Computationally efficient and adaptive scalable video coding

Li, He

2012

Li, H. (2012). Computationally efficient and adaptive scalable video coding. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/50647>

<https://doi.org/10.32657/10356/50647>

Computationally Efficient and Adaptive Scalable Video Coding

Li He

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University

in fulfillment of the requirement for the degree of

Doctor of Philosophy

2012

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.....

Date

.....

Li He

To my parents, my husband and my lovely twin sons.

Acknowledgements

First and foremost, I would like to thank my ever supporting parents for all of love and support they have given me through the years, without whom, nothing would have been possible. I would like to dedicate this work to them.

I would also like to express my sincere gratitude to my supervisor, Professor Wen Chang Yun, for his continuous guidance and support. His advice and encouragement have been absolute invaluable to me at every stage of my process through the Ph.D. program. I have benefited greatly from knowing him, not only from his technical insight and teaching excellence, but also from his reliability, integrity and sincerity.

Next, I would like to thank my co-supervisor, Dr. Li Zheng Guo, for his valuable advice and inspirational enthusiasm for my research. His unique insights and perspectives have enriched my research experience. His enthusiasm for the research is a constant source of encouragement throughout my entire candidature. This research work would not have been possible without his endless supports and encouragements.

Finally, I wish to extend my appreciation to my friends in the Nanyang Technological University, for creating such a pleasant environment. The good times we spent together is unforgettable in my memory. In addition, I would like to thank

the technical staffs in the Computer Control Lab for the uncountable assistance.

Contents

Acknowledgements	iv
Summary	x
List of Figures	xii
List of Tables	xviii
1 Introduction	1
1.1 Background	1
1.2 Motivations	4
1.3 Summary of Major Contributions	7
1.4 Thesis Organization	9
2 Overview of H.264/Advanced Video Coding and Scalable Video Coding	11
2.1 Video Coding Standards	11
2.2 Hybrid Video Coding Scheme	13
2.3 H.264/Advanced Video Coding	17
2.4 Scalable Video Coding	23

2.4.1	Spatial Scalability	28
2.4.2	SNR Scalability	30
2.4.3	Temporal Scalability	31
2.4.4	Combined Scalability	31
2.5	Complexity Analysis of H.264/AVC and SVC	33
2.5.1	Rate Distortion Optimization	33
2.5.2	Macroblock Inter-Prediction Modes	35
2.5.3	Macroblock Intra-Prediction Modes	36
2.5.4	Multiple Reference Frames	39
2.5.5	Quarter-pixel Precision for Motion Estimation and Compensation	40
2.5.6	Scalabilities	43
3	Fast Mode Decision Schemes for Inter-Frame Coding in Fully Scalable Video Coding	45
3.1	Introduction	45
3.2	Spatial Scalable Video Coding	49
3.2.1	Inter-Layer Prediction	52
3.2.2	Correlations Between the Base Layer and Enhancement Layers	56
3.2.3	Proposed Fast Mode Decision Scheme for Spatial Scalability	60
3.3	SNR Scalable Video Coding	62
3.3.1	Coarse Grain Scalability	62
3.3.2	Correlation Between the Base Layer and Enhancement Layers	63
3.3.3	Proposed Fast Mode Decision Scheme 1 for SNR Scalability	67
3.3.4	Enhanced Fast Mode Decision Scheme 2 for SNR Scalability	71
3.4	Temporal Scalable Video Coding	73
3.4.1	Prediction Structure with Hierarchical B-pictures	75

3.4.2	Correlation Between the Base Layer and Enhancement Layers	78
3.4.3	Proposed Fast Mode Decision Scheme for Temporal Scalability	79
3.5	Experimental Results	84
3.5.1	Spatial Scalability	87
3.5.2	SNR Scalability	90
3.5.3	Temporal Scalability	97
3.5.4	Spatial and SNR Scalability	100
3.5.5	Spatial and Temporal Scalability	104
3.5.6	Experimental Comparisons of Fast Mode Decision Algorithms	106
3.6	Conclusions	115
4	Adaptive Decoder Complexity Reduction for Coarse Grain Scalability	121
4.1	Introduction	121
4.2	Rate-Distortion-Complexity Optimization	126
4.2.1	Problem Formulation	126
4.2.2	Adaptive Choices of λ_{MODE} and λ_{MOTION}	128
4.3	Interpolation Complexity Modelling	130
4.4	Experimental Results	132
4.5	Conclusions	134
5	Adaptive SNR Enhancement Layer Motion Estimation and Motion Compensation Schemes	136
5.1	Introduction	136
5.2	Base Layer Motion Estimation and Mode Decision Process	139
5.3	Proposed Enhancement Layer Motion Estimation and Motion Compensation Schemes	144

5.3.1	Proposed Enhancement Layer Refinement Scheme A	145
5.3.2	Proposed Enhancement Layer Refinement Scheme B	148
5.4	Experimental Results	153
5.4.1	Comparison of SNR Scalability for 4 Schemes	153
5.4.2	SNR Scalable Coding vs Non-scalable Coding	157
5.4.3	Layer Separation Characteristic	158
5.4.4	Actual Bit Rate	160
5.4.5	Three Layers SNR Scalable Coder	160
5.5	Conclusions	163
6	Enhanced Intra Coding in Scalable Video Coding	164
6.1	Introduction	164
6.2	Overview of Intra Coding in SVC	166
6.2.1	Intra Prediction Methods in SVC	167
6.2.2	Entropy Coding of Prediction Modes in SVC	169
6.3	Our Proposed Intra Coding Scheme	171
6.3.1	Proposed Intra Prediction Algorithm	171
6.3.2	Proposed Entropy Coding of Prediction Modes	175
6.4	Experimental Results	177
6.5	Conclusions	181
7	Conclusions and Future Research Directions	182
7.1	Conclusions	182
7.2	Future Research Directions	184
	Author's Publications	191
	Bibliography	193

Summary

Current digital video applications require delivery of digital video content to different clients over heterogeneous networks. These clients may have different system resources and bandwidth capabilities. Thus, delivering of digital video data compressed at specific spatial resolution, quality levels, frame rates and encoding those video data at specific bit rates according to the client resources and the available network is essential. The characteristics of current video transmission system pose many problems for video delivery and scalable video coding is a highly attractive solution. This thesis investigates the existing video coding standards, especially the newest H.264/advanced video coding and its scalable extension: scalable video coding. In this thesis, we address two issues related to the state-of-the-art scalable video coding: 1) computational complexity reduction; and 2) video coding efficiency enhancement.

In the first part of this thesis, we address the complexity issue of spatial, signal-to-noise ratio (SNR), temporal and combined scalable video coding. Our goal is to find a good tradeoff between computational complexity and video coding efficiency. We first establish the need to reduce computational complexity for current H.264/advanced video coding and scalable video coding framework. We then evaluate the mode distribution correlations between the base layer and its

enhancement layers for different video scalabilities. After the exhaustive search over all possible block partitions is performed at the base layer, the number of candidate modes for luma and chroma blocks in a macroblock that take part in rate distortion optimization calculation at enhancement layers could be greatly reduced based on the correlations. Finally, adaptive fast mode decision schemes for spatial, SNR and temporal scalable video coding are presented. Our schemes could achieve consistent and significant computational complexity reduction with negligible loss in objective and subjective video quality and insignificant increments in bit rate consumption.

The second part of this thesis is to provide a solution for the problem of high computational complexity of scalable video decoders. An adaptive encoding algorithm is proposed to reduce decoder complexity in coarse grain SNR scalable video coding. Accordingly, light weight bit streams are generated at the encoder, and those bit streams require less interpolation computation at the decoder side and thus the decoding complexity is reduced significantly. The experimental results show that our proposed scheme can efficiently reduce interpolation computation at the decoder side. Hence the decoder complexity is reduced significantly with insignificant visual quality loss and minor impact on the encoder complexity.

The third part is to develop SNR enhancement layer refinement schemes in order to achieve high scalability for SNR scalable video coding without a significant loss in coding efficiency. We first identify the need to develop motion information refinement algorithms at SNR enhancement layers. Two novel enhancement layer motion estimation and motion compensation methods are then proposed. Consequently, a scalable bit stream containing a scalable representation of motion data and residual data can be obtained. Our schemes could obtain a good tradeoff between motion information and residual information at all SNR layers.

Finally, the fourth part of this thesis is devoted to presenting a novel intra coding scheme as an enhancement of scalable video coding. In our proposed intra coding scheme, the prediction is performed in pixel domain, where the current pixel can be predicted from the reconstructed pixels in surrounding blocks as well as previously predicted pixels in the current block. With this scheme, more effective prediction modes for 4×4 and 16×16 blocks are proposed, respectively. To reduce the bit rate cost, we further develop an efficient algorithm, which successfully predicts the most probable prediction mode for the current block. The results show that the methodologies presented in this part are efficient and have potential for further development.

In comparison to the existing scalable video coding scheme, the video coder with our proposed fast mode decision algorithms achieves significant computational gain for spatial, SNR, temporal and combined scalability. In addition, light weight bit streams, which require significantly less decoding complexity, could be generated at the SNR scalable video encoder with proposed adaptive encoding scheme. Furthermore, the video coder with proposed rate control scheme for SNR scalability and intra coding scheme achieves better subjective and objective visual quality.

List of Figures

2.1	A brief history of video coding standards	13
2.2	Data structure: from video sequence to blocks	15
2.3	A typical sequence with I-, P- and B-frames	15
2.4	The encoder side of hybrid video coding scheme	16
2.5	The decoder side of hybrid video coding scheme	17
2.6	Compare the coding efficiency of H.264/AVC and other standards. ‘H.263 HLP’ is the H.263 high latency profile, ‘ASP’ is the MPEG-4 advanced simple profile, ‘H.26L’ is the H.264 main profile [1].	24
2.7	General concept of scalable video coding	27
2.8	SVC provides different types of scalability: spatial, SNR, temporal and combined scalability	28
2.9	An example of spatial scalability in SVC	29
2.10	An example of SNR scalability in SVC	30
2.11	An example of temporal scalability in SVC	31
2.12	An example of combined scalability in SVC	32
2.13	The macroblock partitions of H.264/AVC	37
2.14	The intra prediction methods for 4×4 partition in H.264/AVC . . .	38

2.15	The intra prediction methods for 16×16 partition in H.264/AVC	39
2.16	The method of multiple reference frames for H.264/AVC	40
2.17	Interpolation of luma half-pixel positions in H.264/AVC	41
2.18	Interpolation of luma quarter-pixel positions in H.264/AVC	42
2.19	Computation complexity of each coding tool in SVC [2]	44
3.1	Bit stream with M layers of spatial scalability	50
3.2	High-level block diagram of spatial scalability in SVC [3]	51
3.3	Layer coding for spatial scalability in SVC	53
3.4	Upsampling of motion information	54
3.5	Inter-layer residual prediction	56
3.6	Inter-layer intra-prediction	57
3.7	4-tap filter: $[-3, 19, 19, -3]$	58
3.8	Flowchart of proposed fast mode decision scheme for spatial scalability	61
3.9	Flowchart of proposed selective intra mode prediction for SNR scalability	69
3.10	Flowchart of selective reduction of candidate modes scheme for SNR scalability	70
3.11	Flowchart of proposed enhanced fast decision scheme for SNR scalability	72
3.12	Hierarchical prediction structures for enabling temporal scalability. (a) Coding with hierarchical B-pictures. (b) Nondyadic hierarchical prediction structure. (c) Hierarchical prediction structure with a structural encoding/decoding delay of zero.	76
3.13	Temporal scalability with GOP size is 16	77
3.14	Average percentage of fine partitioned MBs for <i>MOBILE</i> sequence .	79
3.15	Definition of the dominant block from the motion estimated block .	82

3.16	Flowchart of proposed scheme for temporal scalability	83
3.17	Rate distortion curve for <i>FOREMAN</i> in spatial scalable video coding	89
3.18	Rate distortion curve for <i>FOOTBALL</i> in spatial scalable video coding	90
3.19	Rate distortion curve for <i>BUS</i> in spatial scalable video coding . . .	91
3.20	The comparison of reconstructed video quality, the 2nd, 4th and 6th frames of <i>FOOTBALL</i> . (a) reconstructed with our scheme for spatial scalability, (b) reconstructed with JSVM encoder.	92
3.21	Rate distortion curve for <i>CITY</i> in SNR scalable video coding . . .	93
3.22	Rate distortion curve for <i>HARBOUR</i> in SNR scalable video coding	94
3.23	The comparison of reconstructed video quality, the 20th, 21st, 22nd and 23rd frames of <i>FOREMAN</i> . (a) reconstructed with our scheme for SNR scalability, (b) reconstructed with JSVM encoder.	96
3.24	Rate distortion curve for <i>FOOTBALL</i> in temporal scalable video coding	99
3.25	Flowchart of modified scheme for temporal scalability for high QP value	100
3.26	The comparison of reconstructed video quality, the 60th, 61st, 62nd and 63rd frames of <i>BUS</i> . (a) reconstructed with our scheme for tem- poral scalability, (b) reconstructed with JSVM encoder.	102
3.27	Rate distortion curve for <i>FOREMAN</i> in combined spatial and CGS scalable video coding	103
3.28	Rate distortion curve for <i>FOOTBALL</i> in combined spatial and CGS scalable video coding	104
3.29	Rate distortion curve for <i>FOREMAN</i> in spatial and temporal scal- able video coding	106
3.30	Rate distortion curves for <i>AKIYO</i> in CGS scalable video coding . .	113

