

國立交通大學

電子工程學系 電子研究所

博士論文

用於H.264/MPEG-4 AVC可調式視訊編碼

標準之快速編碼演算法設計

Fast Encoding Algorithm Design for

H.264/MPEG-4 AVC Scalable Video Coding

Standard

研究生：林鴻志

指導教授：杭學鳴

中華民國九十九年六月

用於H.264/MPEG-4 AVC可調式視訊編碼標準之快速編碼

演算法設計

Fast Encoding Algorithm Design for H.264/MPEG-4 AVC
Scalable Video Coding Standard

研究生：林鴻志

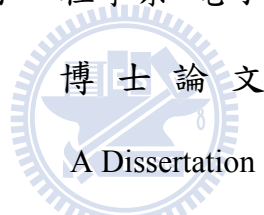
Student: Hung-Chih Lin

指導教授：杭學鳴博士

Advisor: Dr. Hsueh-Ming Hang

國立交通大學

電子工程學系 電子研究所



Submitted to Department of Electronics Engineering

& Institute of Electronics

College of Electrical and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Electronic Engineering

June 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

用於H.264/MPEG-4 AVC可調式視訊編碼 標準之快速編碼演算法設計

研究生：林鴻志

指導教授：杭學鳴 博士

國立交通大學 電子工程學系 電子研究所博士班

摘 要

為了使視訊影像能夠穩健地在異質網路環境中傳輸，H.264/MPEG-4 AVC 視訊編碼標準(H.264/AVC)已擴展出可調式視訊編碼標準(H.264/SVC)。在 H.264/SVC 視訊編碼標準中，主要提供了三種可調特性，包含了時間上、空間上與畫質上之可調特性。H.264/SVC 視訊編碼標準能夠在一次壓縮視訊影響的前提下，根據不同的儲存、傳輸需求，擷取出部分位元流(bit-stream)並解碼出較低畫面率或低解析度之視訊影像。H.264/SVC 視訊編碼標準採用了層次時域預測(hierarchical temporal prediction)編碼架構達到時間可調特性，含有兩個單方向預測與一個雙方向預測(bi-directional prediction, BI prediction)。此外，亦採用了層級編碼架構(layered coding approach)來實現空間與畫質之可調性，而各編碼層以層次時域預測為其基本結構。為了讓編碼效能達到最佳，H.264/SVC 視訊編碼標準會評估編碼參數的所有組合可能，其中包含了H.264/AVC 視訊編碼標準的編碼工具組與新提出的層間預測(inter-layer mechanism)機制。然而，此選定編碼參數作法會招致相當龐大之運算量。根據實驗數據指出，模式決定(mode decision)過程與其所需要之動作估計(motion estimation)之程序占了絕大部份之編碼運算量。因此，發展用於減少 H.264/SVC 視訊編碼標準運算複雜度之快速演算法是必要的。

首先，針對二指數(dyadic)層次時域預測架構，我們提出了一套有效率的選擇時域預測模式(temporal prediction type)之快速演算法。根據 16x16 切割模式所選定之最佳時域預測模式，

利用其高度相關繼承特性，可以有效地避免在大切割模式中(含 16x8、8x16 與 8x8)的非必要之雙方向預測計算。此外，我們也謹慎地找出單方向預測與雙方向預測，兩者的誤差(distortion)與動作碼率(motion rate)數值之關係，用以設定出一組適應性調整之臨界值，排除不必要之雙方向預測運算。而在小切割模式中(含 8x4、4x8 與 4x4)，根據我們的分析，不僅其最佳之時域預測模式可以參考 8x8 切割模式而得知，而且雙方向預測模式在編碼效能提升上是非常有限的。因此，這些分析可以有效地用來屏除層次時域預測架構中的無效之雙方向預測計算。

接著，在全幀內(intra-only)預測之可調編碼架構下，因內部 4x4 預測(intra 4x4)與內部 8x8 預測(intra 8x8)之誤差與碼率(rate)在層級間具有對數線性(log-linear)之關係。利用此特性與基本層(base layer)所選定之最佳內部預測模式，可以大量地減少加強層(enhancement layer)之內部預測測試個數。此外，在較平滑之影像區域，我們保留了內部 16x16 預測(intra 16x16)的評估效應。

最後，在幀間(inter)預測之可調編碼架構下，考慮了時間與畫質兩種可調性組合，提出了一幀內/幀間模式與動作向量選擇演算法。我們觀察不同切割模式的編碼效能與其切割模式在層級之間的條件機率分布，對於幀內模式而言，基於參考基本/參考層(reference layer)之資訊，加強層可以節省至少一半以上的測試個數；另一方面，對於幀間模式而言，藉由層級間量化參數(quantization parameter)的差異，調整加強層所要查詢的切割模式表。另外，為了減少動作搜尋的計算量，基本層的參考畫面位置也可以被選擇性地使用，而且基本層所選定之動作向量(motion vector)亦可被拿來當作加強層中的起始搜尋點。

綜合而言，本論文藉由分析與觀察層級之間的高度相關性，排除罕見的編碼模式組合。實驗數據指出，與 H.264/SVC 之標準參考軟體相比，我們所提出之快速演算法可以在維持鮮少之效能損失下，節省 65%~85%之編碼時間。

Fast Encoding Algorithm Design for H.264/MPEG-4 AVC Scalable Video Coding Standard

Student: Hung-Chih Lin

Advisor: Dr. Hsueh-Ming Hang

Department of Electronics Engineering and Institute of Electronics
National Chiao Tung University

Abstract

To enable robust video transmission over heterogeneous networks, the H.264/MPEG-4 AVC (H.264/AVC) has developed an extension of scalable video coding scheme (H.264/SVC). In the H.264/SVC, there are three main modalities of scalability, consisting of temporal, spatial, and quality scalability. The H.264/SVC can compress the video signal once but enable partially decoding the encoded bit-streams with lower temporal frame rate or spatial resolutions, depending on the storage and transmission requirements. To achieve the temporal scalability, the H.264/SVC uses the coding structure of the hierarchical temporal prediction, in which there are two uni-directional predictions and one bi-directional (BI) prediction. In addition, the spatial and quality scalabilities are realized by adopting the layered coding approach, where the hierarchical temporal prediction forms a basic coding structure in each coding layer. In order to provide high coding efficiency, the H.264/SVC exhaustively evaluates all possible combinations of encoding parameters, including the conventional coding tools in the H.264/AVC and the novel inter-layer prediction mechanism. However, the procedure of selecting optimal coding parameters dramatically results in huge computational complexity. The experimental results show that the mode decision process with related motion

estimations significantly dominates the overall encoding time. Hence, it is necessary to develop fast encoding algorithms to reduce the encoding computations in the H.264/SVC.

First, we propose a fast algorithm that efficiently selects the temporal prediction type for the dyadic hierarchical-B prediction structure in the H.264/SVC temporal scalable video coding. Referring to the best temporal prediction type of 16x16, we utilize the strong correlations of prediction type inheritance to eliminate the unnecessary computations for the BI prediction in the finer partitions, 16x8/8x16/8x8. In addition, we carefully examine the relationship of motion-rate costs and distortions between the BI and the two uni-directional temporal prediction types. As a result, we construct a set of adaptive thresholds to remove the unnecessary BI calculations. Moreover, our analysis points out that the coding efficiency of the BI prediction is limited in small partitions. For the block partitions smaller than 8x8, one of the two uni-directional temporal predictions is skipped based upon the information of an 8x8 partition. Hence, these analyses can be used to efficiently reduce the extensive computations burden in performing the BI prediction.

Second, we make use of the log-linear rate-distortion relationship of inter-dependent layers to predict the better performer among the Intra4x4 and Intra8x8 prediction types at the enhancement layers for intra-only scalable video coding. Based upon the base-layer chosen prediction type, we can further reduce the number of candidate modes. In addition, to ensure the best trade-off between complexity and coding efficiency, the Intra16x16 prediction is retained and enabled only for coding high-resolution videos with smooth image contents.

Finally, we provide a layer-adaptive intra/inter mode decision algorithm and a motion search scheme for the hierarchical B-frames in the H.264/SVC with combined coarse-grain quality scalability (CGS) and temporal scalability. We examine the rate-distortion performance contributed by different coding modes at the enhancement layers and the mode conditional probabilities at different temporal layers. For the intra prediction on inter frames, the number of Intra4x4/Intra8x8 prediction modes can be reduced by 50% or more, based on the reference/base layer intra prediction directions. For the enhancement-layer inter prediction, the look-up tables containing inter prediction

candidate modes are designed to use the macroblock coding mode dependence on and the reference/base layer quantization parameters (Qp). In addition, to avoid checking all motion estimation reference frames, the base-layer reference frame index is selectively reused. And according to the enhancement-layer macroblock partition, the base-layer motion vector can be used as the initial search point for the enhancement-layer motion search.

In conclusion, our proposed algorithms efficiently eliminate the unlikely combinations of coding options. The experiments show that our approaches can reduce 65%~85% encoding time with a similar coded quality, as compared to the reference software of the H.264/SVC.



誌 謝

首先，我要感謝我的指導教授杭學鳴博士在這六年中給我的諸多指導，不僅在學術研究的方向上不斷地引領我走在專業領域的先端。六年來，每星期的討論一點一滴地醞釀我向前進步的潛能，亦讓我培養了獨立解決問題與清楚表達想法的能力。除此之外，杭老師日常的身教與言教，更是讓學生學習的優良典範。

此外也要感謝彭文孝老師在可調視訊編碼專業領域與論文寫作上的諸多指導與建議，彭老師對於研究方面的專業與態度，讓我學習到很多視訊壓縮的知識，感謝彭老師在我疑惑時給予可行的研究方向並提供正確的研究目標。

從碩士班入學、逕讀博士班到現在拿到學位，回想這幾年的點點滴滴，彷彿回到了記憶裡的時光隧道。研究所初期，為了進入視訊壓縮的研究領域，在杭老師與實驗室學長的帶領下，開始了我的研究生涯。進入了通訊電子暨訊號處理實驗室(Commlab)這個大家庭，接觸了許多優秀的學長姐、同學與學弟妹，大家一起做研究、討論功課與聊天打屁…等，這些酸甜苦辣交雜的回憶，在我的生命中，都是珍貴且不可抹滅的。Commlab實驗室提供了一個極佳的研究環境，讓我在研究實驗中有充足的資源可以運用。也感謝實驗室全體成員(張峰城、洪崑健、蔡家揚、蔡彰哲、李志鴻、洪朝雄、蔡崇諺、呂家賢、黃育彰、陳旻弘、劉建志、陳勇竹、陳治傑、鄭凱庭、陳錫祺、林耀企、陳威年、葉尚諭、張順成、江清德、陳豐進、王志偉、曾劭學、陸凱暉、吳崇豪、吳思賢、陳呈毓、周正偉、李兆軒、柯俊言、翁郁婷…等)，營造了一個充滿活力、溫馨與和諧氣氛的環境，一直是身為實驗室一員所自豪的，感謝實驗室成員這些日子以來的照顧與幫助，有你們的陪伴，讓我的研究生生活過得更多采多姿，也希望實驗室夥伴都能在未來的人生路上，一切順利。

另外，我也要感謝我的口試委員：成大電機系的楊家輝教授、東華電機系的陳美娟教授、清大電機系的林嘉文教授、交大電子系的王聖智教授與交大資工系的彭文孝教

授。感謝您們在百忙之中能抽空給予我指導，也因您們的寶貴建議，使得論文能夠更加完備。

謝謝這些在研究上不斷幫助我的貴人，讓我能夠以更謙卑的態度來看待這個學位，期待這個博士學位能成為我以後不斷督促我進步的動力。

最後，我要感謝我的家人，感謝你們在這幾年來的照顧、協助與包容。沒有他們的鼓勵與支持，也就沒有我今天的成就。因此，謹以此論文獻給所有愛我的人與我所愛的人。

林鴻志

謹誌於台灣新竹交通大學

西元 2010 年 6 月



Table of Contents

摘要	i
Abstract	iii
誌謝	vi
Table of Contents	viii
List of Figures	xi
List of Tables	xiii
List of Symbols	xv
Chapter 1 Introduction	- 1 -
Section 1.1 Motivation	- 2 -
Section 1.2 Research Contributions	- 3 -
Section 1.3 Dissertation Organization	- 5 -
Chapter 2 Introduction to H.264/MPEG-4 SVC Coding System	- 7 -
Section 2.1 H.264/MPEG-4 AVC Architecture	- 8 -
2.1.1 Architecture Overview	- 8 -
2.1.2 Basic Coding Tools	- 9 -
2.1.2.1 Intra Prediction	- 9 -
2.1.2.2 Inter Prediction, Motion Estimation and Motion Compensation	- 12 -
2.1.2.2.1 Variable Block-Size Motion Compensation	- 12 -
2.1.2.2.2 Hierarchical-B Prediction Structure with Bi-directional Motion Compensation	- 13 -
2.1.2.3 Transform, Scaling, and Quantization	- 16 -
2.1.2.4 In-loop Deblocking Filter	- 18 -
2.1.2.5 Entropy Coding	- 20 -
Section 2.2 Additional Coding Tools in H.264/SVC	- 21 -
2.2.1 Overview of Layered Coding Structure	- 22 -
2.2.2 Inter-Layer Prediction Tools	- 23 -
2.2.2.1 Inter-Layer Motion Prediction	- 24 -
2.2.2.2 Inter-Layer Residual Prediction	- 25 -
2.2.2.3 Inter-Layer Intra Texture Prediction	- 25 -
Section 2.3 Rate-Constrained Coder Control in H.264/AVC-Based Video Standards	- 26 -
2.3.1 Optimization Using Lagrangian Schemes	- 27 -
2.3.2 Lagrangian Optimization in Hybrid Video Coding	- 28 -
2.3.2.1 Rate-Constrained Motion Estimation – Selection Process in Temporal Prediction Type	- 30 -

2.3.2.2	Rate-Constrained Mode Decision Process.....	- 32 -
Section 2.4	Problem Statement.....	- 33 -
2.4.1	Complexity Analysis in H.264/SVC Coder.....	- 34 -
2.4.2	Our Goal.....	- 37 -
Chapter 3	Fast Temporal Prediction Selection in H.264/AVC Temporal Scalable Video Coding	- 39 -
Section 3.1	Literature Review.....	- 40 -
Section 3.2	Observations and Analysis on Temporal Prediction at Temporal Enhancement Layers	- 44 -
3.2.1	Inheritance of Temporal Prediction Types.....	- 44 -
3.2.1.1	Prediction Type Distributions.....	- 44 -
3.2.1.2	Elimination of BI for Large Partitions.....	- 49 -
3.2.1.3	Consistency of FW and BW in Small Partitions.....	- 52 -
3.2.2	Rate-Distortion Contribution by BI.....	- 53 -
3.2.3	Rate-Distortion Relationships between Uni-directional Predictions and Bi-directional Prediction.....	- 56 -
3.2.3.1	Motion Vector Difference.....	- 57 -
3.2.3.2	Motion-Rate Cost.....	- 64 -
3.2.3.3	Distortion Relationship.....	- 66 -
Section 3.3	Proposed Schemes – Temporal Prediction Inheritance with Adaptive Thresholds for Bi-directional Prediction Selection.....	- 67 -
3.3.1	Adaptive Thresholds.....	- 68 -
3.3.2	Algorithm Overview.....	- 70 -
3.3.2.1	Early Termination on BI for Large Partitions.....	- 73 -
3.3.2.2	Adaptive Prediction Type Selection for Small Partitions.....	- 73 -
Section 3.4	Experimental Results and Discussions.....	- 74 -
3.4.1	Test Conditions.....	- 74 -
3.4.2	Performance Measures.....	- 74 -
3.4.3	Performance Comparison with JSVM.....	- 76 -
3.4.4	Performance Comparison with State-of-the-art Fast Algorithms.....	- 83 -
Chapter 4	Fast Mode Decision Algorithm for Intra-only Scalable Video Coding with Combined Coarse Granular Scalability (CGS) and Spatial Scalability	- 85 -
Section 4.1	Literature Review.....	- 86 -
Section 4.2	Statistical Analysis of Intra Predictions.....	- 87 -
4.2.1	Mode Correlation of Base and Enhancement Layers.....	- 87 -
4.2.2	Rate-distortion Profile of Intra Prediction.....	- 89 -
Section 4.3	Proposed Macroblock-Adaptive Rate-Distortion Estimation Algorithm [82].	- 91 -
4.3.1	Algorithm Overview.....	- 91 -

4.3.2	MacroblocK-Adaptive Rate-Distortion Estimation	- 93 -
4.3.3	Layer-Adaptive Intra Mode Selection	- 94 -
Section 4.4	Experiments	- 95 -
Chapter 5	Fast Mode Selection and Motion Search for Scalable Video Coding with Combined Coarse Granular Scalability (CGS) and Temporal Scalability.....	- 100 -
Section 5.1	Literature Review.....	- 101 -
Section 5.2	Correlations between Base and Enhancement Layers	- 105 -
5.2.1	Distributions of Intra Prediction Mode in CGS	- 105 -
5.2.2	Distributions of Inter Prediction Mode in CGS	- 110 -
5.2.3	Temporal Reference Frames between Coding Layers	- 118 -
5.2.4	Inter-Layer Residual Prediction in Transform/Pixel Domain	- 121 -
Section 5.3	Proposed Approaches – Layer-Adaptive Mode Decision and Motion Search-	126 -
5.3.1	Layer-Adaptive Mode Decision for Hierarchical-B Frames.....	- 132 -
5.3.1.1	Intra Mode Selection.....	- 132 -
5.3.1.2	Inter Mode Selection.....	- 133 -
5.3.2	Layer-Adaptive Reference Frame and Motion Reuse.....	- 135 -
Section 5.4	Experiments and Discussions	- 139 -
5.4.1	Test Conditions	- 139 -
5.4.2	Performance Measures.....	- 140 -
5.4.3	Simulation Results	- 142 -
5.4.4	Performance Comparison with State-of-the-art Fast Algorithms	- 149 -
Chapter 6	Conclusions and Future Work.....	- 153 -
Section 6.1	Concluding Remarks.....	- 153 -
6.1.1	Fast Bi-directional Prediction Selection in H.264/AVC Temporal Scalable Video Coding -	153 -
6.1.2	Fast Mode Decision Algorithm with MacroblocK-Adaptive Rate-Distortion Estimation for Intra-only Scalable Video Coding.....	- 154 -
6.1.3	Fast Context-adaptive Mode Decision Algorithm for Scalable Video Coding with Combined Coarse-grain Quality Scalability (CGS) and Temporal Scalability.....	- 155 -
Section 6.2	Future Work	- 156 -
Appendix	Distribution of the Approximated Distortion $\tilde{\mathcal{D}}_{BI}$.....	- 158 -
Bibliography	(in order of appearance).....	- 165 -
Curriculum Vitae	- 175 -
Publication	- 177 -

List of Figures

Fig. 2-1 H.264/AVC encoder structure [4]	- 9 -
Fig. 2-2 Directional prediction modes of intra 4x4 and the reference prediction pixels A to M....	- 10 -
Fig. 2-3 The scan order of sixteen 4x4 sub-blocks.....	- 11 -
Fig. 2-4 Prediction modes of intra 16x16	- 12 -
Fig. 2-5 Variable block-size macroblock partition.....	- 13 -
Fig. 2-6 An example of hierarchical-B prediction structure with GOP size = 16.....	- 15 -
Fig. 2-7 Syntax elements and their combinations for the inter-layer prediction in the coarse-grain quality scalability (CGS) [2][7]	- 23 -
Fig. 2-8 Selection process for choosing the best temporal prediction type.....	- 31 -
Fig. 2-9 Flowchart of mode decision at enhancement layer for hierarchical-B frames in JSVM 9.11 [10].....	- 35 -
Fig. 3-1 Distribution of temporal prediction types (FW, BW, and BI) at different temporal enhancement layers.....	- 48 -
Fig. 3-2 The performance index $\gamma_{N \times N}$ for individual hierarchical-B frame.....	- 55 -
Fig. 3-3 PDFs and CDFs of the motion vector difference d_T for 16x16 and 8x8 blocks with two selected Qp values	- 61 -
Fig. 3-4 Distributions of motion-rate costs \mathcal{R}_{BI} and \mathcal{R}_{FW+BW}	- 63 -
Fig. 3-5 Fast selection algorithm for temporal prediction types.....	- 70 -
Fig. 3-6 Comparisons in rate-distortion curve with GOP = 8.....	- 80 -
Fig. 3-7 Comparisons in rate-distortion curve with GOP = 16.....	- 81 -
Fig. 4-1 Block address mapping for intra direction mode: (a) CGS, (b) spatial scalability with 1-to-1 mapping, and (c) spatial scalability with 1-to-4 mapping.....	- 88 -
Fig. 4-2 Probability profiles of “similarity” between coding layers: (a) CGS and (b) spatial scalability. (FOREMAN)	- 89 -
Fig. 4-3 Rate-distortion profiles between CGS layers for (a) Intra4x4 and (b) Intra8x8 (FOREMAN)	- 90 -
Fig. 4-4 Fast mode decision algorithm with rate-distortion estimation and layer-adaptive mode selection	- 92 -
Fig. 4-5 Inter-layer dependency settings of H.264/SVC encoder [2] for (a) CGS, (b) dyadic spatial scalability, and (c) combined scalability.....	- 96 -
Fig. 5-1 Distribution of intra prediction types at CGS enhancement layers.....	- 108 -
Fig. 5-2 One-to-one block address mapping of CGS	- 108 -
Fig. 5-3 Similarity probability profiles of intra direction mode at CGS enhancement layer with poor-quality base layer ($Qp_B = 40$) and high-quality base layer ($Qp_B = 30$).....	- 109 -
Fig. 5-4 Conditional probability of inter partition mode at CGS enhancement layers for $Qp_B = 40$,	

Q_{pE} between 20 to 40, and GOP size = 16..... - 114 -

Fig. 5-5 Four regions representing different degrees of mode correlations between coding layers- 117 -

Fig. 5-6 Agreement in selecting reference frames between base layer and enhancement layer... - 120 -

Fig. 5-7 Inter-layer dependency structure in our scheme: (a) two-layer case, and (b) four-layer case- 127 -

Fig. 5-8 Flowchart of the proposed inter mode decision algorithm for CGS enhancement layers- 127 -

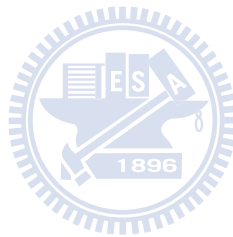
Fig. 5-9 Layer-adaptive mode set selection - 128 -

Fig. 5-10 Comparison of rate-distortion performance of JSVM 9.11 [10] at enhancement layers- 130 -

Fig. 5-11 Layer-adaptive selection in reference frame index and initial search point for hierarchical-B frames..... - 131 -

Fig. 5-12 Rate-distortion curves of JSVM 9.11 [10] and our approaches - 147 -

Fig. A-1 The average χ^2 test-statistic value for individual hierarchical-B frame..... - 162 -



List of Tables

Table 2-1 Determination of Boundary-Strength [27]	- 19 -
Table 2-2 Complexity ratio compared to IPPP coding structure (for a GOP).....	- 34 -
Table 3-1 Conditional probabilities of $p_{(16 \times 8 16 \times 16) \notin \text{BI}}$, $p_{(8 \times 16 16 \times 16) \notin \text{BI}}$, and $p_{(8 \times 8 16 \times 16) \notin \text{BI}}$	- 51 -
Table 3-2 Conditional probabilities of $p_{(4 \times 4 8 \times 8) \in \text{FW}}$ and $p_{(4 \times 4 8 \times 8) \in \text{BW}}$	- 52 -
Table 3-3 Average $\gamma_{N \times N}$ for 16x16, 8x8, and 4x4 blocks in each temporal enhancement layer (in percentage).....	- 55 -
Table 3-4 Optimal β^* value for the linear regression model (3.11)	- 65 -
Table 3-5 k value in $\Gamma(k, \theta)$ and ξ value for derivation of \hat{D}_{BI}	- 70 -
Table 3-6 Testing conditions	- 74 -
Table 3-7 Individual time saving contributed by TP and AT.....	- 77 -
Table 3-8 Performance comparisons with JSVM 9.11 [10] when GOP size is 8.....	- 78 -
Table 3-9 Performance comparisons with JSVM 9.11 [10] when GOP size is 16.....	- 79 -
Table 3-10 Overall time saving with various Qp values.....	- 83 -
Table 4-1 Look-up table for layer-adaptive intra mode selection	- 94 -
Table 4-2 Testing conditions	- 96 -
Table 4-3 Performance comparisons	- 98 -
Table 4-4 Layer complexity ratio of enhancement-layer encoding time to base-layer encoding time- 99 -	- 99 -
Table 5-1 Turning off the inter-layer residual prediction in transform domain for hierarchical-B frames (JSVM 9.11 [10])	- 122 -
Table 5-2 Turning off the inter-layer residual prediction in pixel domain for hierarchical-B frames (JSVM 8.0 [77]).....	- 123 -
Table 5-3 Encoding procedures on the hierarchical-B frames at CGS enhancement layers.....	- 124 -
Table 5-4 Coding type agreement between base layer and enhancement layer in hierarchical-B frames.....	- 129 -
Table 5-5 Candidate modes of inter prediction for $Qp_{n-1} > 30$	- 129 -
Table 5-6 Candidate modes of inter prediction for $Qp_{n-1} \leq 30$	- 130 -
Table 5-7 Candidate modes of sub-MB of inter prediction for all Qp values.....	- 130 -
Table 5-8 Testing conditions	- 139 -
Table 5-9 Average time saving of MD and MR/RF	- 143 -
Table 5-10 Performance comparisons with Qp setting of $(Qp_B, Qp_{E1}, Qp_{E2}, Qp_{E3}) = (40, 30, 20, 10)$	- 144 -
Table 5-11 Performance comparisons with Qp setting of $(Qp_B, Qp_{E1}, Qp_{E2}, Qp_{E3}) = (30, 20, 10, 0)$	- 145 -
Table 5-12 Average complexity ratio of the base layer to one CGS enhancement layer	- 146 -
Table 5-13 Performance comparisons with Li's method [80].....	- 151 -

Table 5-14 Performance comparisons with Li’s methods [80] and [88]..... - 152 -

Table 5-15 Performance comparisons with Ren’s method [89] - 152 -

Table A-1 The average Kolmogorov-Smirnov test-statistic value for temporal enhancement layer - 161 -

Table A-2 The average χ^2 test-statistic value for temporal enhancement layer..... - 161 -



List of Symbols

(in order of appearance)

BI	Bi-directional temporal prediction
CGS	Coarse-grain scalability
Q_p	Quantization parameter
GOP	Group of picture
T_i	The i -th temporal layer (T_0 : temporal base layer; T_i : temporal enhancement layer where $i > 0$)
FW	Forward temporal prediction
BW	Backward temporal prediction
Q_p^{init}	Initial quantization parameter
$Q_p^{(i)}$	Quantization parameter of temporal layer T_i
ABT	Adaptive block-size transforms
KLT	Karhunen-Loeve transform
DCT	Discrete cosine transform
Q_s	Quantization step size
B_s	Boundary strength used in the loop filter
CAVLC	Context-adaptive variable length coding
CABAC	Context-adaptive binary arithmetic coding
\mathbf{mv}	Motion vector
\mathcal{T}	Temporal prediction type (FW, BW, or BI)
$J_{\mathcal{T}}$	The Lagrangian cost of temporal prediction \mathcal{T} in rate-constrained motion estimation
$\mathcal{D}_{\mathcal{T}}$	Distortion measured as the sum of the absolute differences (SAD)
$R_{\mathcal{T}}$	The number of bits representing motion vector(s)
λ_{MOTION}	The Lagrangian multiplier in rate-constrained motion estimation
\mathcal{M}	Block partition mode
$J_{\mathcal{M}}$	The Lagrangian cost of partition mode \mathcal{M} in rate-constrained mode decision
$\mathcal{D}_{\mathcal{M}}$	Distortion measured as the sum of the absolute differences (SAD)
$R_{\mathcal{M}}$	The number of bits resulted from entropy coding
λ_{MODE}	The Lagrangian multiplier in rate-constrained mode decision
Ω	The set of uni-directional predictions (FW and BW)
$\mathbf{N} = (M, N)$	The sub-block (pixel set) specified by block mode \mathcal{M}
\mathbf{mv}_p	Predictive motion vector
$f(\mathbf{x})$	The current frame pixel value

$f'_{\text{FW}}(\mathbf{x})$	The pixel values of the forwardly reconstructed frame
$f'_{\text{BW}}(\mathbf{x})$	The pixel values of the backwardly reconstructed frame
\mathbf{mv}_{FW}	The motion vector found by FW
\mathbf{mv}_{BW}	The motion vector found by BW
$\mathbf{mv}_{\text{BI} \leftarrow \text{FW}}$	The motion vectors found by BI
$\mathbf{mv}_{\text{BI} \leftarrow \text{BW}}$	
$\mathcal{R}_{\mathcal{T}}$	The motion-rate cost defined by $\mathcal{R}_{\mathcal{T}} \triangleq \lambda_{\text{MOTION}} \times R_{\mathcal{T}}$
Φ	The set of all possible partition modes
ME_{R}	Motion estimation dedicated to the motion search with residual prediction
ME_{M}	Motion estimation dedicated to the motion search with motion prediction
$\text{ME}_{\text{R+M}}$	Motion estimation dedicated to the motion search with both residual and motion predictions
ME_{O}	Motion estimation without residual and motion predictions
$p(\mathcal{M} 16 \times 16) \notin \text{BI}$	The probability of both 16×16 partition and partition mode \mathcal{M} not belonged to BI, defined by $\Pr\{\mathcal{M} \notin \text{BI} 16 \times 16 \notin \text{BI}\}$, where $\mathcal{M} \in \{16 \times 8, 8 \times 16, 8 \times 8\}$
$p(4 \times 4 8 \times 8) \in \mathcal{T}$	The probability of temporal prediction inheritance for small block partitions, defined by $\Pr\{4 \times 4 \in \mathcal{T} 8 \times 8 \in \mathcal{T}\}$, where $\mathcal{T} \in \Omega$
$\omega_i \mathcal{T}^*$	The relative rate-distortion improvement of the best temporal prediction \mathcal{T}^*
$\mathcal{W}_{\mathcal{T}}$	The sum of the relative rate-distortion improvement from those blocks selecting the temporal prediction \mathcal{T}
$\gamma_{N \times N}$	BI performance index of $N \times N$ blocks
$d_{\mathcal{T}}$	The motion vector difference (Euclidean distance) of $\mathbf{mv}_{\mathcal{T}}$ and $\mathbf{mv}_{\text{BI} \leftarrow \mathcal{T}}$, where $\mathcal{T} \in \Omega$
PDF	Probability distribution function
CDF	Cumulative distribution function
$\mathcal{R}_{\text{FW+BW}}$	The sum of \mathcal{R}_{FW} and \mathcal{R}_{BW}
$\widehat{\mathcal{R}}_{\text{BI}}$	An estimation of \mathcal{R}_{BI}
β	The slope value representing the linear relationship of motion-rate costs
$\widetilde{\mathcal{D}}_{\text{BI}}$	An approximation of \mathcal{D}_{BI} , in which the prediction signal is the average of the two reference blocks found by FW and BW
$mv_{\mathcal{T}}^x$	The x-direction motion vector found by temporal prediction \mathcal{T} , where $\mathcal{T} \in \Omega$
$mv_{\mathcal{T}}^y$	The y-direction motion vector found by temporal prediction \mathcal{T} , where $\mathcal{T} \in \Omega$
$e_{\mathcal{T}}(x, y)$	The prediction error in the corresponding location (x, y) , defined by $f(x, y) - f'_{\mathcal{T}}(x - mv_{\mathcal{T}}^x, y - mv_{\mathcal{T}}^y)$
$\mathcal{D}_{\text{FW+BW}}$	An upper bound of \mathcal{D}_{BI} which is the average of \mathcal{D}_{FW} and \mathcal{D}_{BW}
$\Gamma(k, \theta)$	Gamma distribution where k is the shape parameter and θ is the scale parameter

$\widehat{\mathcal{D}}_{\text{BI}}$	An estimation of \mathcal{D}_{BI} by taking the mean value of a Gamma distribution
$\text{CDF}_{\Gamma(k,\theta)}^{-1}$	The inverse CDF of a Gamma distribution $\Gamma(k, \theta)$
$\xi_{M \times N}$	The ratio of $\widehat{\mathcal{D}}_{\text{BI}}$ to $\mathcal{D}_{\text{FW+BW}}$ for partition mode $M \times N$
$\delta_{M \times N}$	An adaptive threshold (for partition mode $M \times N$) used to eliminate inefficient BI computation
BDP (dB)	The averaged Y-PSNR loss by the Bjontegaard metric
BDR (%)	The averaged bit-rate increase by the Bjontegaard metric
TS	The overall time saving in encoding process
$\text{TS}_{\text{HBvsIP}}$	The additional computation of our approach as compared to the IPPP coding structure in JSVM reference software
Qp_B	Qp value at the base layer
Qp_E	Qp value at the enhancement layer
\mathcal{I}	Denote the intra prediction type, Intra4x4 or Intra8x8
$\mathcal{D}_E^{\mathcal{I}}(Qp_E)$	The distortion of an enhancement-layer macroblock for intra type \mathcal{I} and the enhancement-layer Qp is Qp_E
$\widehat{\mathcal{D}}_E^{\mathcal{I}}(Qp_E)$	An estimation of $\mathcal{D}_E^{\mathcal{I}}(Qp_E)$
$\alpha_{\mathcal{D}}^{\mathcal{I}}$	The decay ratio of distortion between CGS intra-coding layers
$\mathcal{R}_E^{\mathcal{I}}(Qp_E)$	The rate of an enhancement-layer macroblock for intra type \mathcal{I} and the enhancement-layer Qp is Qp_E
$\widehat{\mathcal{R}}_E^{\mathcal{I}}(Qp_E)$	An estimation of $\mathcal{R}_E^{\mathcal{I}}(Qp_E)$
$\alpha_{\mathcal{R}}^{\mathcal{I}}$	The increase ratio of rate between CGS intra-coding layers
ΔP	PSNR difference between our approach and the JSVM
ΔR	Bit-rate difference between our approach and the JSVM
Mode_{BL}	The optimal base-layer partition mode
Mode_{EL}	The optimal enhancement-layer partition mode
$\text{SubMode}_{\text{EL}}$	The optimal finer partition mode when Mode_{EL} is 8x8
Qp_{n-1}	The reference-layer Qp value
ref_{BL}	The reference frame indices of the best block mode at the base layer
ref_{EL}	The reference frame indices of the best block mode at the enhancement layer
r_0	The reference frame index of forward prediction
r_1	The reference frame index of backward prediction
$\text{kept_ref}_{\text{BL}}$	The reference frame indices of the sub-optimal block mode at the base layer
TS_E	The time reduction at the enhancement layers
T_{BL}	The base-layer encoding time
$J_{\mathcal{M}}^{N \times N}$	$J_{\mathcal{M}}$ with integer transform size of $N \times N$
\mathcal{K}	The Kolmogorov-Smirnov test-statistic value
\mathcal{Q}	The χ^2 -test statistic value

$\mathcal{G}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ The bivariate Gaussian distribution
 $\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ The bivariate Laplacian distribution



Chapter 1 Introduction

In the past few decades, the delivery of motion pictures over various channels, including the wired and wireless networks, becomes one of the popular applications, for instance, video on cell-phone and digital television broadcasting [1]. To resolve the constraints due to the heterogeneous network environments and the capability of terminals, a desirable video coding scheme encodes the video resource only once at the *highest resolution* that allows a partial decoding of the coded bit-stream for a specific target (bit-rate, frame rate, and resolution). Such a coding scheme is the so-called scalable video coding, which was recently developed and adopted by the international MPEG video standards. Currently, there are two major approaches on scalable video coding: one is the DCT-based scheme; the other is the wavelet-based coding method. The coding concepts of these two approaches are rather similar, particularly in removing the temporal redundancy. The scalable video coding extension of H.264/MPEG-4 AVC is a conventional block-based coding scheme and has been accepted as the ITU-T/MPEG standard in 2007 [2]. On the other hand, the newly coding structure realized by the wavelet technique potentially has its advantages, as mentioned in [3]. In this dissertation, we only focus on the scalable video coding extension [2] of H.264/MPEG-4 AVC [4] (referred hereafter as H.264/SVC), especially on the encoding complexity analysis and fast encoding parameter selection.

Section 1.1 Motivation

In response to the increasing demand for scalability features in many applications, the Joint Video Team has recently, based upon H.264/MPEG-4 AVC [4] (referred hereafter as H.264/AVC), standardized a scalable video coding standard [2] that furnishes spatial, temporal, quality (also termed as SNR) and their combined scalabilities within a fully scalable bit-stream. By employing *multilayer coding* along with *hierarchical temporal prediction* [5][6], H.264/SVC [2] 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, H.264/SVC [2] inherits all the coding tools of H.264/AVC [4] and additionally it incorporates an *adaptive inter-layer prediction* mechanism [7] for reducing the coding efficiency loss relative to the state-of-the-art single-layer coding [8][9]. 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 H.264/SVC scheme [2] from the scalable systems in the prior video coding standards.

An H.264/SVC encoder [2], 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 [10] employs a *bottom-up encoding process* that adopts the *exhaustive mode search* for encoding parameter selection. The exhaustive search strategy, though providing a good rate-distortion performance, spends a large amount of computations on evaluating

each possible coding option and it turns out that most of these options have little benefit in increasing coding efficiency.


Our study reveals that a large percentage of computations come from encoding the temporal/spatial/quality enhancement layers; more specifically, a quality enhancement layer requires approximately *three* times the computations of its base layer due to the extra motion search for inter-layer motion estimation and residual prediction. Fast encoding parameter selection algorithms are thus desirable and advisable for reducing the enhancement-layer computational complexity without significantly sacrificing the rate-distortion performance.

Section 1.2 Research Contributions

The contributions of this dissertation mainly focus on the development of fast parameter selection algorithms, as described below:

- The coding parameters of temporal prediction type in temporal scalability: Our proposed scheme provides up to 66% reduction in encoding time, or equivalently, 3x speed-up.
- The hierarchical prediction structure of H.264/SVC requires additional 250% and 120% computations, as compared to the low-delay IPPP coding structure and the hierarchical prediction structure without the bi-directional prediction, respectively. That is, the bi-directional prediction consumes more than 50% of overall encoding time.
- The temporal prediction type is heritable in the large block partitions (from 16x16 to

8x8).

- The proposed measure index of relative rate-distortion improvement shows that the bi-directional prediction does not offer much compression efficiency in the finer partition modes (smaller than 8x8). That is, the two uni-directional predictions are sufficient.
- The statistical goodness-of-fit test reveals that the two sets of prediction error generated by the two uni-directional predictions tend to be jointly Laplacian distributed.
- The coding parameters of the intra prediction mode both in CGS and spatial scalabilities: Our proposed algorithm averagely achieves 63% complexity reduction, nearly, 3x speed-up.
 - The optimal intra mode selected by the enhancement coding layer is usually the one chosen by the base layer or its adjacent modes.
 - The values of rate and distortion in the Lagrangian cost statistically have the log-linear relationship between the coding layers.
- The coding parameters of inter-prediction macroblock partition mode and inter-layer predictions in the combined CGS and temporal scalability: Our proposed method achieves 84% time saving in overall encoding process (almost 6x speed-up) and 20x speed-up in encoding the enhancement layers.

- One CGS enhancement layer requires more than 200% computations, as compared to that of the base layer. The additional evaluation is due to the selection in encoding parameters of inter-layer predictions.
- The coding mode of an inter-frame enhancement-layer macroblock can determine whether it should be intra-coded or inter-coded by referring to the coding type of the reference layer.
- Referencing to the inter partition mode at the reference layer is effective in reducing the parameter space of inter block mode. By observing the conditional mode distributions, various look-up tables are designed for use at the enhancement layer. Moreover, the partitions smaller than 8x8 is inefficient.
- The selection of reference frame list between coding layers is highly correlated. Moreover, the motion starting point can be adaptively selected by the motions determined at the base layer or the original motion vector predictor.
- The coding efficiency of the inter-layer residual prediction is limited when the base layer is in poor quality.

Section 1.3 Dissertation Organization

The rest of this dissertation is organized as follows. In the Chapter 2, the commonly used coding tools adopted by H.264/AVC [4] and the additional inter-layer prediction mechanisms [7] in H.264/SVC [2] are briefly described. The concept of rate-distortion optimization is also reviewed.

Chapter 3 statistically analyzes the heritage of the temporal prediction type in the hierarchical block partition and its strong correlation to the finer partition modes, and theoretically constructs a set of thresholds, all of which efficiently avoid the unnecessary evaluations. The rates, distortions, and the prediction directions of the intra prediction are highly correlated between the coding layers, as examined in Chapter 4. Chapter 5 investigates the consistency of the coding parameters from the base layer to enhancement layers. Lastly, in Chapter 6, this dissertation is concluded by summarizing our proposed algorithms and the future work as well.



