{E2}, Qp_{E3}) = (40, 30, 20, 10) and (30, 20, 10, 0). As can be seen, when the MD is coupled with the MR/RF, an average saving of 83% for the overall encoding time is achieved. Moreover, when considering only the EL, where our schemes actually take effect, we can observe an up to 20× speedup (which amounts to a maximal time saving of 95%). The improvement is achieved with a negligible change in both bit-rate and PSNR, as confirmed by the BDP/BDR values in the tables and the R-D curves in Fig. 14. Interestingly, the overall time saving with three reference frames differs only slightly from that with one reference frame, even though we expect the MR/RF mechanism would benefit more on the multiple reference frame cases. The result is attributed to the fact that the MV search operations are significantly reduced after the MD mechanism is activated and thus percentagewise the amount of computation further reduced by the MR/RF mechanism is relatively small. A more detailed discussion is as follows.

Let us begin with the encoding time ratio of coding a BL to coding an EL in Table XI. We can see that the running time for coding an EL is about 3.24 times that for coding its BL when only one reference frame is in use. In the four-layer case (i.e., one BL + three EL), the EL encoding time represents 90.7% of the overall computation time. From Table VIII, 79.8% of the computation can be skipped when our MD scheme is applied. Thus, only 90.7% − 79.8% = 10.9% are left to the next step improvement—MR/RF in our case. A similar number (∼10.7%) is obtained for the case with three reference frames. According to Amdahl's law and the average TS_E in Table VIII, it is not surprising to see that the MR/RF mechanism is less influential on the overall performance improvement, no matter how many reference frames are used.

TABLE XI
AVERAGE COMPLEXITY RATIO OF THE BL TO ONE EL

Sequences	Six (Qp_B, Qp_E) Settings: Qp_E = 40 with Qp_E = 30, 20, 10 Qp_B = 30 with Qp_E = 20, 10, 0			
	Reference Frame = 1		Reference Frame = 3	
	JSVM 9.11	Proposed	JSVM 9.11	Proposed
CP	1:3.12	1:0.29	1:2.66	1:0.18
CTN	1:3.38	1:0.26	1:2.81	1:0.16
SZ	1:3.19	1:0.27	1:2.67	1:0.16
AK	1:3.42	1:0.25	1:2.85	1:0.15
FB	1:2.67	1:0.24	1:2.38	1:0.14
SF	1:3.19	1:0.30	1:2.63	1:0.17
CT	1:3.76	1:0.29	1:2.68	1:0.14
CR	1:3.20	1:0.29	1:2.71	1:0.16
SC	1:3.25	1:0.28	1:2.69	1:0.15
Avg.	1:3.24	1:0.27	1:2.68	1:0.16

Another interesting fact to be noted is that with our schemes the latency for coding three EL is almost the same as that for coding one BL with the exhaustive search. This phenomenon does not change much with the GOP size. This is because a large portion of the overall speedup comes from the coding of the highest two temporal layers and they constitute 75% of the frames in a GOP. An exception is when GOP size = 2, of which the highest temporal frame number is 1, and thus its percentage reduces to only 50%, namely

$$\begin{cases} \dfrac{2^{N-2}+2^{N-1}}{1+2^0+2^1+\cdots+2^{N-1}} = \dfrac{2^{N-2}+2^{N-1}}{2^N} = 75\%, & N \geq 2 \\ 50\%, & N = 1 \end{cases}$$

where the GOP size is 2^N.

D. Performance Comparison With State-of-the-art Fast Algorithms (Li's Methods and Ren's Method)

In addition to the exhaustive search, we also compare our approaches with the state-of-the-art fast algorithms, Li's methods [16], [18] and Ren's method [19], in which only one reference frame in each prediction direction is considered for the dyadic hierarchical temporal prediction. For a fair comparison, the same number of reference frame (one reference frame in each reference list) is configured in our schemes. As shown in Tables XII–XIV, our methods can achieve a higher time saving (7–41% more) in comparison with [16], [18], and [19] and, in the meanwhile, have a lower Y-PSNR loss and bit-rate increase. The coding loss of our scheme is slightly larger when the coding layers have large Qp difference. Moreover, the time saving of Ren's method [19] has a wide range from 28.6% to 55.6% but Li's [16], [18] and ours have more consistent time savings with a variation of less than 10%.