3.31	Rate distortion curves for <i>FOOTBALL</i> in CGS scalable video coding	114
3.32	Rate distortion curves for <i>FOREMAN</i> in CGS scalable video coding	115
3.33	Rate distortion curves for <i>AKIYO</i> for spatial scalable video coding	118
3.34	Rate distortion curves for <i>FOREMAN</i> for spatial scalable video coding	119
3.35	Rate distortion curves for <i>MOBILE</i> for spatial scalable video coding	120
4.1	Block diagram of a SVC decoder	124
4.2	λ_{SAD} versus λ_{MOTION} for <i>FOREMAN</i> in CGS scalable video coding	129
4.3	Sub-pixel locations in SVC	131
4.4	Rate distortion curve for <i>FOREMAN</i>	134
4.5	Rate distortion curve for <i>FOOTBALL</i>	135
5.1	Flowchart of the base layer motion estimation and mode decision process	139
5.2	Original (a) and base layer reconstructed (b) frame of <i>FOREMAN</i> .	143
5.3	Flowchart of proposed SNR refinement scheme A	145
5.4	Motion information correlations in the base layer and enhancement layers	146
5.5	Original (a), reconstructed base layer (b) and reconstructed en- hancement layer (c) frames of <i>FOREMAN</i>	148
5.6	Flowchart of scheme B for prediction mode determination for each P_i	150
5.7	Flowchart of proposed scheme B for macroblock partition determi- nation	152
5.8	Coded PSNR per frame of enhancement layer SNR scalability for <i>FOREMAN</i>	154
5.9	Coded PSNR per frame of enhancement layer SNR scalability for <i>MOBILE</i>	155

5.10	Coded PSNR per frame of enhancement layer SNR scalability for <i>BUS</i>	156
5.11	Scalable coding versus non-scalable coding for <i>FOREMAN</i>	158
5.12	PSNR responses of enhancement layers as a function of layer separation for base layer bit rate at 20Kbps and 64Kbps by using proposed scheme $A + B$	161
5.13	Comparison of the coding efficiency of various of three layer SNR scalable coders	162
6.1	Video encoder for intra frame coding in SVC	167
6.2	(a) Prediction modes for $INTRA_4 \times 4$, and (b) prediction modes for $INTRA_16 \times 16$ defined in SVC	168
6.3	Labelling of samples for 4×4 intra luma prediction	168
6.4	Current and neighboring 4×4 blocks	170
6.5	(a) Newly proposed prediction modes for $INTRA_4 \times 4$, and (b) newly proposed prediction modes for $INTRA_16 \times 16$	172
6.6	Direction correlations for neighboring 4×4 blocks	176
6.7	Rate distortion curve for <i>FOREMAN</i>	179
6.8	Rate distortion curve for <i>CITY</i>	180
7.1	Illustration of two neighboring macroblock A and B of current macroblock C	187

List of Tables

2.1	Average bit rate savings compared with various prior decoding schemes	
	[1]	24
2.2	Comparison between video coding standards	25
2.3	The calculation complexity at different sub-pixel position.	42
2.4	Encoding time comparison between SVC with and without inter-layer prediction	43
3.1	Probability of a MB to be intra coded at the enhancement layer if the corresponding MB at the base layer is intra coded	59
3.2	Statistical analysis of inter modes distributions for <i>FOREMAN</i> . .	64
3.3	Statistical analysis of inter modes distributions for <i>FOOTBALL</i> . .	65
3.4	Statistical analysis of inter modes distributions for <i>AKIYO</i>	65
3.5	Statistical analysis of inter modes distributions for <i>NEWS</i>	65
3.6	Statistical analysis of inter modes distributions for <i>BUS</i>	66
3.7	Statistical analysis of inter modes distributions for <i>CITY</i>	66
3.8	Statistical analysis of SPR value	71
3.9	Experimental Setup	73
3.10	Simulation condition	85

3.11 Simulation condition for spatial scalability	87
3.12 Simulation results for spatial scalability with QP value of 10	87
3.13 Simulation results for spatial scalability with QP value of 20	88
3.14 Simulation results for spatial scalability with QP value of 30	88
3.15 Simulation results for spatial scalability with QP value of 40	88
3.16 Simulation condition for fast mode decision scheme 1 in SNR scalability	91
3.17 Simulation results in SNR scalability with QP value of 10	92
3.18 Simulation results in SNR scalability with QP value of 20	93
3.19 Simulation condition for enhanced fast mode decision scheme 2 in SNR scalability	94
3.20 Simulation results with $QP_b = 40$ and $QP_e = 30$ for SNR scalable video coding	95
3.21 Simulation results with $QP_b = 30$ and $QP_e = 20$ for SNR scalable video coding	96
3.22 Simulation condition for temporal scalable video coding	97
3.23 Simulation results in temporal scalability with QP value of 15	97
3.24 Simulation results in temporal scalability with QP value of 25	98
3.25 Simulation results in temporal scalability with QP value of 35	98
3.26 Simulation results in temporal scalability with QP value of 40	101
3.27 Simulation results in temporal scalability with QP value of 30	101
3.28 Simulation results in temporal scalability with QP value of 20	101
3.29 Simulation condition for combined spatial and SNR scalability	102
3.30 Simulation results for combined spatial and CGS scalability	103
3.31 Simulation condition for combined spatial and temporal scalability	104

3.32	Simulation results for combined spatial and temporal scalability with QP value is 25	105
3.33	Simulation results for combined spatial and temporal scalability with QP value is 35	105
3.34	Simulation condition	111
3.35	Comparison of performance on CGS	112
3.36	Simulation condition	116
3.37	Comparison of performance on spatial scalability	117
4.1	Filter coefficients for different sub-pixel positions	125
4.2	Simulation condition	132
4.3	Simulation results with $QP_e = 10$	133
4.4	Simulation results with $QP_e = 20$	133
4.5	Simulation results with $QP_e = 30$	134
5.1	Setting of QPs for different SNR scalability schemes	153
5.2	Comparison of the average PSNR for <i>FOREMAN</i> at different bit rates	155
5.3	Comparison of the average PSNR for <i>BUS</i> at different bit rates . .	156
5.4	Comparison of the average PSNR for <i>MOBILE</i> at different bit rates	157
5.5	Performance of SNR scalability as a function of layer separation at base layer rates of 20Kbps for HHI scheme and scheme $A + B$. . .	159
5.6	Performance of SNR scalability as a function of layer separation at base layer rates of 64Kbps for HHI scheme and scheme $A + B$. . .	160
5.7	Performance of three layers SNR scalable coder for various schemes	162
6.1	Choice of prediction mode	171
6.2	Test sequences in our experiments	178

6.3	Simulation results for proposed intra prediction algorithm with QP value of 40	178
6.4	Simulation results for proposed intra prediction algorithm with QP value of 30	178
6.5	Simulation results for proposed intra prediction algorithm with QP value of 20	179
6.6	Encoding time comparison	180
7.1	Context models	187
7.2	Context models for bin 1	187
7.3	Human interest regions for various video sequences	189

Introduction

1.1 Background

Due to the growth and success of the ever-present, ever-growing Internet, the explosive development of wireless communications, and the ever-increasing demand of multimedia representation and communications, data digitization, data compression and data streaming have drawn tremendous attention from both academia and industry. Analog signals are sampled and digitized so that they can be transmitted and stored in errorless format. However, digitization process will generate a large amount of information, among which video is one of the most powerful forms of multimedia because of the extensive information it delivers. Digital video data rates are extremely large. The bandwidth of digital video streams that match the current visual resolution of current television systems exceeds 100 megabits/s (mbps). Modern television formats such as high-definition television (HDTV) can multiply this by a factor of 4 or more [4]. As a result, bandwidth, computing resources and storage consumption can be extremely high in the multimedia services with data rates of this magnitude. In addition to high bandwidth, computing

resources and storage requirements, significant costs will be added to the hardware and systems that process digital video by using uncompressed video data. Thus, digital video coding is extremely crucial even with exponentially increasing bandwidth, computing resources and storage capabilities.

Video coding is an essential application of digital video processing [5]. The goal of video coding is to reduce the data rate of a video sequence, so that it is feasible to be transmitted in real time over a given communication channel. Video coding algorithms operate by removing temporal, spatial and frequency redundancies [6]. By removing these three types of redundancies, it is possible to compress the data significantly at the expense of a certain amount of information loss (distortion). The spatial redundancy could be reduced by transform coding, such as discrete cosine transform (DCT) [7] or discrete wavelet transform (DWT) [8]. The temporal redundancy could be explored by either the traditional motion estimation and motion compensation technique of MPEG standards [9–11], or the recently proposed motion compensated temporal filtering [12–14]. The statistical redundancy could be removed by entropy coding methods, such as Huffman coding [15], arithmetic coding [16], run-level coding [17] and variations of Golomb-Rice coding [6] [18]. Golomb-Rice coding is optimal for the data with geometric progression distribution probability. It is a special case of Huffman coding.

With development of video coding technology, coding efficiency is improved significantly. The latest international video coding standard, H.264/MPEG-4 (Part 10) advanced video coding (H.264/AVC) could achieve a highly improved coding efficiency. Compared with previous standards, H.264/AVC could deliver an average bit rate reduction of 50% given a fixed video quality [19]. Error control schemes are provided in H.264/AVC as well. As a result, transmission errors over

various communication networks are controlled and tolerated. Moreover, low latency capabilities are provided in H.264/AVC so that better quality for higher latency is provided. However, such improvements are achieved at the expense of extremely high computing complexity caused by the new technologies adopted in H.264/AVC [20].

Besides the coding efficiency, current digital video applications require delivery of digital video to different clients over heterogeneous networks and these clients have diverse system resources and different bandwidth capabilities. Thus, delivering of digital video data compressed at specific spatial resolution, quality levels and frame rates, and encoding those video data at specific bit rates according to the client resources and the available network is essential. For example, if the conventional video coding standards are used to encode a video sequence, such as H.26X [21–24] and MPEG-1, -2, -4, [25–27]. Transcoding techniques are used to deliver the content at lower quality, bit rates, spatial resolutions and frame rates using the bit stream previously encoded at higher quality, bit rates, spatial resolution and frame rate. Transcoding costs additional system resources and causes delays [28]. An efficient alternative is scalable video coding, in which the bandwidth allocated to a lower resolution scale is partially reused to code the higher resolution scale. Scalable video coding (SVC) could increase the adaptability of video coding system for various networks and client conditions. Furthermore, if an algorithm or system is designed to be scalable, it can be used over a wide range of operating points without redesign.

Scalable video coding is necessary in many applications [29, 30]: Internet video, wireless LAN video, mobile wireless video for conversational, video on demand, live broadcasting purposes, end-to-end internet/wireless video delivery, multi-channel content production and distribution, storage applications, multi-point surveillance

systems, and so on. However, the flexibility and high coding efficiency in SVC is gained at the expense of an increase in computational complexity in both encoder and decoder. This will generate implementation limitations for many electronic devices, especially for mobile devices having limited battery life and memory size such as digital cameras, digital camcorders and portable media players.

1.2 Motivations

The basic mission of a scalable design is twofold [29]: 1) to minimize the coding efficiency loss relative to single-layer coding; and 2) to minimize the complexity increase (especially for decoders) relative to single-layer coding. Likewise, the main goal of research embodied in this thesis is to develop methods to reduce computational complexity at both video coding encoder and decoder, as well as to improve video coding efficiency of current scalable video coding systems. It motivates us to address the above two problems from the following four aspects:

- (i) *Fast mode decision for inter-frame coding in fully scalable video coding systems*

H.264/AVC and SVC video coding standard achieves much higher coding efficiency as compared to other standards. Unfortunately, high coding efficiency is achieved at a cost in considerably increased complexity at the encoder since motion estimation and mode decision is quite resource consuming. In H.264/AVC and SVC, Lagrangian rate-distortion optimization (RDO) technique is employed to decide the coding mode for a macroblock. In order to select the best coding mode for a macroblock, video encoder calculates the rate-distortion cost of every possible mode and chooses the one with the minimum value. This brute force searching technique achieves

the highest possible coding efficiency. However, it results in extremely long encoding time that prevents it from practical use. In order to reduce the computational complexity of H.264/AVC, many efforts have been proposed to explore the fast algorithms in motion estimation and mode decision for H.264/AVC video coding. Actually, fast motion estimation is one of the well-studied topics, and many fast methods without much loss of video quality were derived [31, 32]. Besides various fast motion estimation algorithms, many fast mode decision strategies have been proposed to further reduce the computation load of H.264/AVC [33–42]. All these methods are efficient in reducing the computational complexity with acceptable quality degradation in H.264/AVC encoder. However, these fast mode decision algorithms cannot be applied directly to SVC. In SVC, enhancement layers motion estimation and mode decision occupies most of the encoding time. By exploring the correlation of mode distribution of the base layer and its enhancement layers, the computational complexity could be substantially reduced. Fast mode decision for inter-frame coding in SVC is a new topic, and little was found in the literature so far, even though it plays a extremely significant role in reducing the overall complexity of scalable video coding.