Chapter 2

Introduction to H.264/MPEG-4 SVC Coding System

The H.264/SVC [2] encoding system [10] provides three modalities of scalability, including, spatial, quality, and temporal scalability. The spatial and quality scalabilities are realized by adopting the layered approach, in which the bottom coding layer is processed followed by the upper coding layers. By taking the advantage of hybrid coding scheme [11], each coding layer inherits the conventional H.264/AVC [4] coding tools to form its basic coding structure, in which the hierarchical-B prediction structure is employed to achieve the temporal scalability and produce bit-streams with different frame rates. Such a layered coding scheme can greatly improve the coding efficiency by exploiting the hierarchical-B prediction structure to remove the temporal redundancy and with additional inter-layer coding tools to reduce the redundancy between coding layers. However, the improved coding quality comes at a high cost of increased computations due mainly to the exhaustive search in finding the optimal coding parameters.

Therefore, in the following sections, we briefly reviewed the main coding tools in H.264/AVC [4] in Section 2.1. Section 2.2 introduces the added inter-layer coding tools with the new syntax elements. In Section 2.3, to obtain high efficiency in video coding, a commonly used approach, the Lagrangian techniques, can find the optimal tradeoff in terms of rate-distortion performance. Finally, the increased computational complexity is statistically analyzed in Section 2.4.

Section 2.1 H.264/MPEG-4 AVC Architecture

H.264/AVC [4] is the latest video coding standard, which is also known as MPEG-4 Part 10 or MPEG-4 Advanced Video Coding (AVC). This state-of-the-art coding standard is developed by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC Moving Picture Experts Group (MPEG) as the Joint Video Team (JVT). Moreover, the standardization of H.264/AVC [4] has been completed in 2003.

2.1.1 Architecture Overview

The block diagram of the H.264/AVC encoder [4] is shown in Fig. 2-1. Its main components are composed of intra prediction, motion estimation, motion-compensation, deblocking filter, transform, quantization, and entropy coding. The motion estimation and motion compensation remove the temporal redundancy; the intra prediction, transform, and quantization remove the spatial redundancy; and the entropy coding eliminates the syntax redundancy. In addition, the deblocking filter reduces the blocking artifact.

The encoding procedure of H.264/AVC [4] can be briefly described below. Firstly, an input frame is split into 16x16 pixels macroblocks. Intra-prediction or inter-prediction (motion estimation) is employed on each block to generate the residual block. These residual data are then transformed and quantized for further entropy coding. In addition, the residual data are also passed through the reconstruction loop, including the inverse transform, de-quantization, and deblocking filter, in order to reconstruct the reference frame for motion estimation.

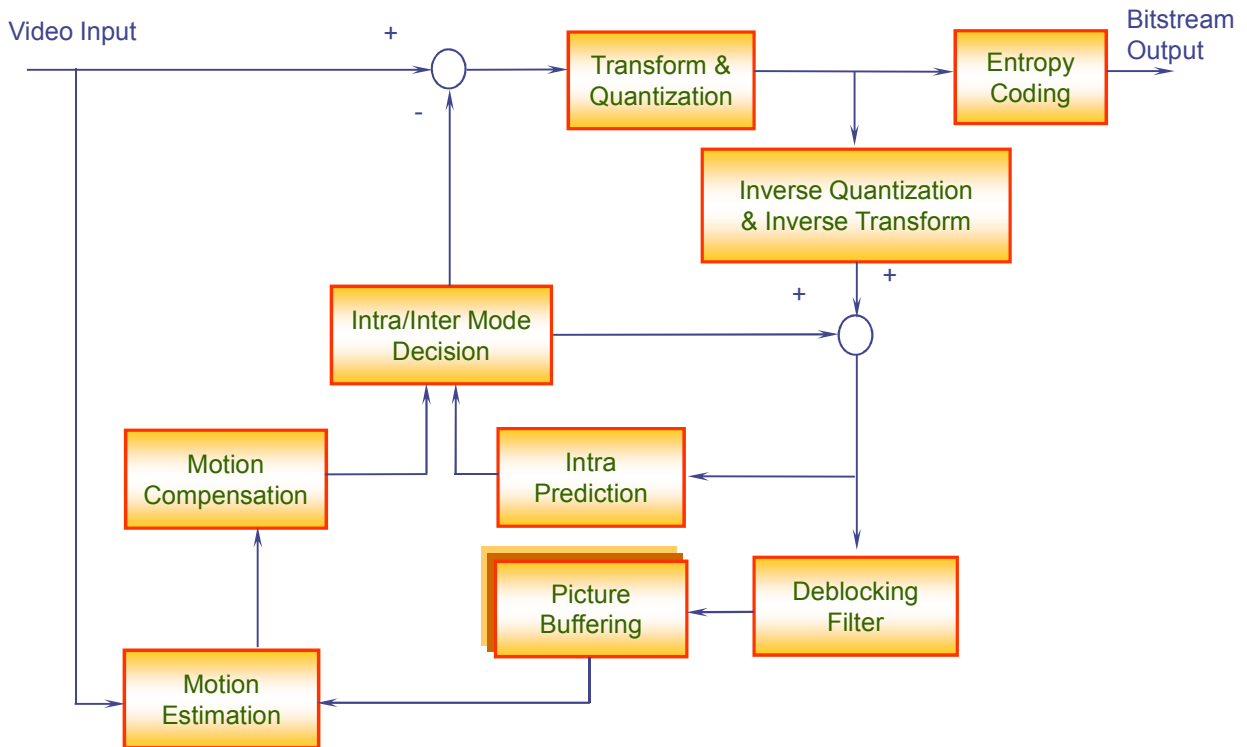


Fig. 2-1 H.264/AVC encoder structure [4]



2.1.2 Basic Coding Tools

With the high efficiency coding tools, H.264/AVC [4] can outperform the earlier MPEG-4 and H.263 standards, which provides better compression of video images. Because of its superior performance, H.264/AVC [4] is becoming the worldwide digital video standard for consumer electronics and personal computers. In the following, we briefly overview the concepts of the main coding tools in the H.264/AVC standard [4].

2.1.2.1 Intra Prediction

The correlation of neighboring area within a video frame is remarkably high. By using the intra prediction, the spatial redundancy of the neighboring region could be reduced. In H.264/AVC [4],

the basic intra-prediction element of luminance samples is 4x4 blocks, 8x8 blocks, or 16x16 blocks, and the basic intra-prediction element of chrominance samples is 8x8 blocks. The intra prediction for a macroblock generates the prediction values from its adjacent blocks borders (top-left, top, top-right, and left).

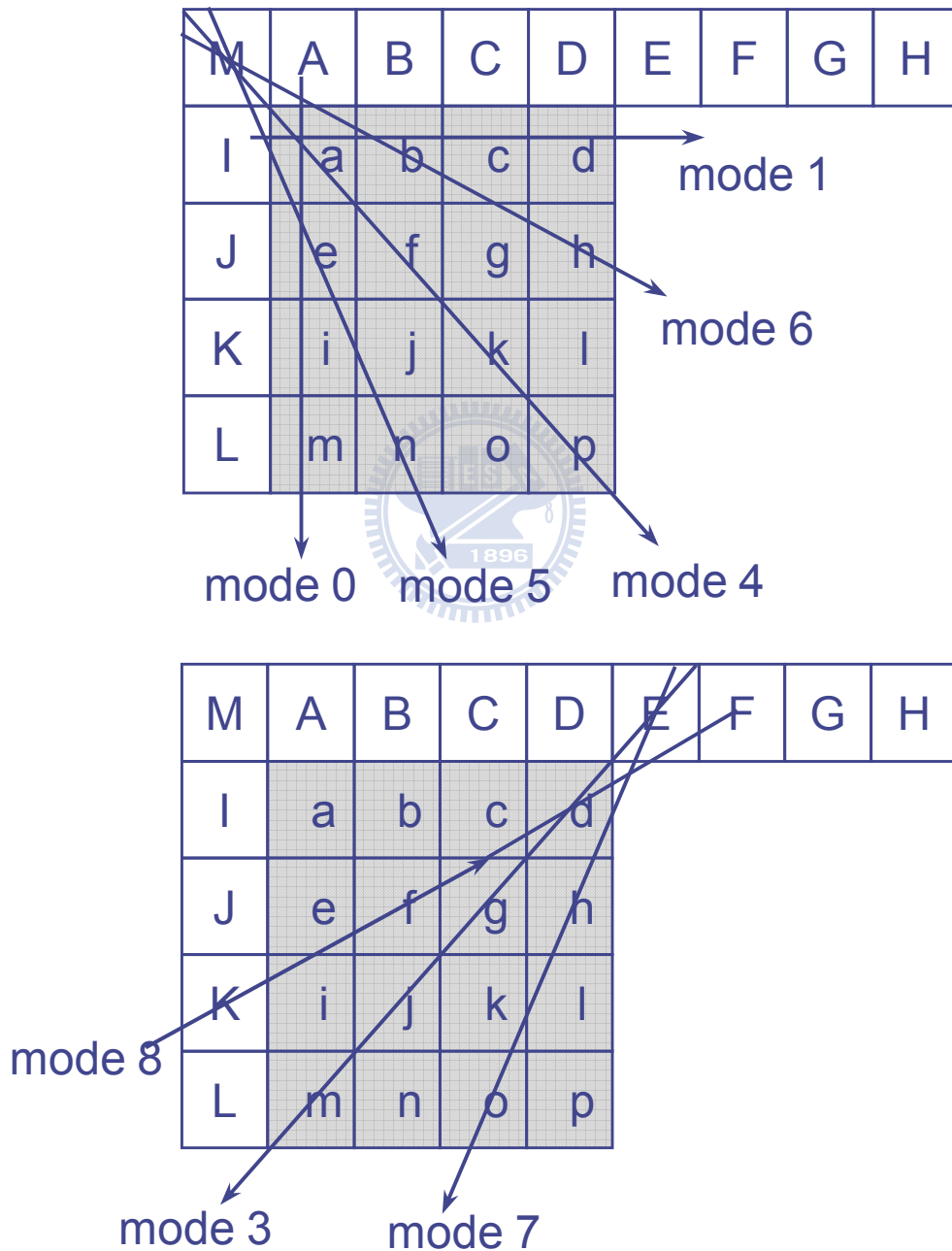


Fig. 2-2 Directional prediction modes of intra 4x4 and the reference prediction pixels A to M

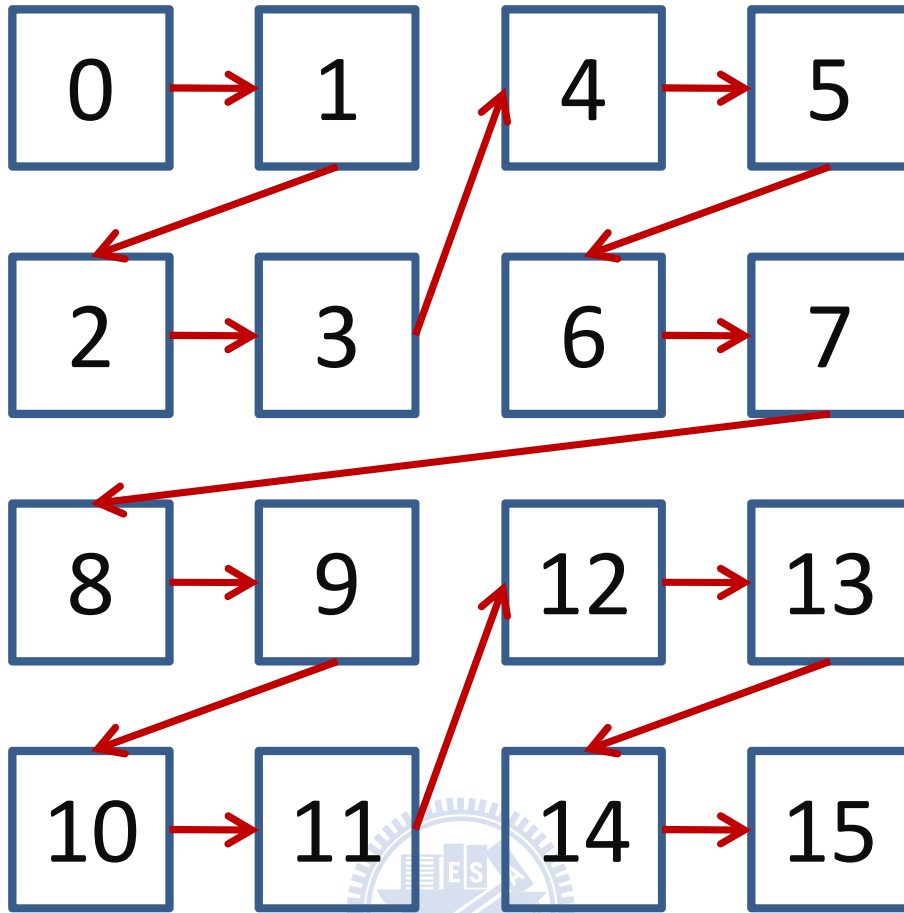


Fig. 2-3 The scan order of sixteen 4x4 sub-blocks

When a macroblock selects the intra 4x4 prediction for encoding, it is divided into sixteen 4x4 blocks. For each 4x4 sub-block, it chooses one of the nine modes, including eight directional prediction modes and the DC mode, to obtain high compression efficiency. As illustrated in Fig. 2-2, the pixels labeled A to M are the pixels of the adjacent blocks that have previously been encoded and reconstructed to form a prediction reference. The sixteen 4x4 blocks in a macroblock are processed in the pre-defined scan order as shown in Fig. 2-3. Moreover, the procedure of intra 8x8 prediction is similar to that of intra 4x4 prediction.

As an alternative to the intra 4x4 and intra 8x8, the whole macroblock could be predicted in a

single operation by selecting the intra 16x16 prediction mode. As shown in Fig. 2-4, there are four available prediction modes, which are vertical, horizontal, DC, and plane mode.

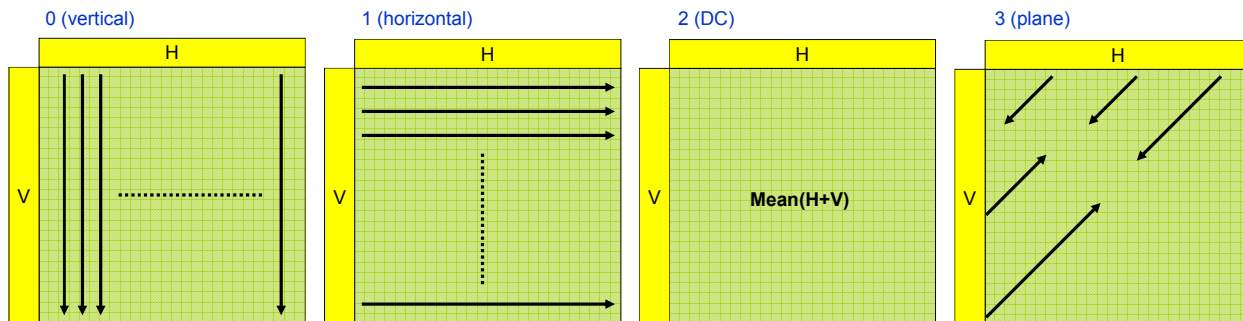


Fig. 2-4 Prediction modes of intra 16x16

2.1.2.2 Inter Prediction, Motion Estimation and Motion Compensation

In a high frame rate video sequence, the successive frames in a short time interval are very likely to be similar. Therefore, the concept of the inter prediction aims to generate predicted pixels from previous decoded frame. That is, an intelligent way to reduce the temporal redundancy is to transmit the difference between successive frames. Such a concept has been widely used in most video compression standards.

2.1.2.2.1 Variable Block-Size Motion Compensation

H.264/AVC [4] adopts a variable block-size macroblock partition to offer better compression quality as compared to the previous video standards. As depicted in Fig. 2-5, each macroblock could be divided in one of the four large block partitions, which are 16x16, 8x16, 16x8, and 8x8. Furthermore, each 8x8 block can be decomposed into finer partitions of 8x8, 4x8, 8x4, and 4x4 blocks if the 8x8 partition is preferred. The large partition might be chosen in smooth regions while the small partition

might work better in texture-complicated areas.

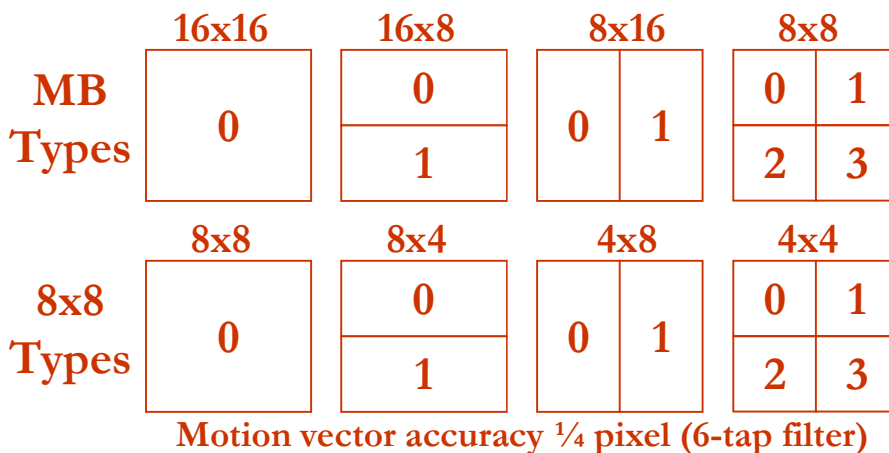


Fig. 2-5 Variable block-size macroblock partition

The partitioned sub-block inside an inter-coded macroblock are predicted from the same size partition region in the reference frame. The distance between these two regions is denoted by the motion vector, which needs to be transmitted to the decoder side. In H.264/AVC [4], the motion accuracy can be up to a $\frac{1}{4}$ -pixel resolution for luma components and $\frac{1}{8}$ -pixel resolution for chroma components. The pixel values of the fractional positions are computed by using interpolation filters, including a six-tap filter and a bilinear filter. The theoretical analysis of the fractional-pixel motion compensation can be found in [12][13].

2.1.2.2.2 Hierarchical-B Prediction Structure with Bi-directional Motion Compensation

With the support of variable block-size macroblock partition, the prediction structure also noticeably affects the coding efficiency. Recently, H.264/AVC [4] and its scalable extension adopt a widely used prediction structure, the hierarchical prediction structure, to offer an excellent coding performance if no any constraints in encoding delay.

Currently, H.264/AVC [4] and its scalable extension [2] can perform the dyadic hierarchical prediction [5][6] in the encoding configurations of their reference software. In this case, a set of successive images is partitioned into groups, so-called *Group of Pictures* (GOP), whose size is typically power of two. Furthermore, a GOP 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, \dots, T_n . The anchor frame of each GOP forms the temporal base layer which is either I- or P-frame. The remaining frames, located at the temporal enhancement layers, are coded as hierarchical-B frames. The B-frames have three candidates in temporal prediction, consisting of two uni-directional predictions, forward prediction (FW) and backward prediction (BW), and a bi-directional prediction (BI) mode. Moreover, the BI 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. 2-6 demonstrates an example of encoding frames by hierarchical-B prediction structure with GOP size = 16. If only pictures and anchor frames I_0/P_0 are transmitted, the reconstructed sequence at the decoder side has $\frac{1}{16}$ of the temporal resolution of the input video sequence. By additionally transmitting frames B_1 , the decoder can reconstruct the frame sequence that has one-eighth of the temporal resolution of the input video. Finally, if the remaining frames $B_2 \sim B_4$ are transmitted, a reconstructed video with the full temporal resolution is obtained.

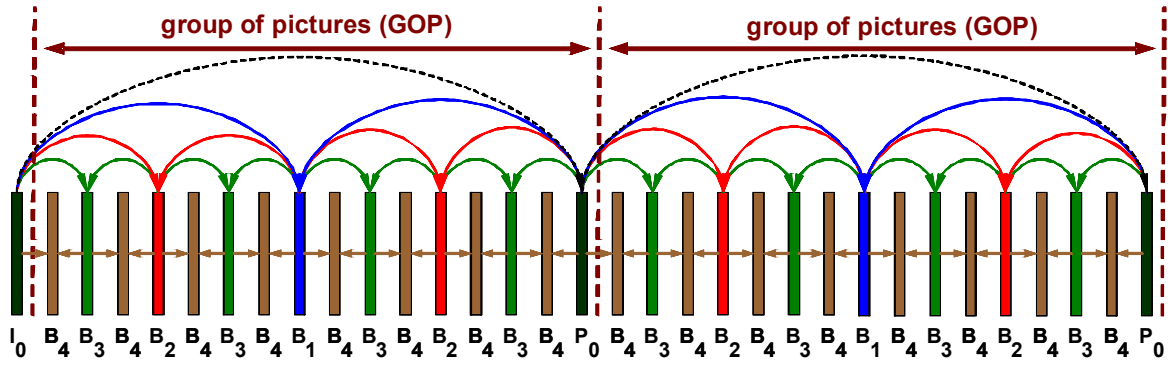


Fig. 2-6 An example of hierarchical-B prediction structure with GOP size = 16

Moreover, the coding efficiency for the hierarchical-B prediction structure is highly dependent on the assignment of quantization parameters (Qp) to the temporal layers [8]. With the GOP size being 2^n and a given initial quantization parameter Qp^{init} , the quantization parameters $Qp^{(t)}$ for a temporal layer T_t ($0 \leq t \leq n - 1$) are determined by

$$\begin{cases} Qp^{(X)} = Qp^{init} + (t \neq 0 ? 2 : 0) - 1.7(n - 1 - t) \\ Qp^{(t)} = \min(51, \max(0, \text{round}(Qp^{(X)}))) \end{cases} \quad (2.1)$$

With this Qp assignment strategy, temporal scalability achieved 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 more than 1.0 dB and 2.0 dB, respectively. Moreover, in this structure, each reference list with one reference frame is satisfactory to obtain a sufficiently high performance. Empirically, the maximum coding efficiency occurs when the GOP size is between 8 and 32, as reported in [9].

For the theoretical analysis of the B-frame and its related works, [14]–[20] provide the detailed explanations.

2.1.2.3 Transform, Scaling, and Quantization

Similar to previous video coding standards, H.264/AVC [4] makes use of block transform coding of the prediction error to remove the inter-pixel redundancy. Moreover, the concept of adaptive block-size transforms (ABT) is adopted for improving the subjective and objective quality. The ABT can apply transforms for the block sizes of 4x4, 4x8, 8x4, and 8x8 pixels. The basic transform coding process is very similar to that of previous standards. At the encoder, this process is composed of a forward transform, zig-zag scanning, scaling, and rounding as the quantization process followed the entropy coding. At the decoder, the inverse of the encoding process is performed, except for the rounding.

The transform coding in H.264/AVC [4] uses the separable 2D transform to process 2D block signal, which is written as

$$\mathbf{C} = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{T}^T \quad (2.2)$$

where \mathbf{S} denotes a matrix representing the prediction error of N pixels and N lines, \mathbf{T} is the $N \times N$ transform matrix, and \mathbf{C} is the transform coefficient matrix. This transform matrix consists of a set of orthogonal base functions. To minimize the computational complexity in transform coding, H.264/AVC [4] restricts the transformation computed exactly in integer arithmetic, avoiding inverse mismatch problems. In addition, the designed transform matrix should be close to the statistically optimal Karhunen-Loeve transform (KLT) and the discrete cosine transform (DCT) [21] can well approximate to the KLT [22]. Hence, according to these two constraints, a 4x4 transform [23] and an

8x8 transform [24] are specified by

$$\mathbf{T}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}, \mathbf{T}_4^{-1} = \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & -1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \quad (2.3)$$

and

$$\mathbf{T}_8 = \begin{bmatrix} 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\ 19 & 15 & 9 & 3 & -3 & -9 & -15 & -19 \\ 17 & 7 & -7 & -17 & -17 & -7 & 7 & 17 \\ 9 & 3 & -19 & -15 & 15 & 19 & -3 & -9 \\ 13 & -13 & -13 & 13 & 13 & -13 & -13 & 13 \\ 15 & -19 & -3 & 9 & -9 & 3 & 19 & -15 \\ 7 & -17 & 17 & -7 & -7 & 17 & -17 & 7 \\ 3 & -9 & 15 & -19 & 19 & -15 & 9 & -3 \end{bmatrix}, \mathbf{T}_8^{-1} = \mathbf{T}_8^T \quad (2.4)$$

Both of these two transforms are division free and can be implemented in a butterfly structure by using additions, bit-shift operations, and a few multiplications.

Furthermore, H.264/AVC [4] employs a hierarchical transform [25] to achieve a better inter-pixel de-correlation; that is, the DC coefficients of the adjacent 4x4 transform blocks are grouped into another 4x4 block and transformed again by a second-level transform.

After the ABT, the quantization process is the step that introduces signal loss to remove the psychovisual redundancy. For a given Qp value, the encoder performs quantization and scaling, in which the details can be referred to [23][26]. The quantization parameter Qp , which ranges from 0 to 51, is used to determine the quantization step size Qs for quantizing the transform coefficients \mathbf{C} . The quantization parameter Qp and quantization step size Qs are related by

$$Qs = 2^{Qp/6} \times \nu(Qp \bmod 6) \quad (2.5)$$

where

$$\left\{ \begin{array}{l} \nu(0) = 0.675 \\ \nu(1) = 0.6875 \\ \nu(2) = 0.8125 \\ \nu(3) = 0.875 \\ \nu(4) = 1.0 \\ \nu(5) = 1.125 \end{array} \right.$$

These Q_p values are formulated so that an increase of unity in Q_p means an increase of Q_s by approximately 12%. It can be also noticed that an increase of unity in Q_p roughly reduces the coding bit-rate by 12%.

2.1.2.4 In-loop Deblocking Filter

In encoding the successive frames by H.264/AVC [4], two coding tools mainly causes the blocking artifacts [27]. The most significant one is the block-based transform coding where the coarse quantization usually introduces the visual discontinuities at the block boundaries. The other is the motion compensated prediction where the reference blocks are usually copied from the interpolated data from different areas of different reference frames. The copying process carries the existing edges into the interior of the block to be compensated. Although the small-size transform in H.264/AVC [4] can marginally reduce this phenomenon, a deblocking filter is still an advantageous coding tool to maximize the coding performance.

Two common schemes can integrate the feature of the deblocking filter into video coding standards: post filters or in-loop deblocking filters. The post filter operates in displaying and it is outside of the coding loop. Moreover, it needs an additional frame buffer to pass the filter frames to the display device. On the other hand, the so-called in-loop deblocking filter is applied within the

coding loop to generate the filtered frames as the reference for motion compensation.

Employing the filtering inside the coding loop is superior to the post filtering, as listed below.

- The in-loop deblocking filter can preserve a certain level of quality in coded frames.
- As compared to the post filter, there does not need any extra frame buffer while decoding.

In the method of in-loop deblocking filter, it is realized by macroblock-wise checking the edge strength and the filtered data are directly stored in the reference frame buffer.

- The empirical experiments demonstrate the in-loop deblocking filter can improve the objective and subjective quality.

Table 2-1 Determination of Boundary-Strength [27]

Block modes and coding parameters	Bs
One of the blocks is Intra and the edge is a macroblock edge	4
One of the blocks is Intra	3
One of the blocks has coded residuals	2
Difference of block motion ≥ 1 luma sample distance	1
Motion compensation form different reference frames	1
Else	0

As mentioned, a Boundary-Strength (Bs) parameter, ranging from 0 to 4, is assigned to each edge between two adjacent 4x4 luminance pixel blocks. Depending on coding modes and the coding

parameters of these two blocks, Table 2-1 determines how the B_s value can be obtained. Then, according to the B_s value, the in-loop deblocking filter detects and analyzes the blocking artifacts, and attenuates them by employing a selected filter. Further information in the concept the filter design can be referred to [28]–[40].

2.1.2.5 Entropy Coding

In H.264/AVC [4], two methods of entropy coding are supported to remove the coding redundancy in representing the transmitted signal. The simpler entropy coding method adopts a single infinite-extent codeword table for all syntax elements, except for the quantized transform coefficients. The chosen single codeword table is an Exp-Golomb code which is very simple and regular to decode. While coding the residual data, a block of transform coefficients is mapped into a 1D data using a pre-defined scanning pattern, such as the zia-zag scan.

For representing the quantized transform coefficients, a more efficient method called Context-Adaptive Variable Length Coding (CAVLC) is employed. The VLC tables for various syntax elements are switched depending on prior coded syntax elements. Since the VLC tables are designed to match the corresponding conditional statistics, the entropy coding performance is improved in comparison to those using a single VLC table.

In the CAVLC process, the following items are coded in a proper order: number of nonzero coefficients, sign marks of trailing ones, levels of remaining nonzero coefficients, number of total zeros, and runs of zeros between nonzero coefficients. In encoding process, these coefficients should

be scanned in the reversed zig-zag order before coding.

The efficiency of entropy coding can be further improved if the Context-Adaptive Binary Arithmetic Coding (CABAC) is used [41]. The usage of arithmetic codes [42] can most easily resolve the inter-symbol redundancy, which allows the assignment of a non-integer number of bits to each symbol of an alphabet. Moreover, it is extremely beneficial for symbol probabilities that are greater than 0.5.

In H.264/AVC [4], the arithmetic coding core engine consists of three elementary steps: binarization, context modeling, and binary arithmetic coding. The binarization uniquely maps a given non-binary valued syntax element to a binary sequence. Another important feature of CABAC, the context modeling, estimates conditional probabilities based on the statistics of prior coded syntax elements. Then, these conditional probabilities are used for switching several estimated probability models. Finally, the arithmetic coding core engine and its estimated probability model are specified as multiplication-free and low-complexity methods by using only shifts and table lookups. As compared to CAVLC, the CABAC typically provides a bit-rate saving between 5% to 15%. More details on CABAC can be found in [41].

Section 2.2 Additional Coding Tools in H.264/SVC

To have a better understanding of our coding algorithms, this section explains the basic concepts of H.264/SVC [2] and its coder control. Some degree of familiarity with H.264/AVC [4] is assumed herein. The reader is referred to the overview paper [9] for details of H.264/AVC [4] and its scalable

extension [2].

2.2.1 Overview of Layered Coding Structure

In order to support the spatial, temporal, and fidelity (SNR) scalabilities, H.264/SVC [2] 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 chapter, however, we consider only the dyadic temporal scalability case so that we can use the current release of JSVM software [10].

In the spatial dimension, H.264/SVC [2] adopts the conventional approach of image pyramid to represent a source video sequence at various spatial resolutions [43]. The spatial encoding process begins with a multi-resolution decomposition of the original high-resolution sequence. The lowest-resolution sequence is coded by H.264/AVC [4] as the base layer, and each higher resolution sequence is coded sequentially as a spatial enhancement layer. 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 base layer and the

enhancement layers have identical spatial resolutions, but different quantization step sizes.

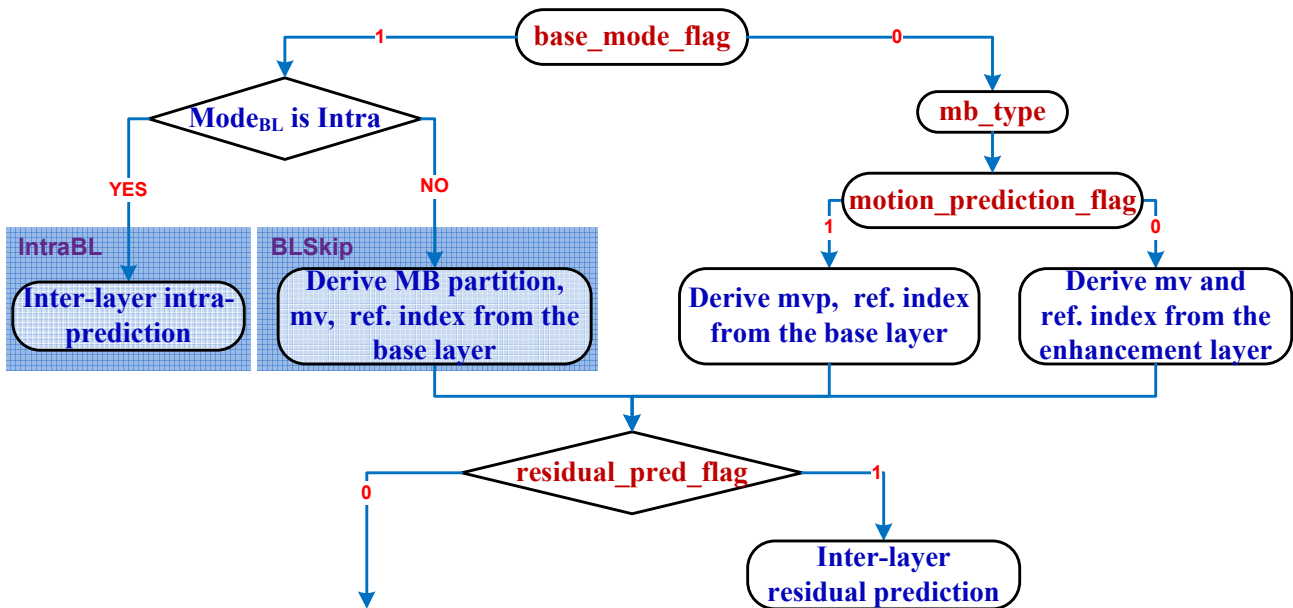


Fig. 2-7 Syntax elements and their combinations for the inter-layer prediction in the coarse-grain

quality scalability (CGS) [2][7]



2.2.2 Inter-Layer Prediction Tools

To achieve the high coding efficiency goal, H.264/SVC [2] has an adaptive inter-layer prediction mechanism [7], which enables the usage of as much decoded information of the reference/base layer as possible to be reused for the enhancement layers. H.264/SVC [2] adopts the inter-layer prediction tools, which are *inter-layer motion prediction*, *inter-layer residual prediction*, and *inter-layer intra-prediction*.

Despite certain restrictions, these inter-layer prediction tools can be combined together to form a number of coding modes for each enhancement-layer macroblock. Fig. 2-7 shows all possible combinations of the `base_mode_flag`, `motion_prediction_flag`, and `residual_prediction_flag`, as well

as their associated coding modes. The detailed information is described in the following subsections.

2.2.2.1 Inter-Layer Motion Prediction

To avoid repeatedly sending the same motion parameters in the cases when the enhancement layer cannot benefit from motion refinement, a flag (*base_mode_flag*) can be sent for each non-skipped macroblock to indicate whether its motion parameters (partition mode, reference 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 macroblock 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 are predicted from the reference/base layer.

For CGS/spatial enhancement layers, H.264/SVC [2] has the syntax element *base_mode_flag* to signal a new macroblock type, termed as BLSkip mode, in which only the residual signal needs to be transmitted. When the *base_mode_flag* is equal to 1 and the reference-layer macroblock is inter-coded, the enhancement-layer macroblock is also inter-coded. In this case, the partition mode of the enhancement-layer macroblock with the associated reference indices and motion vectors are derived from the information of the co-located reference-layer block.

In addition, the (scaled) motion vectors of the reference-layer co-located block can be used to be the motion vector predictor to the enhancement-layer macroblock if it is conventionally inter-coded. In this case, the syntax element *motion_prediction_flag* is set to 1 and the reference indices of the co-located reference-layer block are reused.

2.2.2.2 Inter-Layer Residual Prediction

To enhance the coding efficiency of inter-coded macroblock within the framework of *single-loop decoding*, the residual prediction, which subtracts the residual signal of the reference/base layer from that of the enhancement layer, can be adaptively activated by the *residual_prediction_flag*. The inter-layer residual prediction can be applied for all inter-coded macroblocks in each coding layer.

When the *residual_prediction_flag* is set to 1, the residual signal from the corresponding reference-layer block (up-sampled by using a bilinear filter [43]) is used as the prediction of the residual signal of the enhancement-layer macroblock.

2.2.2.3 Inter-Layer Intra Texture Prediction

To provide a better prediction for the enhancement-layer 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 macroblock is an intra-coded macroblock with *constrained intra prediction*, because the *single-loop* structure prohibits the reference/base layer to conduct motion compensation after it being coded.

The inter-layer intra-prediction occurs when an enhancement-layer macroblock is coded with the *base_mode_flag* equal to 1 and the co-located reference-layer block is intra-coded. The enhancement-layer prediction signal is the reconstructed intra-signal (up-sampled by one-dimensional four-tap FIR filters applied horizontally and vertically [43]).

Section 2.3 Rate-Constrained Coder Control in H.264/AVC-Based Video Standards

In most video coding standards including MPEG-2 Visual [44], H.263 [45], MPEG-4 Visual [46], and H.264/AVC [4] as well as its scalable coding standard [2], their specifications only represent the syntax structure of the bit-stream in the decoding process. However, the operational control of the video encoder is an important issue in video compression [47].

The bit allocation can resolve the problem in the operational control for efficient coding. With a motion-compensated hybrid coder, the bit-rate represents the total consumed bits, consisting of the motion vectors, the residual data, and additional side information. Those transmitted data are divided into N independent bit-streams with bit-rate R_i , which yields the overall rate

$$R = R_1 + R_2 + \cdots + R_N. \quad (2.6)$$

At the decoder, there exists distortion \mathcal{D} between the original frame and the reconstructed frame. We assume that the distortion-rate function $\mathcal{D}(R_1, R_2, \cdots, R_N)$ is strictly convex and differentiable everywhere, and that

$$\frac{\partial \mathcal{D}}{\partial R_i} \leq 0, 1 \leq i \leq N. \quad (2.7)$$

That is, increasing the rate to any one of the bit-streams should decrease the distortion. Hence, the optimum bit allocation that minimizes \mathcal{D} subject to a fixed overall rate R can be found by letting

$$d\mathcal{D} = \left(\frac{\partial \mathcal{D}}{\partial R_1} \right) dR_1 + \left(\frac{\partial \mathcal{D}}{\partial R_2} \right) dR_2 + \cdots + \left(\frac{\partial \mathcal{D}}{\partial R_N} \right) dR_N = 0. \quad (2.8)$$

Moreover, from Eq. (2.6), we derive

$$dR = dR_1 + dR_2 + \dots + dR_N = 0. \quad (2.9)$$

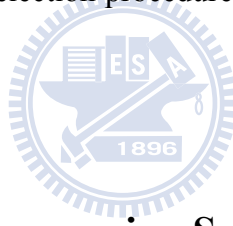
Furthermore, from Eq. (2.8) and Eq. (2.9), we can find the optimum bit allocation condition

$$\frac{\partial \mathcal{D}}{\partial R_1} = \frac{\partial \mathcal{D}}{\partial R_2} = \dots = \frac{\partial \mathcal{D}}{\partial R_N}. \quad (2.10)$$

This result provides an interpretation that we should add bits to the bit-stream with the smallest $\frac{\partial \mathcal{D}}{\partial R_i}$.

This allocation principle would obtain the greatest decrease in distortion.

However, the application of this bit allocation scheme to control a hybrid video coder is not straightforward. In the following subsections, the concept of Lagrangian optimization schemes is briefly reviewed. Its application to control a hybrid video coder is then introduced. Particularly, based on the Lagrangian approach, the selection procedures of the temporal prediction types and the best block modes are described in detail.



2.3.1 Optimization Using Lagrangian Schemes

The task of coder control aims to determine a set of coding parameters and produces the corresponding bit-stream such that the optimal coding efficiency can be achieved with a given rate constraint. Recently, the most widely adopted approach is the Lagrangian bit allocation scheme due to its effectiveness and simplicity.

The Lagrangian optimization approaches transfer the constrained problem (see (2.11)) to an unconstrained problem (see (2.12)) by introducing a new variable λ , called the Lagrangian multiplier.

$$\min_{\mathcal{C}} \mathcal{D}(\mathcal{C}) \text{ subject to } R(\mathcal{C}) \leq R_c \quad (2.11)$$

$$\mathcal{C}^* = \arg \min_{\mathcal{C}} J(\mathcal{C}|\lambda) \text{ with } J(\mathcal{C}|\lambda) = \mathcal{D}(\mathcal{C}) + \lambda R(\mathcal{C}) \quad (2.12)$$

(where \mathcal{C} is the combination of coding options, R_c is the given rate constraint, and J denotes the cost function)

Because the combination of coding options is finite, the discrete optimum solution to this unconstrained problem was introduced in [48].

2.3.2 Lagrangian Optimization in Hybrid Video Coding

During the video encoding, a variety of coding parameters such as motion vectors, block partition modes, transform coefficient levels and transform block sizes have to be determined. Thus, those coding parameters can be intelligently selected with respect to the rate-distortion efficiency by using the Lagrangian optimization techniques [49]. Minimizing the Lagrangian cost has to proceed over the space of the coding parameters for all blocks in the entire video sequence, introducing a great amount of computation while encoding.