In terms of the overall speedup, our schemes do not seem to have a drastic improvement over the two previous works

TABLE XII
PERFORMANCE COMPARISONS WITH LI'S METHOD [16]

Sequences	(Qp_B, Qp_E)	Li's Method [16]			Proposed		
		ΔPSNR (dB)	ΔBitrate (%)	TS (%)	ΔPSNR (dB)	ΔBitrate (%)	TS (%)
Football	(40, 20)	0.05	0.85	47.5	−0.06	2.31	64.6
	(40, 15)	0.09	1.26	48.4	−0.04	2.32	64.5
	(40, 10)	0.06	1.03	49.0	−0.03	2.05	64.4
City	(40, 20)	−0.11	0.21	38.7	−0.01	0.31	67.3
	(40, 15)	−0.11	0.00	39.9	0.00	0.31	67.2
	(40, 10)	−0.09	0.13	41.2	0.00	0.49	67.0
Harbour	(40, 20)	−0.10	0.30	42.6	0.00	0.30	66.7
	(40, 15)	−0.08	0.37	40.4	0.00	0.28	66.6
	(40, 10)	−0.06	0.29	44.1	0.00	0.13	66.5
Avg.		−0.04	0.49	43.5	−0.02	0.94	66.1

TABLE XIII
PERFORMANCE COMPARISONS WITH LI'S METHODS [16] AND [18]

Sequences	(Qp_B, Qp_E)	Li's Method [16]			Li's Method [18]			Proposed		
		ΔPSNR (dB)	ΔBitrate (%)	TS (%)	ΔPSNR (dB)	ΔBitrate (%)	TS (%)	ΔPSNR (dB)	ΔBitrate (%)	TS (%)
Bus	(40, 30)	0.02	1.00	41.7	−0.13	0.42	58.2	−0.04	0.69	66.5
	(30, 20)	0.03	1.84	44.2	−0.06	1.19	56.1	−0.01	0.20	66.1
Football	(40, 30)	0.15	3.42	46.0	0.00	1.84	59.9	−0.07	0.81	64.8
	(30, 20)	0.13	3.15	49.9	−0.01	1.09	58.8	−0.01	0.28	66.1
City	(40, 30)	0.02	0.83	39.3	−0.14	−0.27	64.1	−0.01	0.25	68.3
	(30, 20)	0.00	0.62	40.9	−0.10	0.23	61.8	0.00	0.28	71.0
Crew	(40, 30)	0.07	2.40	42.6	−0.13	0.59	62.8	−0.01	0.46	66.9
	(30, 20)	0.13	3.43	45.7	−0.05	1.22	58.2	0.00	0.19	69.3
Avg.		0.07	2.09	43.8	−0.08	0.79	60.0	−0.02	0.40	67.4

TABLE XIV
PERFORMANCE COMPARISONS WITH REN'S METHOD [19]

Sequences	Ren's Method [19]			Proposed		
	BDP (dB)	BDR (%)	TS (%)	BDP (dB)	BDR (%)	TS (%)
Hall	−0.16	2.99	49.4	−0.01	0.14	70.9
Foreman	−0.23	4.13	37.7	−0.01	0.17	68.9
Mobile	−0.18	2.55	28.6	−0.01	0.08	69.1
News	−0.34	3.87	55.6	−0.01	0.14	70.8
Silent	−0.23	3.00	48.9	−0.01	0.08	70.2
Avg.	−0.23	3.31	44.0	−0.01	0.12	70.0

Video resolution is QCIF, GOP size is 16, Qp_B = 22, 27, 32, 27 and Qp_E = 19, 24, 29, 34.

[16], [18]. This is because the BL coding time is fixed in our study and it becomes the dominant portion of the overall running time when 90% of the EL calculations are removed. According to Table XI, the EL coding occupies 76.4% of the entire computation in the two-layer case. This part is reduced to 76.4% − 49% = 27.4% with Li's methods [16], [18] and 76.4% − 67% = 9.4% with ours (see Tables XII and XIII). Therefore, if we consider the EL speed-up only, which is our focus; our schemes actually have a relative improvement of (27.4% − 9.4%)/27.4% = 65.7% over the Li's methods.