(ii) *Adaptive computational complexity reduction scheme in coarse grain scalable (CGS) video decoding systems*

The excellent performance of coarse grain scalable video coding is achieved at the expense of computational complexity increment at both encoder and decoder. Thus, it is essential to obtain a good tradeoff between computational complexity and compression efficiency for both scalable encoder and decoder. For most internet video content, as well as video server and broadcast application, video is encoded once and decoded many times. Decoder complexity

is critical in these cases, because the end user must decode and display the video content in real time. Thus, decoder complexity reduction plays a vital role in reducing the overall complexity of SVC. With a proper encoding strategy, light weight decoder-friendly bit streams could be generated, and those bit streams require less amount of computational complexity at the decoder. It should be pointed out that a number of efforts have been made to explore decoder complexity reduction in H.264/AVC by using the observation that image interpolation and deblocking filter consume the majority of the decoding time [43–49]. All these methods are remarkably efficient in reducing the computational complexity without considerably quality degradation in H.264 decoder. However, these methods are not applicable to all the layers in CGS scalable video coding systems. Moreover, they did not provide any formulation to choose the Lagrangian factor associated with the complexity issue.

(iii) *Rate control for SNR scalable video coding systems*

The main goal for scalable video coding is to yield high scalability, and such high scalability has to be achieved without a significant loss of coding efficiency. SNR scalability supports decoding of moving pictures having quality varying progressively between acceptable and visually lossless in fine grained steps. Many interesting SNR scalability schemes were proposed [50–52]. However, in these schemes, only one motion vector field is generated for all SNR layers. In this way, the bit allocation between motion and residual information is not optimal in the whole bit rate range. Consequently, there exists a peak SNR gap of about 1-3dB compared to existing single layer coding scheme at certain bit rate [53]. Therefore, an optimized rate control scheme between motion information and residual information is required to improve

the coding efficiency for the SNR scalable video coding systems.

(iv) *Enhanced intra coding scheme in scalable video coding systems*

Intra coding plays a crucial role in scalable video coding in the following way. Intra frames can be regarded as anchor points. Without requiring one to decode all previous frames, those intra frames in the video can be accessed randomly. Moreover, intra frames serve as the reference frames for previous and following prediction frames, thus the coding performance of intra frames will affect the coding performance of other types of frames. However, compared with other types of frames, compression ratios for intra frames are low, therefore intra frames contain the most amounts of bits and take up more bandwidth during the transmission. In recent year, a number of efforts have been made to improve the coding performance and bit rate saving of intra coding in H.264/AVC video coding [54–59]. All these methods achieved some bit rate saving compared to the scheme used in H.264/AVC, but the results are not significant and with increasing encoder complexity. Consequently, an effective intra coding scheme is required for current SVC.

1.3 Summary of Major Contributions

The major contributions of this thesis are as follows:

- (a). Proposal of fast mode decision schemes for inter-frame coding in fully scalable video coding systems:

Fast mode decision schemes are proposed for spatial, SNR, temporal and

combined scalable video coding systems, respectively. By exploring the macroblock mode distribution relationship between the base layer and its enhancement layers, the number of candidate modes for luma and chroma blocks in a macroblock that take part in rate distortion optimization calculation has been reduced significantly at the enhancement layers. The proposed fast mode decision schemes could achieve a reduction of 50% to 60% encoding time on average for different scalabilities. Moreover, our proposed schemes could achieve consistent time saving over a large bit rate range with negligible loss in Y-PSNR and insignificant increments in bit rate consumption.

- (b). Description of decoder computational complexity reduction scheme for coarse grain SNR scalable video coding systems:

An adaptive encoding scheme that would generate light weight bit streams at the decoder is proposed, those bit streams require less amount of computational power. Our proposed algorithm is adaptive to all the layers in scalable video coding. This is achieved by using a new rate-distortion-complexity optimized cost function for motion estimation and mode decision in enhancement layers as compared to traditional encoding algorithms. The experimental results show that our proposed scheme could reduce the decoding computational complexity effectively and flexibly with acceptable performance degradation.

- (c). Development of rate control schemes between motion and residual information for SNR scalable video coding systems:

Two novel schemes on enhancement layer motion estimation and motion compensation are proposed for SNR scalable video coding systems. The improvement in scalable coding efficiency is achieved by refining both motion

information and residual information at scalable video coding enhancement layers. Hence, a good tradeoff between motion and residual information can be obtained at all bit rates. Consequently, the coding efficiency of our proposed scheme is improved by up to 1dB in Y-PSNR.

(d). Presentation of a novel intra coding scheme for scalable video coding systems:

An enhanced intra coding scheme for scalable video coding is presented. With this scheme, more effective prediction modes for 4×4 and 16×16 blocks are proposed, respectively. Moreover, the algorithm for predicting the most probable prediction modes is improved. The experimental results show that our proposed scheme can improve bit rate saving as well as coding performance as compared to JSVM encoder, without a substantial increase in the complexity of the encoding and decoding process.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2 first gives an overview of the video coding standards and hybrid video coding structure. It then summarizes the new features utilized in H.264/AVC, and compares H.264/AVC and its previous video coding standards in terms of coding performance. Subsequently, H.264/AVC's scalable extension: scalable video coding is established. The new features in scalable video coding including video scalability and motion compensated temporal filtering are studied. Finally, it addresses the complexity issue in H.264/AVC and scalable video coding standard.

Chapter 3 presents in detail fast mode decision schemes for spatial, SNR, temporal and combined scalable video coding systems. Firstly, the rationales behind the spatial, SNR and temporal scalable video coding systems are introduced. Then

the correlations between the base layer and its enhancement layers for different scalabilities are examined. Based on these correlations, fast mode decision algorithm for each scalability structure is proposed. Finally, experimental results and relevant discussions for each scalability structure are provided.

Chapter 4 addresses the computational complexity issue of scalable video decoding. A new framework of rate-distortion-complexity optimization is formulated at the beginning of this chapter. Then a complexity model for the interpolation process in scalable video coding is provided. With our proposed adaptive encoding algorithm, light weight decoder-friendly bit streams which require less amount of computational complexity are generated and the efficiency of our scheme is supported by experimental results.

Chapter 5 describes rate control schemes for SNR scalable video coding systems. It begins by introducing conventional rate control mechanism and detailing the importance of motion information scalability in scalable video coding. After that, two enhancement layer refinement schemes are proposed. Finally, it provides in detail the experimental results and a study into some of the characteristics of the performance of enhanced SNR scalability.

Chapter 6 proposes a novel intra coding scheme as an enhancement of scalable video coding. It first presents intra prediction methods adopted in current scalable video coding standard. Then an enhanced intra mode prediction scheme is proposed. At the end of this chapter, our proposed intra coding scheme is compared with JSVM encoder with experimental results and discussions.

Chapter 7 concludes this thesis with important remarks and recommendations for future research directions.

Overview of H.264/Advanced Video Coding and Scalable Video Coding

2.1 Video Coding Standards

In order to eliminate picture redundancy, video coding standards have been developed to allow video information to be transmitted and stored in an efficient and compact way. Standardization organizations have their roots in international trade (ISO) and the telecom industry (ITU) [5]. ISO/IEC's Moving Picture Experts Group (MPEG) has since standardized MPEG-1, MPEG-2, MPEG-4 and MPEG-4 AVC also known as H.264. MPEG-1 [25] was released in 1991 targeting the Video Home System (VHS) quality on a CD-ROM or VCD. MPEG-1 is a standard for storing and playing video on a single computer at relatively low bit rate. MPEG-2 [60] is a standard proposed in 1994 for digital TV and meets the requirements for HDTV (High Definition Television) and DVD. MPEG-4 Visual part 2 [27, 61, 62] video coding, which is released in 1999, is the first standard to

consider content-based coding and it introduces the concept of video object, each video frame could be segmented into several objects and coded separately. MPEG-4 Visual part 2 supports all functions which are already provided by MPEG-1 and MPEG-2. ITU-T's Video Coding Experts Group (VCEG) has since standardized H.261, H.263, and H.264, etc. H.261 [21] is developed in 1988-1990, and it is designed for video conferencing and video-telephone applications over ISDN telephone lines, and the targeting bit rate is $p \times 64$ ($1 \leq p \leq 30$) kbps. H.263 [22, 63] was proposed in 1995 and designed for low bit rate (less than 64 kbps) communication. The ITU-T H.264 standard and the ISO/IEC MPEG-4 Part 10 Advanced Video Coding (AVC) standard are technically identical [53, 64]. It is the newest international video coding standards, and it is currently the most powerful and state-of-the-art standard. It was developed by a joint video team (JVT) consisting of VCEG and MPEG, and it is created in 2001. In this thesis, this new standard is simply specified by the name H.264/AVC. Figure 2.1 shows the brief history of video coding standards. Although these standards are operating at different bit rates and targeting for different applications, they all follow a similar hybrid video coding framework [65, 66] in terms of the coding algorithms.

Work is continuing on new video coding standards. A Joint Collaborative Team on video coding (JCT) has been set up by ITU-T and ISO/IEC in January 2010. The purpose was to develop a new video coding standard, currently known as 'High Efficiency Video Coding' (HEVC or H.265), which is more advanced than the current H.264/AVC [67]. It is still on the development stage, and it looks like we might expect to see a $2 \times$ compression improvement with video quality improvement. This bit reduction will be gained at the expense of higher complexity in the encoder computation. However, decoding of HEVC promises to be not much higher in CPU usage than H.264/AVC.

Figure 2.1: A brief history of video coding standards

2.2 Hybrid Video Coding Scheme

Most video coding methods exploit temporal, spatial and statistical redundancy to achieve compression [6]. Temporal redundancy is reduced by exploiting a high correlation between frames of the video sequence that were captured at around the same period. Spatial redundancy can be reduced by utilizing a high correlation between pixels that are close to each other in the same video frame. The major video coding standards, including ISO MPEG-1,-2,-4, ITU-T H.261, H263 and H.264/AVC, have been based on the same generic design of a video codec that incorporates a motion estimation and compensation front end (temporal model), a transform stage (spatial model) and an entropy encoder (statistical model) [6]. This model, which is described as a hybrid coding scheme, has drawn much attention to date. Enormous research efforts have been spawned to surmount the various

problems encountered.

The data structure in a video sequence used in hybrid video coding systems is shown in Figure 2.2. There are six layers in this structure: video sequence, group of pictures (GOP), picture, slice, macroblock and block. Blocks are grouped into macroblocks, and each macroblock is corresponding to a 16×16 luma region and associated chroma sample. In addition, different types of frames such as I-frames, P-frames and B-frames, may be used by an encoder as shown in Figure 2.3 [11]. An I-frame is an intra coded frame. It is compressed discretely and standalone. As it is not dependent on any other frame in a video sequence, it can be decoded with no need of any other frames. I-frame exploits spatial redundancy only, therefore it consumes more bits and takes up more bandwidth, but on the other hand, it does not generate many artifacts. It is also known as key frame, as it can be used to allow random access within a video sequence as a key. P-frame stands for predictive inter frame. The prediction is made from earlier I and/or P frames, so that require less coding data. However, P-frame is dependent on earlier I or P reference frames, therefore it is sensitive to transmission errors. B-frame is bidirectional predictive inter frame. The prediction of B-frame is made from earlier or future reference frames. This kind of prediction method occupies least coding data and achieves the most compression. However, the computation time is the longest.

The generic system diagram of hybrid video encoding and decoding are shown in Figure 2.4 and 2.5, although actual coders may vary from one to another [6]. The key components in a hybrid video encoder include motion estimation, motion compensation, transformation, quantization and entropy coding shown as boxes in the figures. Based on the group of pictures (GOP) structure, the encoder side needs to determine whether the incoming frame is coded as an intra frame (I) or an inter frame (P/B). In Intra mode, each I-frame is divided into macroblocks or

Figure 2.2: Data structure: from video sequence to blocks

Figure 2.3: A typical sequence with I-, P- and B-frames

blocks, each block of pixels is converted through a DCT transform module, and the spatial domain pixels are converted into transform domain coefficients. After that, a quantizer is applied to reduce the number of levels for transformed coefficients. Variable length coding (VLC) is used to encode the quantized coefficients. For B-frames and P-frames, motion estimation could be performed on each block, based on the current frame and the reconstructed reference frame. For B frames, both the forward and the backward motion predictions are needed. As shown in Figure 2.4, current frame F_n is used for encoding and is processed in terms of a macroblock.

After that, F_n is compared with a reference frame, for example the previous encoded frame (F'_{n-1}). Then a motion estimation function is used to find a 16×16 region in the reference frame (or a sub-sample interpolated version of a reference frame) that matches the current block in the current frame. Motion vector (MV) is defined as the offset between the current macroblock position and the selected reference region. After that, a motion compensated prediction P is generated based on the resulted motion vector. By subtracting P from the current macroblock, a residual or difference macroblock D is produced. Then, D undergoes a discrete cosine transform (DCT). By splitting the macroblock into 8×8 or 4×4 sub-blocks, each sub-block is transformed separately. After that, each sub-block is quantized. The resulted DCT coefficients of each sub-block are first reordered and then run-level coded. Finally, the coefficients, motion vector and associated header information for each macroblock are entropy encoded to produce the compressed bit stream [68]. Hybrid video coding provides an effective optimization framework allowing video coders to find compact representation of motion information and prediction errors.