Typically, for each macroblock, the coding block mode with associated parameters is optimized with the given decisions of prior coded blocks only. In other words, the optimization problem

$$\min_{\mathcal{C}} \sum J(\mathcal{C}|\lambda) \quad (2.13)$$

is simplified to

$$\sum \min_{\mathcal{C}} J(\mathcal{C}|\lambda), \quad (2.14)$$

which can be easily solved by independently selecting the coding parameter for each block.

To ensure the high compression performance, the hybrid encoders H.264/AVC [4] and

H.264/SVC [2] adopt the Lagrange multiplier method [49]–[51] to choose the most suitable block partition (mode) that leads to the optimal tradeoff between distortion \mathcal{D} and bit-rate R . In general, two major nested processes are fully checked to obtain the optimal coding parameters: one is the temporal prediction mode and the other is the block mode.

A simple and widely accepted method for these two set of coding options is to firstly search for the motion vector(s) \mathbf{mv} that minimizes the Lagrangian cost $J_{\mathcal{T}}$ expressed as

$$J_{\mathcal{T}} = \mathcal{D}_{\mathcal{T}} + \lambda_{\text{MOTION}} \times R_{\mathcal{T}} \quad (2.15)$$

where $\mathcal{D}_{\mathcal{T}}$ denotes the distortion measured as the sum of the absolute differences (SAD) and $R_{\mathcal{T}}$ is the number of bits representing the motion vector(s). Simultaneously, the optimal temporal prediction type \mathcal{T}^* is selected. Then, with the given \mathcal{T}^* (or the selected motion vector(s)), the Lagrangian mode decision for a macroblock proceeds to produce the best partition mode \mathcal{M}^* by minimizing

$$J_{\mathcal{M}} = \mathcal{D}_{\mathcal{M}} + \lambda_{\text{MODE}} \times R_{\mathcal{M}} \quad (2.16)$$

where $\mathcal{D}_{\mathcal{M}}$ denotes the distortion measured as the sum of the squared differences (SSD) and $R_{\mathcal{M}}$ is bit-rate resulted from entropy coding. In conclusion, the effective coding parameters are decided by minimizing a Lagrangian cost function $J = \mathcal{D} + \lambda \times R$ that weights the distortion \mathcal{D} of a macroblock against the bit usage R using a Lagrangian multiplier λ .

Depending on the use of SAD or SSD, the Lagrangian multipliers λ_{MOTION} and λ_{MODE} has to be adjusted appropriately. In [50] and [51], the empirical results have shown the following

relationship is efficient for H.264/AVC-based coders:

$$\lambda_{\text{MODE}} = 0.85 \times 2^{(Qp-12)/3} \quad (2.17)$$

$$\begin{cases} \lambda_{\text{MOTION}} = \sqrt{\lambda_{\text{MODE}}} & , \text{if SAD is used} \\ \lambda_{\text{MOTION}} = \lambda_{\text{MODE}} & , \text{if SSD is used} \end{cases} \quad (2.18)$$

2.3.2.1 Rate-Constrained Motion Estimation – Selection Process in Temporal Prediction Type

In the hierarchical-B prediction structure, if a specific block mode \mathcal{M} with $M \times N$ size (in pixel) is applied to a macroblock, there are $\frac{16 \times 16}{M \times N}$ sub-blocks inside this macroblock. 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 rate-distortion cost $J_{\mathcal{T}}$:

$$\begin{aligned} \mathcal{T}^* &= \arg \min_{\mathcal{T} \in \Omega \cup \{\text{BI}\}} \{J_{\mathcal{T}}\} \\ &= \arg \min_{\mathcal{T} \in \Omega \cup \{\text{BI}\}} \{\mathcal{D}_{\mathcal{T}}(\mathbf{mv}) + \lambda_{\text{MOTION}} \times R_{\mathcal{T}}(\mathbf{mv} - \mathbf{mv}_p)\} \end{aligned} \quad (2.19)$$

where

$$\Omega = \{\text{FW}, \text{BW}\},$$

λ_{MOTION} , the Lagrange multiplier formula used in the H.264/AVC-based encoders, is specified by Eq. (2.18),

$\mathcal{D}_{\mathcal{T}}(\mathbf{mv})$ is the pixel distortion, usually, SAD using two motion vectors given by

$$\mathcal{D}_{\mathcal{T}}(\mathbf{mv}) = \sum_{\mathbf{x}=\mathbf{0}}^{\mathbf{N}-1} |f(\mathbf{x}) - (1 - \omega)f'_{\text{FW}}(\mathbf{x} - \mathbf{mv}_0) - \omega f'_{\text{BW}}(\mathbf{x} - \mathbf{mv}_1)| \quad (2.20)$$

$\mathbf{N} = (M, N)$ is the sub-block (pixel set) specified by mode \mathcal{M} ,

\mathbf{mv}_p is the *predictive motion vector* generated by a motion vector predictor, (Usually, it is a linear combination of the neighboring motion vectors.)

$R_{\mathcal{T}}(\mathbf{mv} - \mathbf{mv}_p)$ 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, and

$f'_{FW}(\mathbf{x})$ and $f'_{BW}(\mathbf{x})$ are the pixel values of the forwardly and the backwardly reconstructed frames, respectively.

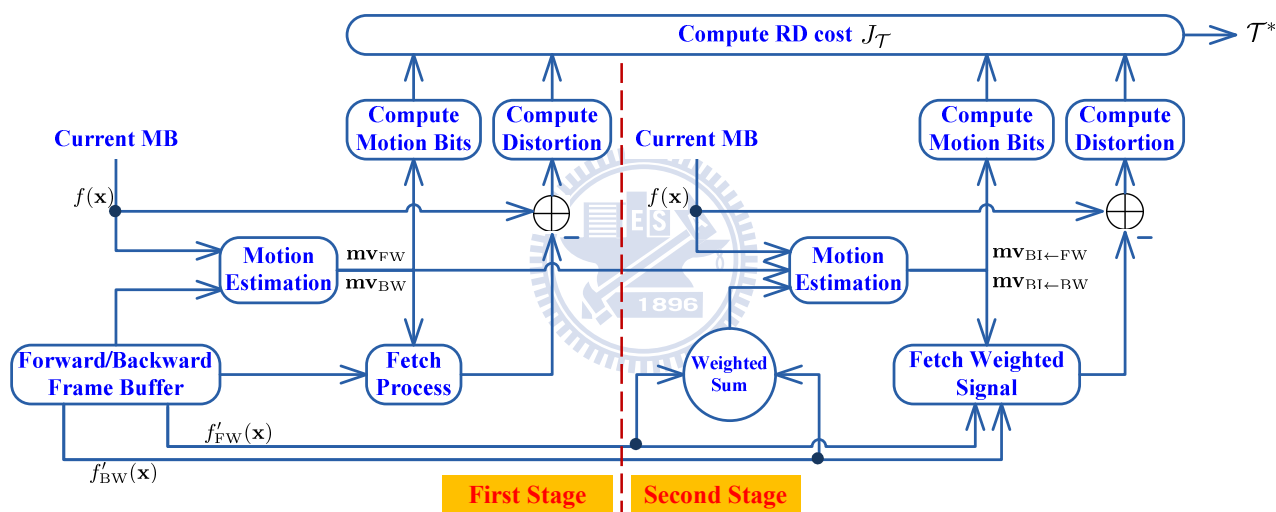


Fig. 2-8 Selection process for choosing the best temporal prediction type

As depicted in Fig. 2-8, the JSVM encoder [10] 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}_{FW} and \mathbf{mv}_{BW} by minimizing the costs J_{FW} and J_{BW} , separately. In computing Eq.

(2.20), we first set $\omega = 0$ and thus $\mathbf{mv}_0 = \mathbf{mv}_{FW}$ for FW; and then set $\omega = 1$ and thus $\mathbf{mv}_1 = \mathbf{mv}_{BW}$ in the case of BW.

- Second stage (BI): As mentioned in [16], the motion vectors found by FW and BW are sub-optimal for BI. Therefore, the JSVM [10] takes the motion vectors \mathbf{mv}_{FW} and \mathbf{mv}_{BW} as the initial search points. Then, these two motion vectors are locally refined to find the optimal motion pair for BI, denoted as $\mathbf{mv}_{BI \leftarrow FW}$ and $\mathbf{mv}_{BI \leftarrow BW}$. Note that the distortion in Eq. (2.20) computes the difference between the current MB and the average of two reference macroblocks by setting $\omega = \frac{1}{2}$, $\mathbf{mv}_0 = \mathbf{mv}_{BI \leftarrow FW}$, and $\mathbf{mv}_1 = \mathbf{mv}_{BI \leftarrow BW}$.

Finally, three sets of distortions $\mathcal{D}_{\mathcal{T}}$ and motion-rate cost $\mathcal{R}_{\mathcal{T}} \triangleq \lambda_{\text{MOTION}} \times R_{\mathcal{T}}$ are collected and compared to produce the best rate-distortion costs $J_{\mathcal{T}}$ and \mathcal{T}^* . For example, there are two sub-blocks in the block mode 16x8. 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 rate-distortion performance of this 16x8 block mode. Next, we pick up the 8x8 mode and there are thus 4 sub-blocks. The motion search process is repeated for each sub-block and at the end we obtain the rate-distortion cost of this 8x8 mode. We try all possible modes and finally, we compare all these candidate modes and select the least rate-distortion cost block mode \mathcal{M}^* .

2.3.2.2 Rate-Constrained Mode Decision Process

For each block mode \mathcal{M} , each of its $\frac{16 \times 16}{M \times N}$ sub-blocks select its own optimal temporal prediction type by the rate-constrained motion estimation. In order to determine the best coding mode for a

macroblock, similarly, the optimal block mode \mathcal{M}^* is chosen via the rate-constrained mode decision process, which minimizes the rate-distortion cost $J_{\mathcal{M}}$:

$$\begin{aligned}\mathcal{M}^* &= \arg \min_{\mathcal{M} \in \Phi} \{J_{\mathcal{M}}\} \\ &= \arg \min_{\mathcal{M} \in \Phi} \{\mathcal{D}_{\mathcal{M}} + \lambda_{\text{MODE}} \times R_{\mathcal{M}}\}\end{aligned}\quad (2.21)$$

where

Φ denotes the set of all possible partition modes,

the Lagrange multiplier λ_{MODE} is defined by Eq. (2.17),

the distortion $\mathcal{D}_{\mathcal{M}}$ is the distance between original macroblock and the reconstructed data in the SSD sense, and

$R_{\mathcal{M}}$ denotes the number of bits representing the quantized residual data and other side information that needs to be transmitted.

Thus, the mode decision from Eq. (2.21) can output the most preferable block mode \mathcal{M}^* with the proper temporal prediction type(s).

Section 2.4 Problem Statement

Although the decoding complexity was well studied and amended during the design phase of H.264/SVC [2], its encoding complexity has rarely been addressed. In the following, we analyze the additional computations required by the hierarchical-B prediction structure, as compared to the low-delay prediction structure. The heavy complexity at the enhancement layers is also studied.

2.4.1 Complexity Analysis in H.264/SVC Coder

In H.264/SVC [2], the temporal scalability realized by the hierarchical-B prediction forms the basic coding structure in each quality or spatial layer. 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.

Table 2-2 Complexity ratio compared to IPPP coding structure (for a GOP)

Test Sequence		Prediction Structure			
		Hierarchical-B		Hierarchical-B with FW and BW 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 $Qp = 40, 35, 30, 25$; Each reference list has one reference frame only.

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 FW and BW only. As listed in Table 2-2, 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 type. For example, there are $\frac{2^n - 1}{2^n} \times 100\%$ of frames (in

percentage) use also BW (and BI) for $GOP = 2^n$. Therefore, a hierarchical prediction structure using a large GOP size results in more computations.

On average, as compared to the IPPP coding structure, the hierarchical-B prediction structure introduces additional 250% computations and the other (FW and BW only) needs only about extra 45% encoding time. More specifically, the complexity ratio of the uni-directional prediction and the BI is as follows.

$$(FW + BW) : BI \cong \begin{cases} 1 : 1.26 & , Qp = 40 \\ 1 : 1.23 & , Qp = 35 \\ 1 : 1.21 & , Qp = 30 \\ 1 : 1.19 & , Qp = 25 \end{cases} \quad (2.22)$$

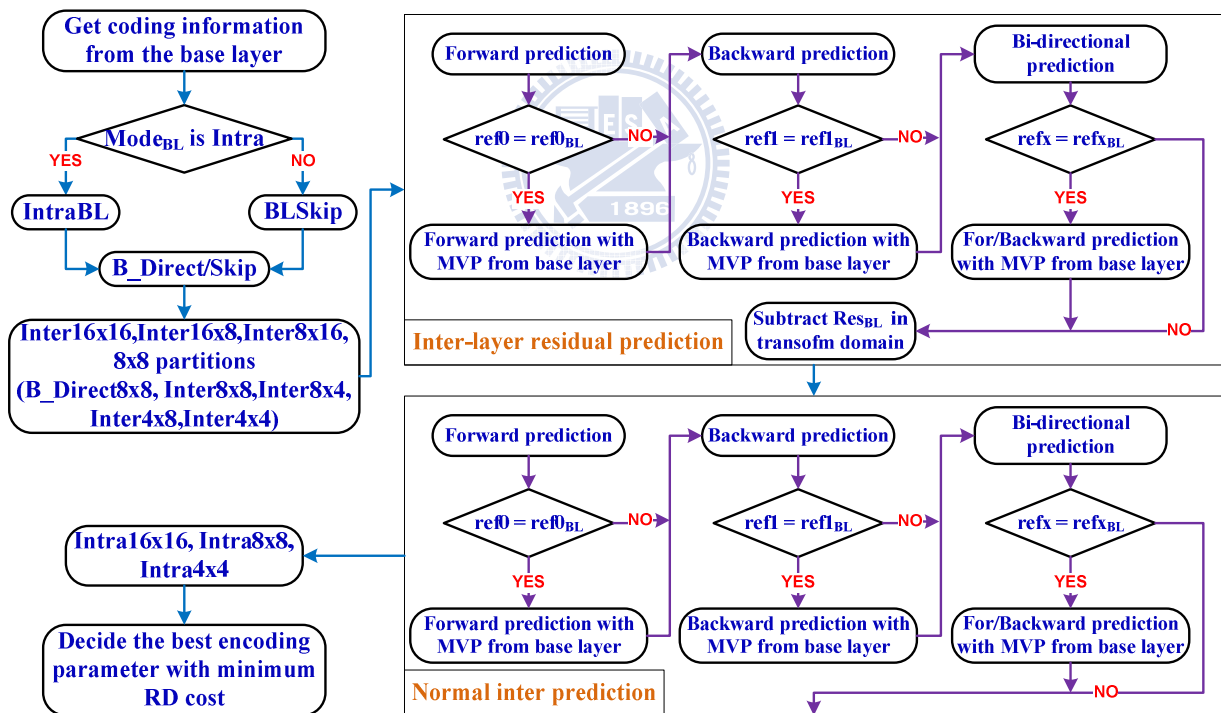
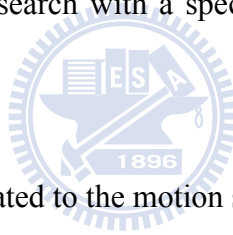


Fig. 2-9 Flowchart of mode decision at enhancement layer for hierarchical-B frames in JSVM 9.11

[10]

On the other hand, Fig. 2-9 describes the mode decision process of JSVM [10], with an elaboration on the motion search procedure. It starts with the inter-layer residual prediction. For each

admissible macroblock partition, the search for its motion parameters begins with a series of motion estimation processes that use J_T as the search criterion. Thus, an optimal combination of motion vectors, reference picture indices, and prediction modes is first found in the inter-layer residual prediction. In the second part of the procedure, all the motion estimation processes are repeated with a replaced search criterion. Now, J_M is used in place of J_T 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 motion vector of the co-located macroblock in the reference/base layer is always examined. In summary, the motion search involves four types of motion estimation processes. Each of them is dedicated to a motion search with a specific use of motion vector prediction and residual prediction:




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

The single-layer coding only perform ME_O , but H.264/SVC [2] can do all four types of motion estimation processes, which explains the prolonged latency needed for H.264/SVC encoding. Based on this observation we expect that the complexity ratio of a CGS enhancement layer 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 [10]. 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.

For example, in a typical encoding experiment with the combined temporal and CGS, it takes about 10 to 40 minutes of CPU time, depending on the number of enhancement layers, to encode a two-second CIF video clip.

2.4.2 Our Goal



In summary, an H.264/AVC-based scalable encoder adopts the layered coding structure and applies the hierarchical prediction structure in order to produce scalable bit-streams and simultaneously maintain high coding efficiency. The expense of the scalable features and the greatly improved coding efficiency, however, is the penalty of heavily increased computations. They mainly come at a cost of the two nested exhaustive search (selection of temporal prediction type and mode partition) in the hierarchical-B frames at the temporal enhancement layers in each coding layer.

Hence, fast encoding algorithms are thus desirable and advisable for eliminating the unnecessarily computational load in evaluating BI and reducing the enhancement-layer computational complexity while achieving a similar level of coded picture quality. In this dissertation, we propose several efficient methods due to the following statistical observations.

- Intra-layer correlations in the hierarchical prediction structure
 - The selection of temporal prediction type in the hierarchical mode partition has a certain consistency in the large block sizes.
 - The two uni-directional predictions can achieve a similar coding performance as compared to that evaluating all three temporal prediction types. It is particularly true in the small block partitions.

- Inter-layer correlations in the mode partition
 - Between coding layers, the intra predictions are dominated by the IntraBL and Intra4x4. Moreover, the intra prediction direction at the enhancement layer usually selects the one chosen at the base layer or its adjacent predictions.
 - Between coding layers, the distortions and the rate costs in intra prediction has the log-linear property.
 - At the enhancement layers, the candidates of block partition can be determined by referring to the selected mode at the reference layer.
 - The temporal predictions between coding layers have a certain consistency.
 - The motion vectors found at the base layer can be a good starting point for the enhancement layers.

Chapter 3

Fast Temporal Prediction Selection in H.264/AVC Temporal Scalable Video Coding

In this chapter, we propose a fast algorithm that efficiently selects the temporal prediction type for the dyadic hierarchical-B prediction structure in the H.264/AVC temporal scalable video coding. Referring to the best temporal prediction type of 16x16, we utilize the strong correlations of prediction type inheritance to eliminate the unnecessary computations for BI prediction in the finer partitions, 16x8/8x16/8x8. In addition, we carefully examine the relationship of motion-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 8x8, either FW or BW is skipped based upon the information of an 8x8 partition. Hence, the proposed schemes can efficiently reduce the extensive computations burden in performing the BI prediction. As Compared to the JSVM 9.11 [10], our method saves the encoding time from 60% to 66% for a great number of test videos over a typical range of coding bit-rates and its coding penalty is negligible.

The rest of this chapter is organized as follows. As compared to the hierarchical-B prediction structure, the prior works related to the multiple reference frames and fast algorithms in mode selection are analyzed in Section 3.1. Section 3.2 contains a brief review of the hierarchical

prediction structure and its decision process of temporal prediction type in JSVM encoder [10]. Its dramatically encoding complexity is also revealed, as compared to the IPPP coding structure. Section 3.3 summaries our observations on the correlations among three temporal prediction types. Based on these analyses, Section 3.4 presents our fast bi-directional prediction selection algorithm. In Section 3.5, our proposed scheme is compared with JSVM 9.11 [10] and the state-of-the-art algorithms [72][73] in terms of complexity reduction and rate-distortion performance.

Section 3.1 Literature Review

To enhance the compression efficiency in the IPPP-coding H.264/AVC [4], a typical way performs the long-term prediction [53], termed the multiple reference frames, to reduce the prediction error potentially. Apparently, its complexity is linearly proportional to the number of used reference frames. Hence, a number of prior studies have been proposed to reduce the extra computations of this strategy. In [54], the accuracy of motion estimation (integer-pel, half-pel, and quarter-pel) is utilized to classify how the location of the moving object locates in each available reference frame. An object labeled as the same accuracy on two or more reference frames implies that the same shifted texture can be found in those references. Therefore, it is sufficient to select the closest one as the candidate. Furthermore, in [55] and [56], they study the continuity in moving objects to construct motion maps among different reference frames. Based on the motion trajectory, the initial starting point on farther references has to be conjectured. After the motion from the most recent frame is known, its reference area may cover several blocks. The motions from other farther references are estimated by the

weighted sum [55] or the median [56] of those motions of the covered blocks. With the temporary predictive motions, the truly motions can be searched in a much smaller search range. These two schemes [55][56] make use of the motion information from the first previous frame. Instead of that, the proposed approach [57] obtains the composed motions by a combination of blocks from difference references to provide a more accurately initial guess. Moreover, in [58]–[60] the selection of multiple reference frame is early determined by a set of termination conditions, such as all-zero block detection, the energy of prediction error, and the optimal block partition in the first previous frame. However, for practical use in H.264/AVC [4], Huang *et al.* [59] empirically illustrate that the coding performance gained by the multiple reference frames is highly dependent on the content of sequences, not on the number of searched references. This observation is then theoretically evidenced in [60]. That is, turning on the multiple reference frames does not usually have noticeable improvement (say, more than 1 dB) in rate-distortion measure, but incurs huge computations in motion search.

On the other hand, a fairly large body of literature has been proposed on the complexity reduction of the H.264/AVC coder [4], based on the estimated rate-distortion cost as thresholds and/or the mode selection. In [61], the candidate modes and the thresholding rate-distortion cost are given by the temporally and spatially neighboring area. Furthermore, the transformed residuals and the corresponding coding bits have a highly linear correlation [62]. Based on the non-zeros quantized transform coefficients, the proposed schemes [62][63] construct the improved rate-distortion

estimator to alleviate the entropy coding and the reconstruction operations during the mode decision process. In order to entirely avoid coding bits computation of residuals, the distortion, the required motions, and the header of block type are used to develop a new cost function of mode selection [64]. Moreover, the so-called early termination is a popular approach in mode selection [65]–[71]. For example, multiple termination criteria eliminate the sophisticated mode search by hierarchically setting from large partitions to small block sizes [65]. In [66] and [67], the sufficient conditions in detecting all-zero blocks are theoretically studied to skip testing unnecessary small partitions. In addition, the spatio-temporal motion characteristics are considered to arrange the mode set [68]–[70]. An object along its moving trajectory determines its motion activity to pick up its dominate modes [68][69], while the spatial motion homogeneity is split into multiple levels to obtain a subset of partition modes [70][71]. All the above schemes are applicable to the IPPP/IBP/IBBP coding structures, in which the coded frame and its reference are very close, but few focus on the hierarchical prediction in the superior H.264/SVC temporal scalability. Moreover, those H.264/AVC-based fast algorithms could not work well, because the correlations between the current frame and its reference may not be sufficiently strong and reliable for being used at lower temporal layers. In [72], 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.* [73] make use of the statistical hypothesis testing to conditionally skip the partitions smaller than 16x16. However, only the encoding parameters of the mode partitions are considered to apply the fast algorithms in these methods.

Up to now, it is surprised that 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 conduct more than triple of the total motion search calculation, as compared to the IPPP coding structure. Therefore, the aim of this paper is to design a fast temporal prediction selection algorithm for the dyadic hierarchical-B prediction structure in H.264/SVC [2]. To achieve this goal, we statistically analyze the correlations of temporal prediction types in large partitions and show that the BI has limited coding benefits in small partitions [74]. The correlations of motion-rates among three temporal prediction types are examined and they are formulated by a first-order regression model. Additionally, the relationships among the distortions are also investigated [75] and the prediction error of the uni-directional temporal predictions have 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 inheritance in the temporal prediction types and the use of adaptive thresholds in the hierarchical-B prediction structure of H.264/SVC [2]. Our simulations show very promising results. On the average, our approaches can provide up to 66% overall encoder time saving over JSVM 9.11 [10], which is equivalent to three times faster in the encoding process.

Section 3.2 Observations and Analysis on 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 H.264/SVC temporal enhancement layers. In the Subsection 3.2.1 we examine the prediction type distributions and their inheritances in the hierarchical blocks from large partitions to small ones. Then, Subsection 3.2.2 analyzes the relative coding efficiency contributed by the BI. In terms of the rate-distortion costs and the motion-rate costs, the last subsection explores their correlations between the uni-directional predictions 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 the experiments. The training set contains three MPEG test videos: FOOTBALL (CIF, fast motion), FOREMAN (CIF, median motion), and MOBILE (CIF, complicated texture).

3.2.1 Inheritance of Temporal Prediction Types

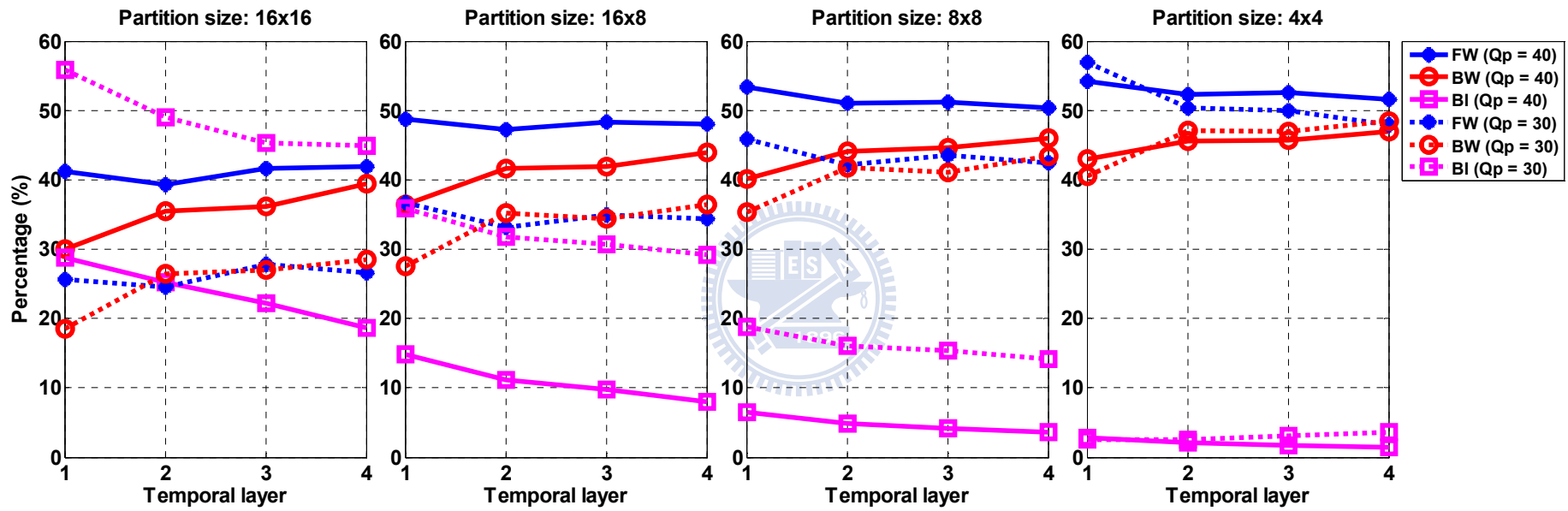
In this subsection, we collect the probability distributions of temporal prediction types.

3.2.1.1 Prediction Type Distributions

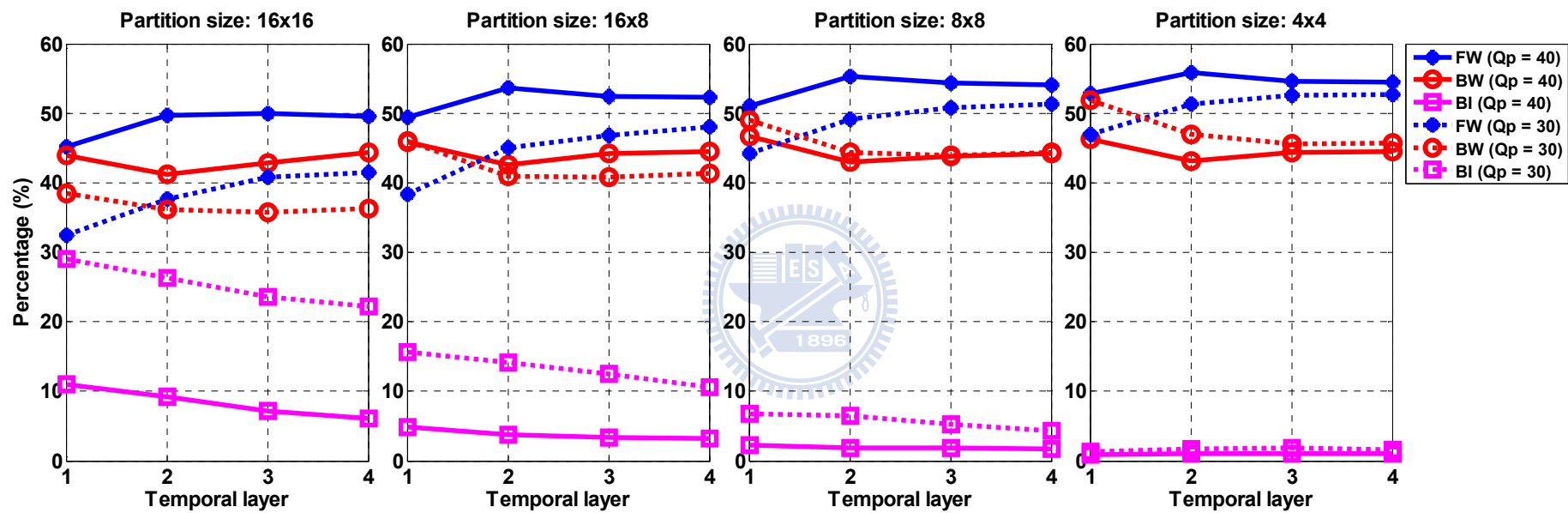
We first gather the probability distribution of temporal prediction types used in several distinct block partitions at various temporal enhancement layers and at different Qp values. In any temporal layer of the hierarchical-B prediction structure, the temporally forward and backward reference frames

have equal distance away from the current frame. If the objects in test have constant movement, 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. 3-1 for, particularly, the MOBILE sequence.

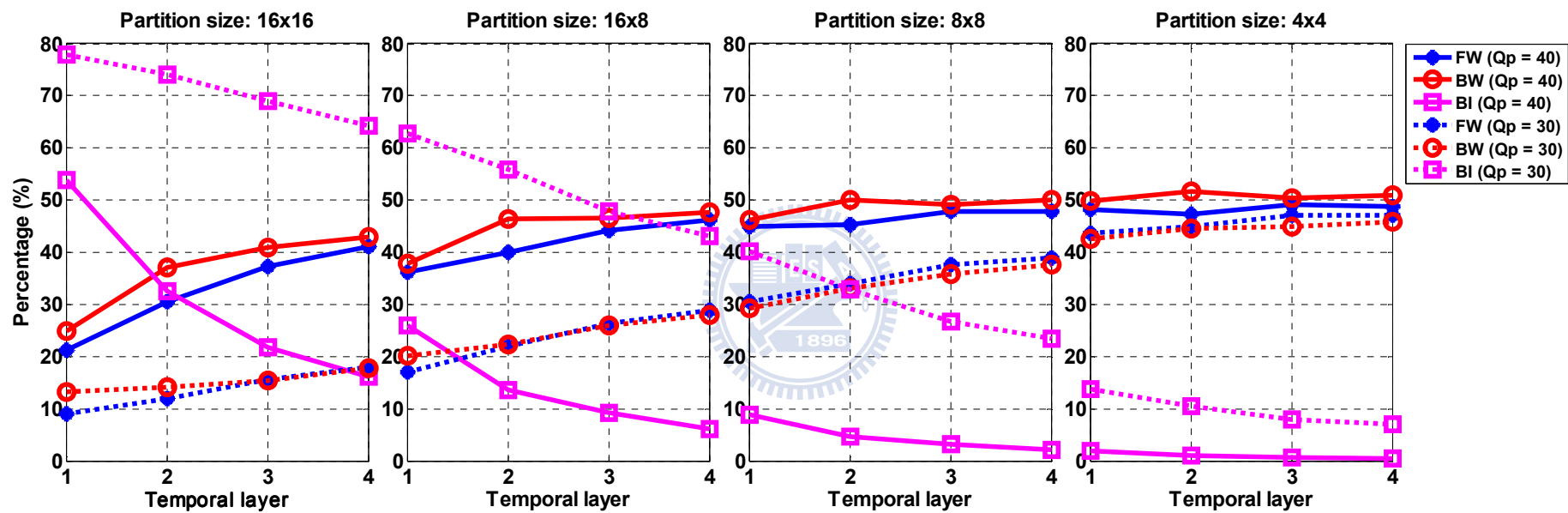




(a) FOOTBALL



(b) FOREMAN



(c) MOBILE

Fig. 3-1 Distribution of temporal prediction types (FW, BW, and BI) at different temporal enhancement layers

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 general, BI is favored at large partitions (16×16 or 16×8) because BI offers more accurate motion compensation at a small motion bit-rate overhead. The distortion reduction by good motion vectors is less significant in the low-motion or motionless areas. Thus, the FORMAN sequence uses fewer BI types. Clearly, distant reference frames reduce the motion compensation effectiveness. Therefore, BI percentage goes down drastically at higher temporal layers. On the other hand, BI needs to transmit more mv's, twice as many as those of the FW/BW. If the reduced distortion provided by BI cannot compensate for the increased motion-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 for the complex-textured sequence MOBILE.

In summary, BI benefits the 16×16 and $16 \times 8/8 \times 16$ 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.

3.2.1.2 Elimination of BI for Large Partitions

Our second observation focuses on the use of BI in the $16 \times 8/8 \times 16/8 \times 8$ partitions. As discussed in

Subsection 3.2.1.1, the BI probability in these partitions is much smaller than that in the 16x16 block size. That is, for instance, $\Pr\{16 \times 8 \in \text{BI}\}$ is less than $\Pr\{16 \times 16 \in \text{BI}\}$, which implies $\Pr\{16 \times 8 \notin \text{BI}\} > \Pr\{16 \times 16 \notin \text{BI}\}$. In order to find out whether or not these two groups $\{16 \times 16 \notin \text{BI}\}$ and $\{16 \times 8 \notin \text{BI}\}$ overlap with each other, we consider three conditional probabilities defined below:

$$p_{(16 \times 8 | 16 \times 16) \notin \text{BI}} \triangleq \Pr\{16 \times 8 \notin \text{BI} | 16 \times 16 \notin \text{BI}\}, \quad (3.1)$$

$$p_{(8 \times 16 | 16 \times 16) \notin \text{BI}} \triangleq \Pr\{8 \times 16 \notin \text{BI} | 16 \times 16 \notin \text{BI}\}, \text{ and} \quad (3.2)$$

$$p_{(8 \times 8 | 16 \times 16) \notin \text{BI}} \triangleq \Pr\{8 \times 8 \notin \text{BI} | 16 \times 16 \notin \text{BI}\}. \quad (3.3)$$

In Table 3-1, these three conditional probabilities are higher than 80% in all cases, and are about 95% on average. Moreover, they are very close to one at higher temporal enhancement layers. This strong correlation indicates that the uni-directional prediction types are inheritable from the 16x16 partition to 16x8/8x16/8x8 partitions. Thus, the information of the prior evaluation on 16x16 BI can provide a very reliable estimate to the use of BI for the 16x8/8x16/8x8 partitions at both low and high bit-rates. We can quite accurately eliminate the use of BI in those partitions.

Table 3-1 Conditional probabilities of $p_{(16 \times 8 | 16 \times 16) \notin BI}$, $p_{(8 \times 16 | 16 \times 16) \notin BI}$, and $p_{(8 \times 8 | 16 \times 16) \notin BI}$

Test Sequence	Qp	$p_{(16 \times 8 16 \times 16) \notin BI}$				$p_{(8 \times 16 16 \times 16) \notin BI}$				$p_{(8 \times 8 16 \times 16) \notin BI}$			
		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
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
AVG.		0.94				0.93				0.96			

3.2.1.3 Consistency of FW and BW in Small Partitions

We now look into the block partitions smaller than 8x8. We find that we only need to evaluate FW and BW; also, the temporal prediction types of the 8x8 partition are strongly correlated to those of the 4x4 partition. As discussed in Subsection 3.2.1.1, the probabilities using BI for the 8x8 and the smaller partitions are often less than 20% and 10%, respectively. We collect the following conditional probabilities of using FW and BW types. One is defined by

$$p_{(4 \times 4 | 8 \times 8) \in \text{FW}} \triangleq \Pr\{4 \times 4 \in \text{FW} | 8 \times 8 \in \text{FW}\}, \quad (3.4)$$

which is equivalent to $1 - \Pr\{4 \times 4 \in \text{BI} | 8 \times 8 \in \text{FW}\} - \Pr\{4 \times 4 \in \text{BW} | 8 \times 8 \in \text{FW}\}$. Typically, the $\Pr\{4 \times 4 \in \text{BI} | 8 \times 8 \in \text{FW}\}$ term is less than 2% in our collected data. The probability $p_{(4 \times 4 | 8 \times 8) \in \text{FW}}$ can thus be approximated by $\Pr\{4 \times 4 \notin \text{BW} | 8 \times 8 \in \text{FW}\}$. Similarly defined $p_{(4 \times 4 | 8 \times 8) \in \text{BW}}$ is close to $\Pr\{4 \times 4 \notin \text{FW} | 8 \times 8 \in \text{BW}\}$.

Table 3-2 Conditional probabilities of $p_{(4 \times 4 | 8 \times 8) \in \text{FW}}$ and $p_{(4 \times 4 | 8 \times 8) \in \text{BW}}$

Test Sequence	Qp	$p_{(4 \times 4 8 \times 8) \in \text{FW}}$				$p_{(4 \times 4 8 \times 8) \in \text{BW}}$			
		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
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
AVG.		0.90				0.88			

Experiments show that the approximated values of $p_{(4 \times 4|8 \times 8) \in \text{FW}}$ and $p_{(4 \times 4|8 \times 8) \in \text{BW}}$ are fairly close to data in Table 3-2. 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 8x8 prediction mode information serves as a good reference to the smaller partitions.

3.2.2 Rate-Distortion Contribution by BI

In this subsection, we address the relative rate-distortion improvement offered by BI in different block modes. As analyzed before, the hierarchical-B prediction structure takes the advantage of using temporal prediction types to improve the coding efficiency. According to the rate-distortion theory, a temporal prediction type with smaller rate-distortion cost provides better coding efficiency. We adopt the rate-distortion cost function defined produced by JSVM [10] (which came from essentially the rate-distortion theory) and collect all J_{FW} , J_{BW} , and J_{BI} for three squared-shape partitions, 16x16, 8x8, and 4x4. For each sub-block \mathcal{S}_i , we define the *relative rate-distortion improvement* $\omega_{i|\mathcal{T}^*}$, offered by the best temporal prediction type \mathcal{T}^* , as follows:

$$\omega_{i|\mathcal{T}^*} = \left(\min_{\mathcal{T} \in (\Omega \cup \{\text{BI}\}) \setminus \mathcal{T}^*} \{J_{\mathcal{T}}\} \right) - J_{\mathcal{T}^*}, \mathcal{S}_i \in \mathcal{T}^* \quad (3.5)$$

The overall relative rate-distortion improvement \mathcal{W} is the sum of $\omega_{i|\mathcal{T}^*}$ of all sub-blocks; that is, $\mathcal{W} \triangleq \mathcal{W}_{\text{FW}} + \mathcal{W}_{\text{BW}} + \mathcal{W}_{\text{BI}}$, where $\mathcal{W}_{\mathcal{T}} = \sum_{\mathcal{S}_i \in \mathcal{T}} \omega_{i|\mathcal{T}}$. 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

$$\gamma_{N \times N} \triangleq \frac{\mathcal{W}_{\text{BI}, N \times N}}{\mathcal{W}_{N \times N}} = \frac{\mathcal{W}_{\text{BI}, N \times N}}{\mathcal{W}_{\text{FW}, N \times N} + \mathcal{W}_{\text{BW}, N \times N} + \mathcal{W}_{\text{BI}, N \times N}}, \quad (3.6)$$

which trivially yields

$$\frac{\mathbf{E}(\mathcal{W}_{\text{BI}, N \times N})}{\mathbf{E}(\mathcal{W}_{\text{FW}, N \times N}) + \mathbf{E}(\mathcal{W}_{\text{BW}, N \times N}) + \mathbf{E}(\mathcal{W}_{\text{BI}, N \times N})}, \quad (3.7)$$

where $\mathbf{E}(\mathcal{W}_{\mathcal{T}, N \times N}) = \frac{\mathcal{W}_{\mathcal{T}, N \times N}}{n_{N \times N}}$ is the average operator and $n_{N \times N}$ is the number of the $N \times N$

sub-blocks. With the Bayes' theorem, this measure index is rewritten as

$$\gamma_{N \times N} = \frac{\mathbf{E}(\mathcal{W}_{N \times N} | \text{BI}) \Pr\{\text{BI}\}}{\sum_{\mathcal{T} \in \Omega \cup \{\text{BI}\}} \mathbf{E}(\mathcal{W}_{N \times N} | \mathcal{T}) \Pr\{\mathcal{T}\}} = \mathbf{E}(\text{BI} | \mathcal{W}_{N \times N}). \quad (3.8)$$

In other words, the term $\gamma_{N \times N}$, ranging from 0 to 1, indicates the percentage of the relative rate-distortion improvement contributed by BI for the totality of $N \times N$ size blocks. 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.