VI. CONCLUSION

In this paper, we have proposed a layer-adaptive intra/inter mode decision algorithm and a motion search scheme for the hierarchical B-frames in SVC with combined CGS and temporal scalability. We examined the R-D performance contributed by different coding modes at EL and the conditional probability distributions of intra/inter modes at different temporal layers. Three types of techniques have been newly proposed or well-extended from the existing proposals. The first technique is to limit the intra prediction candidate modes based on the BL intra mode information. The second technique is to eliminate the infrequent inter modes based on the inter-layer mode correlation. These two techniques were implemented by using look-up tables. A fast layer-adaptive intra/inter mode decision scheme is thus designed. Finally, the third technique is the motion information reuse, including the reference frame in ME and the motion search modes. Using the coded previous-layer information, our approach can provide more than 50% mode reduction with pre-selected reference frame indices, and no extra computation is needed to derive the candidate mode set. The massively heavy computational complexity introduced at the EL encoding process is remarkably reduced. Compared to the exhaustive-search mode decision algorithm in JSVM 9.11 [6], our proposed approach provides an average saving of 80% or higher in the overall encoding time and up to 95% time reduction for encoding the CGS EL. And the penalty on

R-D performance is negligible. The average bit-rate increase is below 1% and the average Y-PSNR loss is below 0.05 dB. Our scheme is up to 41% faster than the existing methods in [16], [18], and [19].

Although specifically designed for the combined CGS and dyadic temporal scalability, our algorithms can also find their applications in the spatial and the non-dyadic temporal scalability. However, these must be adjusted in a number of ways to fit into the special scalability structure. For instance, two important issues need to be addressed for the spatial scalability: 1) the change in statistics due to the multiple-to-one MB mapping from a spatial EL to its BL; and 2) the aliasing effect due to the interpolation of residual and motion signals. The former may decrease the dependence of the EL coding mode/type on its BL counterpart, and the latter could affect the reliability of the BL motion parameters. In contrast, the application of our schemes to the non-dyadic temporal scalability is straightforward. It is expected that the statistics in the non-dyadic case are similar to those in the dyadic one. In practice, the non-dyadic temporal scalability is seldom used.

REFERENCES

[1] T. Wiegand, G. Sullivan, and A. Luthra, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 |ISO/IEC 14496-10 AVC)*, document JVT-G050r1.doc, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Mar. 2003.

[2] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz, and M. Wien, *Joint Draft 10 of SVC Amendment*, document JVT-W201.doc, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Apr. 2007.

[3] 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.

[4] H. Schwarz, D. Marpe, and T. Wiegand, *Hierarchical B Pictures*, document JVT-P014.doc, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Jul. 2005.

[5] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2006, pp. 1929–1932.

[6] J. Reichel, H. Schwarz, and M. Wien, *Joint Scalable Video Model JSVM-9*, document JVT-V202.doc, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Jan. 2007.

[7] A.-C. Tsai, A. Paul, J.-C. Wang, and J.-F. Wang, "Intensity gradient technique for efficient intra-prediction in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 5, pp. 694–698, May 2008.

[8] A.-C. Tsai, J.-F. Wang, J.-F. Yang, and W.-G. Lin, "Effective subblock-based and pixel-based fast direction detections for H.264 intra prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 975–982, Jul. 2008.

[9] C. Kim and C.-C. Jay Kuo, "Feature-based intra/inter coding mode selection for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 441–453, Apr. 2007.

[10] 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.

[11] 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.

[12] 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, Feb. 2008.

[13] I. Choi, J. Lee, and B. Jeon, "Fast coding mode selection with rate-distortion optimization for MPEG-4 Part-10 AVC/H.264," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1557–1561, Dec. 2006.

[14] 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.

[15] B.-G. Kim, "Fast selective intra-mode search algorithm based on adaptive thresholding scheme for H.264/AVC encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 127–133, Jan. 2008.

[16] H. Li, Z.-G. Li, and C. Wen, "Fast mode decision for coarse grain SNR scalable video coding," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. II. 2006, pp. 545–548.