Figure 2.4: The encoder side of hybrid video coding scheme

Figure 2.5: The decoder side of hybrid video coding scheme

2.3 H.264/Advanced Video Coding

H.264 or Advanced Video Coding (H.264/AVC) is the latest standard for video compression, and it was published as Part 10 of MPEG-4 and ITU-T Recommendation H.264 in May 2003 by the joint video team of ISO/IEC's MPEG and ITU-T's VCEG [24,69]. It outperforms the earlier MPEG-4 and H.263 standards, providing better video quality at substantially lower bit rates, without considerably increasing the complexity of design that it would be impractical or excessively too expensive to implement.

The intent of this standardization is to create a standard capable of providing enhanced compression performance, more error robustness and seamless integration of video coding into all current protocols with a simple and straightforward video coding design. An additional goal is to provide enough flexibility to allow the standard to be applied to a wide variety of applications on a wide variety of networks and systems, including 'conversational'(video telephony) and 'non-conversational'(storage, broadcast or streaming) applications [19].

H.264/AVC supports a broad range of bit rates and picture sizes. It addresses video coding capabilities ranging from extremely low quality, low bit rate, low frame rate, 'postage stamp' resolution video for mobile and dail-up devices, through to entertainment-quality standard-definition television services, HDTV, and beyond.

In addition, a flexible system interface for the coded video is specified in H.264/AVC so that it will enable the adaptation of video content for use over this full variety of network and channel-type environments. Therefore, H.264/AVC could provide a technical solution appropriate for a broad range of applications including but not limited to the following [1, 70]:

- Direct broadcast satellite video services;
- Multimedia services over packet networks;
- Real-time conversational services (videoconferencing, videophone, etc.);
- Digital terrestrial television broadcasting, cable modem, DSL;
- Interactive or serial storage on optical and magnetic storage devices (optical disks, DVD, etc.);
- Multimedia mailing;
- Serial storage media(digital VTR, etc.);
- Digital Cinema, Content contribution, content distribution, studio editing, post processing;
- Remote video surveillance;
- Video Streaming over the Internet;
- Cable TV on optical networks, copper, etc.;
- High definition transmission and storage.

A number of new features used in H.264/AVC allow it to compress video much more effectively than previous video coding standards and to offer more flexibility

for applications to a variety of network environments. In particular, some such key features including but not limited to the following [19, 71]:

- Improved inter-prediction and motion estimation:

(i) *Fine-grained motion estimation and compensation*

More flexibility in the selection of matching block size is supported in H.264/AVC. Motion estimation may also identify motion vectors associated with matching sub-macroblock of 4×4 , 4×8 , 8×4 , 8×8 , 8×16 , 16×8 , or the full 16×16 . In addition, the quarter-sample-accurate motion estimation and compensation with reducing the interpolation complexity is used in H.264/AVC.

(ii) *Multiple reference frames*

In MPEG-2, the motion estimation and compensation is carried out between the current frame and only one previous reconstructed frame. H.264/AVC provides flexibility for the encoder to choose the reference frames from up to 16 multiple reconstructed frames. By using this technique, more precise inter prediction can be achieved, and robustness to lost picture data can be improved.

(iii) *Unrestricted motion search*

In MPEG-2 and its previous video coding standards, only the pixels within the previous reconstructed frames are used as reference for motion estimation and compensation. In H.264/AVC, motion search region in reference frames may be partly outside the picture by spatially predicting the missing data from boundary data.

(iv) *Motion vector prediction*

In H.264/AVC, motion vectors may be accurately predicted and only

their residuals transmitted explicitly in the bit stream.

(v) *Weighted prediction*

H.264/AVC allows the motion compensated prediction signal to be weighted and offset by amounts specified by the encoder. This method could dramatically improve coding efficiency for scenes containing fades, and could be used flexibly for other purpose as well.

- Flexible interlaced-scan video coding features:

(a) *Macroblock-adaptive frame-field coding*

This technique uses a macroblock pair structure for pictures coded as frames, allowing 16×16 macroblocks in field mode.

(b) *Picture-adaptive frame-field coding*

This technique allows precise placement of residual signals with little of the ‘ringing’ often found with prior codec designs.

- Improved intra spatial prediction and transform:

(a) *Coarse and fine intra prediction*

Intra prediction may be performed either on 4×4 blocks, or 16×16 macroblocks. The latter is more efficient for uniform areas of a picture.

(b) *Small block-size transform*

H.264/AVC enables the encoder to transform the spatial signal data based on a 4×4 block. This allows the encoder to represent signals in a more locally-adaptive fashion, which reduces artifacts known as ‘ringing’.

(c) *Directional dependent intra prediction*

In MPEG-2 Part 2, DC-only prediction is used for spatial prediction of

intra coding blocks. The transform coefficient prediction is used in H.263 and MPEG-4 Part 2. In H.264/AVC, a more flexible intra prediction method of extrapolating the edges of the previously reconstructed blocks is applied for intra coding blocks.

(d) *Integer transforms*

By implementing the traditional discrete cosine transform as an integer transform that requires no multiplications, computational efficiency in both computation and bit rate is achieved.

(e) *Hierarchical block transform*

Transform of variable size blocks with the same level of flexibility as motion estimation blocks is supported in H.264/AVC. More flexibility and further reduction of bit rates are achieved by using this technique.

(f) *Exact-match inverse transform*

H.264/AVC is the first standard to achieve exact equality of decoded video content from all decoders.

- In-loop deblocking filter:

Most of the popular video coding standards utilize block-based video coding techniques. One drawback of this kind of coding structure is that the discontinuities at the block boundaries are visible, and these discontinuities produce blocking artifacts [72]. In H.264/AVC, deblocking filter is not an optional additional feature. It helps to prevent the blocking artifacts, resulting in better visual appearance and compression efficiency.

- Improved entropy coding algorithms:

(a) *Context-adaptive variable length coding (CAVLC)*

CAVLC employs multiple variable length codeword tables to encode

transform coefficients, which consume the bulk of bandwidth.

(b) *Context-adaptive binary arithmetic coding (CABAC)*

CABAC provides an extremely efficient encoding scheme when it is known that certain symbols are much more likely than others. H.264/AVC introduces CABAC to improve performance relative to prior standard designs [73].

- Improved error containment and scalability:

(a) *Data partitioning*

This feature is supported to allow higher priority data (e.g., motion vectors, header information and other prediction information) to be separated from lower priority data for transmission.

(b) *Flexible Macroblock Ordering (FMO)*

A new ability to partition a frame into regions called slice groups has been utilized in H.264/AVC. FMO will decide to which slice each macroblock of the image belongs. As a result, a packet loss, that will affect a large region will be reduced. Error concealment is enabled by ensuring that the neighboring macroblocks will be available for prediction of a missing macroblock.

(c) *Arbitrary Slice Ordering (ASO)*

By subdividing each frame into one or more slices, each slice can be decoded separately of the other slices in a frame. H.264/AVC is designed to enable the video coding system to send and receive the slices of the frame in any order relative to each other. This feature can improve end-to-end delay in real-time applications.

(d) *Redundant frames*

The H.264/AVC design allows an encoder to send redundant representations of regions of frames when some important parts of the frame is lost during transmission.

(e) *Parameter set structure*

H.264/AVC provides a parameter set structure to separate the key information (such as sequence header or frame header information) for handling in a more flexible and specialized manner since the loss of this key information could have a severe negative impact on the decoding process.

(f) *NAL unit syntax structure*

H.264/AVC provides a logical data packet called NAL unit, so that each syntax structure can be placed into this packet. It allows greater customization of the method of carrying the video content in a manner appropriate for each specific network.

With the above mentioned features, H.264/AVC outperforms previous standards in terms of coding efficiency and network adaption. Figure 2.6 gives an example of comparing the coding efficiency of H.264/AVC and other standards [1,74]. H.264/AVC can deliver better image quality at the same compressed bit rate or a lower compressed bit rate for the same image quality as shown in Table 2.1. The comparison of specific coding algorithms among H.264/AVC and previous video coding standards is given in Table 2.2.

2.4 Scalable Video Coding

Scalable Video Coding (SVC) is a remarkable feature added to the H.264/AVC standard, and it is specified in Annex G of H.264/AVC [75]. In October 2003,

Figure 2.6: Compare the coding efficiency of H.264/AVC and other standards. ‘H.263 HLP’ is the H.263 high latency profile, ‘ASP’ is the MPEG-4 advanced simple profile, ‘H.26L’ is the H.264 main profile [1].

Table 2.1: Average bit rate savings compared with various prior decoding schemes [1]

Coder	MPEG-4 ASP	H.263 HLP	MPEG-2
H.264/AVC	38.62%	48.80%	64.46%
MPEG-4 ASP	-	16.65%	42.95%
H.263 HLP	-	-	30.61%

the MPEG group issued a call for proposals on SVC technology. During the 68th MPEG meeting in April 2004, 14 proposals including 12 based on compression by wavelets were proposed and 2 remaining proposals based on extensions of H.264/AVC [76, 77]. Wavelet based three-dimensional sub-band video coding was proposed by Microsoft Research Asia and scalable extension of H.264/AVC was proposed by Fraunhofer Institute for Telecommunications - Herinrich Hertz Institute (HHI). In January 2005, MPEG of the ISO/IEC and VCEG of the ITU-T agreed to finalize the SVC project as an amendment of their H.264/AVC standard [78] jointly, and the scalable extension of H.264/AVC was selected as the first Working Draft [79]. In July 2007, scalable extension of H.264/AVC was jointly

Table 2.2: Comparison between video coding standards

Feature/Standard	MPEG-2	MPEG-4 Part 2	H.264/AVC
Macroblock size	16×16 (frame mode), 16×8 (field mode)	16×16	16×16
Block size	8×8	16×16 , 16×8 , 8×8	16×16 , 8×16 , 16×8 , 8×8 , 4×8 , 8×4 , 4×4
Macroblock prediction with $1/4$ pixel	No, $1/2$ pel only	Yes, $1/2$ pel and $1/4$ pel	Yes, $1/4$ pel only
Intra prediction in blocks of intra MB	Fixed prediction of DC coefficient	Adaptive prediction of DC coefficient, and first row/column of AC coefficients	Adaptive spatial prediction of 4×4 or 16×16 pixel blocks
Direct prediction mode in B pictures	No	1 Mode only: temporal direct	2 Modes: temporal direct no mv update, spatial direct
Bidirectional prediction mode	Forward/backward	Forward/backward	Forward/forward forward/backward backward/backward
Block artifact reduction	Postprocessing often used but no suggested filter	Postprocessing suggested with provided optional filter	Mandatory in-loop filter, post processing may also be used
Intra DC nonlinear quant, intra AC directional scans and improved chroma quant	No	Special nonlinear quant, MB level adaptive directional scan, improved chroma quant	Nonlinear DC quant, horizontal and vertical scans, improved chroma quant
Frame types	I, P, B	I, P, B	I, P, B, SP, SI, and I, P and B slices
Flexible picture prediction structure and stored B picture	Basic, no stored B-picture	Basic, no stored B-picture	Yes, allowed
Multi reference prediction	No	No	Yes
Weighted prediction in P/B	Usual ($1/2$, $1/2$) weighting of forward and backward prediction in B-pictures	Usual ($1/2$, $1/2$) weighting of forward and backward prediction in B-pictures	Yes, very flexible
Motion estimation & compensation	Yes	Yes	Yes, more flexible up to 16 MVs per MB
ME accuracy	Integer, $1/2$ -samples	Integer, $1/2$ -sample, $1/4$ -sample	Integer, $1/2$ -sample, $1/4$ -sample
Unrestricted motion vectors	No	Yes	Yes
Motion vector prediction	Simple	Better, uses median	Uses median, and segmented
Arbitrary slice order and flexible macroblock ordering	No	No	Yes
Transform	8×8 DCT	8×8 DCT/wavelet	Approximation of 4×4 DCT
Quantization	Scalar quantization with step size of constant increment	Vector quantization	Scalar quantization with step size increase at the rate of 12.5%
Efficient quantizer overhead	No	Yes	Yes
Entropy coding	VLC	VLC	VLC, CAVLC, CABAC
Adaptive VLC coding	No	Yes, uses 2 tables	Yes, very content adaptive
Adaptive arithmetic coding	No	No, not for DCT coefficients	Yes, very content adaptive
Stream switching, splicing and random access	Basic, intra pictures	Basic, intra pictures	Intra pictures/slices, SI/SP switching pictures/slices
Error robustness	Data partitioning, FEC for important packet transmission	Synchronization, Data partitioning, Header extension, Reversible VLCs	Data partitioning, parameter setting, flexible macroblock ordering, redundant slice, switched slice
Transmission rate	2 – 15 Mbps	64kbps – 2Mbps	64kbps – 240Mbps
Encoder complexity	Low	Medium	High

published by MPEG and VCEG, which makes the scalable extension to be the state-of-the-art video codec. In this thesis, our contributions for scalable video coding are based on the scalable extension of H.264/AVC proposed by HHI. The key features and requirements of SVC including [29, 80]:

- Hierarchical prediction structure;
- Layered coding scheme with switchable inter-layer prediction mechanisms;
- Base layer compatibility with H.264/ MPEG-4 AVC;
- Fine granular quality scalability using progressive refinement slices;
- Usage and extension of the NAL unit concept of H.264/MPEG-4 AVC;
- Support for multiple sub-streams with spatial, temporal and SNR scalability;
- Single video file-multiple sub-streams;
- Single loop decoding;
- Advanced and simple header format that facilitates the aggregation of multiple layers into a single bit stream;
- Similar coding efficiency compared to a single layer coding;
- Little increase in decoder complexity compared to a single layer coding;
- Much better coding efficiency than that of simulcasting coding;

Compared to the H.264/AVC standard, scalable video coding provides spatial, SNR and temporal scalability [81]. It allows decoding of bit streams at quality levels, spatial resolutions and frame rates lower than those parameters of the originally encoded video sequence.