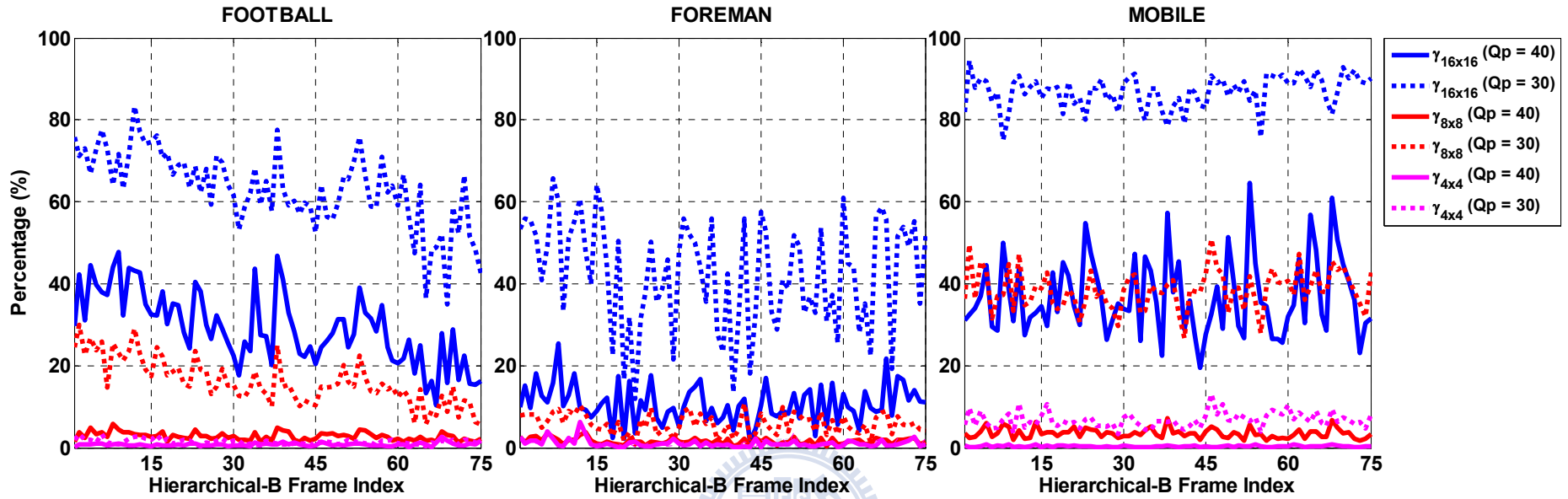


Fig. 3-2 The performance index $\gamma_{N \times N}$ for individual hierarchical-B frame

Table 3-3 Average $\gamma_{N \times N}$ for 16x16, 8x8, and 4x4 blocks in each temporal enhancement layer (in percentage)

Test Sequence	Q_p	$\gamma_{16 \times 16}$				$\gamma_{8 \times 8}$				$\gamma_{4 \times 4}$			
		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
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
AVG.		46.5				12.0				1.9			

Fig. 3-2 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 16x16 partition. On the other hand, the BI type usually furnishes less than 20% in terms of $\gamma_{8 \times 8}$ and $\gamma_{4 \times 4}$, even at high coding rates.

Table 3-3 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 $\gamma_{16 \times 16}$ has an average value of 46.5%; that is, BI plays an important role in improving coding efficiency for large partitions. For the 8x8 partition, the effect of BI plunges to 12.0% on average, which says that the two uni-directional predictions are sufficient to provide good compression. Furthermore, when the partition is getting finer to 4x4, the contribution of BI can be ignored because $\gamma_{4 \times 4}$ is less than 2% typically. However, some test videos such as MOBILE need BI to achieve better rate-distortion performance for both 16x16 and 8x8 partitions, since its $\gamma_{16 \times 16}$ and $\gamma_{8 \times 8}$ values reach 88.6% and 40.0%, respectively. In conclusion, the BI prediction type offers little coding gain for the block partitions smaller than 8x8.

3.2.3 Rate-Distortion Relationships between Uni-directional Predictions and Bi-directional Prediction

In this subsection, we are interested in the relationships between the uni-directional predictions and the BI in motion-rate cost and residual distortion. We collect the following information in our

experiments: (a) the motion vector difference, (b) the motion-rate cost $\mathcal{R}_{\mathcal{T}}$, and (c) the distortion $\mathcal{D}_{\mathcal{T}}$ for the three temporal prediction types. The statistical observations and theoretical analyses on the experimental results are reported below.

3.2.3.1 Motion Vector Difference

In order to find out the correlations of two cost terms $\mathcal{R}_{\mathcal{T}}$ and $\mathcal{D}_{\mathcal{T}}$ between these temporal prediction types, we examine the motion vector differences after the motion vectors are refined by the BI search process. We look at two square block partitions, 16x16 and 8x8. On the JSVM 9.11 platform [10], we search for the best motion vectors of different prediction types for a specified block partition $N \times N$. Their notations are as follows:

$\mathbf{mv}_{\text{FW},N \times N}$: the FW motion vector of $N \times N$ blocks

$\mathbf{mv}_{\text{BW},N \times N}$: the BW motion vector of $N \times N$ blocks

$\mathbf{mv}_{\text{BI} \leftarrow \text{FW},N \times N}$: the forward motion vector refined by BI for $N \times N$ blocks

$\mathbf{mv}_{\text{BI} \leftarrow \text{BW},N \times N}$: the backward motion vector refined by BI for $N \times N$ blocks.

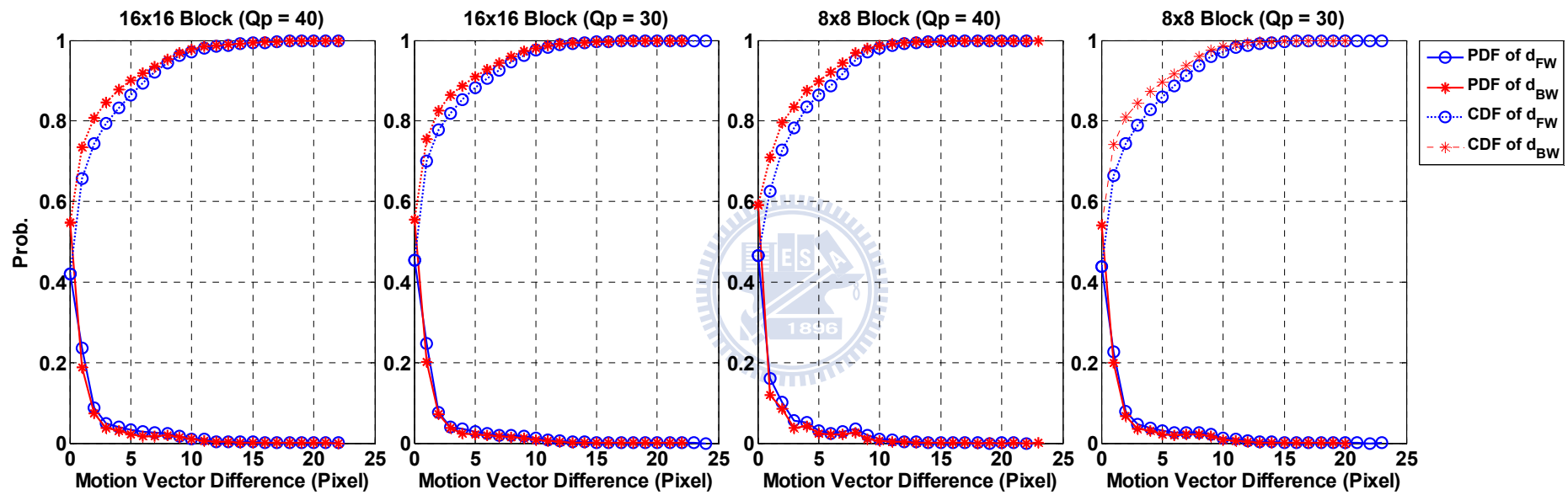
As described earlier, the BI search process takes \mathbf{mv}_{FW} and \mathbf{mv}_{BW} as its initial search points for motion estimation. The Euclidean distance to measure the motion vector difference and

$$d_{\mathcal{T},N \times N} = \|\mathbf{mv}_{\mathcal{T},N \times N} - \mathbf{mv}_{\text{BI} \leftarrow \mathcal{T},N \times N}\|, \mathcal{T} \in \Omega. \quad (3.9)$$

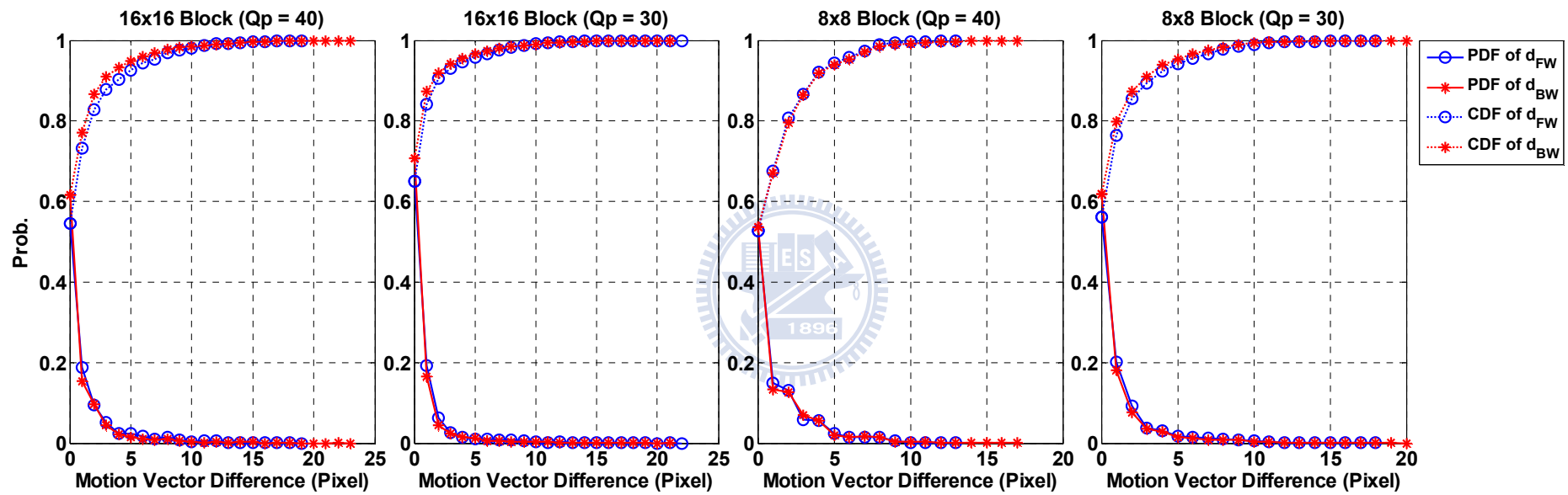
We statistically gather the 16x16 and 8x8 blocks that choose BI as their best temporal prediction type for generating the probability distribution functions (PDF) and cumulative distribution functions

(CDF) of d_{FW} and d_{BW} , as shown in Fig. 3-3.

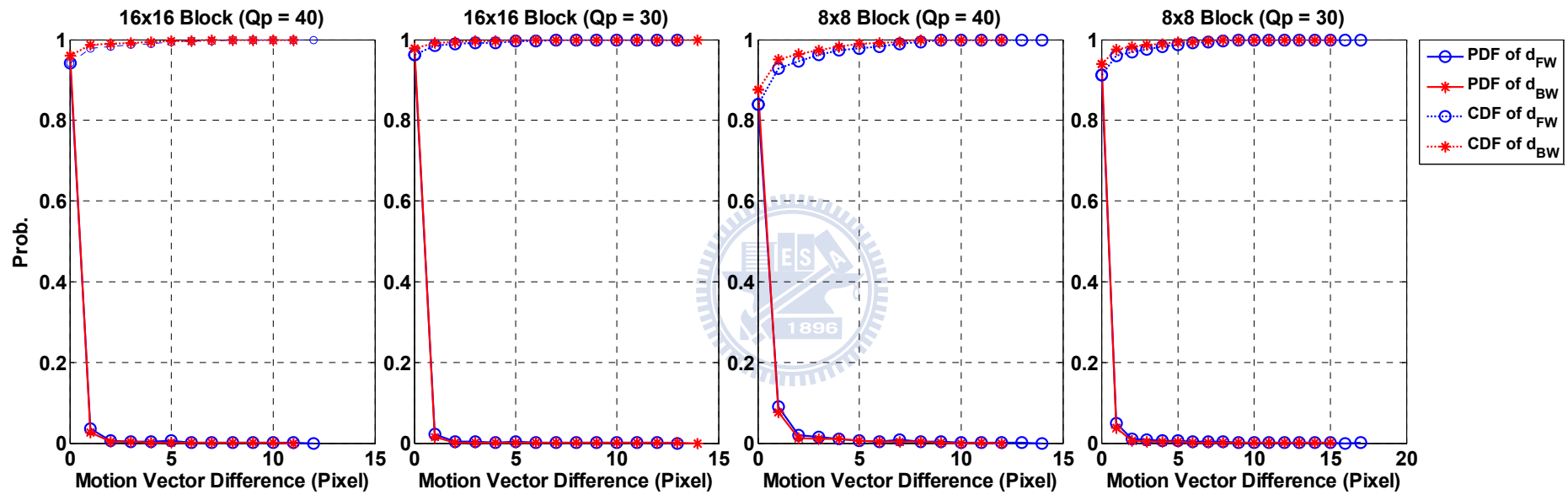




(a) FOOTBALL

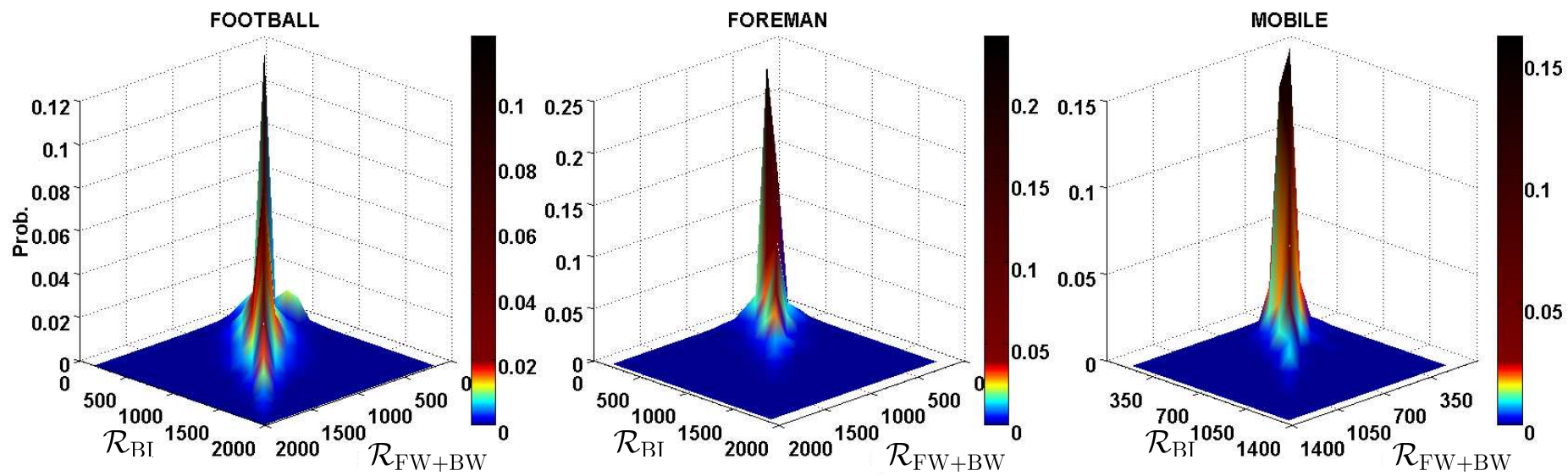


(b) FOREMAN

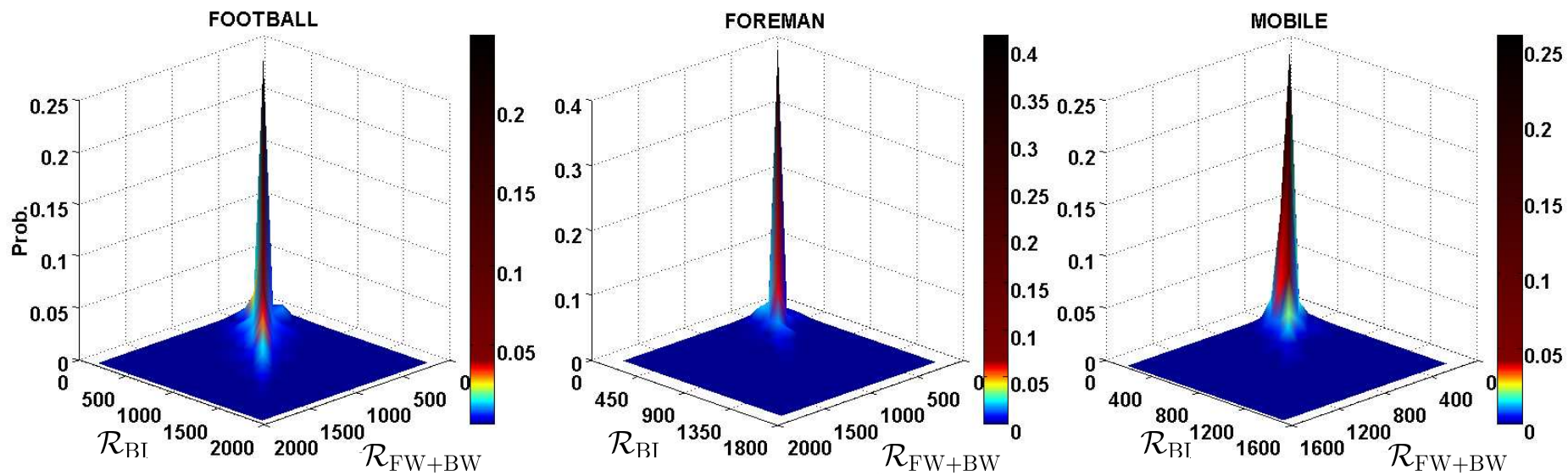


(c) MOBILE

Fig. 3-3 PDFs and CDFs of the motion vector difference $d_{\mathcal{T}}$ for 16x16 and 8x8 blocks with two selected Q_p values



(a) 16x16 partition size with $Qp = 40$



(b) 8x8 partition size with $Qp = 40$

Fig. 3-4 Distributions of motion-rate costs \mathcal{R}_{BI} and \mathcal{R}_{FW+BW}

As shown, the PDFs of the motion vector difference are strongly clustered around the starting search points. Particularly, the one-pixel probability $\Pr\{d_{\mathcal{T}} < 1 \text{ (pixel)}\}$ 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}_{FW} and \mathbf{mv}_{BW} . Typically, the MV differences are within three pixels; the CDFs of motion vector differences less than three pixels usually reach 80% or more. Our experiments show that different block partitions have similar probability distributions. The similarity in the two motion pairs $(\mathbf{mv}_{FW}, \mathbf{mv}_{BI \leftarrow FW})$ and $(\mathbf{mv}_{BW}, \mathbf{mv}_{BI \leftarrow BW})$ is the foundation of the following analysis.

3.2.3.2 Motion-Rate Cost

Our second study tries to identify the correlation of the motion-rate costs $\mathcal{R}_{\mathcal{T}}$ between uni-directional predictions and BI. As discussed in Subsection 3.2.3.1, the motion vectors $\mathbf{mv}_{BI \leftarrow FW}$ and $\mathbf{mv}_{BI \leftarrow BW}$ produced by BI are often close to \mathbf{mv}_{FW} and \mathbf{mv}_{BW} , respectively. In addition, the motion-rate cost is proportional to the motion vector length. Based on those two observations, we anticipate that there exists a strong correlation in motion-rate cost between uni-directional predictions and BI. More specifically, since the BI operation needs two motion vectors to fetch two reference blocks for prediction, we collect the motion-rate costs of three temporal prediction types, \mathcal{R}_{FW} , \mathcal{R}_{BW} , and $\mathcal{R}_{BI} \triangleq (\mathcal{R}_{BI \leftarrow FW} + \mathcal{R}_{BI \leftarrow BW})$ to find out the relationship between \mathcal{R}_{BI} and the combined cost of two uni-directional predictions, denoted as $\mathcal{R}_{FW+BW} \triangleq (\mathcal{R}_{FW} + \mathcal{R}_{BW})$.

As depicted in Fig. 3-4, the distributions of \mathcal{R}_{FW+BW} and \mathcal{R}_{BI} are noticeably concentrated along a straight line; this high correlation is foreseen because $\mathbf{mv}_{FW} \approx \mathbf{mv}_{BI \leftarrow FW}$

and $\mathbf{mv}_{\text{BW}} \approx \mathbf{mv}_{\text{BI} \leftarrow \text{BW}}$, as discussed earlier. This implies that $\mathcal{R}_{\text{FW}} \approx \mathcal{R}_{\text{BI} \leftarrow \text{FW}}$ and $\mathcal{R}_{\text{BW}} \approx \mathcal{R}_{\text{BI} \leftarrow \text{BW}}$. Therefore, we make use of the first-order regression to represent the motion-rate cost \mathcal{R}_{BI} by $\mathcal{R}_{\text{FW}+\text{BW}}$. Here, the motion-rate cost \mathcal{R}_{BI} is modeled as an affine function of $\mathcal{R}_{\text{FW}+\text{BW}}$. This regression for \mathcal{R}_{BI} based on $\mathcal{R}_{\text{FW}+\text{BW}}$ is thus formulated as

$$\widehat{\mathcal{R}}_{\text{BI}} = \alpha + \beta \cdot \mathcal{R}_{\text{FW}+\text{BW}}, \quad (3.10)$$

where α and β are the regression parameters. Furthermore, we assume that $\alpha \approx 0$ because the motion-rate cost r_{BI} should be nearly to zero because $\mathcal{R}_{\text{FW}} \approx 0$ when $\|\mathbf{mv}_{\text{FW}}\| \approx 0$ and $\mathcal{R}_{\text{BW}} \approx 0$ when $\|\mathbf{mv}_{\text{BW}}\| \approx 0$. The first-order model is thus simplified to a linear function,

$$\widehat{\mathcal{R}}_{\text{BI}} = \beta \cdot \mathcal{R}_{\text{FW}+\text{BW}}. \quad (3.11)$$

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

$$\beta^* = \frac{\mathbf{E}(\mathcal{R}_{\text{BI}} \mathcal{R}_{\text{FW}+\text{BW}})}{\mathbf{E}(\mathcal{R}_{\text{FW}+\text{BW}}^2)}. \quad (3.12)$$

Table 3-4 Optimal β^* value for the linear regression model (3.11)

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
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		0.95	0.95	0.94	0.92	0.96	0.93	0.92	0.90
AVG.		0.93							

As tabulated in Table 3-4, the optimal β^* for 16x16 and 8x8 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, $\widehat{\mathcal{R}}_{\text{BI}} = \overline{\beta^*} \cdot \mathcal{R}_{\text{FW+BW}}$, is a rather good approximation to the real \mathcal{R}_{BI} because the percentage error $\mathbf{E} \left(\left| \frac{\widehat{\mathcal{R}}_{\text{BI}} - \mathcal{R}_{\text{BI}}}{\mathcal{R}_{\text{BI}}} \right| \right)$ is around 11%.

3.2.3.3 Distortion Relationship

Our last study addresses on the correlations in the distortions of different prediction types. We obtain an upper bound of \mathcal{D}_{BI} and give a summary of the approximated distribution of \mathcal{D}_{BI} (denoted as $\widetilde{\mathcal{D}}_{\text{BI}}$).

For a given $M \times N$ partition mode, the SAD metric used to evaluate its distortions of uni-directional predictions is defined by Eq. (2.20), unfolded as

$$\mathcal{D}_{\mathcal{T}}(\mathbf{mv}_{\mathcal{T}}) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |f(x, y) - f'_{\mathcal{T}}(x - mv_{\mathcal{T}}^x, y - mv_{\mathcal{T}}^y)| = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |e_{\mathcal{T}}(x, y)|, \mathcal{T} \in \Omega. \quad (3.13)$$

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 $(mv_{\mathcal{T}}^x, mv_{\mathcal{T}}^y)$ 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, $f(x, y) - f'_{\mathcal{T}}(x - mv_{\mathcal{T}}^x, y - mv_{\mathcal{T}}^y)$. In the BI case with equal weighted prediction, its distortion value is defined by

$$\mathcal{D}_{\text{BI}} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left| f(x, y) - \frac{1}{2} \sum_{\mathcal{T} \in \Omega} f'_{\mathcal{T}}(x - mv_{\text{BI}|\mathcal{T}}^x, y - mv_{\text{BI}|\mathcal{T}}^y) \right|. \quad (3.14)$$

Because the motion vectors finally adopted by BI are close to those produced by the uni-directional predictions, the value of \mathcal{D}_{BI} may be approximated by $\widetilde{\mathcal{D}}_{\text{BI}}$, namely,

$$\mathcal{D}_{\text{BI}} \approx \tilde{\mathcal{D}}_{\text{BI}} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left| f(x, y) - \frac{1}{2} \sum_{T \in \Omega} f'_T(x - mv_T^x, y - mv_T^y) \right|. \quad (3.15)$$

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

$$\begin{aligned} \tilde{\mathcal{D}}_{\text{BI}} &= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left| f(x, y) - \frac{1}{2} \sum_{T \in \Omega} f'_T(x - mv_T^x, y - mv_T^y) \right| \\ &= \frac{1}{2} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left| [f(x, y) - f'_{\text{FW}}(x - mv_{\text{FW}}^x, y - mv_{\text{FW}}^y)] + [f(x, y) - f'_{\text{BW}}(x - mv_{\text{BW}}^x, y - mv_{\text{BW}}^y)] \right| \\ &= \frac{1}{2} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |e_{\text{FW}}(x, y) + e_{\text{BW}}(x, y)| \leq \frac{1}{2} \sum_{T \in \Omega} \left(\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |e_T(x, y)| \right) \\ &= \frac{1}{2} (\mathcal{D}_{\text{FW}} + \mathcal{D}_{\text{BW}}) \triangleq \mathcal{D}_{\text{FW+BW}} \end{aligned}$$

(3.16)

In addition to derive an upper bound of $\tilde{\mathcal{D}}_{\text{BI}}$, we verify the probability distribution of $\tilde{\mathcal{D}}_{\text{BI}}$ by two goodness-of-fits tests, as described in the Appendix. The two statistical tests indicate that the pair of $(e_{\text{FW}}, e_{\text{BW}})$ tends to be bivariate-Laplacian distributed. This distribution model is used to construct a set of adaptive thresholds in Section 3.3.

Section 3.3 Proposed Schemes – Temporal Prediction Inheritance with Adaptive Thresholds for Bi-directional Prediction Selection

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 3.2. We deduce a set of

adaptive thresholds that efficiently eliminate unnecessary BI evaluations in Subsection 3.3.1. Then, with the adaptive thresholds, our proposed schemes are detailed in Subsection 3.3.2.

3.3.1 Adaptive Thresholds

The highly correlated motion-rate costs and distortions between the uni-directional predictions and the BI are used to develop a set of thresholds for block partitions from 16x16 to 8x8. According to the information from FW and BW, we can separately build a BI motion-rate cost estimator and a BI distortion estimator for a specified $M \times N$ block.

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

$$\hat{\mathcal{R}}_{\text{BI},M \times N} = \bar{\beta}^* \cdot \mathcal{R}_{\text{FW+BW},M \times N}, \quad (3.17)$$

where $\bar{\beta}^* = 0.93$.

Next, we try to estimate \mathcal{D}_{BI} from the probability model of $\tilde{\mathcal{D}}_{\text{BI}}$ and the percentage of the exception case of ($\mathcal{D}_{\text{BI}} > \mathcal{D}_{\text{FW+BW}}$). Occasionally, $\mathcal{D}_{\text{FW+BW}}$ is not an upper bound of \mathcal{D}_{BI} if the chosen BI is inferior in terms of distortion. (Note that the motion vector chosen by the BI has the better combined rate-distortion value, not based only on the distortion value.) From our collected

data, the exception ($\mathcal{D}_{\text{BI}} > \mathcal{D}_{\text{FW+BW}}$) is 1%~5% on the average for different block partitions.

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

$$\hat{\mathcal{D}}_{\text{BI}} = k\theta. \quad (3.18)$$

As discussed earlier, $\mathcal{D}_{\text{BI}} \approx \tilde{\mathcal{D}}_{\text{BI}}$ (Normally, $\mathcal{D}_{\text{BI}} < \tilde{\mathcal{D}}_{\text{BI}}$ indicates the motion vector refinement in BI is effective.) and $\Pr\{\Gamma(k, \theta) = \theta\Gamma(k, 1) > \mathcal{D}_{\text{FW+BW}}\} = 1\% \sim 5\%$ for different block sizes. The probability $\Pr\{\Gamma(k, \theta) = \theta\Gamma(k, 1) > \hat{\mathcal{D}}_{\text{BI}}\}$ can be calculated for a fixed k . Therefore, we can determine the relationship between $\mathcal{D}_{\text{FW+BW}}$ and $\hat{\mathcal{D}}_{\text{BI}}$ without any knowledge of θ , as listed in Table 3-5. For example, the probability $\Pr\{\mathcal{D}_{\text{BI},16 \times 16} > \mathcal{D}_{\text{FW+BW},16 \times 16}\}$ is 3.6% from our experimental data.

$$\begin{aligned} 0.036 &= \Pr\{\mathcal{D}_{\text{BI},16 \times 16} > \mathcal{D}_{\text{FW+BW},16 \times 16}\} \\ &\approx \Pr\{\hat{\mathcal{D}}_{\text{BI},16 \times 16} > \mathcal{D}_{\text{FW+BW},16 \times 16}\} \\ &= \Pr\{\Gamma(k, \theta) > \mathcal{D}_{\text{FW+BW},16 \times 16}\} \\ &= \Pr\{\theta\Gamma(k, 1) > \mathcal{D}_{\text{FW+BW},16 \times 16}\} \end{aligned} \quad (3.19)$$

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

$$\frac{\hat{\mathcal{D}}_{\text{BI},16 \times 16}}{\mathcal{D}_{\text{FW+BW},16 \times 16}} = \frac{k}{\text{CDF}_{\Gamma(k,1)}^{-1}(0.964)} \stackrel{k:=256}{=} \frac{256}{285.5196} = 0.90. \quad (3.20)$$

Finally, using the ξ definition in Table 3-5, an estimate of \mathcal{D}_{BI} (for the $M \times N$ partition mode) is represented by

$$\hat{\mathcal{D}}_{\text{BI},M \times N} = \xi_{M \times N} \cdot \mathcal{D}_{\text{FW+BW},M \times N}. \quad (3.21)$$

Table 3-5 k value in $\Gamma(k, \theta)$ and ξ value for derivation of $\hat{\mathcal{D}}_{\text{BI}}$

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

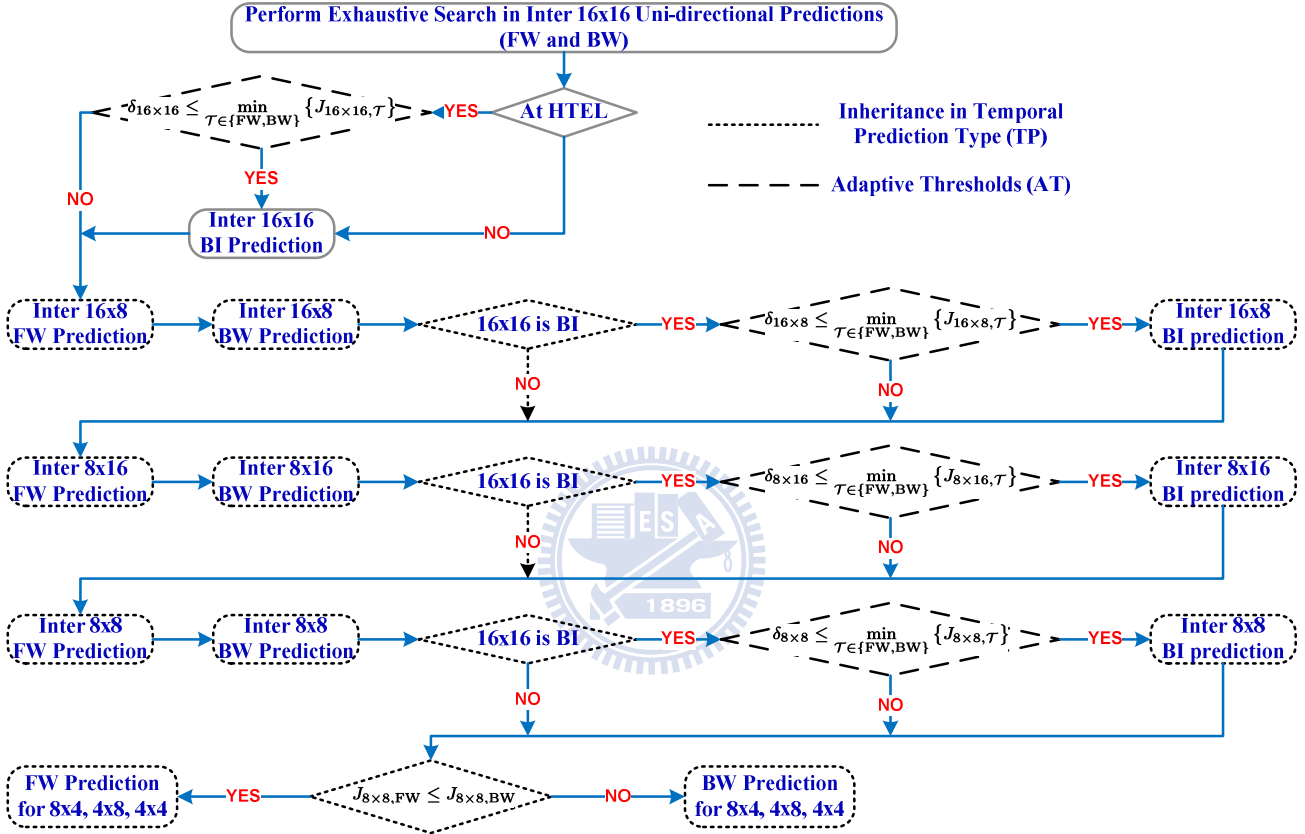


Fig. 3-5 Fast selection algorithm for temporal prediction types

3.3.2 Algorithm Overview

The flowchart in Fig. 3-5 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, detailed below.

Stage 1: Conditionally Exhaustive Temporal Prediction Type Search for Inter16x16. The Inter16x16 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 macroblock, the uni-directional temporal predictions FW and BW are evaluated in order to collect their distortions (\mathcal{D}_{FW} and \mathcal{D}_{BW}) and motion-rate costs (\mathcal{R}_{FW} and \mathcal{R}_{BW}) that will be used in Step 1.3. If the macroblock 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 $\delta_{16 \times 16}$ can be obtained by

$$\begin{aligned}
 \delta_{16 \times 16} &= \widehat{\mathcal{D}}_{\text{BI},16 \times 16} + \widehat{\mathcal{R}}_{\text{BI},16 \times 16} \\
 &= \xi_{16 \times 16} \cdot \mathcal{D}_{\text{FW}+\text{BW},16 \times 16} + \overline{\beta}^* \cdot \mathcal{R}_{\text{FW}+\text{BW},16 \times 16} \\
 &= 0.9 \left(\frac{\mathcal{D}_{\text{FW},16 \times 16} + \mathcal{D}_{\text{BW},16 \times 16}}{2} \right) + 0.93(\mathcal{R}_{\text{FW},16 \times 16} + \mathcal{R}_{\text{BW},16 \times 16})
 \end{aligned} \tag{3.22}$$

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

Stage 2: Early Termination on BI for Large Partitions. Before getting into the Stage 2, the best

temporal prediction type $\mathcal{T}_{16 \times 16}^*$ is determined in Stage 1. Two steps of this stage predict whether or not the BI type in $16 \times 8 / 8 \times 16 / 8 \times 8$ partitions have an inferior rate-distortion performance and thus can be excluded from testing. Subsection 3.3.2.1 details the early termination procedure.

Step 2.1: Exhaustive Uni-directional Prediction Type Search for Partitions $16 \times 8 / 8 \times 16 / 8 \times 8$.

To gather the distortions (\mathcal{D}_{FW} and \mathcal{D}_{BW}) and motion-rate costs (\mathcal{R}_{FW} and \mathcal{R}_{BW}) of partitions $16 \times 8 / 8 \times 16 / 8 \times 8$, the uni-directional temporal predictions FW and BW are calculated.

Go to Step 2.2.

Step 2.2: Pre-decided BI Elimination for Partitions $16 \times 8 / 8 \times 16 / 8 \times 8$. If the pre-determined

information $\mathcal{T}_{16 \times 16}^*$ 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

$$\begin{aligned} \delta_{M \times N} &= \widehat{\mathcal{D}}_{\text{BI}, M \times N} + \widehat{\mathcal{R}}_{\text{BI}, M \times N} \\ &= \xi_{M \times N} \cdot \mathcal{D}_{\text{FW}+\text{BW}, M \times N} + \overline{\beta}^* \cdot \mathcal{R}_{\text{FW}+\text{BW}, M \times N} \end{aligned} \quad (3.23)$$

where $M \times N \in \{16 \times 8, 8 \times 16, 8 \times 8\}$. The BI type is calculated if the specifications

$\delta_{M \times N} < J_{\text{FW}, M \times N}$ and $\delta_{M \times N} < J_{\text{BW}, M \times N}$ 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 Subsection 3.3.2.2 elaborates our implementation.

3.3.2.1 Early Termination on BI for Large Partitions

Based on our observations discussed earlier, the prediction type information of Inter16x16 partition mode is useful in skipping the BI type for large block partitions. More precisely, the conditional probabilities, $p_{(16 \times 8 | 16 \times 16) \notin \text{BI}}$, $p_{(8 \times 16 | 16 \times 16) \notin \text{BI}}$, and $p_{(8 \times 8 | 16 \times 16) \notin \text{BI}}$, can suggest whether the unnecessary computations of BI of 16x8/8x16/8x8 partitions can be avoided if the 16x16 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 16x16 partition. Furthermore, for the case that $\mathcal{T}_{16 \times 16}^*$ is BI, the remaining superfluous BI can be detected by the $\delta_{M \times N}$ thresholds in Step 2.3.

3.3.2.2 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 the improved compression efficiency comparing to FW and BW, and (c) small partitions often have the same prediction types as that of their inherited 8x8 parent block. These three observations help us in developing an adaptive prediction selection algorithm for small partitions.

The 8x8 or smaller blocks are seldom coded with BI. As discussed earlier, the prediction type of a smaller partition can be reliably estimated by its 8x8 parent partition. Thus, each smaller partition refined from an 8x8 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 8x8 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.

Table 3-6 Testing conditions

Sequence	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), MOBICAL (MC), SHIELDS (SH), STOCKHOLM (ST)
Qp_i	$Qp_1 = 40$, $Qp_2 = 36$, $Qp_3 = 32$, and $Qp_4 = 28$	
Encoder Configuration	Motion search range: ± 32 pixels with $\frac{1}{4}$ -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	

Section 3.4 Experimental Results and Discussions



3.4.1 Test Conditions

For performance assessment, we have implemented the proposed algorithms in JSVM 9.11 [10] and have tested 15 typical video sequences in three resolutions (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 frame prediction structure. The detailed encoder parameters are given in Table 3-6 and the other parameters are the default values set by the reference software JSVM 9.11 [10].

3.4.2 Performance Measures

To show the change in rate-distortion performance, we adopt the Bjontegaard metric [76] which needs four rate-distortion points to measure the averaged Y-PSNR [BDP (dB)] and bit-rate differences [BDR (%)] between the two rate-distortion curves produced by JSVM 9.11 [10] and by our schemes, respectively. Moreover, we use $\Delta\text{FileSize}$ (%) to show the total file size increase (in percentage) at these four rate-distortion points, defined by

$$\Delta\text{FileSize} = \frac{1}{4} \sum_{i=1}^4 \frac{\text{FileSize}_{\text{Proposed}}(Qp_i) - \text{FileSize}_{\text{JSVM9.11}}(Qp_i)}{\text{FileSize}_{\text{JSVM9.11}}(Qp_i)} \times 100\%, \quad (3.24)$$

where the $\text{FileSize}_{\text{JSVM9.11}}(Qp_i)$ and $\text{FileSize}_{\text{Proposed}}(Qp_i)$ are, respectively, the number of bits coded by JSVM 9.11 [10] and by our schemes with $Qp = Qp_i$.