[17] 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.

[18] H. Li, Z.-G. Li, C. Wen, and S. Xie, "Fast mode decision for coarse granular scalability via switched candidate mode set," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2007, pp. 1323–1326.

[19] J. Ren and N. Kehtarnavaz, "Fast adaptive termination mode selection in H.264 scalable video coding," *J. Real-Time Image Process.*, vol. 4, pp. 13–21, Mar. 2009.

[20] L. Xiong, *Reducing Enhancement Layer Directional Intra Prediction Modes*, document JVT-P041.doc, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Jul. 2005.

[21] L. Yang, Y. Chen, J. Zhai, and F. Zhang, *Low Complexity Intra Prediction for Enhancement Layer*, document JVT-Q084.doc, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Oct. 2005.

[22] H.-C. Lin, W.-H. Peng, H.-M. Hang, and W.-J. Ho, "Layer-adaptive mode decision and motion search for scalable video coding with combined coarse granular scalability (CGS) and temporal scalability," in *Proc. IEEE Int. Conf. Image Process.*, 2007, pp. II-289–II-292.

[23] H.-C. Lin, W.-H. Peng, and H.-M. Hang, "A fast mode decision algorithm with macroblock-adaptive rate-distortion estimation for intra-only scalable video coding," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2008, pp. 765–768.

[24] H. Schwarz, D. Marpe, and T. Wiegand, *Inter-Layer Prediction of Motion and Residual Data*, ISO/IEC JTC 1/SC 29/WG11/M11043, Jul. 2004.

[25] J. Reichel, H. Schwarz, and M. Wien, *Joint Scalable Video Model JSVM-8*, document JVT-U202.doc, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Oct. 2006.

[26] L. Fan, S. Wa, and F. Wu, "Overview of AVS video standard," in *Proc. IEEE Int. Conf. Multimedia Expo*, vol. 1. 2004, pp. 423–426.

[27] Z.-G. Li, S. Rahardja, and H. Sun, "Implicit bit allocation for combined coarse granular scalability and spatial scalability," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1449–1459, Dec. 2006.

[28] Y.-W. Huang, B.-Y. Hsieh, T.-C. Chen, and L.-G. Chen, "Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame coder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 378–401, Mar. 2005.

[29] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33.doc, Apr. 2001.

Hung-Chih Lin was born in Nantou, Taiwan, in 1982. He received the B.S. and M.S. degrees in electrical control engineering and electronics engineering from National Chiao-Tung University (NCTU), Hsinchu, Taiwan, in 2004 and 2005, respectively. He is currently working toward the Ph.D. degree from the Department of Electronics Engineering, NCTU.

He has worked on the software optimization of the MPEG video coding standards since his M.S. program. His research topics focus mainly on the fast algorithm designs in the H.264/AVC video coding standard and its scalable extension. His current research interests include digital image processing, video signal processing, and scalable video compression, particularly on fast algorithm designs and their implementations on the digital signal processing platforms.

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

From 2000 to 2001, he was an Intern with the Intel Microprocessor Research Laboratory, Santa Clara, CA, 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 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 performance 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.

Hsueh-Ming Hang (F'02) received the B.S. and M.S. degrees in control engineering and electronics engineering from National Chiao-Tung University (NCTU), Hsinchu, Taiwan, in 1978 and 1980, respectively, and the Ph.D. degree in electrical engineering from the 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 Department of Electronics Engineering, NCTU, in 1991. From 2006 to 2009, he took a leave from NCTU and was appointed as the Dean of the Electrical Engineering and Computer Science College, National Taipei University of Technology, Taipei, Taiwan. He is currently a Distinguished Professor with the Department of Electronics Engineering, NCTU. He holds 16 patents in the U.S., China, and Japan, and has published over 160 technical papers related to image compression, signal processing, and video codec architecture. 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 of the IEEE TRANSACTIONS ON IMAGE PROCESSING from 1992 to 1994, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 1997 to 1999. He is a Co-Editor and Contributor of the *Handbook of Visual Communications* (New York: Academic). He is a recipient of the IEEE Third Millennium Medal and is a Fellow of the Institution of Engineering and Technology and a Member of the Sigma Xi.