Figure 2.7: General concept of scalable video coding

As shown in Figure 2.7, the coded stream is arranged in a number of layers, including a ‘base’ layer and one or more ‘enhancement’ layers. In this figure, decoder A receives the stream only from the base layer and can decode a ‘basic’ quality version of the scene, whereas decoder B receives streams from all layers and decodes a higher quality version of the scene. This technique can be used in many applications. For instances, the base layer bit stream may be decoded by a low-complexity decoder. Then a low-rate bit stream may be extracted for transmission over a network with limited capability.

Scalable video coding has a number of advantages [51]. Firstly, it allows service interworking. Users can receive different quality versions of the same video service by using a scalable approach. This is realized by having the encoder produce two or more bit streams. The base layer bit stream contains the most important information, and enhancement layers contain coded information that can be decoded by a decoder to a higher resolution version of the service. Scalable encoder is carefully designed to ensure that the overall performance of this encoder is much superior to a simulcast approach that provides the same services. The second advantage of the scalable coding approach is that the coded video data is divided into two or more layers of different levels of importance, and the most critical information is

contained in the base layer. This high priority information can then be transmitted over a more reliable transmission network when error resilience is important. Current digital video applications require at least three types of scalability features [82]: spatial resolution scalability, SNR and temporal scalability as shown in Figure 2.8.

Figure 2.8: SVC provides different types of scalability: spatial, SNR, temporal and combined scalability

2.4.1 Spatial Scalability

Spatial scalability is a technique to code a video sequence into two or more layers at the same frame rate, but different spatial resolutions. It should be able to support

Figure 2.9: An example of spatial scalability in SVC

a variety of resolutions, including QCIF, CIF, 4CIF, and also higher resolutions such as 1436×1080 to 3610×1536 [3]. As shown in Figure 2.9, for spatial scalability, the base layer contains a reduced-resolution version of each coded frame. Moreover, it is coded independently. The reconstructed base layer frame is upsampled to form the prediction for the high resolution frame in the enhancement layer. The enhancement layer is coded based on predictions formed from the base layer frames and previously encoded enhancement layer frames. A low resolution video sequence can be produced by decoding the base layer alone. A higher resolution video sequence is produced by decoding the base layer together with enhancement layers. In case of the spatial resolution of the base layer is the same as that of the enhancement layer, i.e., the upsampling ratio is 1, this spatial scalable decoder is considered as an SNR scalable decoder.

Figure 2.10: An example of SNR scalability in SVC

2.4.2 SNR Scalability

SNR scalability is a technique to code a video sequence into two or more layers at the same frame rate and the same spatial resolution, but different quantization accuracy. This scalability supports decoding of moving pictures having quality varying progressively between acceptable and visually lossless in fine grained steps [83]. As shown in Figure 2.10, decoding the base layer provides a low-quality version of the reconstructed image. Further decoding the enhancement layers results in quality increases of the reconstructed images, up to the highest quality. The base layer is obtained by applying a coarse quantizer to the original image or in a transform (e.g., DCT) domain. The second layer contains the quantized difference between the original image and that reconstructed from the base layer, using a quantizer that is finer than that used to produce the first layer. Similarly, each of the subsequent layers contains the quantized difference between the original image and that reconstructed from the previous layers, using an increasingly finer quantizer.

2.4.3 Temporal Scalability

Figure 2.11: An example of temporal scalability in SVC

In temporal scalability, as opposed to multi-spatial resolution coding, the input sequence is divided into multiple temporal resolution. The temporal scalability supports decoding of moving pictures with frame rates ranging between 5Hz and 75Hz [84]. For instance, for multi-channel content production and distribution, the same stream will be viewed on a variety of devices having different temporal resolutions. The temporal decomposition frameworks of hierarchical B frames and motion compensated temporal filtering inherently provide temporal scalability. By using n decomposition stages, up to n levels of temporal scalability can be provided.

2.4.4 Combined Scalability

Spatial, SNR and temporal scalability can be combined to reach finer granularity. The coding structure depends on the scalability space that is required by the application. An example of the combined scalable video coding scheme is provided

Figure 2.12: An example of combined scalability in SVC

in Figure 2.12. The sequence with QCIF spatial resolution is first encoded at a frame rate of 7.5Hz as Layer 1. By using the inter-layer prediction method, Layer 2 with CIF spatial resolution is encoded from the up-sampled information of Layer 1. Layer 3 is achieved based on Layer 2 by the SNR refinement. The temporal frame rate is refined to 15Hz in Layer 4. In this case, spatial, SNR and temporal resolution of the reconstructed image can be improved by progressively transmitting and decoding the bit stream. The order is to improve image quality at a given spatial resolution until the best quality is achieved at that resolution, and then to increase the spatial resolution to a higher level and improve the quality

gain, finally to increase the frame rate until the desired frame rate is achieved. In this thesis, spatial scalability is provided using a layered approach. For SNR scalability, an embedded quantization approach is pursued. Temporal scalability is enabled by the Hierarchical B structure.

2.5 Complexity Analysis of H.264/AVC and SVC

High compression efficiency is one of key advantages of H.264/AVC. Compared with H.263 or MPEG-4, H.264/AVC permits a reduction in bit rate by up to 50% for a similar degree of encoder optimization at most bit rates [19, 64]. However, the high compression efficiency comes at a price. The implementation complexity of H.264/AVC decoder is roughly two times that of MPEG-2. The computational complexity for SVC is even higher than H.264/AVC because many of the decisions, such as motion estimation and compensation are ordinarily carried out only for the base layer have to be repeated on all the enhancement layers. In addition, inter layer prediction will incur excessive encoding time as illustrated in the following section.

2.5.1 Rate Distortion Optimization

Rate distortion optimization refers to the optimization of the amount of distortion against the amount of bits required to encode the video stream. In previous video coding standards, such as MPEG-1, MPEG-2, H.261, H.263, motion vectors are determined by the sum of absolute difference (SAD) or sum of squared difference (SSD), related to the distortion of the motion prediction as following:

$$mv = \arg \min_{mv \in r} \{D_{mv}\} \quad (2.1)$$

where mv is the determined macroblock motion vector, r is the motion search range, and D_{mv} is SAD or SSD between the current macroblock and the reference macroblock. Equation 2.2 and 2.3 describe two energy measures, SAD and SSD. The motion compensation block size is $N \times N$ samples; C_{ij} and R_{ij} are current and reference area samples respectively [15].

$$\text{SAD} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \quad (2.2)$$

$$\text{SSD} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \quad (2.3)$$

As shown in equation 2.1, the bit rate is not considered in these criteria since the QP value is not involved in the motion estimation step. Hence, it is necessary to adjust the QP value between the motion compensation step and the spatial quantization step to meet the required coding bits. Based on the classic rate distortion theory [85], H.264/AVC utilizes the Lagrangian optimization to adjust the bit allocation between the motion information and residual information in the motion estimation and mode decision process automatically [86]. Rate distortion optimization refers to the optimization of the amount of distortion (loss of video quality) against the amount of data required to encoder the video, the rate [87]. The motion estimation and mode decision cost function J_{mv} and J_{mode} are given by:

$$J_{mv} = D_{mv|QP} + \lambda_{mv} R_{mv|QP} \quad (2.4)$$

$$J_{mode} = D_{mode|QP} + \lambda_{mode} R_{mode|QP} \quad (2.5)$$

where D is the SAD or SSD between the current macroblock and the motion compensated matching block. R_{mv} is the bit cost to encode the motion vector, and R_{mode} is the bit cost to encode the quantized residual coefficients. λ_{mv} and λ_{mode} are the Lagrange multipliers for motion estimation and mode decision, respectively. The contribution of the motion bits in the total cost functions is controlled by these weight parameters. The above λ_{mv} and λ_{mode} are defined as a function of the QP empirically:

$$\lambda_{mode} = 0.85 \times 2^{QP/3-4} \quad (2.6)$$

$$\lambda_{mv} = \sqrt{\lambda_{mode}} \quad (2.7)$$

However, the optimal choice of λ should be dependent on the characteristics of each macroblock as well as QP [88]. Hence, in H.264/AVC, QP takes part in the bit allocation between motion information and residual information. Clearly, a large quantization step size results in a large value of λ and thus a low bit rate range and a large amount of distortion [82]. On the other hand, a small quantization step size results in a small value of λ . Therefore, a high bit rate range and a small amount of distortion.

2.5.2 Macroblock Inter-Prediction Modes

Macroblock is an image compression component. It contains coded data corresponding to a 16×16 sample region of the video frame including 16×16 luma samples, 8×8 Cb and 8×8 Cr samples. Every coded macroblock in an H.264/AVC frame is predicted from previously-encoded data [74]. For intra macroblock, samples within the block are predicted from neighboring samples in the current frame that have already been encoded, decoded and reconstructed. For inter macroblock,

samples are predicted from previously-encoded macroblocks.

H.264/AVC defines seven different macroblock partitions to represent accurate motion field of moving objects to reduce the residual error as shown in Figure 2.13 [6]. For inter prediction, the luminance component of each macroblock is split up in four ways and motion compensated either as one 16×16 macroblock partition, two 16×8 partitions, two 8×16 partitions or four 8×8 partitions as shown in Figure 2.13(a). H.264/AVC encoder calculates the rate distortion cost of all the possible partition modes and select the prediction mode with minimal rate distortion cost. If the 8×8 mode is chosen, each of the four 8×8 sub-macroblocks within the macroblock may be split in further four ways, either as one 8×8 sub-macroblock partition, two 8×4 sub-macroblock partitions, two 4×8 sub-macroblock partitions or four 4×4 sub-macroblock partitions as shown in Figure 2.13(b) [6]. These macroblock partitions and sub-macroblock partitions provide a large number of possible combinations with each macroblock.

2.5.3 Macroblock Intra-Prediction Modes

H.264/AVC defines several spatial prediction block partitions for intra prediction including 16×16 and 4×4 for luma and 8×8 for chroma. In addition to provide the ability for the encoder to select among the various block sizes, the encoder can also select among the various prediction directions for each block size. For a 4×4 block, H.264/AVC defines nine prediction modes as shown in Figure 2.14 [6].

Mode 0 (Vertical) The upper samples A, B, C, D are extrapolated vertically.

Mode 1 (Horizontal) The left samples I, J, K, L are extrapolated horizontally.

Mode 2 (DC) All samples in previously encoded and reconstructed block are predicted by the mean of samples A...D and I...L.

Figure 2.13: The macroblock partitions of H.264/AVC

Mode 3 (Diagonal Down-Left) The samples are interpolated at a 45 degree angle between lower-left and upper-right.

Mode 4 (Diagonal Down-Right) The samples are extrapolated at a 45 degree angle down and to the right.

Mode 5 (Vertical-Right) Extrapolation at an angle of approximately 26.6 degree to the left of vertical ($\text{width/height} = 1/2$).

Mode 6 (Horizontal-Down) Extrapolation at an angle of approximately 26.6

degree below horizontal.

Mode 7 (Vertical-Left) Extrapolation at an angle of approximately 26.6 degree to the right of vertical.

Mode 8 (Horizontal-Up) Interpolation at an angle of approximately 26.6 degree above horizontal.

Figure 2.14: The intra prediction methods for 4×4 partition in H.264/AVC

For 16×16 macroblock, H.264/AVC defines four prediction modes as shown in Figure 2.15 [6].

Mode 0 (Vertical) Extrapolation from upper samples (H).

Mode 1 (Horizontal) Extrapolation from left samples (V).

Mode 2 (DC) Mean of upper and left-hand samples (H+V).

Mode 3 (Plane) A linear ‘plane’ function is fitted to the upper and left-hand samples H and V. This works well in areas of smoothly-varying luminance.

All modes are directional predictions and the directions are indicated by the arrows. The encoder typically selects the prediction mode for each block that

Figure 2.15: The intra prediction methods for 16×16 partition in H.264/AVC

minimizes the difference between previously encoded, reconstructed block and the block to be encoded.