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

(1) The overall time saving TS is defined as

$$\text{TS} = \frac{1}{4} \sum_{i=1}^4 \frac{T_{\text{JSVM9.11}}(Qp_i) - T_{\text{Proposed}}(Qp_i)}{T_{\text{JSVM9.11}}(Qp_i)} \times 100\%, \quad (3.25)$$

where $T_{\text{JSVM9.11}}(Qp_i)$ and $T_{\text{Proposed}}(Qp_i)$ denotes the encoding time of JSVM 9.11 [10] and that of our schemes by setting $Qp = Qp_i$, respectively.

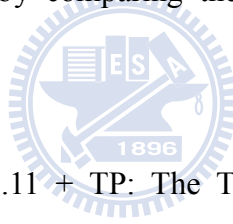
(2) In this $\text{TS}_{\text{HBvsIP}}$ measure, the denominator is the additional computing time due to the use the hierarchical-B frames. The numerator is the additional computing time of using our fast algorithms. That is,

$$TS_{HBvsIP} = \frac{1}{4} \sum_{i=1}^4 \frac{T_{Proposed}(Qp_i) - T_{JSVM9.11 \text{ with IPPP}}(Qp_i)}{T_{JSVM9.11}(Qp_i) - T_{JSVM9.11 \text{ with IPPP}}(Qp_i)} \times 100\%, \quad (3.26)$$

where $T_{JSVM9.11 \text{ with IPPP}}(Qp_i)$ represents the encoding time of JSVM 9.11 [10] with the IPPP coding structure and $Qp = Qp_i$.

3.4.3 Performance Comparison with JSVM

Table 3-7 to Table 3-9 present the time savings of the proposed schemes in comparison with JSVM 9.11 [10]. Listed in Table 3-7 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 vs. JSVM 9.11 + TP: The TP setting makes use of the information produced by the 16x16 partition size to skip the BI type in 16x8/8x16/8x8 partitions. For the block sizes smaller than 8x8, they only evaluate one of the uni-directional predictions, depending on the encoding information of 8x8.

Setting #2: JSVM 9.11 vs. 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 16x6 to 8x8 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 2x, whereas the AT offers only a moderate time saving of 20%~28%.

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.

Table 3-7 Individual time saving contributed by TP and AT

Test Sequence	GOP = 8				GOP = 16			
	TP		AT		TP		AT	
	TS (%)	TS _{HBvsIP} (%)	TS (%)	TS _{HBvsIP} (%)	TS (%)	TS _{HBvsIP} (%)	TS (%)	TS _{HBvsIP} (%)
AK	55.2	20.1	27.6	60.1	55.8	21.3	28.4	59.9
BU	46.1	34.8	23.2	67.2	46.3	36.2	23.5	67.7
FO	43.8	38.2	23.9	66.3	43.9	39.6	24.1	66.9
FM	50.6	28.0	27.6	60.7	51.0	29.4	28.1	61.2
MO	44.1	36.4	21.8	68.6	44.5	37.5	22.3	68.7
SF	46.5	33.6	23.3	66.8	47.0	34.8	23.5	67.4
CT	51.3	25.9	24.2	65.1	51.7	27.3	25.0	64.8
CR	49.3	28.7	26.7	61.4	49.9	29.9	27.1	61.8
HA	46.1	33.6	24.6	64.6	46.6	34.7	24.8	65.3
IC	53.5	23.1	27.6	60.2	54.1	24.2	28.1	60.6
SC	50.0	28.7	23.9	65.8	50.4	30.1	24.7	65.8
BS	52.7	23.5	25.0	63.7	53.3	24.7	25.7	63.7
MC	52.0	23.5	20.5	69.8	52.6	24.8	21.9	68.8
SH	52.1	24.1	21.8	68.3	52.5	25.6	22.6	68.0
ST	52.9	22.9	20.0	70.8	53.5	24.1	20.6	70.8
AVG.	49.7	28.3	24.1	65.3	50.2	29.6	24.7	65.4

To see their combined effects, Table 3-8 and Table 3-9 provide 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 63% for the overall encoding time is achieved. In other words, we can observe an approximated 3x

speedup. The improvement is achieved with a negligible change in both bit-rate and Y-PSNR, as confirmed by the BDP/BDR values in the tables and the rate-distortion curves in Fig. 3-6 and Fig. 3-7. As discussed before, the BI examination in the encoding time is about $\frac{1.22}{1+1.22} = 55\%$. That is, the improvement in TS 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 may calculate only one of the uni-directional temporal predictions. Furthermore, the additional computation required by the hierarchical-B prediction structure can be reduced up to 95% and the average TS_{HBvsIP} is around 10%.

Table 3-8 Performance comparisons with JSVM 9.11 [10] when GOP size is 8

Test Sequence	BDP (dB)	BDR (%)	Δ FileSize (%)	TS (%)	TS_{HBvsIP} (%)
AK	-0.02	0.42	-0.06	65.5	5.1
BU	-0.18	3.61	1.44	61.2	13.3
FO	-0.19	3.86	1.16	59.4	16.2
FM	-0.09	2.17	0.48	64.2	8.6
MO	-0.11	2.84	0.83	61.4	11.3
SF	-0.20	4.44	2.41	61.7	11.9
CT	-0.03	0.92	0.14	63.0	9.0
CR	-0.07	2.56	0.52	62.6	9.4
HA	-0.06	1.72	0.26	62.6	9.8
IC	-0.09	2.34	0.85	64.5	7.2
SC	-0.07	1.74	0.52	62.2	11.4
BS	-0.01	0.42	-0.03	64.0	7.2
MC	-0.01	0.32	-0.10	62.0	8.8
SH	-0.01	0.51	0.10	61.8	10.0
ST	-0.01	0.45	0.00	61.7	10.1
AVG.	-0.08	1.89	0.57	62.5	10.0

Table 3-9 Performance comparisons with JSVM 9.11 [10] when GOP size is 16

Test Sequence	BDP (dB)	BDR (%)	Δ FileSize (%)	TS (%)	TS _{HBvsIP} (%)
AK	-0.02	0.56	0.01	66.2	6.6
BU	-0.19	3.93	1.58	61.5	15.4
FO	-0.20	4.05	1.22	60.0	17.6
FM	-0.11	2.57	0.61	64.9	10.3
MO	-0.13	3.67	1.27	62.0	12.8
SF	-0.22	4.96	2.62	62.0	13.8
CT	-0.03	0.98	0.01	63.9	10.2
CR	-0.08	2.92	0.65	63.2	11.1
HA	-0.07	2.09	0.37	63.3	11.4
IC	-0.11	2.78	1.01	65.2	8.7
SC	-0.08	2.18	0.78	62.9	12.8
BS	-0.01	0.50	-0.02	64.9	8.3
MC	-0.01	0.30	-0.09	63.1	9.7
SH	-0.02	0.79	0.23	62.8	11.1
ST	-0.01	0.63	0.04	62.6	11.2
AVG.	-0.09	2.19	0.69	63.2	11.4

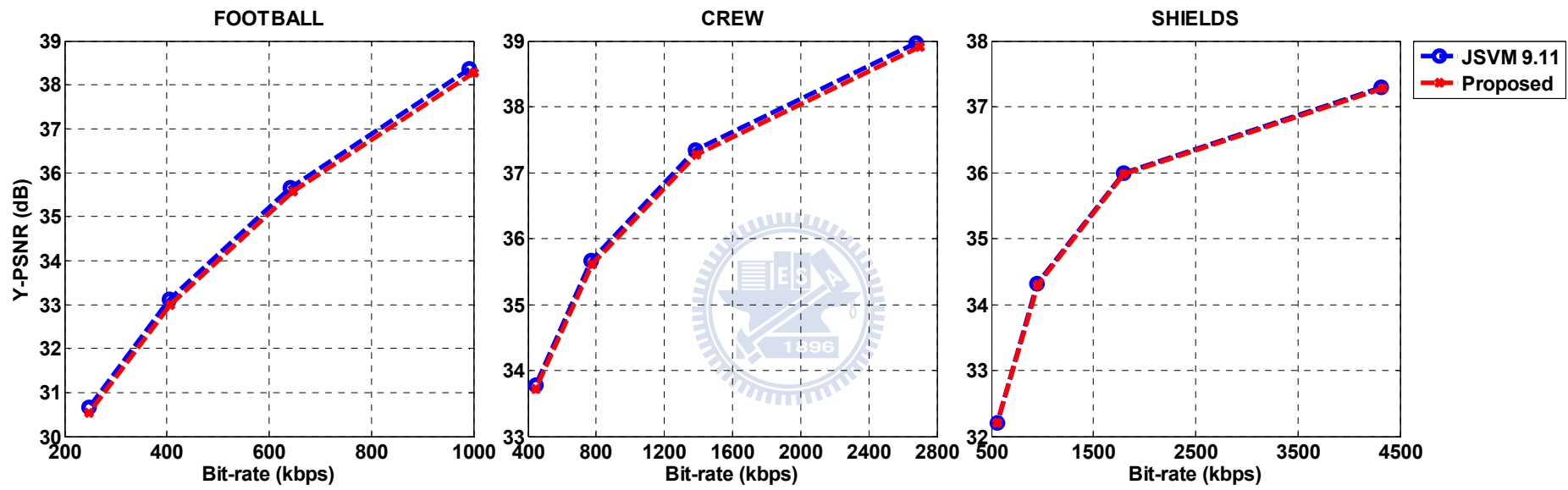


Fig. 3-6 Comparisons in rate-distortion curve with GOP = 8

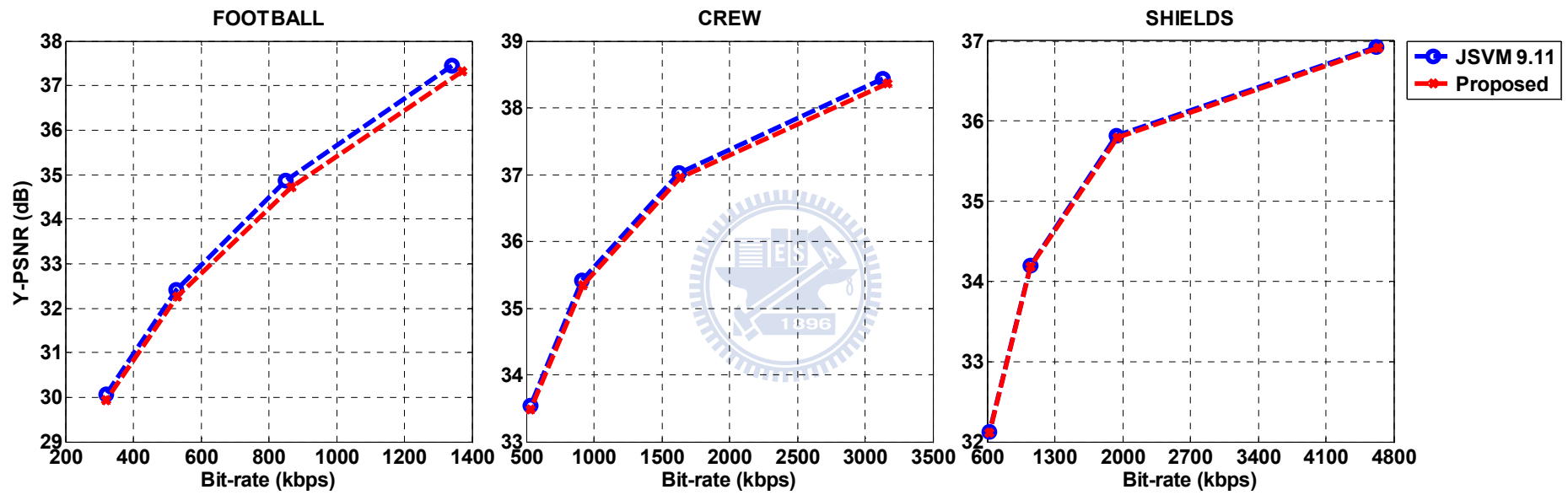


Fig. 3-7 Comparisons in rate-distortion curve with GOP = 16

Interestingly, the overall time savings in Table 3-8 and Table 3-9 are not the sum of the results from Table 3-7. In our approach, we set two successive criterions to conditionally eliminate the BI computation, as illustrated in Fig. 3-5; one is TP, and the other is AT. However, their contributions are overlapped for some macroblocks; that is, if these macroblocks satisfy the first TP criterion, the second 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 13% improvement. The additional improvement comes from the cases when the 16x16 partition is not BI, and the AT condition is satisfied. For example, in Fig. 3-1(a), about 70% of MBs do not select BI at T_1 (with $Qp = 40$) when 16x16 partition is examined. In this case, our algorithm skips these 70% of BI calculations. In total, there are 85% of 16x8 partitions do not prefer BI in our collected data. Therefore, only the remaining 15% of 16x8 partition blocks are further checked by the AT criterion for further complexity reduction.

Moreover, in Table 3-10, the overall time saving decreases as the Qp value becomes small. The complexity reduction goes down from 66% to 57% as Qp decreases. This is due to the combined rate-distortion cost $J_T = \mathcal{D}_T + \lambda_{\text{MOTION}} \times \mathcal{R}_T$ that affects the selection of temporal prediction type. Such an optimization principle tends to minimize the motion-rate term \mathcal{R}_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}_T at a small Qp . Thus, BI is used more often for small Qp values because BI is effective in minimizing distortion. Hence, fewer BI blocks can be skipped by our

approach.

Table 3-10 Overall time saving with various Qp values

Test Sequence	TS (%) with GOP = 8				TS (%) with GOP = 16			
	Qp_1	Qp_2	Qp_3	Qp_4	Qp_1	Qp_2	Qp_3	Qp_4
AK	66.0	65.9	65.6	64.5	66.8	66.6	66.3	65.3
BU	62.9	61.8	60.8	59.5	63.2	62.1	61.0	59.6
FO	60.5	59.8	59.1	58.3	61.1	60.3	59.6	59.0
FM	64.8	64.7	64.2	63.3	65.2	65.2	64.9	64.1
MO	65.7	63.0	60.0	57.0	66.5	63.7	60.4	57.4
SF	63.8	62.6	61.1	59.1	64.4	63.1	61.2	59.5
CT	64.9	64.3	62.9	59.9	65.8	65.4	63.8	60.6
CR	63.6	63.1	62.5	61.2	64.2	63.7	63.1	62.0
HA	65.2	63.9	62.1	59.4	65.7	64.6	62.8	60.1
IC	65.3	65.0	64.8	63.1	65.8	65.7	65.3	64.0
SC	63.4	62.7	62.1	60.5	64.0	63.5	62.7	61.3
BS	65.6	64.9	63.8	61.5	66.5	66.0	64.7	62.2
MC	64.8	63.5	61.5	58.3	66.0	64.4	62.6	59.3
SH	63.8	62.7	61.4	59.0	65.0	64.0	62.2	59.9
ST	64.2	62.8	60.9	58.9	65.4	63.8	61.6	59.3
AVG.	64.3	63.4	62.2	60.2	65.0	64.1	62.8	60.9

3.4.4 Performance Comparison with State-of-the-art Fast Algorithms

In addition to the exhaustive search, we also compare our approaches with the state-of-the-art fast algorithms, Li's method [72] and Lee's method [73]; both reduce the massive computations only on the mode decision. For a fair comparison, the same number of reference frame (one reference frame in each reference list) is configured in the experiment.

As reported in [72], the Li's method can averagely achieve 38% time reduction and has Y-PSNR loss of 0.14 dB and 2.15% bit-rate increase. The Lee's method [73] provides a higher time saving,

which speeds up the encoding process at about two times on average. However, the complexity reduction resulted from the Lee's method [73] is sequence-dependent, widely ranging from 21% to 69%.

In comparison to these two schemes [72][73], our approaches, at a similar level of quality loss in terms of rate-distortion performance, averagely achieve up to 65% time saving. In addition, our algorithm has a nearly constant complexity reduction rate with less than 10% variation for 15 different sequences.



Chapter 4

Fast Mode Decision Algorithm for Intra-only Scalable Video Coding with Combined Coarse Granular Scalability (CGS) and Spatial Scalability

In this chapter, we propose a fast mode decision algorithm with macroblock-adaptive rate-distortion estimation for intra-only scalable video coding. We make use of the log-linear rate-distortion relationship of inter-dependent layers to predict the better performer among the Intra4x4 and Intra8x8 prediction types at the enhancement layers. Based upon the base-layer chosen prediction type, we can further reduce the number of candidate modes. In addition, to ensure the best trade-off between complexity and coding efficiency, the Intra16x16 prediction is retained and enabled only for coding high-resolution videos with smooth image contents. Comparing to the Joint Scalable Video Model v.8 (JSVM 8) [77], an encoder time saving from 49% to 64% depending on the encoder configurations, is achieved with negligible penalty in coding efficiency.

This chapter is organized as follows. The prior works related to the fast intra mode decision is presented in Section 4.1. Section 4.2 analyzes the statistics of intra prediction modes at different coding layers. Based on this analysis, Section 4.3 presents our rate-distortion estimation model and the layer-adaptive mode selection algorithm. Section 4.4 compares the proposed schemes with JSVM 8 [77] in terms of encoding latency and rate-distortion performance.

Section 4.1 Literature Review

Several approaches have thus been proposed to simplify the mode decision process. In [78] and [79], the correlations of the intra prediction modes between coding layers are considered for fast mode decision. Both schemes suggest skipping the Intra16x16 and Intra8x8 prediction types as they offer little coding gain at the enhancement layers. In [78], the prediction is further restricted to use only *Vertical*, *Horizontal*, and *DC* modes. A similar approach is also adopted by [80] and [81] to search for the best intra prediction mode in inter-frame coding. By using a reduced-size mode set, these schemes speedup the encoding process.

However, we find that excluding certain prediction types may result in poor rate-distortion performance, especially for coding high-resolution videos with smooth image contents. To relieve this coding loss, we propose a layer-adaptive intra mode/type selection algorithm. Different from the previous approaches, our scheme retains all the intra prediction types/modes and makes use of the inter-layer correlation to adaptively select the candidate modes. Based on the log-linear rate-distortion relation of inter-dependent layers, we propose a rate-distortion estimation model that predicts the better performer among the Intra4x4 and Intra8x8 prediction types. The intra prediction is then constrained to take the same or similar prediction directions of the corresponding coding block at the base/reference layer. Particularly, to achieve a better trade-off between complexity and coding efficiency, the Intra16x16 prediction is enabled only if the base layer is coded with IntraBL/Intra16x16 or the spatial resolution is higher than CIF. Comparing to JSVM 8 [77], the

proposed schemes provide up to 64% saving on the overall encoder time and a 69% time reduction for encoding the enhancement layers with negligible coding loss.

Section 4.2 Statistical Analysis of Intra Predictions

In this section, we examine the statistical correlation of intra prediction modes between coding layers.

Without loss of generality, the encoder is configured to consist of one base layer along with one CGS or spatial enhancement layer. We identify strong correlations between layers as described below.

4.2.1 Mode Correlation of Base and Enhancement Layers

Our first study aims at investigating the correlation of intra prediction modes between an enhancement layer and its base layer. To quantitatively characterize the correlation, we first compare the prediction directions of two layers when a block is coded by either Intra4x4 or Intra8x8. Specifically, an enhancement coding block is said to have a *similar* prediction mode as its counterpart at the base layer if the best prediction comes from the same or neighboring directions, or is the *DC* mode. For instance, if the coding block at the base layer uses the *Vertical* mode and the corresponding enhancement layer block takes *Vertical*, *Vertical Right*, *Vertical Left*, or *DC* predictions, then these two blocks are called *similar* in prediction mode. This similarity check requires a location check at the base layer. As depicted in Fig. 4-1, the process is implemented by a unique, 1-to-1 block address mapping in CGS. For the spatial scalability, there are two possibilities depending on the prediction type: (1) a 1-to-1 mapping that associates each 8x8 prediction block with

a 4x4 base block or (2) a 1-to-4 mapping that relates a group of adjacent 4x4 prediction blocks to a 4x4 base block. In both cases, the enhancement layer always refers to the base-layer prediction mode of Intra4x4 because the minimum coding block size is 4x4.

Observation 1: Strong Correlation of Intra Direction Modes between Layers. Fig. 4-2 shows the

probability profiles for the base and enhancement layers of *similar* prediction modes, given a fixed base-layer Qp_B and a set of enhancement-layer Qp_E ranging from 10 to 40. It is evident that the prediction modes of the two layers have strong correlation and that more than 70% of the block pairs use *similar* prediction modes. Moreover, in Fig. 4-2(a), the correlation becomes even stronger when Qp_E is closer to Qp_B . After examining the data, the reason that the correlation still gets higher with increasing Qp_E , shown in Fig. 4-2(b), is that the *DC* mode probability goes up at the enhancement layer. The results are consistent over different video sequences and scalability configurations.

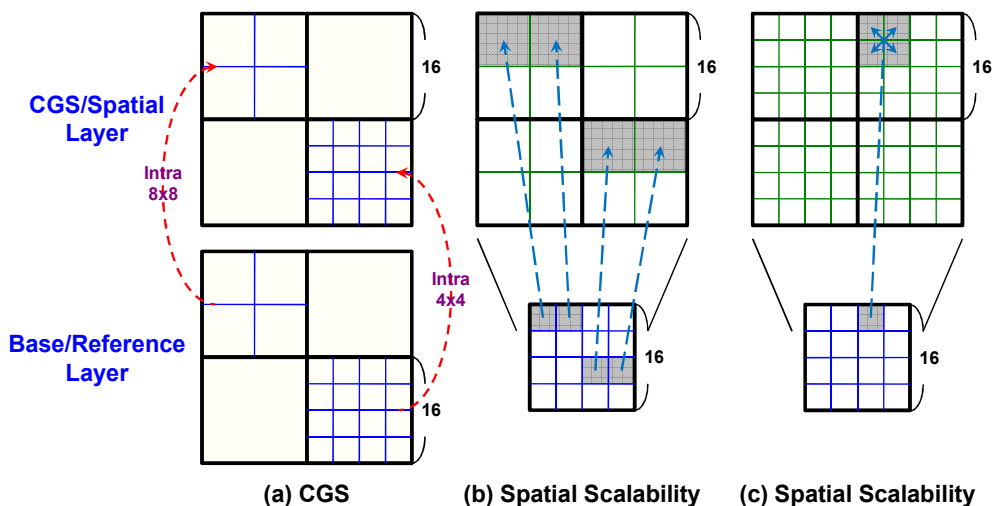
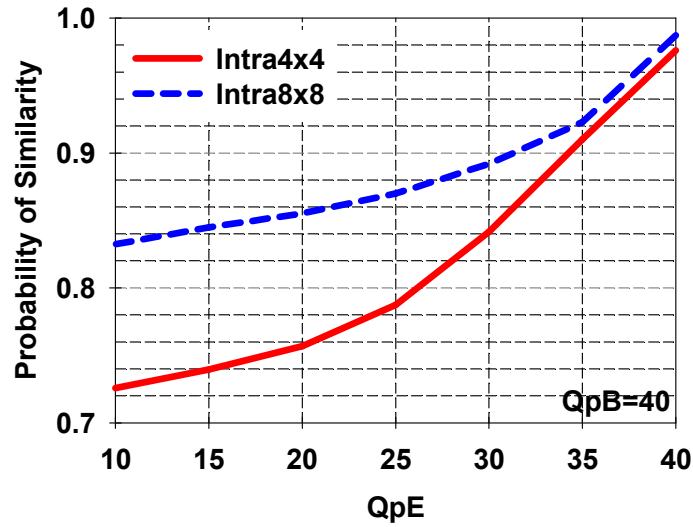
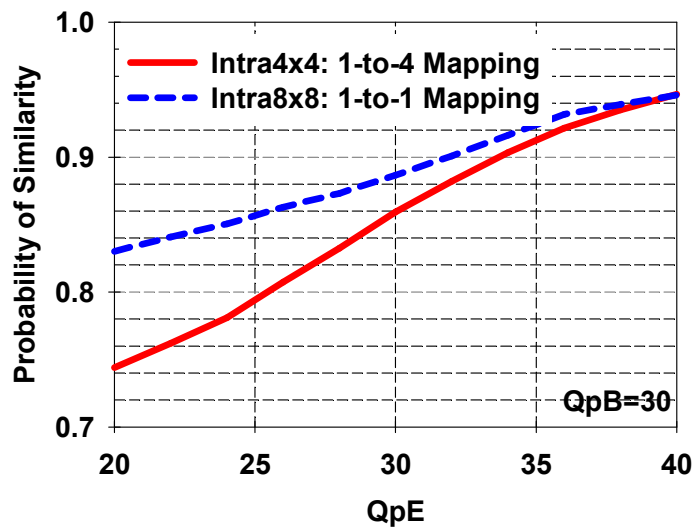


Fig. 4-1 Block address mapping for intra direction mode: (a) CGS, (b) spatial scalability with 1-to-1 mapping, and (c) spatial scalability with 1-to-4 mapping



(a)



(b)

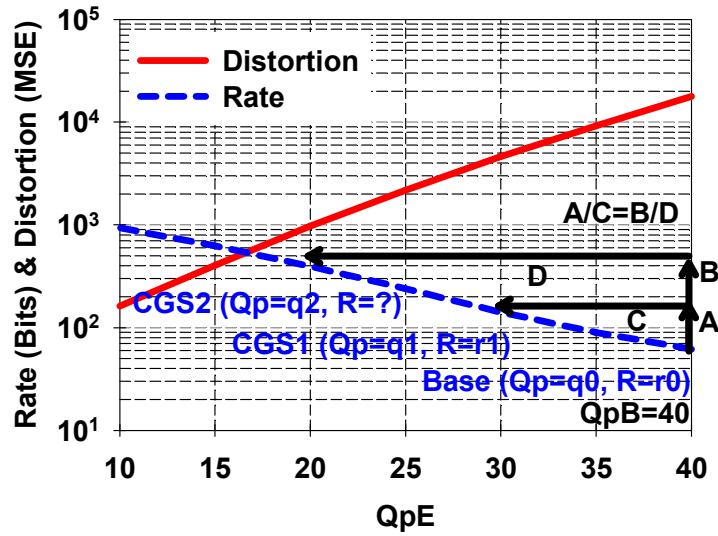
Fig. 4-2 Probability profiles of “similarity” between coding layers: (a) CGS and (b) spatial scalability.

(FOREMAN)

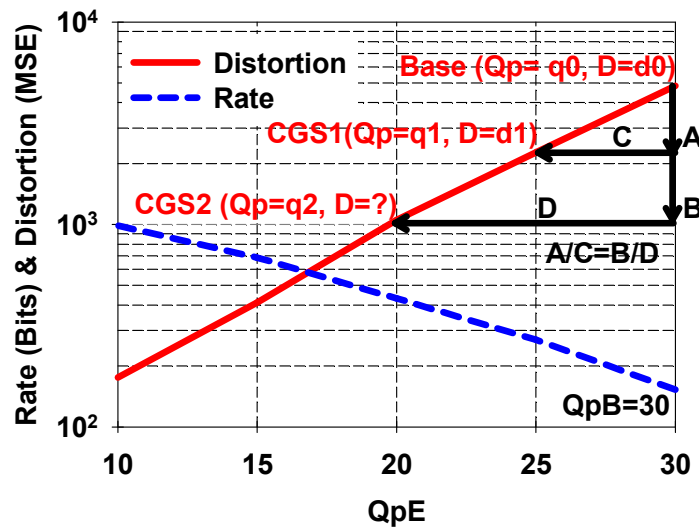
4.2.2 Rate-distortion Profile of Intra Prediction

Our second study addresses the question that how the rate-distortion costs of different CGS layers differ from one another in terms of their Qp values and intra prediction types. Similar to the previous experiment, in this study, the base-layer is coded by a fixed Qp_B and the

enhancement-layer Q_{pE} value varies from 10 to 40.



(a)



(b)

Fig. 4-3 Rate-distortion profiles between CGS layers for (a) Intra4x4 and (b) Intra8x8 (FOREMAN)

Observation 2: Log-linearity of Rate and Distortion. Fig. 4-3 depicts the rate-distortion profiles of the Intra4x4 and Intra8x8 predictions. It shows that the logarithmic rate-distortion costs in both cases are linear functions of the enhancement-layer Q_{pE} , which implies the rate-distortion functions of CGS can be well approximated by the log-linear parametric rate-distortion models:

$$\log \widehat{D}_E^{\mathcal{I}}(Qp_E) = \alpha_{\mathcal{D}}^{\mathcal{I}} \times Qp_E + \beta_{\mathcal{D}}^{\mathcal{I}} \quad (4.1)$$

$$\log \widehat{R}_E^{\mathcal{I}}(Qp_E) = \alpha_{\mathcal{R}}^{\mathcal{I}} \times Qp_E + \beta_{\mathcal{R}}^{\mathcal{I}} \quad (4.2)$$

where $\mathcal{I} \in \{4 \times 4, 8 \times 8\}$. Eq. (4.1) comes at no surprise since the distortion function of a uniform, scalar quantizer is well known to be a quadratic function of the quantization step-size, and the quantization step-size in H.264/SVC [2] is an exponential function of parameter Qp . Together, the above distortion model is anticipated. Similarly, the rate model in Eq. (4.2) can be explained using the classical rate-distortion theory.

In summary, the best intra prediction mode at the enhancement layer is highly correlated to the mode used by the base layer, and the log-linear rate-distortion models can well approximate the rate-distortion functions for CGS. The results inspire us to develop a layer-adaptive mode decision algorithm based on the coding state of the base layer.

Section 4.3 Proposed Macroblock-Adaptive Rate-Distortion Estimation Algorithm [82]

In this section, the fast mode decision algorithm for combined CGS and spatial scalability is proposed based on the observations in Section 4.2.

4.3.1 Algorithm Overview

The proposed algorithm, with flowchart in Fig. 4-4, consists of four major steps, which adaptively evaluate the Intra4x4, Intra8x8, and Intra16x16 prediction modes and the default IntraBL mode.

Step 1: Exhaustive Base Layer Prediction. The base layer performs the exhaustive search to identify the best Intra4x4 and Intra8x8 prediction modes together with their rate-distortion costs for use in the rate-distortion estimation and the layer-adaptive mode selection.

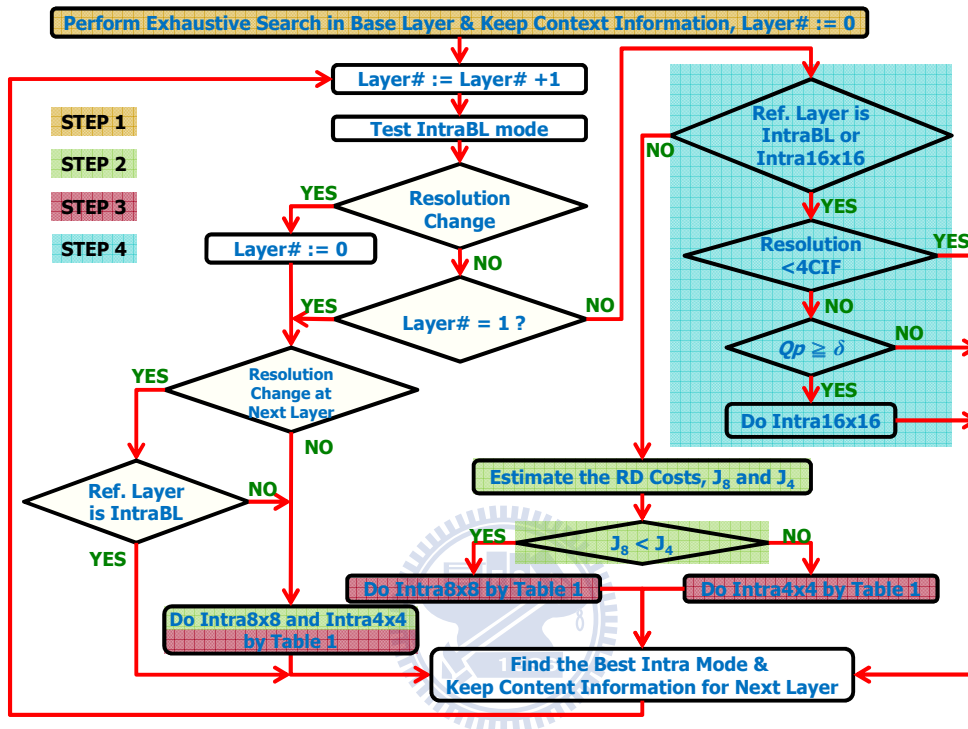


Fig. 4-4 Fast mode decision algorithm with rate-distortion estimation and layer-adaptive mode selection

Step 2: Macroblock-Adaptive Rate-Distortion Estimation of Intra4x4/8x8. The rate-distortion estimation step predicts which of the Intra4x4 and Intra8x8 prediction types at CGS layers has the inferior rate-distortion performance and should be excluded from testing at the very beginning. We achieve a computing time saving by eliminating the estimated weak modes. Section 4.3.2 details the procedure of rate-distortion estimation.

Step 3: Layer-Adaptive Mode Selection. Due to the strong correlation between the coding layers

in the Intra4x4 and Intra8x8 modes, the enhancement layers skip less probable prediction modes to reduce complexity by checking their base/ reference layer coding states. Section 4.3.3 elaborates our implementation.

Step 4: Conditional Intra16x16 Prediction. The conditional Intra16x16 prediction provides a better trade-off between coding efficiency and complexity. As found in our investigation, the Intra16x16 offers comparable coding efficiency at a much lower complexity for high-resolution video sequences with smooth image contents.

4.3.2 Macroblock-Adaptive Rate-Distortion Estimation

Based on our observations discussed earlier, we notice that (a) the rate-distortion behavior can be estimated by the log-linear rate-distortion models and (b) the estimation should be adjusted to match the varying statistics of input data. These two conclusions help us in developing a macroblock-adaptive rate-distortion estimation.

The log-linear rate-distortion models suggests that given any two CGS layers their distortion reduction (or bit rate increase) ratio with respect to the change of Qp is roughly a constant, as illustrated by the two triangles in Fig. 4-3. Numerically, this statement is expressed by Eq. (4.2), in which the enhancement-layer rate-distortion estimators, $\widehat{\mathcal{D}}_E^I(Qp_E)$, $\widehat{\mathcal{R}}_E^I(Qp_E)$, are functions of the measured rates and distortions of the base and the first CGS layers, $\widehat{\mathcal{D}}_B^I(Qp_B)$, $\widehat{\mathcal{R}}_B^I(Qp_B)$, $\widehat{\mathcal{D}}_E^I(Qp_{E1})$, $\widehat{\mathcal{R}}_E^I(Qp_{E1})$, and their Qp values, Qp_B and Qp_{E1} .

$$\log \widehat{\mathcal{D}}_E^I(Qp_E) = \alpha_D^I \times (Qp_B - Qp_E) + \log \mathcal{D}_B^I(Qp_B), \quad (4.3)$$

$$\log \widehat{\mathcal{R}}_E^I(Qp_E) = \alpha_{\mathcal{R}}^I \times (Qp_B - Qp_E) + \log \mathcal{R}_B^I(Qp_B) \quad (4.4)$$

where

$$\alpha_{\mathcal{D}}^I = \frac{\log \mathcal{D}_E^I(Qp_{E1}) - \log \mathcal{D}_B^I(Qp_B)}{Qp_{E1} - Qp_B} \quad \text{and} \quad \alpha_{\mathcal{R}}^I = \frac{\log \mathcal{R}_E^I(Qp_{E1}) - \log \mathcal{R}_B^I(Qp_B)}{Qp_{E1} - Qp_B}.$$

For the first two layers, to better match the local statistics, the average rate-distortion costs are used in Eq. (4.3) and Eq. (4.4) for each macroblock. The estimated rate-distortion costs of both Intra4x4 and Intra8x8 predictions are then used in the Lagrange multiplier to identify the best prediction type.

Table 4-1 Look-up table for layer-adaptive intra mode selection

Candidate Modes		Prediction Modes of Reference Layers								
		Intra4x4/Intra8x8								
		0	1	2	3	4	5	6	7	8
Intra4x4/Intra8x8	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 {●, ○}.

4.3.3 Layer-Adaptive Intra Mode Selection

As suggested by our statistical analysis, 70% or higher Intra4x4/8x8 coding blocks choose their prediction modes similar to their counterparts at the base/reference layers. This strong correlation is used to design Table 4-1 for the layer-adaptive mode selection. As shown, each macroblock is tested

with 4 or fewer prediction modes; each must possess the same or similar prediction directions as the base layer. If the base layer is encoded by DC, Vertical, or Horizontal mode, the diagonal direction predictions are also omitted for further complexity reduction. Similarly, only one prediction mode is retained if both the previous two layers choose the identical mode. Nevertheless, the Intra4x4 and Intra8x8 modes are both skipped if Intra16x16 or IntraBL is in use at the base/reference layer.

Unlike the Intra4x4 and Intra8x8 modes, the Intra16x16 is adaptively enabled not only for increasing coding efficiency, it is more so for complexity reasons. From the statistical analysis, Intra16x16 is shown to be more efficient at low bit rates and/or in high-resolution video sequences. As a result, it is included in testing if (1) the base layer adopts Intra16x16 or IntraBL, (2) the spatial resolution is higher than CIF, and (3) the Q_p is greater than a threshold δ , which is set empirically or is adjusted by users. In practice, the threshold δ in this paper is set to 30.

The IntraBL, on the other hand, is a part of the default because it offers superior rate-distortion performance comparing to the other modes, especially when the base and the enhancement layers are coded at similar quality.

Section 4.4 Experiments

In the following experiments, the proposed algorithm is implemented on JSVM 8 [77] and with the test video sequences covering a broad range of visual characteristics. We run tests of CGS, dyadic spatial scalability, and the combination of these two scalabilities. The layer dependency in three tests is shown in Fig. 4-5. The detailed encoder configurations are specified by Table 4-2.

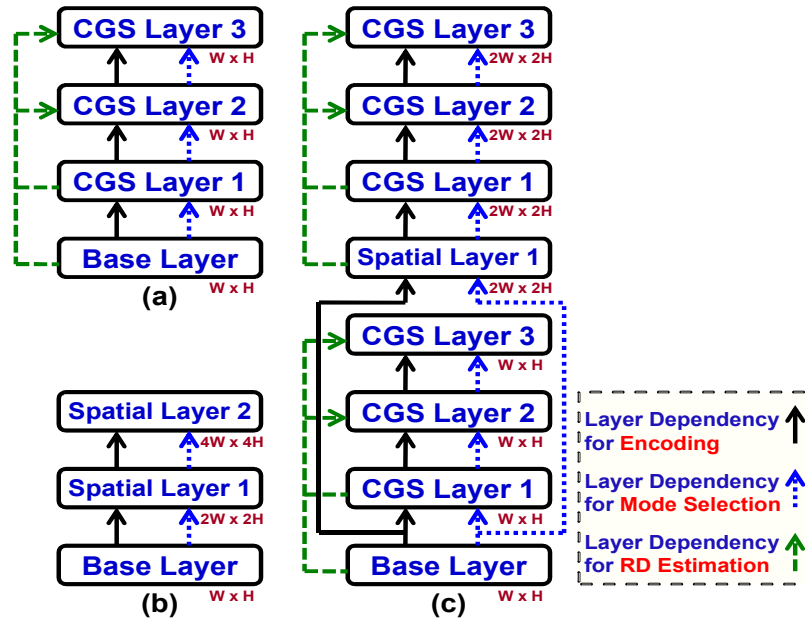


Fig. 4-5 Inter-layer dependency settings of H.264/SVC encoder [2] for (a) CGS, (b) dyadic spatial scalability, and (c) combined scalability

Table 4-2 Testing conditions

Sequence	QCIF	BUS (BU), NEWS (NS), STEFAN (SF)
	CIF	AKIYO (AK), FOOTBALL (FB), FOREMAN (FM), MOBILE (MB)
	4CIF	CITY (CT), CREW (CR), HARBOUR (HB), ICE (IC), SOCCER (SC)
	HD	DUCKSTAKEOFF (DTO), OLDTOWNCROSS (OTC), PARKLOY (PJ)
Q_p	(a)	I $Q_{p_{i+1}} = Q_{p_i} - 6; Q_{p_0} = 42$
	(b)	II $Q_{p_0} = 40; Q_{p_1} = 43; Q_{p_2} = 46$
	(c)	III $Q_{p_{i+1}} = Q_{p_i} - 6; Q_{p_0} = 40; Q_{p_4} = 43$
		IV $Q_{p_{i+1}} = Q_{p_i} - 6; Q_{p_0} = 40; Q_{p_4} = 46$
		V $Q_{p_{i+1}} = Q_{p_i} - 6; Q_{p_0} = 30; Q_{p_4} = 36$
Misc.	RDO+CABAC+Adaptive Inter-Layer Prediction	

Structures (a)-(c) are specified in Fig. 4-5.

In Table 4-3, the proposed schemes are compared with JSVM 8 [77] in terms of the average

Y-PSNR loss (ΔP) and bit rate increase (ΔR), and the overall time saving (TS). Our scheme provides up to 64% overall encoder time saving over JSVM 8 [77], which is about 3 times faster in the encoding process. Moreover, the average bit rate increase is less than 0.9% and the PSNR loss is no more than 0.05 dB. The changes in PSNR and bit-rate are so small that their output sequences cannot be distinguished between the proposed scheme and JSVM 8 [77]. In Table 4-4, the complexity ratio is defined as the ratio of the enhancement layers encoding time and the base layer encoding time. As shown, the proposed scheme provides up to 69% time saving for encoding the enhancement layers.

In addition to JSVM 8 [77], we also compare our scheme with the other two state-of-the-art fast algorithms [78][79], in which a reduced mode set offers a reduction of 58% in encoding complexity. However, without considering the Intra16x16, these two schemes result in higher Y-PSNR loss and bit rate increase, especially in the CREW and ICE sequences.

Also from Fig. 4-5(b) and Table 4-3, it is interesting to note that the speed-up in the spatial scalability is only slightly higher than 50%. This is mostly due to the fact that the spatial interpolation, as required by the spatial scalability, involves computation-intensive filtering operations that are not accelerated in the proposed scheme. Another factor is that the rate-distortion estimation is currently applied to the CGS case only. As a result, the gain comes solely from our layer-adaptive mode selection.

Table 4-3 Performance comparisons

Sequence	Xiong's method [78]			Yang's method [79]			Proposed		
	ΔP (dB)	ΔR (%)	TS (%)	ΔP (dB)	ΔR (%)	TS (%)	ΔP (dB)	ΔR (%)	TS (%)
Structure (a) with Qp Setting I and $\delta = 30$									
BU	0.00	0.01	49.3	0.00	0.04	34.9	0.00	-0.01	57.5
NS	-0.03	0.15	47.7	-0.01	0.29	34.4	-0.02	0.23	55.8
FM	-0.04	0.48	48.5	-0.01	0.48	36.0	-0.02	0.28	55.7
MB	0.00	0.00	50.2	0.00	0.01	35.9	0.00	0.00	57.5
CR	-0.03	1.46	49.2	-0.03	1.74	35.6	-0.02	0.71	51.0
SC	0.00	0.19	50.7	0.00	0.32	37.0	0.00	0.01	52.5
AVG.	-0.02	0.38	49.3	-0.01	0.48	35.6	-0.01	0.20	55.0
Structure (b) with Qp Setting II									
DTO	0.00	-0.20	45.1	-0.01	-0.18	33.9	0.00	-0.07	51.4
OTC	0.00	0.01	44.2	0.00	0.02	33.4	0.00	0.05	49.2
PJ	0.00	-0.04	44.4	0.00	0.03	33.5	0.00	-0.01	50.3
AVG.	0.00	-0.08	44.6	0.00	-0.06	33.6	0.00	-0.01	50.3
Structure (c) with Qp Setting III and $\delta = 30$									
CT	-0.01	0.20	56.7	-0.01	0.30	41.2	-0.01	0.05	59.5
CR	-0.04	1.45	55.0	-0.04	1.71	40.6	-0.02	0.84	57.1
IC	-0.08	0.78	53.7	-0.05	0.87	40.1	-0.05	0.46	55.1
AVG.	-0.04	0.81	55.1	-0.03	0.96	40.6	-0.03	0.45	57.2
Structure (d) with Qp Setting IV and $\delta = 30$									
CR	-0.04	1.29	54.6	-0.04	1.54	40.4	-0.01	0.71	56.6
HB	-0.01	0.04	57.1	-0.01	0.11	40.7	-0.01	-0.04	59.6
IC	0.00	0.22	55.5	0.00	0.35	40.6	0.00	-0.04	57.6
AVG.	-0.02	0.52	55.7	-0.02	0.67	40.6	-0.01	0.21	57.9
Structure (e) with Qp Setting V and $\delta = 30$									
CT	0.00	0.01	58.0	0.00	0.03	41.1	0.00	-0.03	63.7
CR	-0.01	0.32	55.9	-0.01	0.43	40.3	0.00	0.22	62.3
SC	0.00	0.01	57.2	0.00	0.05	40.9	0.00	-0.02	63.1
AVG.	0.00	0.11	57.0	0.00	0.17	40.8	0.00	0.06	63.0

Table 4-4 Layer complexity ratio of enhancement-layer encoding time to base-layer encoding time

Structure	JSVM 8 [77]	Xiong's method [78]	Yang's method [79]	Proposed
(a)	3.99	1.53	2.21	1.25
(b)	25.18	13.52	16.38	12.02
(c)	23.57	9.79	13.57	8.93



Chapter 5

Fast Mode Selection and Motion Search for Scalable Video Coding with Combined Coarse Granular Scalability (CGS) and Temporal Scalability

To speed up the H.264/SVC encoder [2], we propose a layer-adaptive intra/inter mode decision algorithm and a motion search scheme for the hierarchical-B frames in H.264/SVC [2] 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 performance contributed by different coding modes at the enhancement layers and the mode conditional probabilities at different temporal layers. For the intra prediction on inter frames, we can reduce the number of Intra4x4/Intra8x8 prediction modes by 50% or more, based on the reference/base layer intra prediction directions. For the enhancement-layer inter prediction, the look-up tables containing inter prediction candidate modes are designed to use the macroblock coding mode dependence on and the reference/base layer quantization parameters (Qp). In addition, to avoid checking all motion estimation reference frames, the base layer reference frame index is selectively reused. And according to the enhancement-layer macroblock partition, the base-layer motion vector can be used as the initial search point for the enhancement-layer motion estimation. Compared with JSVM 9.11 [10], our proposed algorithm provides a 20x speedup on encoding the enhancement layers 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.

The rest of this chapter is organized as follows. Section 5.1 contains a brief review of the prior works related to the fast mode decision algorithms both in H.264/AVC [4] and H.264/SVC [2] coding structure. Section 5.2 analyzes the correlation between the mode distributions of the base layer and enhancement layers. Section 5.3 describes our context-adaptive mode decision algorithm, and also presents our motion search strategy. Section 5.4 compares the proposed schemes with JSVM and the other state-of-the-art algorithms in terms of complexity reduction and rate-distortion performance.



Section 5.1 Literature Review

An effective way to reduce the encoding complexity is to restrict the number of candidate modes. There exists a large body of literature devoted to the studies on mode reduction for H.264/AVC [4]. For example, Tsai *et al.* in [83] 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 sub-block and pixel levels [84]. Another example of using macroblock features to predict mode sets can be found in [85]. They first classify macroblocks 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.* [69] pick up the mode set for each macroblock based on its motion activity. There are some other mode reduction approaches that exploit the spatial and temporal correlation between partition modes. Their processes usually predict the most probable macroblock mode by observing the coding mode of its nearby macroblocks [61] or of its co-located macroblock in the previous frame [68]. 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 [86]. This type of techniques has often been generalized to a hierarchical decision process with multiple termination criteria in [68], [65] and [87]. All these methods are equally applicable to the intra-layer mode reduction in H.264/SVC [2].

Thus far, little research has been devoted to the study of the H.264/SVC [2] fast mode decision. Most of published articles use the *inter-layer correlation* to confine the mode search at the ELs. Li *et al.* [80][88], for example, observe that owing to the Lagrange rate-distortion optimization process, the inter macroblock motion partition at enhancement layers tends to be the same as or smaller than that of its corresponding base-layer macroblock. This observation is used in conjunction with the base layer mode decision to design a fast mode search for the ELs. In [72], 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 [89], Ren *et al.* notice a high correlation exists in spatially neighboring macroblocks. 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 rate-distortion costs to progressively perform the mode search for each macroblock until an early termination condition is satisfied. Some other previous work has been associated with the intra macroblock mode reduction. Yang *et al.* [79] show that the *inter-layer intra prediction* can effectively replace Intra16x16 and Intra8x8 modes. On top of that, Xiong [78] makes an additional simplification by restricting the Intra4x4 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% to 60% (in comparison with JSVM 9.11 [10]) has been reported at the cost of 1% to 4% bit-rate increase for typical test sequences.

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

1. *The effect of layer settings on the mode distribution at enhancement layers.* In our previous studies [81][82], we noticed that the quality of the base layer affects the reliability on the candidate mode prediction, and that an enhancement layer, when coded at a much higher bit-rate than its base layer, 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 multi-layer coding scenarios, where the Qp values and the inter-layer dependency change on a layer-to-layer basis.
2. *The correlation between the motion parameters of base layer and enhancement layer.* As also

shown in our previous studies [81][82], an enhancement-layer (inter) macroblock usually has the same *reference frame index* and *prediction direction* as its co-located macroblock at the base layer, especially when both are coded with the same macroblock partition. In this regard, the exhaustive motion search (adopted by most previous researchers) may not be needed for reaching the target rate-distortion 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 H.264/SVC [2] with combined CGS and temporal scalability. Our scheme distinguishes from the other approaches in two significant ways: (1) the candidate mode set for each enhancement-layer macroblock is chosen according to both local and global contexts—including the coding mode adopted by its co-located macroblock at the base layer, the Qp assigned to the base layer and enhancement layers, as well as its temporal layer index, and (2) the search for motion parameters, for a particular candidate mode, is conducted only when the base-layer motion information is not reusable. That is, the exhaustive motion search is performed only when the BL motion information is judged unreliable for that enhancement layer. Compared with JSVM 9.11 [10], 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 enhancement layers alone is reduced to 10% of that of the JSVM implementation [10]. Compared with the state-of-the-art fast algorithms, [80][88][89], 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.

Section 5.2 Correlations between Base and Enhancement Layers

In this section, we are going to investigate the relationship between the base-layer coding modes and the enhancement-layer 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 enhancement-layer dominating modes; (2) how these modes are distributed when the base-layer mode is given; and (3) which coding modes are most critical to the enhancement-layer rate-distortion performance. In addition, we examine the statistics of the reference frame selection and the inter-layer residual predictor efficiency. Our codec contains one base layer and one CGS enhancement layer and is tested on six video sequences: AKIYO (QCIF), STEFAN (QCIF), FOREMAN (CIF), MOBILE (CIF), CITY (4CIF), and CREW (4CIF). The notations Qp_B and Qp_E denote the quantization parameters of the base layer and enhancement layers, respectively, and AVG. shows the averaged behavior of all six test sequences.

5.2.1 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. 5-1, the distribution of the enhancement-layer 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 the base layer and enhancement layers. When the base layer is coded

with good quality (using a small Q_{pB}), most of the intra macroblocks are coded in the IntraBL type, whose predictor comes from the base-layer intra-coded macroblock. However, when the enhancement-layer quality gradually improves, the intra predictor is switched from the base layer to the enhancement layer. Particularly, the Intra4x4 percentage increases more noticeably than the other two types, Intra8x8 and Intra16x16; together with the IntraBL, it makes up 80% or more of the intra prediction types at enhancement layers. 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 [79]. In addition, the Intra16x16 is preferred for smooth areas, but its presence is usually less than 10% at the CGS enhancement layers because it must compete with the IntraBL mode, which is chosen more often in the smooth areas due to less overhead. As the base-layer quality improves, the Intra8x8 and the Intra16x16 do not seem to offer benefit in coding efficiency.

In addition to the intra prediction type, we compare the nine prediction directions in intra coding when both layers are coded by either Intra4x4 or Intra8x8. Specifically, an enhancement-layer coding block is said to have a *similar* prediction mode to its counterpart at the base layer 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 the base layer selects the *Vertical* mode and the one at the enhancement layer 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 base-layer counterpart of a coding block. As shown by Fig. 5-2, this process can be implemented by a one-to-one block address

mapping in the CGS configuration.



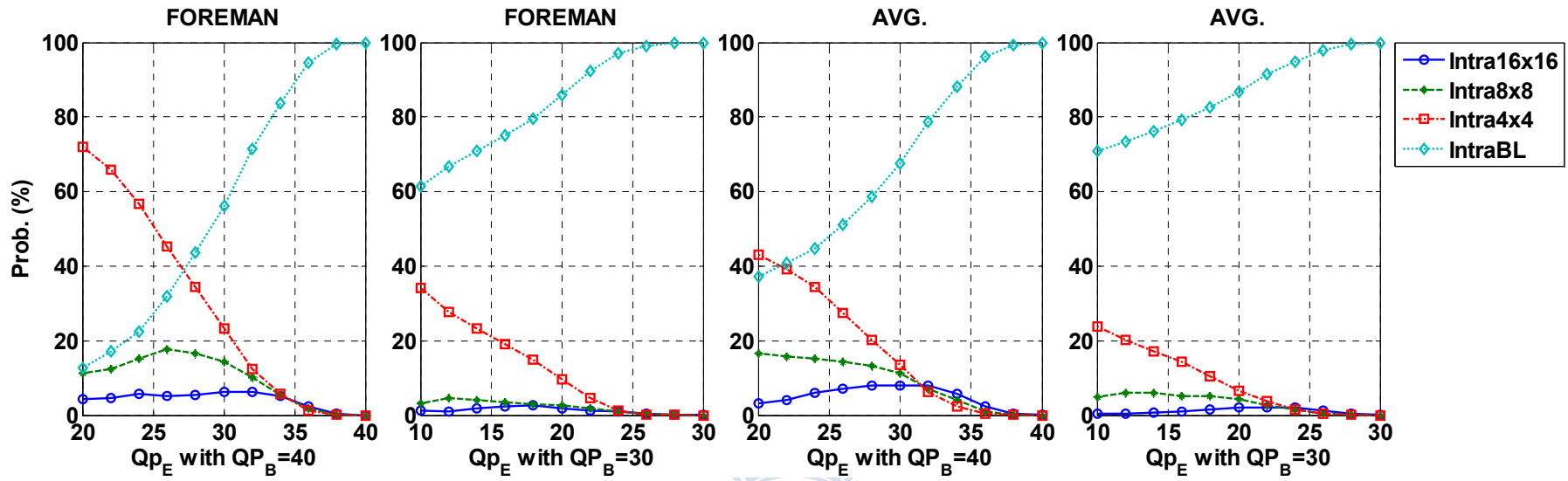


Fig. 5-1 Distribution of intra prediction types at CGS enhancement layers

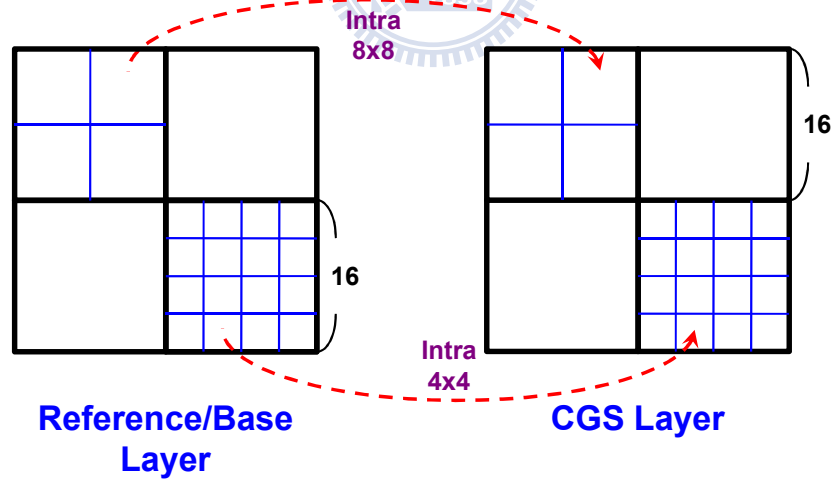


Fig. 5-2 One-to-one block address mapping of CGS

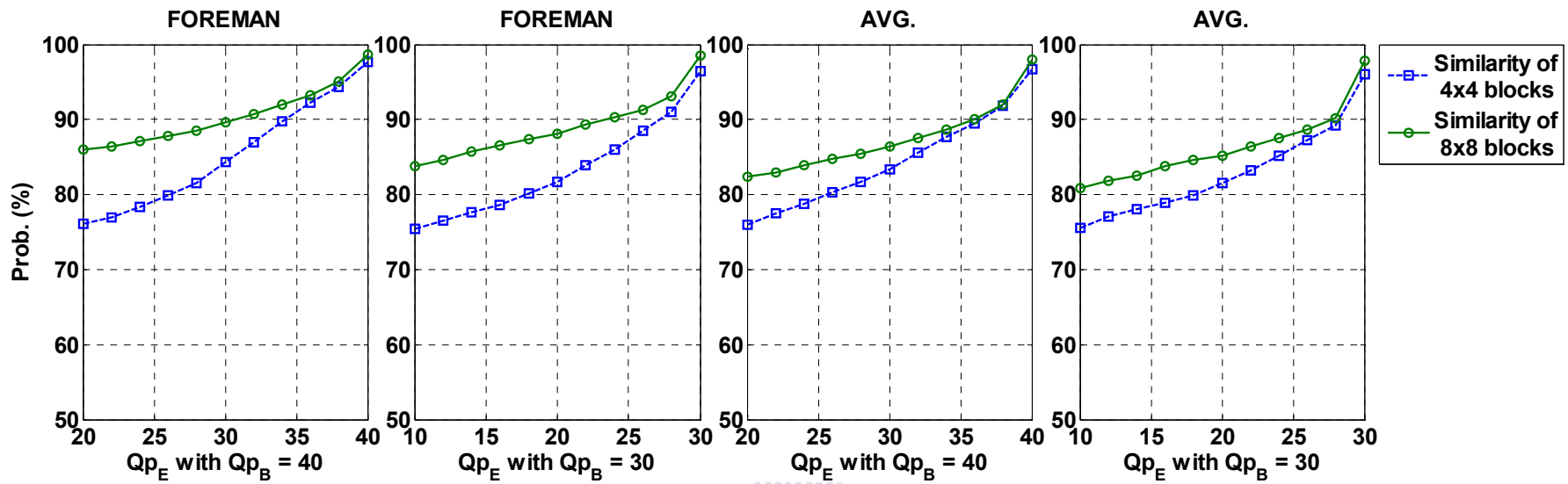


Fig. 5-3 Similarity probability profiles of intra direction mode at CGS enhancement layer with poor-quality base layer ($Qp_B = 40$) and high-quality base layer ($Qp_B = 30$)

Fig. 5-3 shows the probability of the base layer and the enhancement layer having *similar* intra prediction modes for fixed Qp_B and a set of Qp_E values ranging from $(Qp_B - 20)$ to Qp_B . From these data we can conclude that the intra prediction modes between the base layer and the enhancement layer 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 base-layer quality and the test sequence.

5.2.2 Distributions of Inter Prediction Mode in CGS

Next, we investigate the correlation of the motion partition between the base layer and the enhancement layer, 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

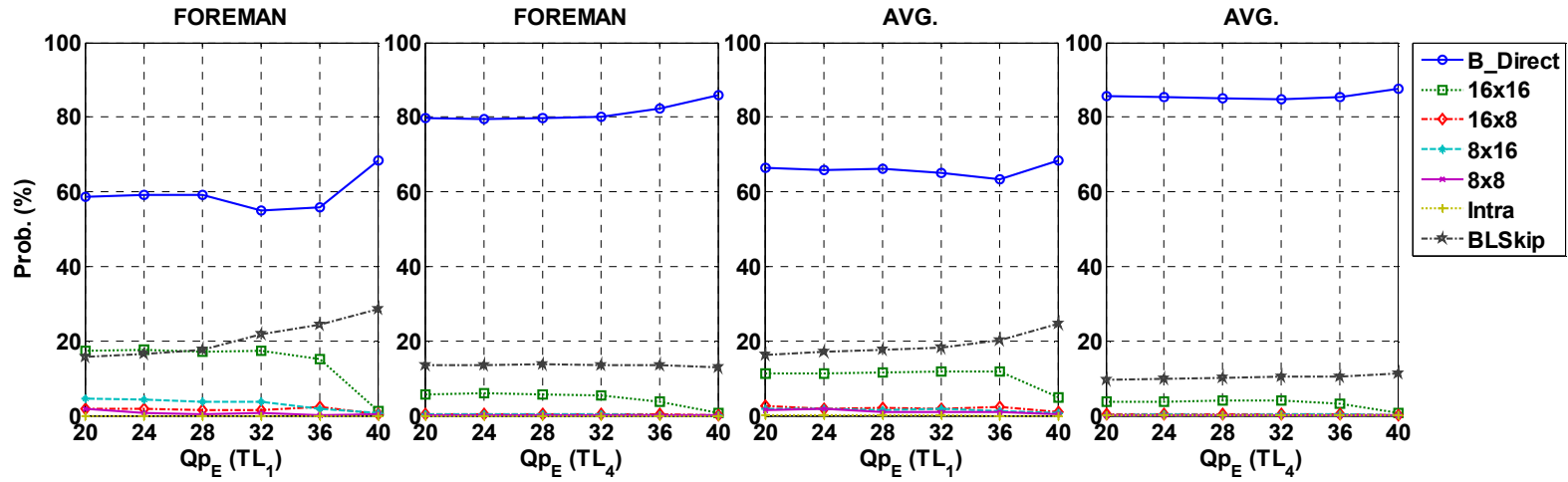
$$\Pr\{\text{Mode}_{\text{EL}}(Qp_E, T_k) = j | \text{Mode}_{\text{BL}}(Qp_B = 40, T_k) = i\}, \quad (5.1)$$

where $\text{Mode}_{\text{BL}}(Qp_B = 40, T_k)$ denotes the best mode selected by the base layer with $Qp_B = 40$ at temporal layer k ; $\text{Mode}_{\text{EL}}(Qp_E, T_k)$ is the optimal mode at the enhancement layer with Qp_E at temporal layer k ; $i \in \{\text{B_Direct/Skip}, \text{Inter16x16}, \text{Inter16x8}, \text{Inter8x16}, \text{Inter8x8}\}$; and $j \in \{\text{B_Direct/Skip}, \text{Inter16x16}, \text{Inter16x8}, \text{Inter8x16}, \text{Inter8x8}, \text{Intra}, \text{BLSkip}\}$. The collected statistics are given in Fig. 5-4 (a)-Fig. 5-4 (e). In addition, in Fig. 5-4 (f), a different conditional probability is defined as

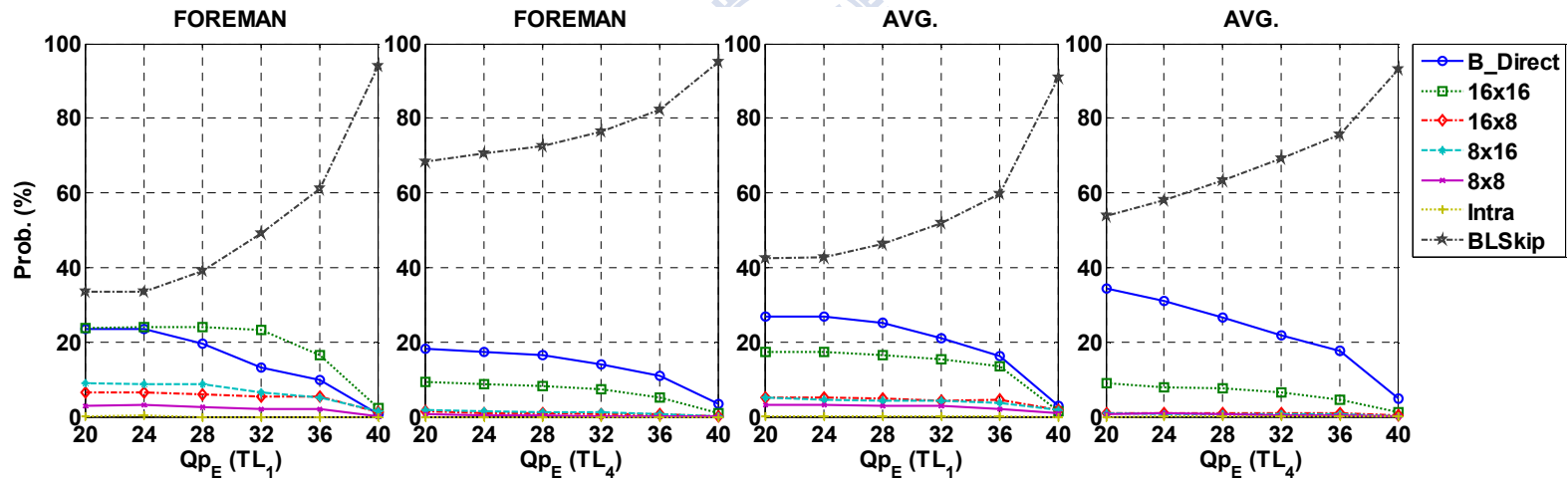
$$\Pr\{\text{SubMode}_{\text{EL}}(Qp_E, T_k) = m | \text{Mode}_{\text{EL}}(Qp_E, T_k) = \text{Inter8} \times 8\}, \quad (5.2)$$

where $m \in \{\text{B_Direct}8 \times 8, \text{Inter}8 \times 8, \text{Inter}8 \times 4, \text{Inter}4 \times 8, \text{Inter}4 \times 4\}$. This conditional probability presents the distribution of the finer partitions including 8×8 and those smaller than 8×8 .

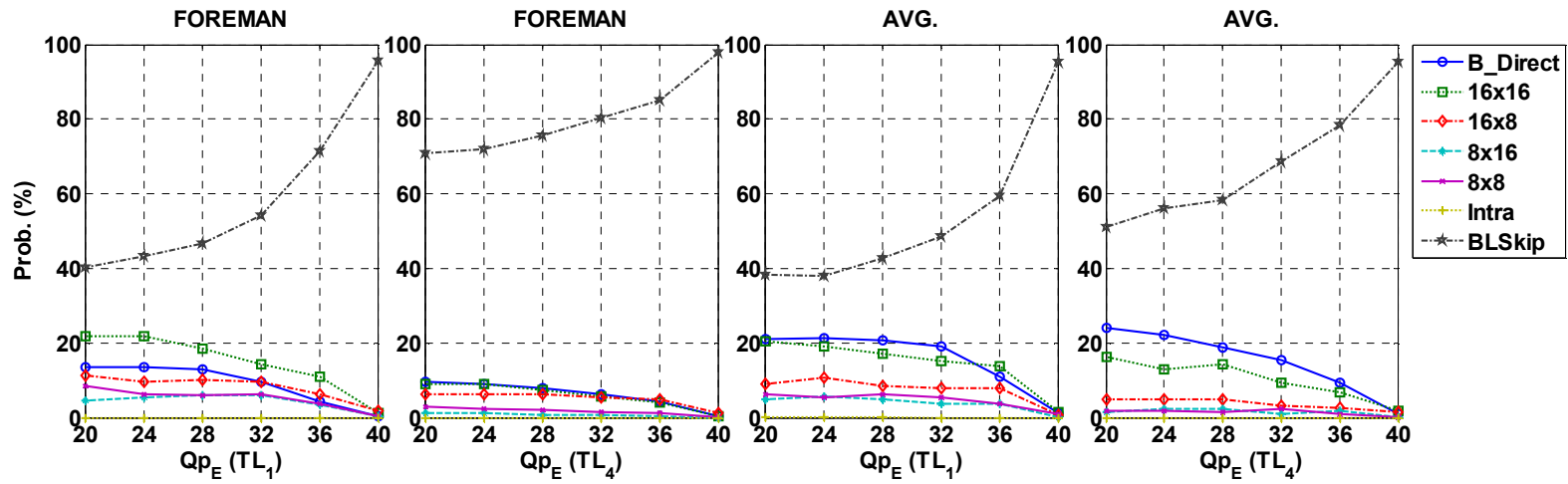




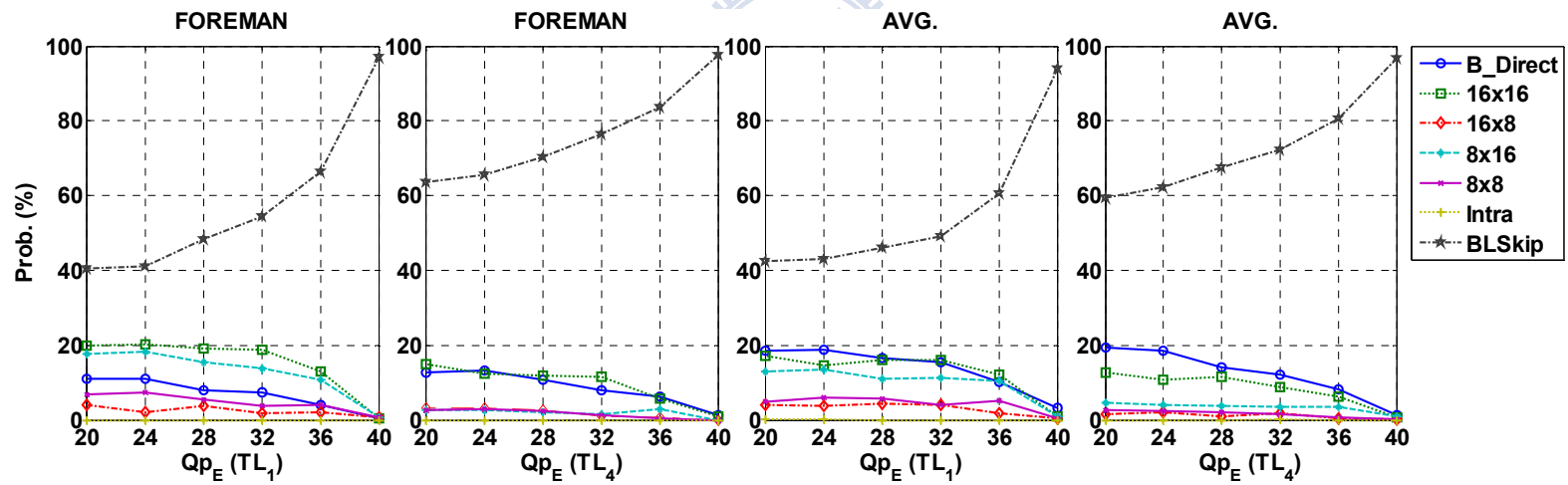
(a) Conditional probability of Mode_{EL} when $\text{Mode}_{\text{BL}} = \text{B_Direct/Skip}$



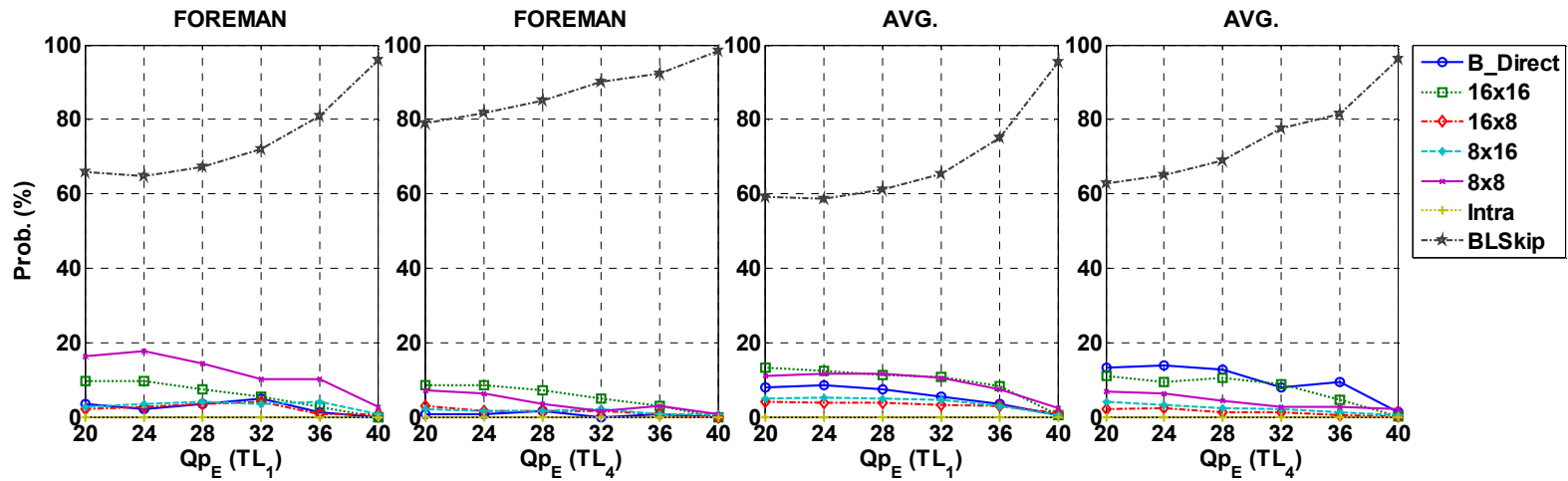
(b) Conditional probability of Mode_{EL} when $\text{Mode}_{\text{BL}} = \text{Inter16x16}$



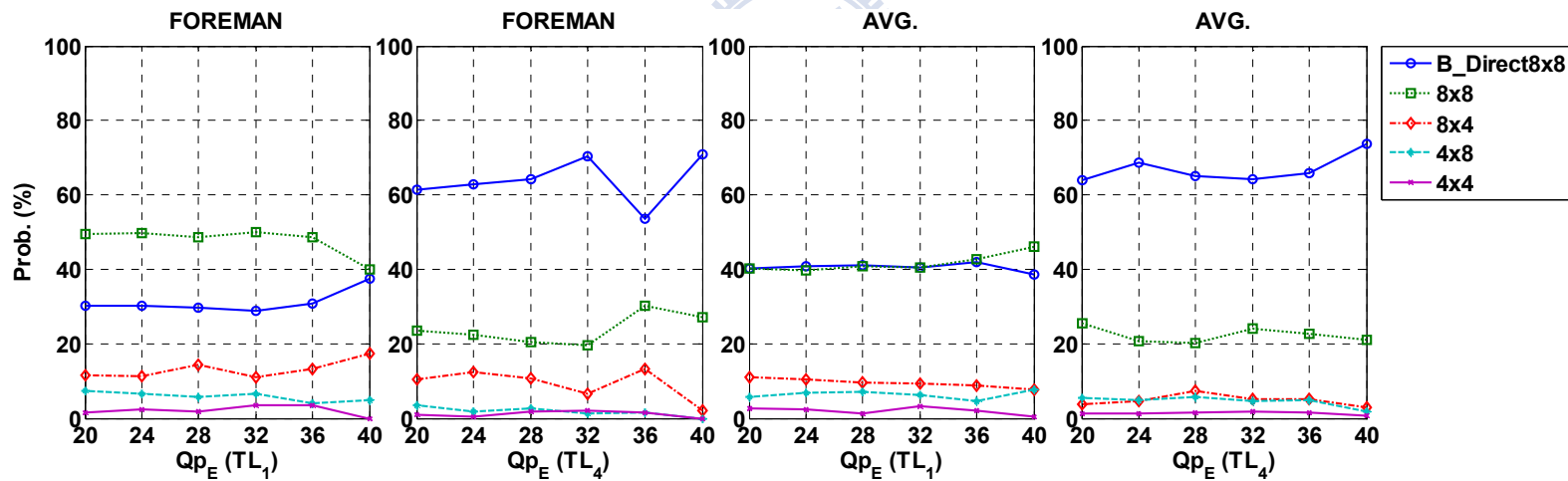
(c) Conditional probability of Mode_{EL} when $\text{Mode}_{\text{BL}} = \text{Inter16x8}$



(d) Conditional probability of Mode_{EL} when $\text{Mode}_{\text{BL}} = \text{Inter8x16}$



(e) Conditional probability of Mode_{EL} when $\text{Mode}_{BL} = \text{Inter8x8}$



(f) Distribution of sub-partition at enhancement layers

Fig. 5-4 Conditional probability of inter partition mode at CGS enhancement layers for $Qp_B = 40$, Qp_E between 20 to 40, and GOP size = 16

From Fig. 5-4, several important observations can be made:

- More than 50% of macroblock pairs choose the same motion partition for both base layer and enhancement layer, namely,

$$\Pr\{\text{Mode}_{\text{EL}}(Qp_E, T_k) = \text{BLSkip} | \text{Mode}_{\text{BL}}(Qp_B = 40, T_k) = i\} + \Pr\{\text{Mode}_{\text{EL}}(Qp_E, T_k) = i | \text{Mode}_{\text{BL}}(Qp_B = 40, T_k) = i\} > 0.5. \quad (5.3)$$

Among them, the enhancement-layer macroblock 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 Inter16x16, respectively.

This observation is slightly different from those in [72][80][88], which suggest the enhancement-layer candidate mode generally does not have partition size larger than its co-located base-layer macroblock mode. Interestingly, if the base-layer macroblock chooses Inter8x8 mode, the choice for the enhancement-layer macroblock is also likely (>70%) to be the same. These results seem to be independent of the Qp difference, $(Qp_B - Qp_E)$.

- When a base-layer macroblock is coded in B_Direct/Skip mode, its co-located enhancement-layer macroblock is often coded in either B_Direct/Skip or Inter16x16.
- If a base-layer macroblock is coded with the 8x16 (or 16x8) partition, it is unlikely that its enhancement-layer counterpart will choose the 16x8 (or 8x16) partition.
- The probability for an enhancement-layer macroblock to be coded in BLSkip mode is

greater than 0.5 at the two highest temporal layers, T_{N-1} and T_N .

- The probability for an enhancement-layer sub-macroblock having a sub-partition finer than 8x8 is usually less than 0.2. Even though the MOBILE and STEFAN have more macroblocks coded with finer partitions, on the average, 70% of sub-macroblocks still select the B_Direct8x8 and Inter8x8 as their sub-partition modes.
- $\Pr\{\text{SubMode}_{\text{EL}}(Qp_E, T_k) = \text{Inter4} \times 4 | \text{Mode}_{\text{EL}}(Qp_E, T_k) = \text{Inter8} \times 8\} < 0.05$: Our experimental data reveal that when an enhancement-layer macroblock is further partitioned into sub-partitions smaller than 8x8, the conditional probability of Inter4x4 is typically less than 0.05, whereas it can increase to 0.1 for the sequences MOBILE and STEFAN.

Fig. 5-4(a) to (e) also show that the most probable mode in the hierarchical-B frames is the BLSkip mode. This is a direct consequence of the Lagrangian rate-distortion optimization process, which looks for a balanced compromise between distortion and coding rate. To achieve a better quality, an enhancement-layer macroblock may search for new motion vectors with the same-size partition or additional motion vectors 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 enhancement layers, especially in the CGS configuration because it has the benefits of reducing the number of candidate modes. This is most obvious when the base layer is coded with good quality using a small Qp_B . In such a case, the conditional probability

$$\begin{aligned} & \Pr\{\text{Mode}_{\text{EL}}(Qp_E, T_k) = i | \text{Mode}_{\text{BL}}(Qp_B = 30, T_k) = i\} + \\ & \Pr\{\text{Mode}_{\text{EL}}(Qp_E, T_k) = \text{BSkip} | \text{Mode}_{\text{BL}}(Qp_B = 30, T_k) = i\} \end{aligned} \quad (5.4)$$

can go higher than 0.9, making it possible to skip more coding modes with different partition size from that at the base layer. Furthermore, the inter-layer relation represented by $\Pr\{\text{Mode}_{\text{EL}}(Qp_E, T_k) = i | \text{Mode}_{\text{BL}}(Qp_B, T_k) = i\}$ becomes stronger as the index of temporal layer T_k increases. We thus divide the mode conditional probabilities into four regions along two dimensions, the temporal layer T_k and the quantization parameter Qp_{n-1} of the reference layer, as illustrated by Fig. 5-5. High conditional probabilities appear at small Qp_{n-1} and 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 four, it is possible that all the temporal enhancement layers belong to the $T_{N-1} \sim T_N$ category.