Therefore, H.264/AVC provides more flexible prediction modes for each macroblock in the inter frames from the set. Both the motion vector and the macroblock partition mode should be determined with rate distortion optimization using the Lagrangian method, and the final macroblock mode is determined with the minimal rate distortion cost. However, rate distortion optimization based motion estimation and mode decision require a series of operations such as motion vector estimation, quantization, integer transform, entropy coding, inverse quantization and inverse integer transform, etc. It is extremely computationally intensive to select the best mode from all the candidate modes due to so many possible partitions of each MB. Subsequently, the resultant computational burden challenges the personal terminals (e.g. handphone and PDA) for real-time video communication.

2.5.4 Multiple Reference Frames

In MPEG-2 and its previous standards, only the frame just before and/or after the current frame could be used as a reference frame. H.264/AVC allows the use of multiple reference frames as shown in Figure 2.16 [1]. Besides the motion vector, the frame reference parameters (Δ) are also transmitted to the decoder

Figure 2.16: The method of multiple reference frames for H.264/AVC

side. This concept could also be extended to bi-directional predicted frames. The extra reference frames can improve compression efficiency and video quality. The maximum number of concurrent reference frames supported by H.264 is 16 [19]. However, in the process of video encoding, many of the decisions such as motion estimation that have to be carried out on all the reference frames instead of one reference frame in the previous video coding standards. Thus, multiple reference frames could considerably increase encoding time. In addition, in the process of video decoding, reference frames need to be stored in memory until they are no longer needed for further decoding. Thus, the memory usage of the decoder for videos with large numbers of reference frames needs to be increased.

2.5.5 Quarter-pixel Precision for Motion Estimation and Compensation

The default level of resolution for motion estimation and compensation for MPEG-4 and its previous standards is half a pixel [89]. H.264/AVC enables quarter-pixel precision instead. The more accurate motion prediction provides the encoder a significant improvement of motion estimation and motion compensation. Similar to half-pixel motion prediction, quarter-pixel motion prediction is achieved through

interpolation. H.264/AVC first uses a 6-tap Finite Impulse Response (FIR) filter with weights $(1/32, -5/32, 5/8, 5/8, -5/32, 1/32)$ for half-pixel interpolation as shown in Figure 2.17 [6]. For example, half-pixel sample **b** is calculated from the 6 horizontal integer samples E, F, G, H, I and J:

$$\mathbf{b} = \text{round}((E - 5F + 20G + 20H - 5I + J)/32) \quad (2.8)$$

Figure 2.17: Interpolation of luma half-pixel positions in H.264/AVC

Once all the half-pixel data are available, H.264/AVC uses simple bilinear sampling to achieve quarter-pixel precision from the half-pixel data as shown in Figure 2.18 [90]. For example, quarter-pixel sample **a** is calculated from the bilinear interpolation:

$$\mathbf{a} = \text{round}((G + b)/2) \quad (2.9)$$

Figure 2.18: Interpolation of luma quarter-pixel positions in H.264/AVC

It is clear that the sub-pixel values at different positions are with different interpolation complexity. The calculation complexity at different sub-pixel positions is given in Table 2.3. Full search motion estimation involves evaluating equation 2.2 at each quarter-pixel, half-pixel, integer-pixel position in the search window. For each possible macroblock partition, the prediction method together with the associated reference indices r_0 and r_1 as well as motion vectors mv_0 and mv_1 are determined by equation 2.10. From the table, we can see that the computational complexity of quarter-pixel is much higher.

$$mv_{0/1}(r_{0/1}) = \arg \min \{D + \lambda(R(r_{0/1}) + R(mv_{0/1}))\} \quad (2.10)$$

Table 2.3: The calculation complexity at different sub-pixel position.

Example position	sub-pixel accuracy	interpolation complexity
H	integer	no
b, h	half	$1 \times (\text{six-tap})$
j	half	$7 \times (\text{six-tap})$
a, c, d, n	quarter	$1 \times (\text{six-tap}) + 1 \times (\text{two-tap})$
e, g, p, r	quarter	$2 \times (\text{six-tap}) + 1 \times (\text{two-tap})$
i, f, k, q	quarter	$7 \times (\text{six-tap}) + 1 \times (\text{two-tap})$

2.5.6 Scalabilities

Similar to H.264/AVC, in order to choose the best coding mode for an MB, SVC calculates the rate distortion cost of every possible mode and selects the one with minimum cost as the best mode. Calculation of the rate distortion cost in SVC needs to execute both forward and backward process of integer transform, quantization, inverse quantization, inverse integer transform and entropy coding and this introduces high computational complexity to the encoder. Figure 2.19 shows that the motion estimation occupies most of the encoding time in SVC at each layer. Meanwhile, the encoding time of the enhancement layer motion estimation is about 10 times of the base layer [2]. Inter layer prediction can improve the rate distortion performance substantially. However, it will incur excessive encoding time especially in spatial scalable video coding. This is shown by the experimental results given in Table 2.4. In the experiment, two spatial layers are used. The base layer has a resolution of QCIF, and the enhancement layer has a resolution of CIF. *FOREMAN* is used as the testing sequence. The test platform is Intel Core 2.13GHz CPU, 2.86G RAM with Windows 7 professional operating system. The total encoding time for the encoder with inter-layer prediction is much longer than that for the encoder without inter-layer prediction.

Table 2.4: Encoding time comparison between SVC with and without inter-layer prediction

QP	Total Encoding Times (sec)	
	Inter-Layer Prediction On	Inter-Layer Prediction Off
15	5844	564
25	4500	520
35	3720	480
45	3556	456

Figure 2.19: Computation complexity of each coding tool in SVC [2]

Fast Mode Decision Schemes for Inter-Frame Coding in Fully Scalable Video Coding

3.1 Introduction

Compared to the previous video coding standards, scalable video coding (SVC) is intended to encode the signal once, but enable decoding from partial streams depending on the specific resolution, quality and bit rate required by a certain application [81]. SVC is designed to extend the hybrid video coding approach used in H264/AVC and efficiently incorporate spatial, SNR and temporal scalability. A layered approach can be used to realize the spatial and SNR scalability. Video data can be represented as one base layer and one or more enhancement layers. The base layer contains a reduced resolution or a reduced quality version of each coded frame. The enhancement layers contain a higher resolution or a higher quality version of each coded frame. They can be predicted from the base layer frames and previously encoded enhancement layer frames. Temporal scalability in SVC

is achieved by using a structure of hierarchical B pictures [91, 92]. A temporal scalable video coding system allows extraction of video of multiple frame rates from a single coded stream.

Current SVC scheme shows significant achievements in terms of coding efficiency [64]. In this coding system, variable block-size matching motion estimation is used to reduce the temporal redundancy between frames. SVC defines 7 macroblock (MB) modes for inter prediction ($MODE_{16 \times 16}$, $MODE_{16 \times 8}$, $MODE_{8 \times 16}$, $MODE_{8 \times 8}$, $MODE_{8 \times 4}$, $MODE_{4 \times 8}$ and $MODE_{4 \times 4}$), 9 prediction modes for $INTRA_{4 \times 4}$, 4 prediction modes for $INTRA_{16 \times 16}$ and $MODE_{SKIP}$ [91]. In order to encode the motion field of an enhancement layer, ‘Base_layer_mode’ and ‘Qpel_refinement_mode’, are added to the modes applicable in the base layer [75]. These two modes are used to indicate that motion and prediction information including the partitioning of the corresponding MB of the base layer are used. In this chapter, we use BL_{pred} to represent these two modes. In order to determine the best coding mode for an MB, rate distortion cost (RDcost) of every possible mode is calculated, and the one with minimum cost is selected as the best mode. Calculation of the RDcost in SVC needs a series of operations, such as forward and backward processes of integer transform, quantization, inverse quantization, inverse integer transform, and entropy coding, and this introduces high computational complexity to the encoder. Therefore, it is desirable to design algorithms to reduce the computational complexity of SVC without compromising the coding efficiency for the implementation of SVC.

A number of efforts have been made to provide solutions to the problem of high computational complexity in current H.264/AVC video coding systems. Among all modules in encoder, motion estimation and mode decision play key roles and contribute most of the computational complexity. There have been many efficient

fast motion estimation techniques proposed in the literatures. Zhu *et al.* [93,94] proposed a non-rectangular search pattern, diamond search pattern, for fast block-matching motion estimation. In contrast to algorithms with limited steps, diamond pattern requires much fewer checking points. A more advanced idea using similar starting pattern, but with dynamic size and adaptive feature, can be found in [95]. In addition, Yi *et al.* [31] proposed a simplified UMHexagonS technique to reduce motion estimation memory consumption and computational complexity while retaining video visual quality for H.264/AVC. Later, they introduced a novel dual-halfway-stop method, in addition to the one-halfway-stop technique, in normalized partial distortion search (NPDS) algorithm to reduce block matching motion estimation complexity [32]. The simulation results showed improvements in terms of speedup and video quality when compared to those of many well-known fast block matching algorithms. Beside motion estimation, macroblock mode decision is also the main culprit that increases the overall computational complexity. Pan *et al.* [33] proposed an algorithm by performing Sobel edge operators to detect the edge direction in the current block, and then utilize the edge direction histogram to decide the best prediction mode. An effort has also been made by Wu *et al.* to use the spatial homogeneity and the temporal stationary characteristics of video objects to guide the fast inter prediction process [34]. Yu *et al.* proposed fast inter mode decision algorithms by making use of a spatial content measure and temporal similarity detection to assign different levels of inter modes to each macroblock depending on its complexity. The lower the complexity of the macroblock content, the few inter modes the encoder has to check, and vice versa [35,36]. The macroblock level neighborhood information has been exploited in [37] along with a set of skip mode conditions for enhanced skip mode decision. In [38], context-based adaptive skipping of unlikely intra prediction modes,

subsampling of matching operations, bit width truncation to reduce the computations and interleaved full-search/partial-search strategy are proposed to reduce the encoding complexity effectively. Early skip mode decision and selective intra mode decision are exploited in [39]. In [40] early termination was used to reduce the number of potential prediction modes. Different from the above mentioned methods, [41] proposed an efficient classifier which is based on the absolute frame different, to decide which block types are to be enabled. In addition, Ahmad *et al.* [42] presented a fast mode decision algorithm to determine the mode of the current macroblock based on the modes previously determined in the previous and current frames. All these methods are efficient in reducing the computational complexity with acceptable quality degradation in H.264/AVC encoder. Furthermore, all these methods could be adopted in SVC encoder to reduce the encoding time for the base layer. However, in SVC, enhancement layers motion estimation and mode decision occupies most of the encoding time. By exploring the correlation of mode distribution of the base layer and its enhancement layers, the computational complexity could be greatly reduced.

We have observed that the mode distribution between the base layer and its enhancement layers has certain correlation. In spatial scalability, for each MB at the base layer, the corresponding upsampled MBs at enhancement layers tend to have the same mode partition [96]. For SNR scalability, each enhancement layer MB tends to have a finer mode partition than the corresponding MB at the base layer [97, 98]. In the case of temporal scalability, the mode partition of MBs in the current frame is most similar to the mode partition of MBs in its reference frames [99]. Motivated by these observations, we propose effective fast mode decision schemes for spatial, SNR and temporal scalable video coding. With our proposed schemes, a good mode partition prediction can be achieved if

we predict the MB mode at the enhancement layer from that of the base layer. Therefore, the presented schemes reduce the number of candidate modes for an MB at enhancement layers by using the mode distribution at the base layer, and hence reduce the computational complexity. Simulation results illustrate that our proposed schemes can achieve up to 61% of encoding time saving with negligible peak signal-to-noise (Y-PSNR) loss and insignificant bit rate increases.

The rest of this chapter is organized as follows. Section 3.2, Section 3.3 and Section 3.4 present in detail the fast mode decision schemes for spatial, SNR and temporal scalable video coding, respectively. Experimental results are presented in Section 3.5 and conclusions are given in Section 3.6.

3.2 Spatial Scalable Video Coding

Spatial scalability provides support for multiple display resolutions within a single compressed bit stream. Figure 3.1 illustrates a bit stream with M layers of spatial scalability [100]. By decoding the base layer, a preview version of the decoded image at a lower resolution is produced. By decoding the second layer, a larger reconstructed image can be generated. Furthermore, by progressively decoding the additional layers, the spatial resolution of the resulted image will be increased, up to the full resolution of the original image. This kind of spatial scalability could be achieved on either a subband decomposition coding framework or the multi-layer hybrid coding framework. The subband decomposition methods such as wavelet-based approaches take advantage of the inherent properties of wavelet transform to achieve spatial scalability [12, 101–103]. On the other hand, hybrid video coding has been a dominant video coding technology. For supporting spatial scalable coding, SVC follows the conventional approach of multi-layer coding, which is also

used in MPEG-2, MPEG-4 Visual and H.263. In SVC, spatial scalable coding of video is considered at multiple resolutions (e.g. 176×144 (common intermediate format or QCIF), 352×288 (CIF) and 704×576 (4CIF)) with a factor of 2 in horizontal and vertical resolution.

Figure 3.1: Bit stream with M layers of spatial scalability

Spatial scalability has become an attractive feature in many video applications [3, 29]. Lower resolution displays with between ten thousand and on hundred thousand pixels are required in applications constrained by size, power and weight. By contrast, high definition displays are becoming common in consumer applications. Consequently, the resolution diversity of current display devices motivates the need for spatial scalability. On the other hand, diverse, limited, and time-varying channel capacity provides a second motivation for spatial scalability.