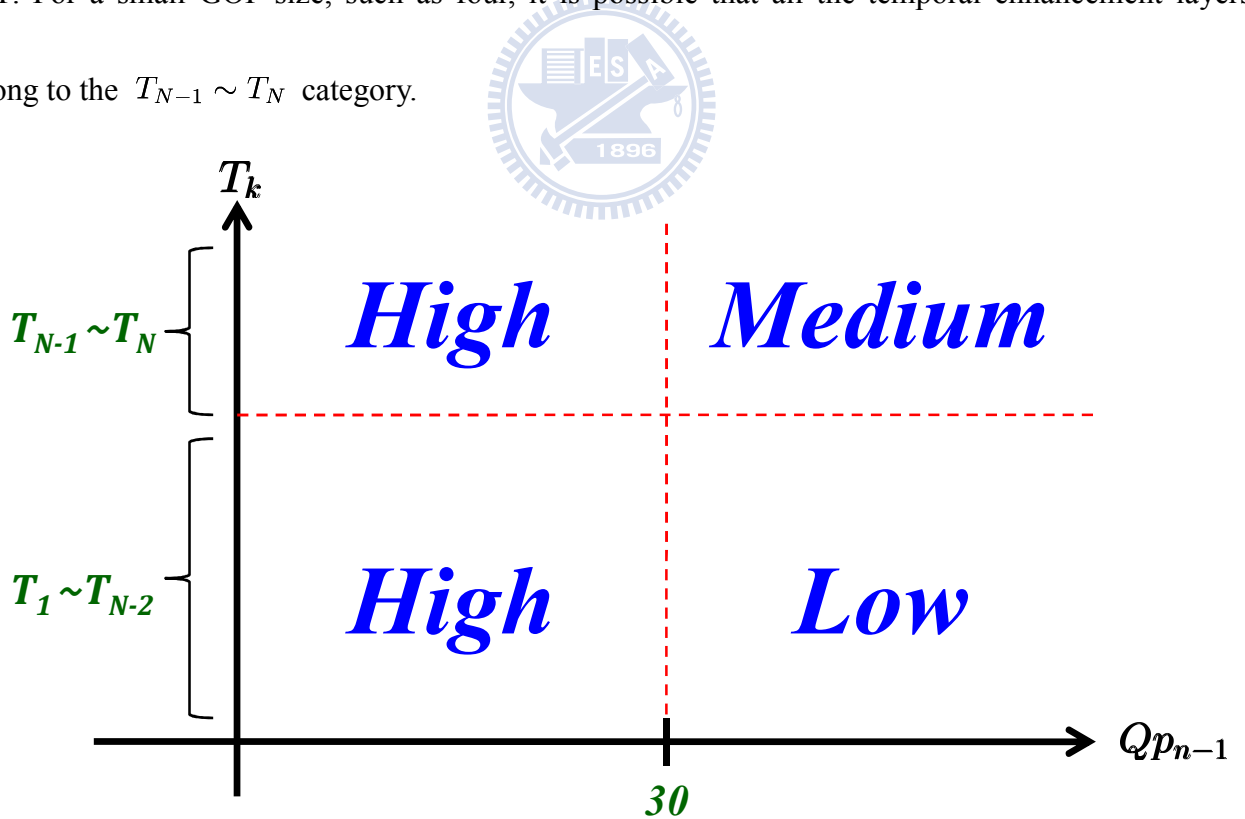


Fig. 5-5 Four regions representing different degrees of mode correlations between coding layers

In summary, the base-layer coding information can be a good reference for predicting the enhancement-layer coding mode in the CGS configuration. Generally, which coding mode would be the best for a base-layer macroblock depends highly on the image texture. However, the conditional probabilities of the enhancement-layer 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 base-layer modes, the distribution of the enhancement-layer modes has a weak dependency on video content. Therefore, in Section 5.3 we will use these observations to design our fast enhancement-layer mode decision algorithm.

5.2.3 Temporal Reference Frames between Coding Layers

As described before, the motion estimation 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 motion vector in all reference frames for each of these temporal predictions. Moreover, the enhancement layer should additionally perform the ME_R , ME_M , and ME_{R+M} modes calculation and selection. Then, based on the rate-distortion cost, the encoder finally chooses the best temporal prediction type and its associated motion vectors for a specified inter coding mode. The current JSVM 9.11 [10] adopts the exhaustive motion search at the enhancement layer, leading to enormously high complexity.

To reduce the enhancement-layer computational load but to maintain good temporal prediction performance, we examine the temporal prediction reference frame selection between the base layer

and enhancement layers. The experiments are performed with reference frame number = 3 and GOP size =16. As shown in Fig. 5-6, 80% or higher enhancement-layer macroblocks choose the same reference frames as their base-layer counterparts. Moreover, the percentage increases as the base-layer quality improves. In other words, the reference frames selected at the base layer can be reliably reused for enhancement-layer macroblocks particularly when the base layer is coded with good quality.



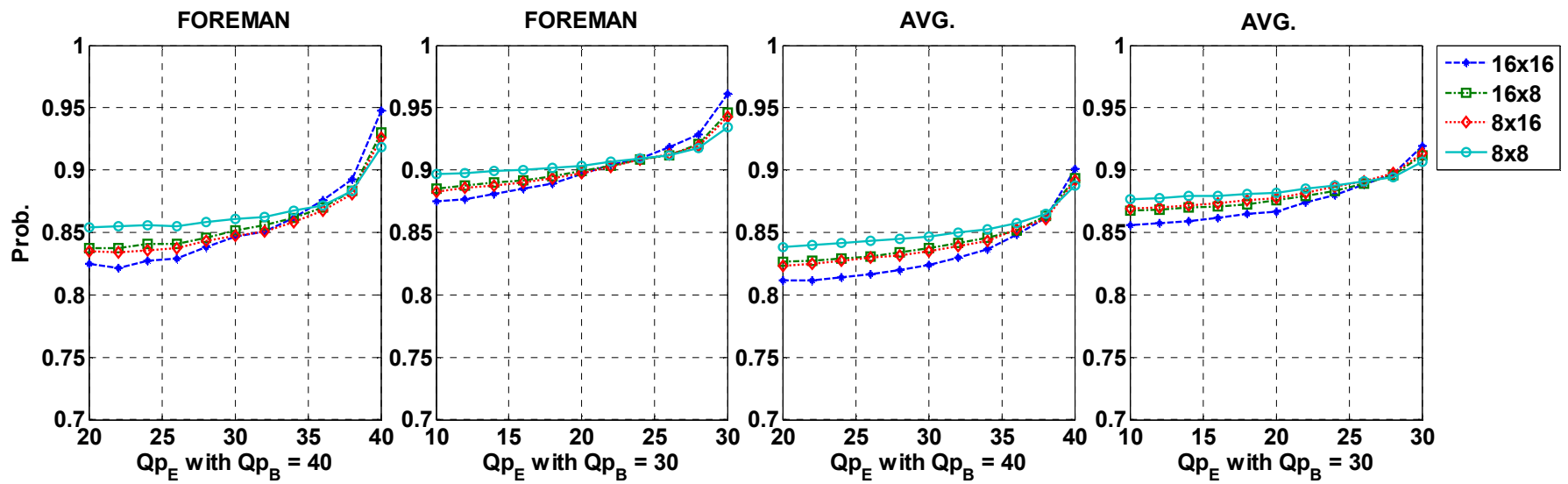


Fig. 5-6 Agreement in selecting reference frames between base layer and enhancement layer

5.2.4 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 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.

Table 5-1 and Table 5-2 tabulate the coding improvement provided by the inter-layer residual predictions, performing in pixel domain and in transform domain respectively, on the hierarchical-B frames at the CGS enhancement layers. The performance metrics are the Y-PSNR loss (ΔP) and bit-rate increase (ΔR) at the CGS enhancement layers, compared with the exhaustive search in the JSVM software. For CGS applications, we find 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 enhancement layers 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 base layer is noise-like signal for the CGS enhancement layers. Similar results can be also observed in the pixel-domain inter-layer residual prediction.

Table 5-1 Turning off the inter-layer residual prediction in transform domain for hierarchical-B frames (JSVM 9.11 [10])

Sequences		High-quality BL			Low-quality BL		
		(Q_{PB}, Q_{PE})	ΔP (dB)	ΔR (%)	(Q_{PB}, Q_{PE})	ΔP (dB)	ΔR (%)
QCIF	BUS	(24,18)	0.00	1.19	(45,39)	0.02	1.02
		(24,12)	0.00	0.43	(45,33)	0.00	0.00
		(24,6)	0.00	0.22	(45,27)	0.00	-0.10
	NEWS	(24,18)	0.00	1.63	(45,39)	0.02	1.40
		(24,12)	0.00	0.49	(45,33)	0.01	0.44
		(24,6)	0.00	0.18	(45,27)	0.00	0.17
CIF	FOREMAN	(24,18)	0.00	0.00	(45,39)	0.01	0.01
		(24,12)	0.00	0.00	(45,33)	0.00	0.00
		(24,6)	0.00	0.00	(45,27)	0.00	0.00
	MOBILE	(24,18)	0.00	0.39	(45,39)	0.01	0.53
		(24,12)	0.00	0.11	(45,33)	0.00	0.06
		(24,6)	0.00	0.01	(45,27)	0.00	0.02
4CIF	CITY	(24,18)	0.00	-0.05	(45,39)	0.00	0.09
		(24,12)	0.00	0.00	(45,33)	0.00	0.00
		(24,6)	0.00	-0.02	(45,27)	0.00	0.03
	CREW	(24,18)	0.00	0.20	(45,39)	0.00	0.23
		(24,12)	0.00	0.07	(45,33)	0.00	-0.02
		(24,6)	0.00	0.04	(45,27)	0.00	0.02

Table 5-2 Turning off the inter-layer residual prediction in pixel domain for hierarchical-B frames

(JSVM 8.0 [77])

Sequences		High-quality base layer			Low-quality base layer		
		(Q_{PB}, Q_{PE})	ΔP (dB)	ΔR (%)	(Q_{PB}, Q_{PE})	ΔP (dB)	ΔR (%)
QCIF	BUS	(24,18)	0.00	2.82	(45,39)	0.00	0.77
		(24,12)	0.00	0.96	(45,33)	0.00	0.12
		(24,6)	-0.02	0.22	(45,27)	0.00	0.08
	NEWS	(24,18)	0.00	1.28	(45,39)	0.01	1.31
		(24,12)	0.00	0.20	(45,33)	-0.01	0.06
		(24,6)	-0.01	0.01	(45,27)	0.00	0.03
CIF	FOREMAN	(24,18)	-0.04	-0.90	(45,39)	0.00	0.24
		(24,12)	-0.08	-1.23	(45,33)	0.00	-0.03
		(24,6)	-0.16	-1.39	(45,27)	0.00	-0.12
	MOBILE	(24,18)	0.00	0.94	(45,39)	0.01	0.66
		(24,12)	-0.01	0.25	(45,33)	0.01	0.22
		(24,6)	-0.05	-0.46	(45,27)	0.00	-0.01
4CIF	CITY	(24,18)	-0.01	-2.05	(45,39)	0.00	-0.01
		(24,12)	-0.35	-4.18	(45,33)	0.00	-0.02
		(24,6)	-0.68	-4.38	(45,27)	0.00	-0.05
	CREW	(24,18)	-0.30	-4.43	(45,39)	0.00	0.23
		(24,12)	-0.40	-3.11	(45,33)	0.00	0.02
		(24,6)	-0.51	-1.98	(45,27)	0.00	-0.05

Table 5-3 Encoding procedures on the hierarchical-B frames at CGS enhancement layers

	JSVM 9.11 [10]	JSVM 8.0 [77]
Step 1	<p>The reconstructed transform coefficients from the base layer: $\mathbf{T}(\epsilon(\mathbf{x}, t))$</p> <p>The predicted signal: Original frame $f(\mathbf{x}, t)$</p> <p>The reference signal: Reference frame $f_E(\mathbf{x}, t^-)$</p>	<p>The reconstructed residual signal (pixel domain) from the base layer: $\epsilon(\mathbf{x}, t)$</p> <p>The predicted signal: $f(\mathbf{x}, t) - \epsilon(\mathbf{x}, t)$</p> <p>The reference signal: Reference frame $f_E(\mathbf{x}, t^-)$</p>
Step 2	Motion estimation to find the best match $f_E(\mathbf{x} - \mathbf{mv}, t^-)$ with the associated MV \mathbf{mv}	
Step 3	<p>Determine the residual signal: $\epsilon_E(\mathbf{x}, t) = f(\mathbf{x}, t) - f_E(\mathbf{x} - \mathbf{d}(\mathbf{x}), t^-)$</p>	<p>Determine the residual signal: $\epsilon_E(\mathbf{x}, t) = [f(\mathbf{x}, t) - \epsilon(\mathbf{x}, t)] - f_E(\mathbf{x} - \mathbf{d}(\mathbf{x}), t^-)$</p>
Step 4	<p>Integer transform: $T_t = \mathbf{T}(\epsilon_E(\mathbf{x}, t)) - \mathbf{T}(\epsilon(\mathbf{x}, t))$</p>	<p>Integer transform: $T_p = \mathbf{T}(\epsilon_E(\mathbf{x}, t))$</p>
⋮	⋮	⋮

Furthermore, the average coding improvement shown in Table 5-2 is slightly greater than that in Table 5-1. 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 5-3. At the enhancement layer, the JSVM 8.0 encoding procedure for inter-layer residual prediction [77] finds the best block match $f_E(\mathbf{x} - \mathbf{mv}, t^-)$ from the reference frame $f_E(\mathbf{x}, t^-)$ for the predicted signal $f(\mathbf{x}, t) - \epsilon(\mathbf{x}, t)$. On the other hand, JSVM 9.11 [10] performs the motion search between $f(\mathbf{x}, t)$ and $f_E(\mathbf{x}, t^-)$, and then subtracts $\mathbf{T}(\epsilon(\mathbf{x}, t))$ in determining the rate-distortion cost. Note that $\mathbf{T}(\cdot)$ indicates the integer transform operation. Thus, in JSVM 9.11 [10], the selected motion vector at the enhancement layer is optimized only for the difference between the current macroblock and its reference macroblock 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 [9], there is no significant coding gain provided by the inter-layer residual prediction for CGS, especially with a low-quality base layer. 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 the base layer is poor.

Section 5.3 Proposed Approaches – Layer-Adaptive Mode Decision and Motion Search

Based on the observations presented in Section 5.2, we develop a fast context-adaptive mode decision algorithm and a motion search scheme for the hierarchical-B frames in the H.264/SVC [2] with combined coarse-grain quality scalability (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 rate-distortion performance at the enhancement layer. Moreover, we skip the coding modes that are not used often and that have a little contribution to the rate-distortion performance. Our algorithms are described by the flowcharts in Fig. 5-7–Fig. 5-9. The base layer is encoded with the exhaustive search (or a fast search 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 enhancement layers are defined by Table 4-1 and Table 5-5–Table 5-7 sorted by the Qp values and the base-layer coding modes. Furthermore, the algorithm in Fig. 5-11 determines the reference frames for motion search and, depending on the partition sizes; the initial point is also adaptively generated by the motion vectors at the base layer or the motion vector predictor at the enhancement layer. These procedures are described in the following sub-sections.

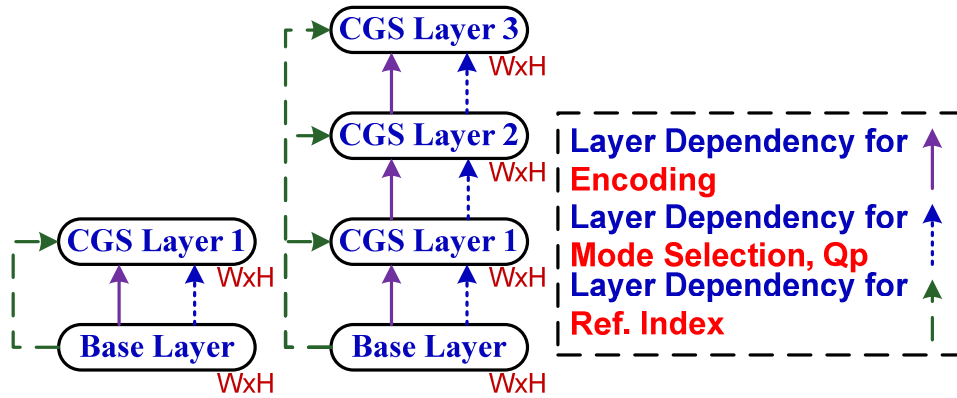


Fig. 5-7 Inter-layer dependency structure in our scheme: (a) two-layer case, and (b) four-layer case

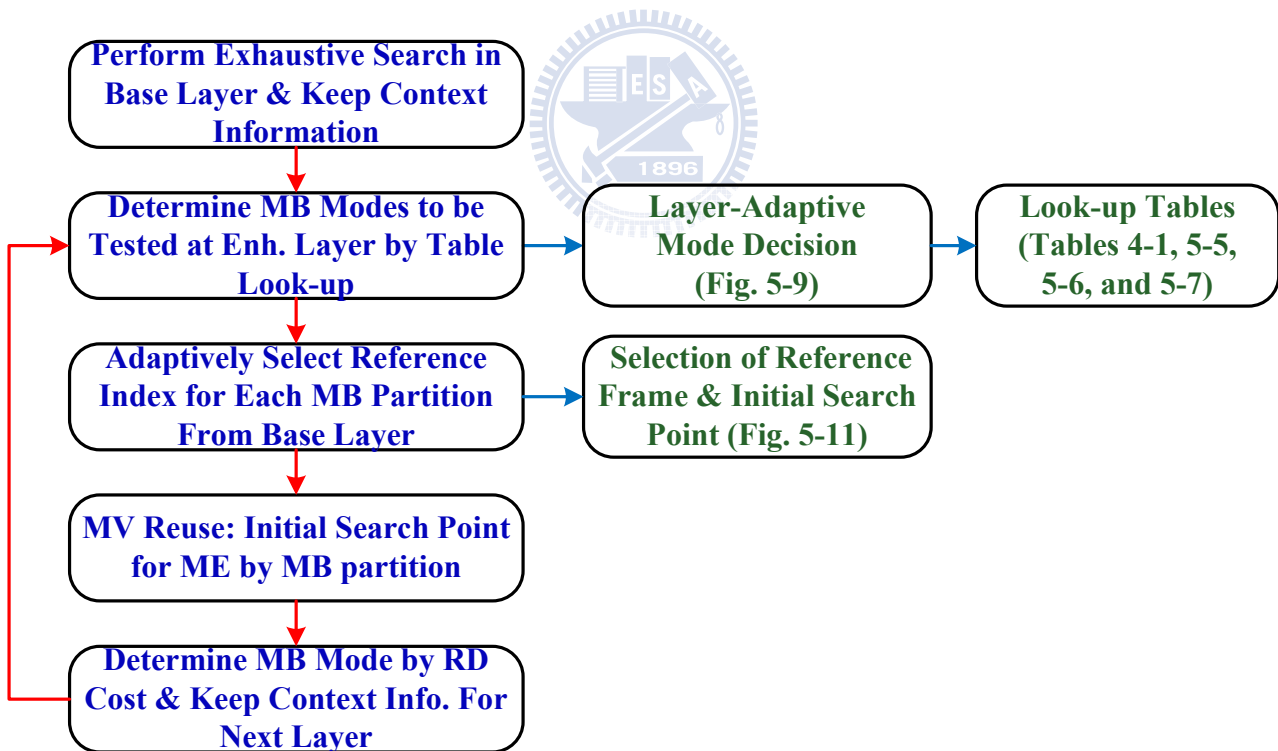


Fig. 5-8 Flowchart of the proposed inter mode decision algorithm for CGS enhancement layers

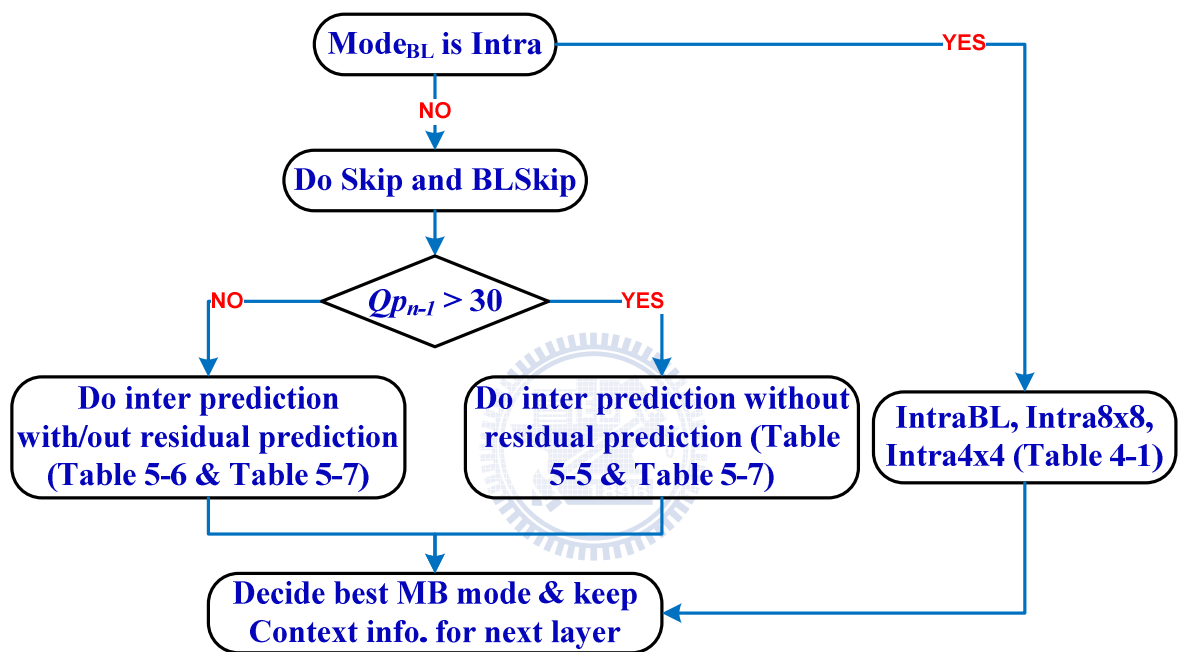


Fig. 5-9 Layer-adaptive mode set selection

Table 5-4 Coding type agreement between base layer and enhancement layer in hierarchical-B frames

Coding Types $Qp_B = 40$	Average Probability at EL, $Qp_E = 20 \sim 40$					
	AKIYO	STEFAN	FOREMAN	MOBILE	CITY	CREW
Mode _{BL} = Mode _{EL} = Intra	> 0.99	0.95	> 0.99	> 0.99	0.92	0.99
Mode _{BL} = Mode _{EL} = Inter	> 0.99	> 0.99	> 0.99	> 0.99	> 0.99	> 0.99

Table 5-5 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	16x16	16x8	8x16	8x8	B_Direct/ Skip	16x16	16x8	8x16	8x8
16x16	○	○	○	○	○	○	○	○	○	○
16x8	○	○	○					○		
8x16	○	○		○					○	
8x8					○					○

Table 5-6 Candidate modes of inter prediction for $Qp_{n-1} \leq 30$

Temporal layer	$T_1 \sim T_N$				
	B_Direct/Skip	16x16	16x8	8x16	8x8
16x16		○			
16x8			○		
8x16				○	
8x8					○

Table 5-7 Candidate modes of sub-MB of inter prediction for all Qp values

Temporal layer	$T_1 \sim T_N$
B_Direct8x8	○
8x8	○
8x4	
4x8	
4x4	

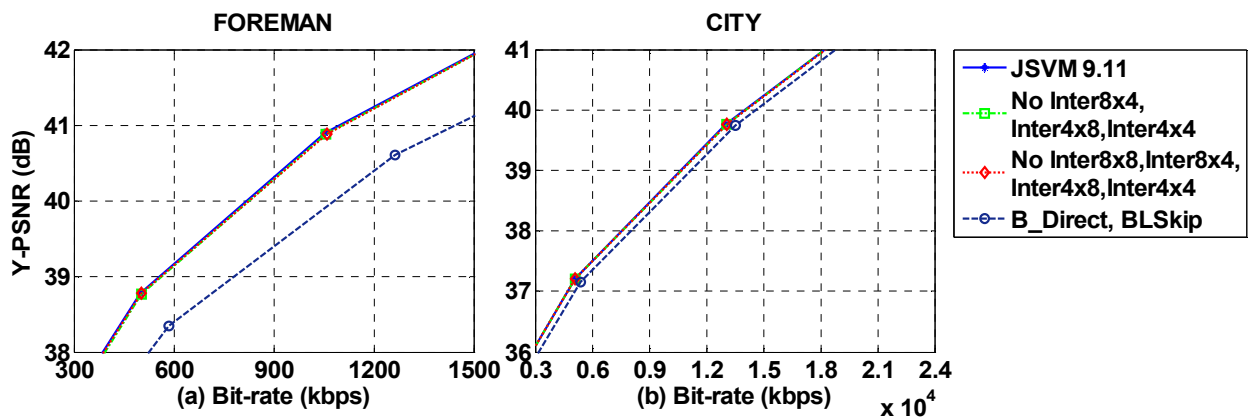


Fig. 5-10 Comparison of rate-distortion performance of JSVM 9.11 [10] at enhancement layers

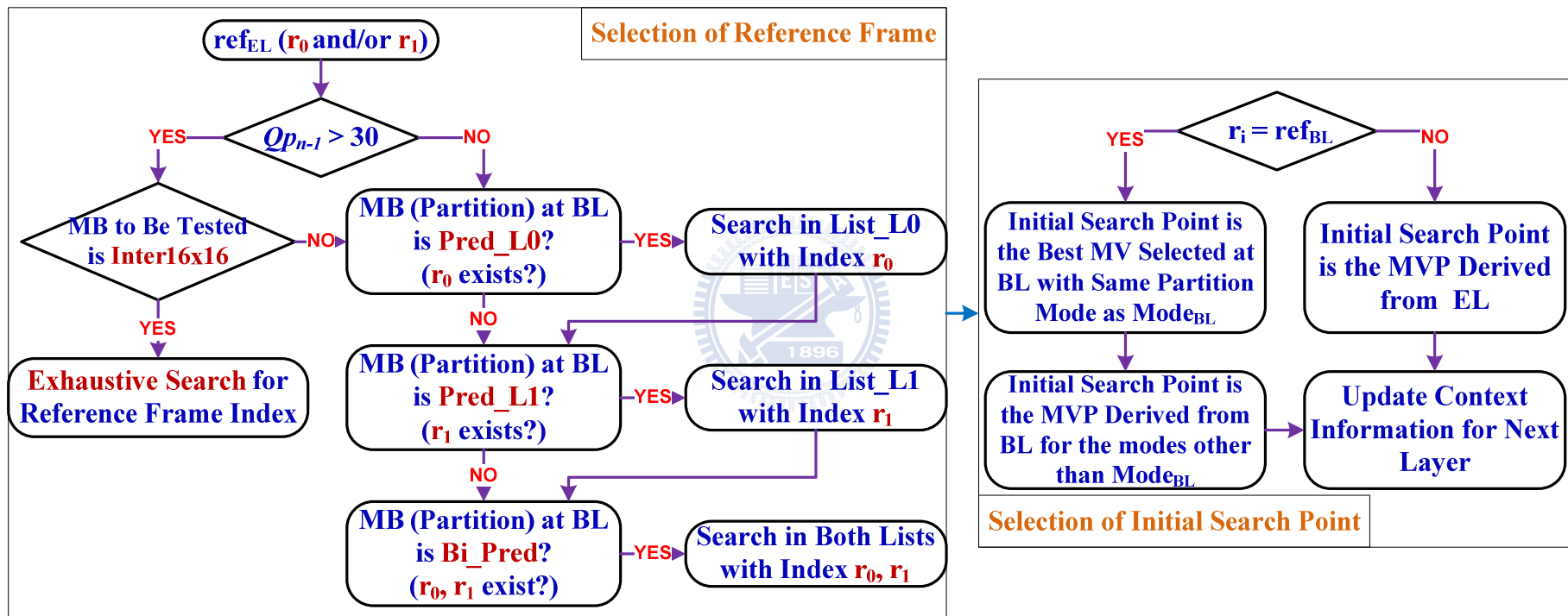


Fig. 5-11 Layer-adaptive selection in reference frame index and initial search point for hierarchical-B frames

5.3.1 Layer-Adaptive Mode Decision for Hierarchical-B Frames

5.3.1.1 Intra Mode Selection

Due to the strong correlation between the coding layers in the Intra4x4 and Intra8x8 modes, the enhancement-layer macroblock skips the less probable prediction modes by checking their reference/base layer coding states. As suggested by our statistical analysis, 75% or more Intra4x4/8x8 coding blocks choose prediction modes *similar* to their counterparts at the reference/base layer. This strong correlation is used to design Table 4-1 for the layer-adaptive intra mode selection. As shown, each macroblock at the enhancement layer 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 base-layer macroblock 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 enhancement-layer Inter/Intra coding types at different Qp_E values in Table 5-4. Obviously, the enhancement-layer macroblock most likely has the same coding type as the BL counterpart. Thus, Fig. 5-9 suggests that an enhancement-layer macroblock only needs to evaluate the intra/inter modes when the base-layer macroblock ($Mode_{BL}$) is also intra/inter-coded.

5.3.1.2 Inter Mode Selection

To achieve greater savings of coding time while minimizing the coding efficiency loss, the layer-adaptive inter candidate mode sets in Table 5-5–Table 5-7 are designed by examining the inter-layer correlation. We consider both the mode conditional distribution, as shown in Fig. 5-4, and the rate-distortion performance, shown in Fig. 5-10.

Based on the statistical data, the less effective macroblock modes that do not contribute much to the coding efficiency are neglected. As mentioned earlier, the best mode (size) at the enhancement layer is likely equal to or larger than that at the reference/base layer. Although the enhancement-layer 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, motion vectors, and reference frame indices. Fig. 5-10 shows that the enhancement-layer coding performance loss is negligible when small-size partitions are disabled, especially for the Inter8x8 mode and those smaller than the 8x8 size. Therefore, the single-layer Advanced Video Standard (AVS) [90] supports only four types of block sizes ranging from 16x16 down to 8x8. In the literature [90], it reports that smaller partitions of the H.264/AVC standard [4] are seldom used, especially in coding high resolution videos. Thus, the coding modes smaller than 8x8 can be skipped at enhancement-layer coding for complexity reduction, as confirmed by the experimental data.

In Table 5-5 and Table 5-6, the proposed algorithm evaluates the Inter8x8 mode only when Mode_{BL} is coded as the Inter8x8 mode. In contrast, for larger partitions, the same and larger

partition sizes, the B_Direct/Skip and Inter16x16 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_Direct8x8 and Inter8x8 modes because the conditional probabilities of the finer partition modes, Inter8x4, Inter4x8, and Inter4x4, are usually less than 0.2. The inter modes with sizes smaller than 8x8 require a high computational complexity, but they provide a very limited coding improvement. Also, statistically they are seldom used at enhancement layers. We thus skip these three coding modes for the enhancement-layer macroblocks, tabulated in Table 5-7.

In addition, we always check the B_Direct/Skip and BLSkip modes at enhancement layers. These two modes provide a significant rate-distortion improvement but only introduce a slight computational load due to their derived motion vectors (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 data in Table 5-1. Similar results can be found in [81]. Moreover, as suggested by the statistical analysis, when a base-layer macroblock is coded with the 16x8 (8x16) partition size, its counterpart at the enhancement layer will not be evaluated with the partition of 8x16 (16x8).

Based on our collected data, we always check the Inter16x16 mode with only one exception when the base-layer macroblock is coded with high quality and $\text{Mode}_{\text{BL}} = \text{B_Direct/Skip}$. For a base-layer macroblock coded with the 8x8 partition, its enhancement-layer counterpart is not tested with any modes (other than 16x16) having a partition size larger than 8x8. If the base-layer

macroblock has a coding mode size larger than 8x8, such as 16x8, 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 base-layer macroblock 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 a macroblock at the reference/base layer is coded with Inter16x16, Inter16x8, Inter8x16, or Inter8x8, only the mode with the same partition is checked. Although Fig. 5-4 indicates that such a design may not be optimal in terms of the mode distribution, the experimental results in Fig. 5-10 show that replacing the 8x8 partition with larger partitions has negligible impact on the coding efficiency, especially when the enhancement layer is coded with high quality.

5.3.2 Layer-Adaptive Reference Frame and Motion Reuse

Similar to the mode selection, the layer-adaptive motion search described by Fig. 5-11 is designed to avoid evaluating four types of motion searches (ME_R , ME_M , ME_{R+M} , and ME_O) at enhancement layers. We check only a selected sub-set of the reference frames and make use of the base-layer motion information. Our scheme is composed of two sequential steps: firstly, select the reference frame candidate indices ref_{EL} , and secondly, determine the motion vector starting point according to the information of ref_{EL} and ref_{BL} (defined later).

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

For each base-layer macroblock, 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 Inter16x8, two 16x8 blocks in a macroblock 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 16x8 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 these data are not the base-layer 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 enhancement-layer macroblocks and are denoted as $\text{kept_ref}_{\text{BL}}$. For instance, the best Inter16x16 is BI prediction with reference indices r'_0 and r'_1 . Then, $\text{kept_ref}_{\text{BL}}$ contains r'_0 and r'_1 , even if Mode_{BL} is Inter16x8. This is because the Inter16x16 may become a candidate mode at the enhancement layer. For the convenience in notation, r'_0 and r'_1 in this example are denoted as r_0 and r_1 in Fig. 5-11.

As discussed before, the best temporal prediction of each (sub-)macroblock at the enhancement layer is highly correlated with that of its base-layer counterpart. This high correlation suggests that the set of ref_{BL} or $\text{kept_ref}_{\text{BL}}$ is sufficient to be the enhancement-layer candidate set. The enhancement-layer reference frame candidate set, ref_{EL} , may take either ref_{BL} or $\text{kept_ref}_{\text{BL}}$ depending on whether or not the evaluated enhancement-layer mode is the same as Mode_{BL} . That is, ref_{EL} in Fig. 5-11(a) can be either ref_{BL} or $\text{kept_ref}_{\text{BL}}$, except when an enhancement-layer macroblock is checked with the Inter16x16 mode and the base layer is of low

quality (i.e., $Qp > 30$). For example, if Mode_{BL} is the Inter16x8 mode, the enhancement-layer candidate mode set is specified by Table 5-5-Table 5-7. Thus, if an enhancement-layer macroblock is evaluated using the Inter16x8 mode, ref_{EL} takes ref_{BL} as its reference frame indices. Otherwise, $\text{kept_ref}_{\text{BL}}$ becomes the reference frame index set, ref_{EL} . To ensure a good interframe prediction performance, the enhancement layer should perform the forward prediction with index r_0 , the backward prediction with index r_1 , and the BI prediction with both indices r_0 and r_1 if ref_{EL} contains r_0 and r_1 .

An exception is that when the base layer is coded at low bit-rate and the base-layer macroblock chooses the partition size of 16x16, 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 Inter16x16 mode may not be reusable for the enhancement-layer macroblock. In this case, the exhaustive search on the reference frames is thus performed for the Inter16x16 prediction mode. In Fig. 5-11(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.

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

- If the reference frame index set ref_{EL} (ref_{BL} or $\text{kept_ref}_{\text{BL}}$) equals to ref_{BL} , the enhancement-layer motion estimation operation does ME_{R} and ME_{O} . Additionally, the

inter-layer prediction performs motion estimation with the motion vector predictor derived from the base layer (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 prediction if the reference frame index set includes only one of r_0 and r_1 .

- Otherwise, the enhancement-layer 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 .

Step 2: Determination of Initial Search Point

After narrowing down the reference frame candidates, we also consider reducing the number of search points in motion estimation. As discussed earlier, the BLSkip mode is the most probable mode when the partition size of $Mode_{BL}$ is smaller than 16×16 . It means that the motion vectors selected from the base layer is reliable and reusable when the enhancement layer checks the same mode (as in $Mode_{BL}$). Moreover, in our previous work [81], we found that the motion vectors of the base layer and enhancement layers are largely correlated. We also reported that the base-layer motion vector would provide a better prediction when the macroblock partition size is greater than 8×8 . More specifically, we compare the initial search points derived by using the base-layer motion vectors and the enhancement-layer motion vector predictor. We examine the motion vector

difference between the initial search point and the final motion vector. The statistics show that the motion vector difference using the base-layer-derived initial search point is, on average, one pixel less than that using the enhancement-layer motion vector predictor. Thus, in Fig. 5-11(b) our scheme suggests that the motion search starting point should be the one determined by the base layer when ref_{EL} is equal to ref_{BL} . Otherwise, the enhancement-layer motion vector predictor provides the motion search starting point. Consequently, except for the inter modes smaller than 8×8 , the motion vectors of the other base-layer coding modes should be stored for possibly being used as the enhancement-layer initial search points.

Table 5-8 Testing conditions

Sequence	CIF	CARPHONE(CP), COASTGUARD(CG), CONTAINER(CT), MOTHERDAUGHTER(MD), SUZIE(SZ)
	4CIF	AKIYO(AK), BUS(BU), FOOTBALL(FB), MOBILE(MB), STEFAN(SF)
	HD (720p)	CITY(CT), CREW(CR), HARBOUR(HB), ICE(IC), SOCCER(SC)
Encoder Configuration	Motion search range: ± 32 pixels with $\frac{1}{4}$ -pel accuracy RDO: Enabled; GOP size: 8 Entropy coding: CABAC Number of reference frame in each reference list: 1 Frame encoded: 1 Intra followed by 80 Inter frames	

Section 5.4 Experiments and Discussions

5.4.1 Test Conditions

For performance assessment, we have implemented our proposed algorithms in JSVM 9.11 [10] 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 enhancement layers. The dyadic hierarchical prediction structure is enabled for the temporal scalability and the CGS enhancement layers are created on top of the base layer 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. 5-7, which depicts the two-layer and four-layer cases. The detailed encoder parameters are given in Table 5-8.

In all simulations, we follow the common practice in setting up the Qp values. According to [91], the accumulated bit-rate of the enhancement layers and the base layer together should be within three times of the base-layer 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% [92]. 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 [10] is from 28 to 40. Therefore, we set Qp_B value to either 30 or 40 in our experiments.

5.4.2 Performance Measures

To measure the speedup performance, we define “time saving (TS)” for the whole encoding process and “enhancement-layer time saving (TS_E)” for coding the enhancement layers only.

(1) The *overall time saving* TS is defined as $TS = \frac{T_{\text{JSVM9.11}} - T_{\text{Proposed}}}{T_{\text{JSVM9.11}}} \times 100\%$, where

$T_{\text{JSVM9.11}}$ and T_{Proposed} denotes the encoding time of JSVM 9.11 [10] and that of our schemes, respectively.

(2) The *enhancement-layer time saving* TS_E is defined as $TS_E = \frac{T_{\text{JSVM9.11}} - T_{\text{Proposed}}}{T_{\text{JSVM9.11}} - T_{\text{BL}}} \times 100\%$,

where T_{BL} is the base-layer encoding time.

To show the change in rate-distortion performance, we adopt the Bjontegaard metric [76] to measure the averaged Y-PSNR [BDP (dB)] and bit-rate differences [BDR (%)] between the rate-distortion curves produced by JSVM 9.11 [10] and by our schemes, respectively. Because the computation of BDP and BDR requires at least four rate-distortion 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 [ΔP (dB)] and bit-rate [ΔR (%)] differences at enhancement layers by the following formulae. In either case, we use $\Delta \text{FileSize}$ (%) to indicate the percentage of the total file size increase.

(1) The *PSNR difference* is defined as $\Delta P = \text{PSNR}_{\text{Proposed}} - \text{PSNR}_{\text{JSVM9.11}}$, where

$\text{PSNR}_{\text{JSVM9.11}}$ and $\text{PSNR}_{\text{Proposed}}$ are the Y-PSNR values obtained by using JSVM 9.11 [10] and our schemes, respectively.

(2) The *bit-rate increase* is defined as $\Delta R = \frac{\text{Bitrate}_{\text{Proposed}} - \text{Bitrate}_{\text{JSVM9.11}}}{\text{Bitrate}_{\text{JSVM9.11}}} \times 100\%$, where

$\text{Bitrate}_{\text{JSVM9.11}}$ and $\text{Bitrate}_{\text{Proposed}}$ correspond to the bit-rate of JSVM 9.11 [10] and that of our schemes, respectively.

5.4.3 Simulation Results

Table 5-9 to Table 5-11 present the time savings of the proposed schemes in comparison with JSVM 9.11. Listed in Table 5-9 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 vs. JSVM 9.11 + MD,

MR/RF Setting: JSVM 9.11 vs. 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 5x), and it gives a higher improvement (up to 90%) in coding enhancement layers. 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.