Figure 3.2: High-level block diagram of spatial scalability in SVC [3]

Figure 3.2 shows a block diagram of a two layer spatially scalable video encoder. A number of advanced features have been introduced by using this coding scheme. Significant achievements in terms of coding efficiency, robustness to a variety of network channels are demonstrated in this coding scheme. In each spatial layer, motion-compensated prediction and intra prediction are employed as for single layer coding. Lower resolution video sequence is taken as the input of the base layer encoder and is encoded with the H.264/AVC coding standard. The enhancement layer encoder takes a higher resolution sequence as input. It can be encoded to a single-layer bit stream and transmitting with the base layer bit stream using the sum of the two bit rates. In order to improve coding efficiency in comparison to simulcasting different spatial resolutions, three prediction mechanisms are incorporated in SVC: 1) Inter-layer prediction as illustrated in Figure 3.3. The value of enhancement layer data is predicted from previously reconstructed data of a lower resolution layer at the same time instance. 2) Inter-frame prediction, it involves prediction operating temporally between different pictures of the same resolution layer. 3) Intra-frame prediction, it involves prediction operating spatially within the same picture of one particular resolution layer. In order to restrict the memory requirements and decoder complexity, the same coding order is used for all supported spatial layers in SVC. As illustrated in Figure 3.3, it is not necessary to present lower layer pictures in all access units; thus it is possible to combine other scalabilities such as temporal and spatial scalability.

3.2.1 Inter-Layer Prediction

In order to improve rate distortion efficiency of enhancement layers, the usage of as much lower layer information as possible is one of the main goal when designing inter-layer prediction tools. The methods employ the reconstructed samples of the

Figure 3.3: Layer coding for spatial scalability in SVC

lower layer signal are the only inter-layer prediction methods supported in MPEG-4 Visual and its previous standards. The prediction signal is either formed by motion-compensated prediction inside the enhancement layer, by upsampling the reconstructed lower layer signal, or by averaging such an upsampled signal with a temporal prediction signal [29]. The following inter-layer prediction techniques are used in the current SVC scheme: 1) Inter-layer motion prediction: The partitioning data of the enhancement layer macroblock together with the associated reference indexes and motion vectors are derived from the upsampled data of the co-located 8×8 block in the reference layer by inter-layer motion prediction. 2) Inter-layer residual prediction: Prediction of the residual signal using the upsampled residual signal of the reference layer. 3) Inter-layer intra-prediction: Prediction using the corresponding upsampled reconstructed intra-signal of the reference layer.

(a) *Inter-Layer Motion Prediction:*

SVC's inter-layer prediction scheme results in many more MB encoding modes than those of the other standards. The available MB modes in SVC include two intra modes, seven inter modes, *MODE_SKIP* and the enhancement layer mode *BL_pred*. When *BL_pred* is set to be active, and the reference layer macroblock is inter coded, the enhancement layer macroblock is also inter coded.

Figure 3.4: Upsampling of motion information

As shown in Figure 3.4, the macroblock partitioning is obtained by upsampling the corresponding mode of the co-located 8×8 block in the base layer. If the co-located 8×8 block is not partitioned into smaller blocks, enhancement layer macroblock is not partitioned, i.e. the macroblock mode is 16×16 . Otherwise, each $M \times N$ sub-macroblock partition in the 8×8 base layer block corresponds to a $2M \times 2N$ macroblock partition in the enhancement layer macroblock. The same reference indexes as for the co-located base layer blocks are used for the upsampled macroblock partitions. In addition, both components of

the associated motion vectors are derived by scaling the corresponding base layer motion vector components by a factor of 2. Take the middle figure as an example. If the corresponding base layer MB is coded in *MODE_SKIP*, *MODE₁₆ × 16*, *MODE₁₆ × 8* and *MODE₈ × 16*, then *MODE₁₆ × 16* is used for the current MB at the enhancement layer. Therefore, corresponding signals of previous layers can be used to predict the intra and inter MBs. Moreover, motion description for each enhancement layer can be predicted from the motion description of proceeding layers. In our scheme, the base layer MB mode is used to predict the corresponding enhancement layer MB mode.

(b) *Inter-Layer Residual Prediction:*

Regardless whether macroblocks are coded using the newly SVC macroblock type or by using any of the conventional macroblock types, inter-layer residual prediction can be employed for all inter-coded macroblocks. In order to signal the usage of inter-layer residual prediction, a flag is added to the macroblock syntax for spatial enhancement layers. When this *residual-prediction-flag* is set to 1, the residual signal of the corresponding 8×8 sub-macroblock in the reference layer is block-wise upsampled using a bilinear filter as shown in Figure 3.5. This bilinear filter is adopted for efficient computation [104]. Upsampled block is used as prediction for the residual signal of the enhancement layer macroblock, so that only the corresponding difference signal need to be coded using a typical transform coding in the enhancement layer.

(c) *Inter-Layer Intra-Prediction:*

When *BL_pred* is set to be active, and the corresponding 8×8 block in the reference layer lies inside an intra-coded macroblock, the prediction signal of

Figure 3.5: Inter-layer residual prediction

the enhancement layer macroblock is obtained by inter-layer intra-prediction. In this mechanism, the corresponding reconstructed intra-signal of the reference layer is upsampled as shown in Figure 3.6. In order to upsample the luma component, one-dimensional 4-tap FIR filters are applied horizontally and vertically [104]. One example of a 4-tap FIR filter is shown in Figure 3.7. The chroma components are upsampled by using a simple bilinear filter. H.264/AVC deblocking filter is applied to the reconstructed intra-signal of the reference layer before upsampling to prevent disturbing signal components in the prediction signal.

3.2.2 Correlations Between the Base Layer 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 spatial scalability configuration. With the intensive experiment results, we found that the best mode of the current macroblock has a strong relationship with the used modes of the co-located reference macroblocks.

Figure 3.6: Inter-layer intra-prediction

- (a). In SVC, if there is a significant change between the reference and current frames (for example, a scene change), it may be more efficient to encode the MB by intra mode. The intra-coded MB at the base layer indicates that one can not find a good match in the reference frames. Similarly, at enhancement layers, it is also difficult for the upsampled MB to find a suitable match in the reference frames. Table 3.1 shows the probability of MBs that are intra coded at the enhancement layer in the following typical video sequences: *FOOTBALL*, *FOREMAN*, *BUS*, *NEWS*, *HARBOUR* and *CREW*. The QP value for both layers is set to be 10 and GOP size is set 16. Base layer resolution is QCIF and enhancement layer resolution is CIF. Table 3.1 indicates that if the MB at the base layer is intra coded, it is probable that the corresponding four MBs at its enhancement layer are intra coded. However,

Figure 3.7: 4-tap filter: $[-3, 19, 19, -3]$

in inter frame coding of SVC, the encoder has to compute rate distortion cost (RDcost) of all intra modes ($INTRA_4 \times 4$ and $INTRA_{16} \times 16$) that involve testing all intra prediction directions for all the MBs. This process is extremely complex, and the number of computing RDcost values is about 5 times higher than the case of inter modes [105]. On the other hand, as statistical data of intra mode indicates, the probability for an MB to have an intra mode in B slice is at most 7% and 4% on average [106], although the exact figure depends on specific input video characteristics. Such a small probability suggests that we should distinguish the intra-coded MBs in B slices at enhancement layers and only compute the RDcosts of intra modes for those MBs [96, 97].

(b). As discussed in Section 3.2.1, when the base layer represents a layer with half

Table 3.1: Probability of a MB to be intra coded at the enhancement layer if the corresponding MB at the base layer is intra coded

Video Sequence	Probability [%]
<i>FOOTBALL</i>	98.32
<i>FOREMAN</i>	97.17
<i>BUS</i>	95.66
<i>NEWS</i>	99.89
<i>HARBOUR</i>	98.38
<i>CREW</i>	97.86

the spatial resolution, according to the inter-layer prediction technique, the motion vector field including the MB partitioning is also upsampled. Therefore, we can reduce the number of candidate modes for certain MBs. If MB mode at the base layer is $MODE_{16 \times 16}$, it is probable that the corresponding MBs at its enhancement layer are coded at the mode of $MODE_{16 \times 16}$ or $MODE_{SKIP}$. The feature is more obvious if the quantization parameter is large.

- (c). After an exhaustive search, many MBs at enhancement layers are coded in the mode of BL_{pred} . This indicates that both the motion and texture information of the base layer is used to predict those at enhancement layers.

It is observed that there are correlations between the base layer and its enhancement layers for spatial scalability. Therefore, a good prediction could be achieved if we predict each MB mode partition at the enhancement layer from the corresponding MB at the base layer.

3.2.3 Proposed Fast Mode Decision Scheme for Spatial Scalability

Based on the considerations above, our fast mode decision procedure is given as follows:

- (a). If the estimated MB mode at the base layer is $INTRA_{4 \times 4}$, the candidate mode subset at its enhancement layer is reduced to BL_{pred} and $INTRA_{4 \times 4}$.
- (b). If the estimated MB mode at the base layer is $INTRA_{16 \times 16}$, the candidate mode subset at its enhancement layer is reduced to BL_{pred} and $INTRA_{16 \times 16}$.
- (c). If the estimated MB mode at the base layer is $INTER_{16 \times 16}$, the candidate mode subset at its enhancement layer is reduced to BL_{pred} and $INTRA_{16 \times 16}$ and $MODE_SKIP$.
- (d). If the estimated MB mode at the base layer is $MODE_{16 \times 8}$, $MODE_{8 \times 16}$, and $MODE_{8 \times 8}$, we will choose the best two modes and pass them to the enhancement layers to build the candidate mode set. These best two modes are the modes with minimum and 2^{nd} minimum estimated rate distortion cost at the base layer. Together with BL_{pred} and $MODE_SKIP$, the mode set is reduced to four in this case.

A flowchart of the overall process of the proposed scheme is shown in Figure 3.8. $MODE_{BL}$ represents the best mode selected at the base layer for certain MB. $MODE_{ELpred1}$ and $MODE_{ELpred2}$ are the best two modes generated at the base layer, and they are part of the prediction mode set for enhancement layers. These

Figure 3.8: Flowchart of proposed fast mode decision scheme for spatial scalability

best two modes can be generated after the rate distortion optimization at the base layer by comparing the rate distortion cost.

As a result, in our scheme, the candidate mode set is reduced at enhancement layers and complexity reduction is achieved.

3.3 SNR Scalable Video Coding

Signal-to-noise ratio (SNR) or bit rate scalability is a technique to code a video sequence into two or more layers at the same frame rate and the same spatial resolution, but different quantization accuracy. For SNR scalability in SVC, the required granularity varies for different applications. Two quality scalable modes namely coarse grain scalability (CGS) and medium-grain scalability (MGS) are supported in SVC. CGS supports bit rate adaptation on a level of coded video sequences. MGS supports bit rate adaptation on a NAL unit level. Moreover, MGS provides a finer granularity level of quality scalability by partitioning a given enhancement layer into several MGS layers [107]. In this thesis, fast mode decision schemes in CGS will be addressed.

3.3.1 Coarse Grain Scalability

Compared with spatial scalability in SVC, CGS employs similar encoding mechanisms except the spatial resolution is kept constant; therefore, it can be viewed as a special case of spatial scalability. More specifically, similar to spatial scalability, CGS employs inter-layer prediction mechanisms, such as prediction of macroblock modes and associated motion parameters and prediction of the residual signal. CGS differs from spatial scalability in that the upsampling operations and the inter-layer intra-prediction deblocking for intra-coded reference layer macroblock are not performed.

At first, the texture information is encoded in an H.264/AVC compatible base layer to provide a minimum quality at a given quantization level. At enhancement layers, quantization step size is decreased, and CGS is achieved by encoding successive refinements of the transform coefficients. Commonly, the quantization

parameter difference (QPD) between adjacent layers is equally spaced as shown in equation 3.1. Here, QPB , QPE_1 , QPE_2 , $\dots$, QPE_n are used to denote the quantization parameter values of the base layer, first enhancement layer, second enhancement layer, and so on.

$$\begin{aligned} QPD &= QPB - QPE_1 = QPE_1 - QPE_2 \\ &= QPE_n - QPE_{n+1} \text{ for } n = 1, 2, \dots \end{aligned} \quad (3.1)$$

Intra and inter MBs can be predicted using the corresponding signals of previous layers. Moreover, the motion description of each layer can be used for a prediction of the motion description for following enhancement layers.

3.3.2 Correlation Between the Base Layer and Enhancement Layers

Similar to H.264/AVC, the motion estimation and mode decision process in SVC is performed by minimizing the rate distortion function:

$$J = D + \lambda R \quad (3.2)$$

A large quantization step size results in a large value of λ and thus a low bit rate range and a large amount of distortion. On the other hand, a small quantization step size results in a small value of λ , thus a high bit rate range and a small amount of distortion [108]. Consequently, in the CGS, the mode partition of each MB at enhancement layers is finer than that of the corresponding MB at the base layer. In addition, quantization parameter difference (QPD) is a very important parameter in analyzing motion correlation between the base layer and its enhancement layers.