Table 5-9 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
CT	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

$(Q_{P0}, Q_{PE1}, Q_{PE2}, Q_{PE3}) = (40, 30, 20, 10)$ and $(30, 20, 10, 0)$

Table 5-10 Performance comparisons with Q_p setting of $(Q_{pB}, Q_{pE1}, Q_{pE2}, Q_{pE3}) = (40, 30, 20, 10)$

Sequences	Reference Frame = 1					Reference Frame = 3				
	BDP (dB)	BDR (%)	Δ FileSize (%)	TS (%)	TS _E (%)	BDP (dB)	BDR (%)	Δ FileSize (%)	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
CT	-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
FT	-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 5-11 Performance comparisons with Qp setting of $(Qp_B, Qp_{E1}, Qp_{E2}, Qp_{E3}) = (30, 20, 10, 0)$

Sequences	Reference Frame = 1					Reference Frame = 3				
	BDP (dB)	BDR (%)	Δ FileSize (%)	TS (%)	TS _E (%)	BDP (dB)	BDR (%)	Δ FileSize (%)	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
CT	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

Table 5-12 Average complexity ratio of the base layer to one CGS enhancement layer

Sequences	Six (Q_{PB}, Q_{PE}) settings: $Q_{PB} = 40$ with $Q_{PE} = 30, 20, 10$ and $Q_{PB} = 30$ with $Q_{PE} = 20, 10, 0$			
	Reference frame = 1		Reference frame = 3	
	J SVM 9.11	Proposed	J SVM 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

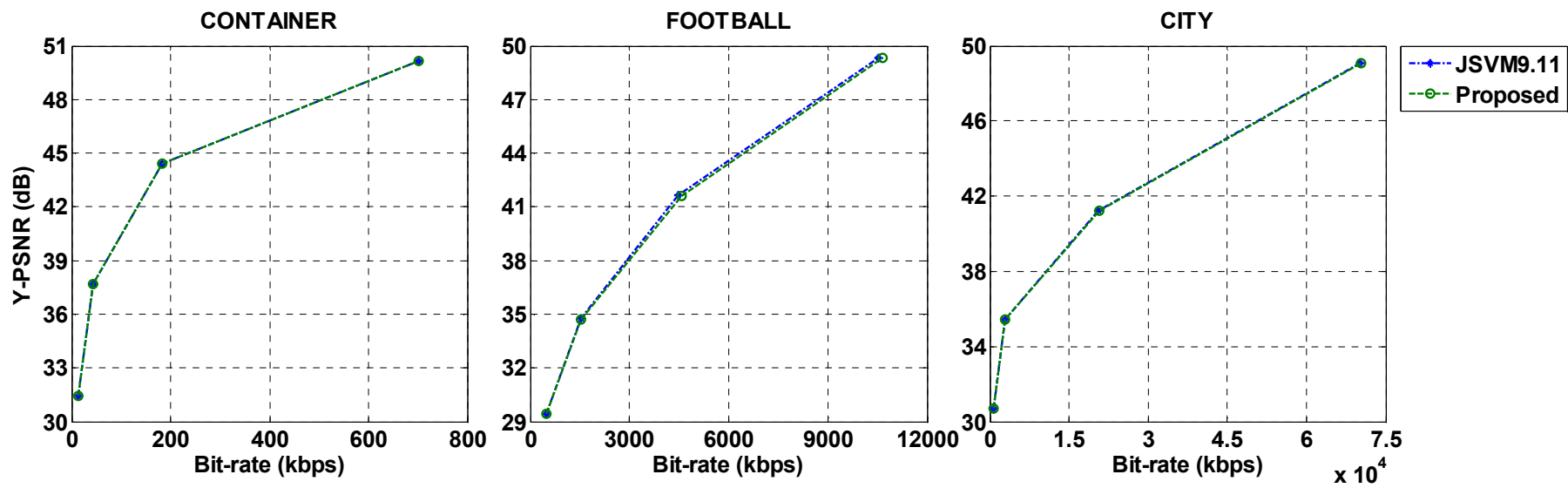


Fig. 5-12 Rate-distortion curves of JSVM 9.11 [10] and our approaches

To see their combined effects, Table 5-10 and Table 5-11 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 Q_p settings: $(Q_{pB}, Q_{pE1}, Q_{pE2}, Q_{pE3}) = (40, 30, 20, 10)$ 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 enhancement layers, where our schemes actually take effect, we can observe an up to 20x 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 rate-distortion curves in Fig. 5-12. 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 frames cases. The result is attributed to the fact that the motion 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 base layer to coding an enhancement layer in Table 5-12. We can see that the running time for coding an enhancement layer is about 3.24 times that for coding its base layer when only one reference frame is in use. In the four-layer case (i.e., one base layer + three enhancement layers), the enhancement-layer encoding time represents 90.7% of the overall computation time. From Table 5-9, 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 ($\sim 10.7\%$) is obtained for the case with three reference frames. According to Amdahl's law and the average TS_E in Table 5-9, 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.

Another interesting fact to be noted is that with our schemes the latency for coding three enhancement layers is almost the same as that for coding one base layer 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%.

$$\left\{ \begin{array}{l} \frac{2^{N-2} + 2^{N-1}}{1 + 2^0 + 2^1 + \dots + 2^{N-1}} = \frac{2^{N-2} + 2^{N-1}}{2^N} = 75\% \quad , N \geq 2 \\ 50\% \quad , N = 1 \end{array} \right.$$

where the GOP size is 2^N .

5.4.4 Performance Comparison with State-of-the-art Fast Algorithms

In addition to the exhaustive search, we also compare our approaches with the state-of-the-art fast algorithms, Li's methods [80][88] and Ren's method [89], 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 Table 5-13–Table 5-15, our methods can achieve a higher time saving (7-41% more) in comparison with [80][88] and [89] 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 [89] has a wide range from 28.6% to 55.6% but Li’s [80][88] 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 [80][88]. This is because the base-layer coding time is fixed in our study and it becomes the dominant portion of the overall running time when 90% of the enhancement-layer calculations are removed. According to Table 5-12, the enhancement-layer 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 ([80] and [88]) and $76.4\% - 67\% = 9.4\%$ with ours (see Table 5-13 and Table 5-14). Therefore, if we consider the enhancement-layer speed-up only, which is our focus; our schemes actually have a relative improvement of $\frac{27.4\% - 9.4\%}{27.4\%} = 65.7\%$ over the Li’s methods.

Table 5-13 Performance comparisons with Li's method [80]

Sequences	(Q_{pB}, Q_{pE})	Li's method [80]			Proposed		
		ΔP (dB)	ΔR (%)	TS (%)	ΔP (dB)	ΔR (%)	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 5-14 Performance comparisons with Li's methods [80] and [88]

Sequences	(Q_{pB}, Q_{pE})	Li's method [80]			Li's method [88]			Proposed		
		ΔP (dB)	ΔR (%)	TS (%)	ΔP (dB)	ΔR (%)	TS (%)	ΔP (dB)	ΔR (%)	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
FOOTBAL L	(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 5-15 Performance comparisons with Ren's method [89]

Sequences	Ren's method [89]			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; $Q_{pB} = 22, 27, 32, 37$ and $Q_{pE} = 19, 24, 29, 34$

Chapter 6 Conclusions and Future Work

In this chapter, we summarize the main idea of our proposed algorithms and review their separate time reduction performance at a similar level of coded quality of the JSVM 9.11 [10]. Moreover, we suggest several items for further study.

Section 6.1 Concluding Remarks

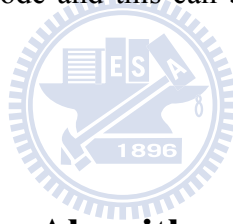
6.1.1 Fast Bi-directional Prediction Selection in H.264/AVC Temporal

Scalable Video Coding

In the Chapter 3, we propose an effective temporal prediction type selection algorithm for the dyadic hierarchical prediction structure in H.264/SVC [2], 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 (1) conditional elimination of BI for large partitions, (2) adaptive thresholds produced by the information obtained in the FW and BW processes, and (3) 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 them are used to decide the execution of BI estimation. To construct a reliable threshold, we examine the correlations of motion-rate costs and the distortions between the uni-directional 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 demonstrates up to 66.2% complexity reduction for the entire encoding process with negligible changes in coding efficiency, as compared to JSVM 9.11 [10]. And, the extra computations in the hierarchical-B prediction can be reduced by up to 94.9%. Hence, our approach can achieve a similar coding efficiency as JSVM 9.11 [10] but with a much lower computational complexity. Moreover, this fast algorithm reduces the complexity only in the temporal prediction types; it does not skip any mode and this can be applied to all the enhancement coding layers.



6.1.2 Fast Mode Decision Algorithm with Macroblock-Adaptive Rate-Distortion Estimation for Intra-only Scalable Video Coding

In the Chapter 4, we propose a fast intra-mode decision algorithm, which consists of a layer-adaptive intra mode selection scheme and a macroblock-adaptive rate-distortion estimation scheme. The correlations between coding layers are intelligently explored for accelerating the encoding process. When our scheme is compared to JSVM 8 [77], it provides up to 64% overall coding time saving and it reduces the computational complexity up to 69% in encoding the enhancement layers. The coding efficiency loss is negligible. The proposed algorithms are particularly useful for intra-only H.264/SVC [2] applications that have real-time requirement.

6.1.3 Fast Context-adaptive Mode Decision Algorithm for Scalable Video Coding with Combined Coarse-grain Quality Scalability (CGS) and Temporal Scalability

In the Chapter 5, we propose a layer-adaptive intra/inter mode decision algorithm and a motion search scheme for the hierarchical-B frames in H.264/SVC [2] with combined CGS and temporal scalability. We examine the rate-distortion performance contributed by different coding modes at enhancement layers and the conditional probability distributions of intra/inter modes at different temporal layers. Three types of techniques have been newly proposed or extended from the existing proposals. The first technique is to limit the intra prediction candidate modes based on the base-layer intra mode information. The second technique is to eliminate the infrequent inter modes based on the inter-layer mode correlation. These two techniques are implemented by 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 motion estimation 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 enhancement-layer encoding process is remarkably reduced. Compared to the exhaustive-search mode decision algorithm in JSVM 9.11 [10], 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

enhancement layers. And the penalty on rate-distortion performance is negligible. The average bit-rate increase is below 1% and the average Y-PSNR loss is below 0.05dB. Our scheme is up to 41% faster than the existing methods in [80][88] and [89].

Section 6.2 Future Work

Our approaches can match the common encoding setting and eliminate the huge computations in selecting the key coding parameters (such as the temporal prediction type and the block partition mode). However, we do not address the determinations of the subordinate coding parameters that may further reduce the encoding time. Moreover, our proposed schemes may not perform well in some specific encoding scenario. Those inadequate considerations are listed as follows.

- In the encoding process, H.264/AVC [4] and H.264/SVC [2] perform macroblock-wise Lagrangian optimization to determine the optimal coding parameters of the current block. Firstly, the temporal prediction type of each block partition is selected by Eq. (2.19). Then, the best partition mode and the preferable transform size (that is, ABT) are chosen simultaneously. When choosing these two coding parameters, the selection process adopts Eq. (2.21) to obtain the optimal parameters by competing $J_{\mathcal{M}}^{4 \times 4}$ and $J_{\mathcal{M}}^{8 \times 8}$, where $J_{\mathcal{M}}^{N \times N}$ denotes the rate-distortion cost applying the transform size of $N \times N$. Hence, determining these two coding parameters needs to (1) calculate the $N \times N$ transform coding, quantization, and the entropy coding to get the actual bits, and (2) perform the reconstruction process so that the distortion (SSD) can be computed. If we can make

decision on the transform size in advance, the computations of $J_{\mathcal{M}}^{N \times N}$ is reduced by half.

- Although our algorithms are specifically designed for the combined CGS and dyadic temporal scalability, they can also be used for the spatial and the non-dyadic temporal scalability. However, these tools must be adjusted properly to fit into the special scalability structure. For instance, two important issues need to be addressed for the spatial scalability: (1) the change in statistical data collection due to the multiple-to-one macroblock mapping from a spatial enhancement layer to its base layer and (2) the aliasing effect due to the interpolation of residual and motion signals. The former may decrease the dependency of the enhancement-layer coding mode/type on its base-layer counterpart, and the latter may affect the probability of the base-layer motion parameters. In general, the extension 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.

Appendix

Distribution of the Approximated Distortion $\tilde{\mathcal{D}}_{\text{BI}}$

In Subsection 3.2.3.3, we have derived the upper bound of $\tilde{\mathcal{D}}_{\text{BI}}$, as shown in Eq. (A.1).

$$\begin{aligned}\tilde{\mathcal{D}}_{\text{BI}} &= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left| f(x, y) - \frac{1}{2} \sum_{\mathcal{I} \in \Omega} f'_{\mathcal{I}}(x - mv_{\mathcal{I}}^x, y - mv_{\mathcal{I}}^y) \right| \\ &= \frac{1}{2} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |e_{\text{FW}}(x, y) + e_{\text{BW}}(x, y)|\end{aligned}\tag{A.1}$$

In addition, we further study its probability distribution in this appendix.

First, we try to find out the distribution of the prediction error $e_{\mathcal{I}}$. The probability distribution of the transform coefficients in the image and/or video coding system have been investigated in the literature [93]–[99]. The earlier studies [93]–[96] found that the transform coefficients of images or video prediction residuals have Laplacian distribution as identified by the goodness-of-fit tests. Later, Lam and Goodman [97] analytically derived this model. A few other research reports [98][99] used more complex probability density functions, such the generalized Gaussian distribution and the mixture of several probability density functions, to improve the modeling accuracy, but these complicated distributions are not widely used in practice because they are difficult for mathematical analysis. Generally, the Laplacian distribution is the most popular one for practical use.

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 [100]. Hence, we assume that the e_{FW} 's inside a block have the i.i.d. Laplacian distribution, and so do the e_{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 (\mathbf{x}, \mathbf{y}) , we like to show that the $e_{\text{FW}}(\mathbf{x}, \mathbf{y})$ and $e_{\text{BW}}(\mathbf{x}, \mathbf{y})$ 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 two popular goodness-of-fit tests to examine the distribution of $(e_{\text{FW}}, e_{\text{BW}})$. They are the Kolmogorov-Smirnov test (KS-test) and the Pearson's χ^2 -test.

Kolmogorov-Smirnov (KS) test: The one-sample KS-test is a non-parametric test, which compares the empirical cumulative probability function (ECDF) with the given model CDF. The KS statistic \mathcal{K} defined in [101] and by Eq. (A.2) quantifies a distance between the ECDF of the data sample and the candidate CDF.

$$\mathcal{K} = \max_{\mathbf{x}} |\text{CDF}_{\text{modeled}}(\mathbf{x}) - \text{ECDF}(\mathbf{x})| \quad (\text{A.2})$$

Moreover, the KS statistic \mathcal{K} ranges from 0 to 1.

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 [102]

$$\mathcal{Q} = \sum_{i=1}^K \frac{(O_i - mp_i)^2}{mp_i} \quad (\text{A.3})$$

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

$$\mathcal{G}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{2\pi |\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) \quad (\text{A.4})$$

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 $\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}$, 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 [103]

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\pi |\boldsymbol{\Sigma}|^{1/2}} K_0\left(\sqrt{2(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})}\right) \quad (\text{A.5})$$

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

In the data fitting process, we need to decide two parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ by adopting the approach of method of moments [104][105]. 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 $(e_{\text{FW}}, e_{\text{BW}})$.

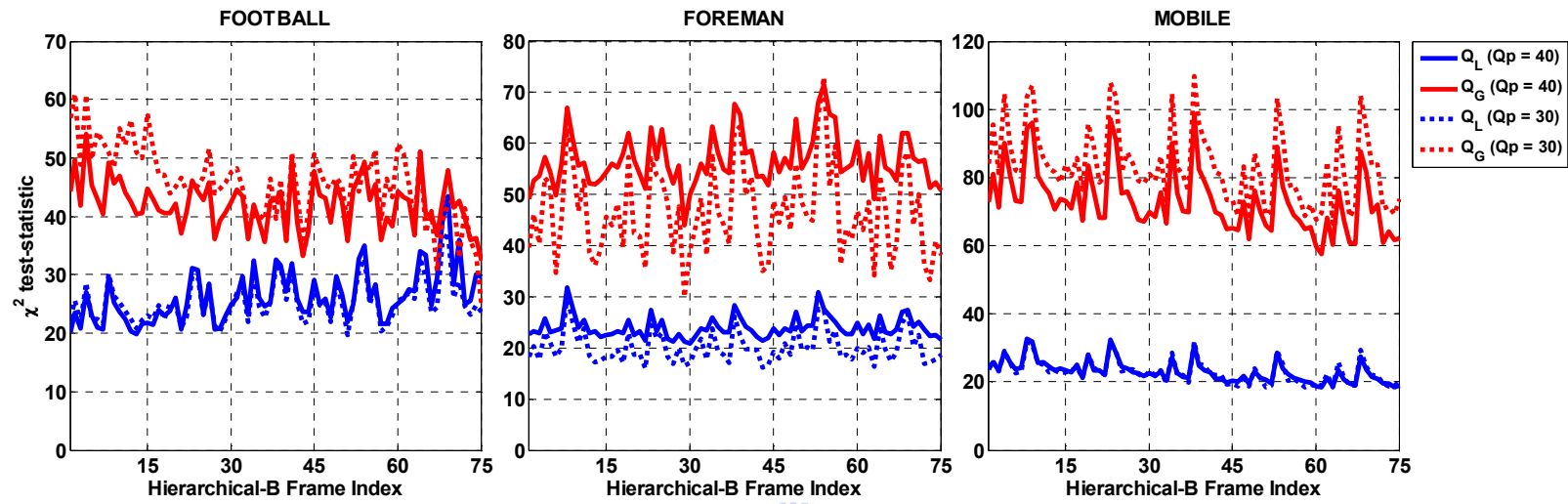
Table A-1. The average Kolmogorov-Smirnov test-statistic values for temporal enhancement layer

Test Sequence	Qp	$\mathcal{K}_{\mathcal{L},16 \times 16}$		$\mathcal{K}_{\mathcal{G},16 \times 16}$		$\mathcal{K}_{\mathcal{L},8 \times 8}$		$\mathcal{K}_{\mathcal{G},8 \times 8}$	
		T_1	T_4	T_1	T_4	T_1	T_4	T_1	T_4
FOOTBALL	40	.083	.074	.098	.093	.075	.070	.096	.091
FOREMAN		.080	.073	.116	.108	.075	.074	.114	.110
MOBILE		.081	.072	.127	.109	.077	.071	.124	.111
FOOTBALL	30	.081	.073	.095	.095	.070	.065	.096	.091
FOREMAN		.075	.065	.108	.092	.068	.062	.104	.089
MOBILE		.080	.069	.136	.116	.075	.067	.131	.115
AVG.		0.076		0.108		0.071		0.106	

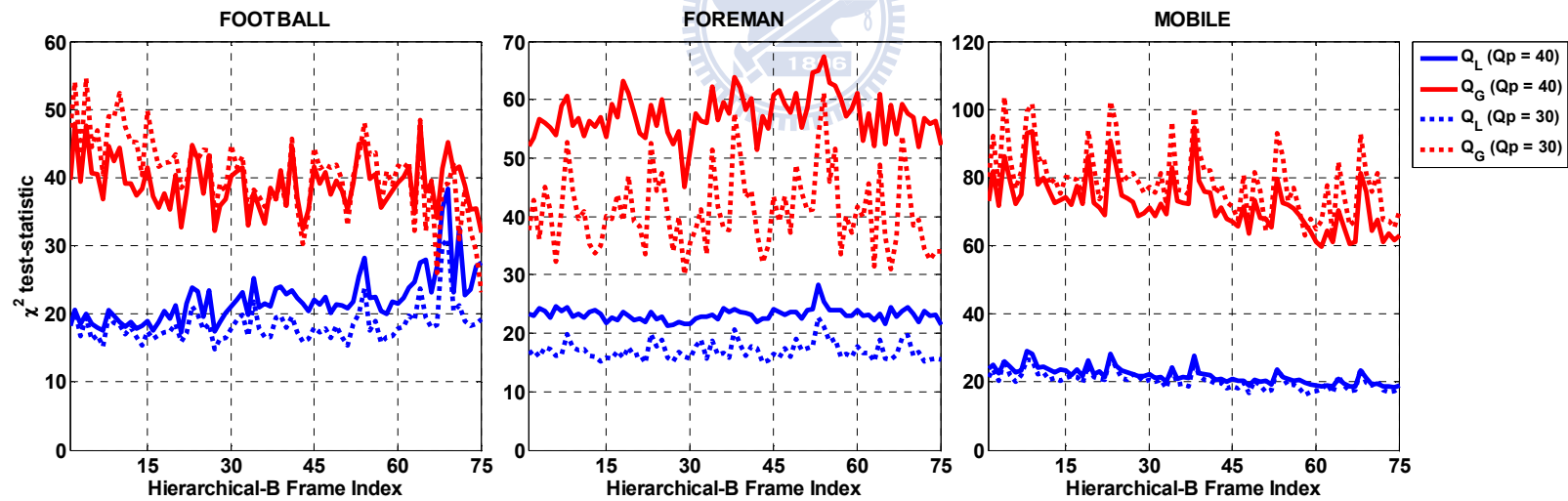


Table A-2. The average χ^2 test-statistic value for temporal enhancement layer

Test Sequence	Qp	$\mathcal{Q}_{\mathcal{L},16 \times 16}$		$\mathcal{Q}_{\mathcal{G},16 \times 16}$		$\mathcal{Q}_{\mathcal{L},8 \times 8}$		$\mathcal{Q}_{\mathcal{G},8 \times 8}$	
		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	



(a) 16x16 block mode



(b) 8x8 block mode

Fig. A-1 The average χ^2 test-statistic value for individual hierarchical-B frame

We examine these two goodness-of-fit tests in two distinct block sizes, 16x16 and 8x8. The empirical data are evaluated against these two selected model distributions. In Table A-1, the reference model \mathcal{L} is better than the other model \mathcal{G} in terms of the KS test-statistic value \mathcal{K} . The \mathcal{K} value of \mathcal{L} usually varies from 0.062 to 0.083, but the \mathcal{K} value \mathcal{G} is about 0.11 on average. The χ^2 -test in Table A-2 show similar results, in which averagely $\mathcal{Q}_{\mathcal{L}}$ is much smaller than $\mathcal{Q}_{\mathcal{G}}$. We thus conclude that the collected data $(e_{\text{FW}}, e_{\text{BW}})$ are closer to the bivariate Laplacian distribution. Furthermore, Fig. A-1 shows the χ^2 test-statistic value of each hierarchical-B frame. The $\mathcal{Q}_{\mathcal{L}}$ value typically ranges from 20 to 40, while the $\mathcal{Q}_{\mathcal{G}}$ value is often more than 40. For the MOBILE, it can be up to 100. Moreover, small partitions usually match the mode distributions better. The test metrics in the small 8x8 partition are slightly smaller than that in the large 16x16 block size.

After we use the jointly Laplacian distribution to model $(e_{\text{FW}}, e_{\text{BW}})$ is, we can derive the distribution of $\frac{1}{2} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |e_{\text{FW}}(x, y) + e_{\text{BW}}(x, y)|$; namely, the distribution of $\tilde{\mathcal{D}}_{\text{BI}}$. According to the property of Laplacian distribution [103], 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

$$e_{\text{FW}+\text{BW}}(x, y) \triangleq e_{\text{FW}}(x, y) + e_{\text{BW}}(x, y) \quad (\text{A.6})$$

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 [106][107].

$$\frac{1}{2} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \underbrace{e_{\text{FW}}(x, y) + e_{\text{BW}}(x, y)}_{\text{jointly Laplacian}} = \frac{1}{2} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \underbrace{e_{\text{FW+BW}}(x, y)}_{\text{Laplacian}} \quad (\text{A.7})$$

Gamma

Exponential

Hence, $\tilde{\mathcal{D}}_{\text{BI}}$ should have a Gamma distribution $\Gamma(k, \theta)$, where $k = MN$ is the shape parameter and θ is the scale parameter.



Bibliography (in order of appearance)

- [1] J.-R. Ohm, "Advances in Scalable Video Coding," *Proc. IEEE*, vol. 93, no. 1, pp. 42–56, Jan. 2005.
- [2] 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.
- [3] R. Leonardi, T. Oelbaum, and J.-R. Ohm, "Status Report on Wavelet Video Coding Exploration," *ISO/IEC JTC1/SC29/WG11 MPEG, N8043*, Apr. 2006.
- [4] 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)," *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-G050r1*, Mar. 2003.
- [5] 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.
- [6] H. Schwarz, D. Marpe, T. Wiegand, "Analysis of Hierarchical B Pictures and MCTF," *IEEE Int'l Conf. on Multimedia and Expo*, pp. 1929–1932, 2006.
- [7] 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.
- [8] M. Wien, H. Schwarz, and T. Oelbaum, "Performance Analysis of SVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1194–1203, 2007.
- [9] 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, 2007.
- [10] 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.
- [11] B. Girod, "The Efficiency of Motion-Compensating Prediction for Hybrid Coding of Video

- Sequences,” *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 7, pp. 1140–1154, 1987.
- [12] B. Girod, “Motion-Compensating Prediction with Fractional-Pel Accuracy,” *IEEE Trans. Commun.*, vol. 41, pp. 604–612, 1993.
- [13] T. Wedi, and H. G. Musmann, “Motion- and Aliasing-Compensated Prediction for Hybrid Video Coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 577–586, 2003.
- [14] B. Girod, “Efficiency Analysis of Multihypothesis Motion-Compensated Prediction for Video Coding,” *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 173–183, 2000.
- [15] M. Flierl, T. Wiegand, and B. Girod, “A Video Codec Incorporating Block-Based Multi-Hypothesis Motion-Compensated Prediction”, in *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, vol. 4067, pp. 238–249, 2000.
- [16] M. Flierl, T. Wiegand, and B. Girod, “A Locally Optimal Design Algorithm for Block-Based Multi-Hypothesis Motion-Compensated Prediction,” *Proc. Data Compression Conf.*, pp. 239–248, 1998.
- [17] M. Flierl, T. Wiegand, and B. Girod, “Rate-Constrained Multi-Hypothesis Motion-Compensated Prediction for Video Coding,” *IEEE Int’l Conference on Image Processing*, vol. III, pp. 150–153, 2000.
- [18] M. Flierl and B. Girod, “Multihypothesis Motion Estimation for Video Coding,” *Proc. Data Compression Conf.*, pp. 341–350, 2001.
- [19] M. Flierl, T. Wiegand, and B. Girod, “Rate-Constrained Multihypothesis Prediction for Motion-Compensated Video Compression,” *IEEE Trans. Circuits Systems Video Technol.*, vol. 12, no. 11, pp. 957–969, 2002.
- [20] M. Flierl and B. Girod, “Generalized B pictures and the Draft H.264/AVC Video-Compression Standard,” *IEEE Trans. Circuits Systems Video Technol.*, vol. 13, no. 7, pp. 587–597, 2003.
- [21] N. Ahmed, T. Natarajan, and K.R. Rao, “Discrete Cosine Transform,” *IEEE Trans. Computers*, vol. C-23, no. 1, pp. 90–93, 1974.

- [22] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [23] H. S. Malcar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-Complexity Transform and Quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598–603, 2003.
- [24] B. Gjontegaard, "Addition of 8x8 Transform to H.26L," *ITU-T SG16, Doc. VCEG-Q15-I-39*, Oct. 1999.
- [25] H. S. Malvar, *Signal Processing with Lapped Transforms*. Boston, MA: Artech House, 1992.
- [26] M. Wien, "Variable Block-Size Transforms for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 604–613, 2003.
- [27] P. List, A. Joch, J. Lainema, B. Gjontegaard, and M. Karczewicz, "Adaptive Deblocking Filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, 2003.
- [28] J. Lainema and M. Karczewicz, "Core Experiment Results on Low Complexity Loop Filtering," *ITU-T SG16, Doc. VCEG-M21*, Apr. 2001.
- [29] "Further Improvements on TML Loop Filtering," *ITU-T SG16 Doc. VCEG-M22*, 2001.
- [30] G. Bjøntegaard and I. Lille-Langoy, "Possible Simplifications of the Present Deblocking Filter in TML 5.9," *ITU-T SG16, Doc. VCEG-M30*, Apr. 2001.
- [31] P. List, "Proposal for a Simplification of the H.26L Loop Filter," *ITU-T SG16, Doc. VCEG-M48*, Apr. 2001.
- [32] "Report of the Ad Hoc Committee on Loop Filter Improvement," *ITU-T SG16, Doc. VCEG-N08r1*, Sep. 2001.
- [33] S. Sun and S. Lei, "Improved TML Loop Filter With Lower Complexity," *ITU-T SG16, Doc. VCEG-N17*, Sep. 2001.
- [34] G. Côté, L. Winger, and M. Gallant, "Lower Complexity Deblocking Filter With In-Place Filtering," *ITU-T SG16, Doc. VCEG-O39*, Dec. 2001.
- [35] P. List, "AHG Report: Loop Filter," *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc.*

JVT-B011r2, Jan. 2002.

- [36] J. Au, B. Lin, A. Joch, and F. Kossentini, “Complexity Reduction and Analysis for Deblocking Filter,” *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. JVT-C094*, May 2002.
- [37] A. Joch, “Improved Loop-Filter Tables and Variable-Shift Table Indexing,” *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. JVT-D038*, Jul. 2002.
- [38] A. Joch, “Loop Filter Simplification and Improvement,” *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. JVT-D037*, Jul. 2002.
- [39] C. Gomila and A. Joch, “Simplified Chroma Deblocking (Revisited),” *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. JVT-E089*, Oct. 2002.
- [40] A. MacInnis and S. Zhong, “Corrections to Loop Filter in the Case of MB-AFF,” *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. JVT-F027*, Dec. 2002.
- [41] D. Marpe, H. Schwarz, and T. Wiegand, “Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, 2003.
- [42] J. Rissanen and G. G. Langdon Jr, “Universal Modeling and Coding,” *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 12–23, 1981.
- [43] C. A. Segall and G. J. Sullivan, “Spatial Scalability within the H.264/AVC Scalable Video Coding Extension,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1121–1135, 2007.
- [44] “Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video,” ITU-T and ISO/IEC JTC1, ITU-T Recommendation H.262—ISO/IEC 13 818-2 (MPEG-2), 1994.
- [45] “Video Coding for Low Bitrate Communication Version 1,” ITU-T, ITU-T Recommendation H.263, 1995.
- [46] “Coding of Audio-Visual Objects—Part 2: Visual,” ISO/IEC JTC1, ISO/IEC 14 496-2 (MPEG-4 visual version 1), 1999.
- [47] B. Girod, “Rate-Constrained Motion Estimation,” in *Proc. SPIE Visual Communications and*

Image Processing, vol. 2308, pp. 1026–1034, 1994.

- [48] H. Everett, “Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources,” *Oper. Res.*, vol. 11, pp. 399–417, 1963.
- [49] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, G. J. Sullivan, “Rate-Constrained Coder Control and Comparison of Video Coding Standards,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–702, 2003.
- [50] G. Sullivan and T. Wiegand, “Rate-Distortion Optimization for Video Compression,” *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, 1998.
- [51] T. Wiegand and B. Girod, “Lagrange Multiplier Selection in Hybrid Video Coder Control,” *IEEE Int’l Conf. on Image Processing*, vol.3, pp. 542–545, 2001.
- [52] 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, 2003.
- [53] 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, 1999.
- [54] A. Chang, O. C. Au, and Y. M. Yeung, “A Novel Approach to Fast Multi-Frame Selection for H.264 Video Coding,” *IEEE Int’l Conf. on Acoustics, Speech and Signal Processing*, pp. II-704–II-707, 2003.
- [55] 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.
- [56] 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, 2006.
- [57] M.-J. Chen, G.-L. Li, Y.-Y. Chiang, and C.-T. Hsu, “Fast Multiframe Estimation Algorithms by Motion Vector Composition for the MPEG-4/AVC/H.264 Standard,” *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 478–487, 2006.
- [58] L. Shen, Z. Liu, Z. Zhang, and G. Wang, “An Adaptive and Fast Multiframe Selection

- Algorithm for H.264 Video Coding,” *IEEE Signal Processing Letters*, vol. 14, no. 11, pp. 836–839, 2007.
- [59] 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, 2006.
- [60] 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, 2008.
- [61] 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, 2009.
- [62] 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, 2006.
- [63] 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, 2010.
- [64] 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, 2009.
- [65] 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, 2008.
- [66] 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, 2007.
- [67] 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, 2009.
- [68] 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.

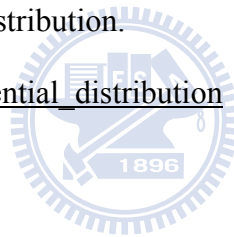
2, pp. 273–279, 2008.

- [69] 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, 2009.
- [70] 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.
- [71] 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, 2009.
- [72] 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, 2006.
- [73] B. Lee *et al.*, “An Efficient Block Mode Decision for Temporal Scalability in Scalable Video Coding,” *Proceedings of SPIE*, vol. 6822, pp. 68220L-1–68220L-9, 2008.
- [74] H.-C. Lin, H.-M. Hang, and W.-H. Peng, “Fast Temporal Prediction Selection for H.264/AVC Scalable Video Coding,” *IEEE Int’l Conf. on Image Processing*, pp. 3425–3428, 2009.
- [75] H.-C. Lin and H.-M. Hang, “Fast Algorithm on Selecting Bi-directional Prediction Type in H.264/AVC Scalable Video Coding,” *IEEE Int’l Symposium on Circuits and Systems*, pp. 113–116, 2010.
- [76] G. Bjontegaard, “Calculation of Average PSNR Differences between RD-curves,” *ITU-T SG 16, Doc. VCEG-M33*, Apr. 2001.
- [77] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz, and M. Wien, “Joint Draft 8 of SVC Amendment,” *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-U201*, Oct. 2006.
- [78] L. Xiong, “Reducing Enhancement Layer Directional Intra Prediction Modes,” *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-P041*, Jul. 2005.
- [79] L. Yang, Y. Chen, J. Zhai, and F. Zhang, “Low Complexity Intra Prediction for Enhancement Layer,” *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-Q084*, Oct. 2005.

- [80] H. Li, Z.-G. Li, and C. Wen, "Fast Mode Decision for Coarse Grain SNR Scalable Video Coding," *IEEE Int'l Conf. on Acoust., Speech, Signal Processing*, pp. II-545–II-548, 2006.
- [81] 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," *IEEE Int'l Conf. on Image Processing*, pp. II-289–II-292, 2007.
- [82] 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," *IEEE Int'l Conf. on Multimedia and Expo*, pp. II-765–768, 2008.
- [83] 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, 2008.
- [84] 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, 2008.
- [85] C. Kim, 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, 2007.
- [86] 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, 2006.
- [87] 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. 127–131, 2008.
- [88] H. Li, Z.-G. Li, C. Wen, and S. Xie, "Fast Mode Decision for Coarse Granular Scalability via Switched Candidate Mode Set," *IEEE Int'l Conf. on Multimedia and Expo*, pp. 1323–1326, 2007.
- [89] J. Ren and N. Kehtarnavaz, "Fast Adaptive Termination Mode Selection in H.264 Scalable

- Video Coding,” *Journal of Real-Time Image Processing*, vol. 4, pp. 13–21, Mar. 2009.
- [90] L. Fan, S. Wa, and F. Wu, “Overview of AVS Video Standard,” *IEEE Int’l Conf. on Multimedia and Expo*, pp. I-423–I-426, 2004.
- [91] 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, 2006.
- [92] 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, 2005.
- [93] R. C. Reininger and J. D. Gibson, “Distributions of the Two-Dimensional DCT Coefficients for Images,” *IEEE Trans. Commun.*, vol. COM-31, no. 6, pp. 835–839, 1983.
- [94] M. Barni, F. Bartolini, A. Piva, and F. Rigacci, “Statistical Modelling of Full Frame DCT Coefficients,” *Proc. Eur. Signal Processing Conf.*, vol. III, pp. 1513–1516, 1998.
- [95] F. Bellifemine, A. Capellino, A. Chimienti, R. Picco, and R. Ponti, “Statistical Analysis of the 2D-DCT Coefficients of the Differential Signal for Images,” *Signal Process. Image Commun.*, vol. 4, pp. 477–488, 1992.
- [96] S. R. Smooth and R. A. Lowe, “Study of DCT Coefficients Distributions,” *Proc. SPIE*, vol. 2657, pp. 403–411, 1996.
- [97] E.Y. Lam and J.W. Goodman, “A Mathematical Analysis of the DCT Coefficient Distributions for Images,” *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1661–1666, 2000.
- [98] F. Muller, “Distribution Shape of Two-Dimensional DCT Coefficients Natural Images,” *Electron. Lett.*, vol. 29, no. 22, pp. 1935–1936, 1993.
- [99] T. Eude, R. Grisel, H. Cherifi, and R. Debrie, “On the Distribution of the DCT Coefficients,” *IEEE Int’l Conf. Acoustics, Speech, Signal Processing*, pp. V-365–V-368, 1994.
- [100] M. Narroschke, “Extending H.264/AVC by an Adaptive Coding of the Prediction Error,” *Proc. of Picture Coding Symposium*, 2006.

- [101] J. Peacock, "Two-dimensional Goodness-of-fit Testing in Astronomy," *Monthly Notices of the Royal Astronomical Society*, vol. 202, pp. 615–627, 1983.
- [102] R. V. Hogg, J. W. McKean, and A. T. Craig, *Introduction to Mathematical Statistics*, 6th ed. Pearson: Prentice Hall, 2004.
- [103] S. Kotz, T. J. Kozubowski, and K. Podgorski, *The Laplace Distribution and Generalizations*. Birkhauser, 2001.
- [104] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, vol. I, New Jersey: Prentice-Hall, 1993.
- [105] T. Sltoft, T. Kim, T.-W. Lee, "On the Multivariate Laplace Distribution," *IEEE Signal Lett.*, vol. 13, no. 5, pp. 300–303, 2006.
- [106] Laplace Distribution. [Online] Available: http://en.wikipedia.org/wiki/Laplace_distribution
- [107] Exponential Distribution. [Online] Available: http://en.wikipedia.org/wiki/Exponential_distribution



Curriculum Vitae

Hung-Chih Lin

CONTACT	Engineering Building IV – 422R	<i>Tel:</i> 03-5712121-54231
INFORMATION	Institute of Electronics	<i>Fax:</i> 03-5731791
	National Chiao Tung University	<i>E-mail:</i> huchlin@gmail.com
	Hsinchu, Taiwan 30010	

RESEARCH INTERESTS	Video Compression, Digital Video Signal Processing, MATLAB [®] Vectorization
--------------------	---

EDUCATION	National Chiao Tung University , Hsinchu, Taiwan Ph. D., Institute of Electronics, July 2010 <ul style="list-style-type: none">- Dissertation: “Fast Encoding Algorithm Design for H.264/MPEG-4 AVC Scalable Video Coding Standard”- Advisor: Hsueh-Ming Hang National Chiao Tung University , Hsinchu, Taiwan B.S., Electrical and Control Engineering, June 2004 <ul style="list-style-type: none">- Ranked 4th Ming-Dao Senior High School , Taichung, Taiwan
-----------	--

ACADEMIC EXPERIENCE	National Chiao Tung University , Hsinchu, Taiwan Ph. D. candidate Sep. 2004 ~ Jun. 2010
---------------------	---

Fast Algorithm Development for H.264/AVC Video Coding Standard

- Fast intra prediction selection on x264 software
- Statistical analysis on inter mode distribution

Fast Mode Decision Algorithm for H.264/AVC Scalable Video Coding Standard

- Fast intra/inter mode decision for coarse-grain quality scalability
- Fast intra-only scalable coder for coarse-grain quality and spatial scalability
- Fast temporal prediction selection for hierarchical prediction
- Fast bi-directional prediction selection for hierarchical prediction
- Fast temporal prediction selection for spatial scalability

Implementation of H.264/AVC Codec on TI DSP Platform

- Software optimization for x264 encoder and decoder
- Software optimization for JSVM reference software

Electrical and Computer Engineering , University of Illinois at Urbana-Champaign
Research Scholar (July 2006)

- Study and implement a multi-processor coding scheme on H.264/AVC coding standard

Electrical and Computer Engineering , University of Illinois at Urbana-Champaign
Short-term Scholar (July 2007)

- Study GPU of Compute Uniform Device Architecture (CUDA)
- Speed up H.264/AVC Intra coder on GPU-based coprocessor

HONORS AND AWARDS Academic Achievement Award and Scholarship (2002 Spring, 2002 Fall, 2003 Spring, 2004 Fall, 2005 Spring)
Ph. D. Student Scholarship, 2005 and 2006 (awarded by EE, NCTU)

PROFESSIONAL ACTIVITY (REVIEWER) Signal Processing: Image Communication (Elsevier)
IEEE Transactions on Image Processing (TIP)
Journal of Visual Communication and Image Representation (JVCIR)
IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)

TEACHING ASSISTANT Electronic Lab. (I) (2005 Fall)
Digital Signal Processing (2006 Spring)
Source Coding (2006 Fall and 2009 Fall)
Information Theory (2007 Fall)
Advanced Digital Signal Processing (2009 Spring)
Multimedia Communications (2010 Spring)



Publication

Journal Paper:

H.-C. Lin, W.-H. Peng, and H.-M. Hang, “Fast Context-adaptive Mode Decision Algorithm for Scalable Video Coding with Combined Coarse-grain Quality Scalability (CGS) and Temporal Scalability,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 5, pp. 732–748, 2010.

International Conference Paper:

H.-C. Lin, Y.-J. Wang, *et al.*, “Algorithms and DSP Implementation of H.264/AVC,” *Asia and South Pacific Design Automation Conference*, Yokohama, Japan, 24-27 Jan., 2006, pp. 742–749.

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. on Image Process.*, San Antonio, Texas, USA, 16-19 Sep., 2007, pp. II-289–II-292.

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*, Hannover, Germany, 23-26 June, 2008, pp. 765–768.

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. on Image Process*, Cairo, Egypt, 7-10 Nov., 2009, pp. 3425–3428.

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.*, Paris, France, 30 May-2 June, 2010, pp. 113–116.

MPEG Standard Contribution:

H.-C. Lin, W.-H. Peng, and H.-M. Hang, “Low-complexity Macroblock Mode Decision Algorithm for Combined CGS and Temporal Scalability,” *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6*, JVT-W029, San Jose, California, USA, 21-27 Apr., 2007.