Large QPD will result in weak mode correlation; On the other hand, small QPD will result in strong mode correlation between layers. Therefore, by examining the QPD, the encoder can choose a suitable candidate mode reduction scheme. In order to examine the importance of QPD on the mode type distribution at the enhancement layers, we set up a series of experiments. We tested six sequences including *FOREMAN*, *FOOTBALL*, *AKIYO NEWS*, *BUS* and *CITY*, with JSVM encoder for a statistical analysis. All the test sequences are 97 frames long, and the GOP size is 8. In the experiment, two CGS layers are evaluated. The QP value for the enhancement layer is set to be 10, 20 and 30. The QP value for the base layer is set to be 40. Statistical results for inter MB modes distribution in two CGS layers are shown in Table 3.2 to Table 3.7. It can be shown from the statistical data that the percentage of fine partitioned MB increases as the quantization step size decreases. Moreover, the mode type distribution at enhancement layers greatly depends on the QPD. When QPD has a small value, the motion information at the base layer and its enhancement layer is highly similar, therefore, majority of MB mode type is *BL_pred* at the enhancement layer. On the other hand, when QPD has a large value, the majority of MB mode type is *MODE_8 × 8*. This is because the resulted bit rate is high when the value of QP is small, and thus the system needs to increase the motion information in order to reduce the distortion.

Table 3.2: Statistical analysis of inter modes distributions for *FOREMAN*

MODE	BL	EL_QP30	EL_QP20	EL_QP10
SKIP	4300	1129	1294	833
16×16	2503	1308	1214	757
16×8	637	490	639	616
8×16	674	740	1028	834
8×8	174	1022	3020	4425
BL_pred	0	3602	1008	552
Total	8288	8291	8203	8017

Table 3.3: Statistical analysis of inter modes distributions for *FOOTBALL*

MODE	BL	EL_QP30	EL_QP20	EL_QP10
SKIP	2585	491	429	249
16×16	2302	1315	1189	753
16×8	1126	510	553	613
8×16	1218	744	845	823
8×8	796	1805	3029	3073
BL_pred	0	3412	1754	1586
Total	8027	8277	7799	7097

Table 3.4: Statistical analysis of inter modes distributions for *AKIYO*

MODE	BL	EL_QP30	EL_QP20	EL_QP10
SKIP	8073	50	66	48
16×16	181	215	292	406
16×8	28	41	85	127
8×16	31	54	138	164
8×8	0	68	358	816
BL_pred	0	7888	7377	6755
Total	8316	8316	8316	8316

It is observed that there are correlations between the base layer and its enhancement layers for CGS scalability. Therefore, a good prediction could be achieved if we predict each MB mode partition at the enhancement layer from the corresponding MB at the base layer. Based on our exhaustive experimental results on various video sequences, the following relationships can be found:

Table 3.5: Statistical analysis of inter modes distributions for *NEWS*

MODE	BL	EL_QP30	EL_QP20	EL_QP10
SKIP	7560	114	130	91
16×16	492	447	591	637
16×8	79	84	131	193
8×16	127	127	239	240
8×8	56	276	762	1548
BL_pred	0	7268	6463	5606
Total	8314	8316	8316	8315

Table 3.6: Statistical analysis of inter modes distributions for *BUS*

MODE	BL	EL_QP30	EL_QP20	EL_QP10
SKIP	2460	630	745	544
16×16	1963	1056	838	610
16×8	867	384	566	513
8×16	539	454	618	602
8×8	396	1120	2342	2978
BL_pred	0	2593	1118	971
Total	6225	6237	6227	6218

Table 3.7: Statistical analysis of inter modes distributions for *CITY*

MODE	BL	EL_QP30	EL_QP20	EL_QP10
SKIP	5654	2167	2074	1425
16×16	1892	1646	1769	1250
16×8	159	265	588	679
8×16	551	462	1121	1026
8×8	60	594	2031	3421
BL_pred	0	3182	733	515
Total	8316	8316	8316	8316

- (a). If the MB mode at the base layer is *INTRA* $_{4 \times 4}$ or *INTRA* $_{16 \times 16}$, it is probable that the corresponding MB at its enhancement layer is intra coded. The intra-coded MB at the base layer indicates that one cannot find a good match in the reference frames. Similarly, at enhancement layers, it is also difficult for the MB to find a good match in the reference frames.
- (b). For CGS, enhancement layers refine the transform coefficients generated at the base layer. Therefore, the partition at enhancement layers for each MB should be finer than that at the base layer.
- (c). The hierarchical B frames at low temporal level are generated with large temporal distance, and they have more motion and texture information than that at a high temporal level.

3.3.3 Proposed Fast Mode Decision Scheme 1 for SNR Scalability

Based on the considerations above, three techniques are proposed for SNR scalability in fast mode decision scheme.

3.3.3.1 Selective Intra-Mode Prediction

This proposed scheme is used to distinguish the MBs encoded with intra mode in B frames at enhancement layers. An enhancement layer is a motion and residual information refinement of its base layer. For CGS enhancement layer, BL_pred and $INTRA_4 \times 4$ are frequently selected modes in most cases. Without $INTRA_16 \times 16$, BL_pred and $INTRA_4 \times 4$ can preserve the accuracy of intra prediction well [109]. Therefore, $INTRA_16 \times 16$ is removed to reduce the high computational load of intra coding while keeping the performance.

Figure 3.9 shows the flowchart of our proposed selective intra-mode prediction method, where $MODE_{BL}$ stands for the optimally selected MB mode at the base layer corresponding to the current MB at the enhancement layer. We divide the modes into two classes: *Class_inter* and *Class_intra*. If $MODE_{BL}$ is $INTRA_4 \times 4$ or $INTRA_16 \times 16$, then the corresponding MB at the enhancement layer belongs to *Class_intra*, and the candidate mode set is reduced to BL_pred and $INTRA_4 \times 4$.

The hierarchical B frames at low temporal level are generated with large temporal distance, and they have more motion and texture information than that at a high temporal level. Therefore, MBs in the low temporal level frames have a high probability to be intra coded. As a result, if $MODE_{BL}$ is not intra coded, for high temporal level frames, the corresponding MB at the enhancement layer belongs to *Class_inter*. In order to decide whether an MB is intra coded in low temporal level

frames, we regard $MODE_8 \times 8$ and $INTRA_4 \times 4$ as the representative block sizes of $Class_inter$ and $Class_intra$. The RDcosts for $MODE_8 \times 8$ (RDcost8) and $INTRA_4 \times 4$ (RDcost4) for the MBs at low temporal level frames are estimated. If RDcost4 is less than RDcost8, we assume that the probability of intra coding is high, and the best mode is set to $INTRA_4 \times 4$. On the other hand, if RDcost8 is less than RDcost4, the best mode would belong to $Class_inter$. Then, the following methods are used in the mode decision process for the MBs in high and low temporal level frames which belong to $Class_inter$.

3.3.3.2 Selective Reduction of Candidate Modes

Since enhancement layers have refined motion and residual information of that at the base layer, we can reduce the number of candidate modes for certain MBs. If the MB mode at the base layer is $MODE_8 \times 8$, then the candidate mode set is reduced to BL_pred and $MODE_8 \times 8$. If the MB mode at the base layer is $MODE_16 \times 8$ (or $MODE_8 \times 16$), then the candidate mode set is reduced to BL_pred , $MODE_8 \times 8$ and $MODE_16 \times 8$ (or $MODE_8 \times 16$). Figure 3.10 shows the flowchart of selective reduction of candidate modes method.

3.3.3.3 Selective Residual Prediction at Enhancement layers

This scheme is used to examine whether the MBs at enhancement layers need residual prediction. For a pixel $p_{i,j}$, we use $R_Y(i, j)$, $R_{Cb}(i, j)$ and $R_{Cr}(i, j)$ to denote the coded previous layer residual for luma and chroma information in an MB. The sum of coded residual in the previous layer (SPR) is given as:

$$SPR = \sum_{i=0}^{15} \sum_{j=0}^{15} |R_Y(i, j)| + \sum_{i=0}^7 \sum_{j=0}^7 |R_{Cb}(i, j)|$$

Figure 3.9: Flowchart of proposed selective intra mode prediction for SNR scalability

$$+ \sum_{i=0}^7 \sum_{j=0}^7 |R_{Cr}(i, j)| \quad (3.3)$$

Residual prediction is preformed at enhancement layer if SPR is greater than a threshold θ . Otherwise, there is no residual prediction. The choice of θ provides a tradeoff between coding speed and quality. For most of video sequences, there is a high probability that the value of SPR for each MB is in two ranges: one is from 0

Figure 3.10: Flowchart of selective reduction of candidate modes scheme for SNR scalability

to 5, and the other is greater than 100. This is shown by the experimental results given in Table 3.8. In the experiment, two CGS layers are used. The QP for the base layer (denoted as BLQP) ranges from 40 to 20, and the QP for enhancement layer is set to 10. Since the coding quality will be degraded when θ is greater than 100, θ is selected as less than or equal to 5.

Table 3.8: Statistical analysis of SPR value

Sequence	BLQP	SPR ≤ 5	5 $<$ SPR <100	SPR ≥ 100
FOREMAN	40	96.46%	0	3.540%
	30	83.55%	0	16.45%
	20	56.35%	8.590%	35.06%
FOOTBALL	40	68.98%	0	31.02%
	30	38.96%	0	61.04%
	20	30.23%	2.309%	67.46%
AKIYO	40	99.93%	0	0.07%
	30	99.33%	0	0.67%
	20	96.63%	1.35%	2.02%
NEWS	40	99.05%	0	0.95%
	30	95.69%	0.07%	4.24%
	20	88.69%	2.69%	8.62%
BUS	40	88.35%	0	11.65%
	30	60.54%	0.20%	39.26%
	20	25.39%	4.38%	70.24%
CITY	40	98.52%	0	1.48%
	30	89.09%	0.34%	10.57%
	20	52.53%	9.97%	35.51%

3.3.4 Enhanced Fast Mode Decision Scheme 2 for SNR Scalability

In this section, substantial improvement is made by proposing an enhanced fast mode decision scheme with adaptation to the QP difference between the base layer and its corresponding enhancement layers. According to different strength of motion correlation, different sets of candidate modes are used to reduce the complexity.

Figure 3.11 shows the flowchart of our proposed scheme. QPD stands for the QP difference of the base layer and its enhancement layer, $MODE_{BL}$ and $MODE_{BL2}$ are the best two modes generated at the base layer, and they are part of the prediction mode set for enhancement layers. These best two modes can be generated after the rate distortion optimization at the base layer by comparing the rate distortion

Figure 3.11: Flowchart of proposed enhanced fast decision scheme for SNR scalability

cost.

In our proposed scheme, after performing the selective intra coding method, the MBs in inter frames can be divided into inter-coded and intra-coded MBs. If the MB is classified into intra-coded one, then the candidate mode set is BL_{pred} and $INTRA_{4 \times 4}$. On the other hand, if the MB is classified into inter-coded one, we need to examine the strength of motion correlation between the base layer

and its enhancement layers. In order to solve this problem, a threshold QP_{thr} is set to distinguish whether the base layer and its enhancement layers have a strong motion correlation:

$$\begin{cases} \text{Weak Motion Correlation} & \text{if } QPD < QP_{thr} \\ \text{Strong Motion Correlation} & \text{otherwise} \end{cases} \quad (3.4)$$

The choice of QP_{thr} provides a trade-off between the coding speed and quality. A suitable value can be determined based on extensive experiments on test sequences. It is set to be 15 in our experiments. The experimental setup is shown in Table 3.9.

Table 3.9: Experimental Setup

Test Sequences		<i>FOREMAN, FOOTBALL, BUS, CITY and CREW</i>
QP Setting	Base	40
	Enhancement	35, 30, 25, 20, 15 and 10
Resolution	Base	QCIF
	Enhancement	CIF

If the layers have strong motion correlation, it is most probable that the MBs at the enhancement layers are coded in BL_{pred} . Therefore, the candidate mode set is reduced to BL_{pred} , $MODE_{BL}$ and $MODE_{BL2}$. Otherwise, the original candidate mode reduction method proposed in Section 3.3.3 will be used.

3.4 Temporal Scalable Video Coding

Temporal scalability is realized by extracting of video of multiple frame rates from a single coded bit stream. The coding efficiency and the degree of temporal scalability are a function of size of a group of pictures (GOP), defined as the number of

consecutive frames that can be decoded independently from the rest of the video sequence. The temporal scalability is achieved by decoding subsets of the GOP consisting of equally-spaced frames.

For hybrid video codecs, temporal scalability can generally be enabled by restricting motion-compensated prediction to reference pictures. The prior video coding standards MPEG-1,-2,-4 and H.263 all support temporal scalability to some degree. H.264/AVC provides a significantly increased flexibility for temporal scalability because of its reference picture memory control. There are three kinds of hierarchical prediction structures for enabling temporal scalability in SVC [29].

(i) **Prediction structure with hierarchical B-pictures:**

Temporal scalability with dyadic temporal enhancement layers can be very efficiently provided with the concept of hierarchical B-pictures [92, 110] as illustrated in Figure 3.12(a). The numbers in this figure specify the coding order and the symbols T_k specify the temporal layers with k representing the corresponding temporal layer level. Only previous pictures of the temporal base layer are used for prediction of the current pictures. For enhancement layer, the current pictures can be bidirectionally predicted by using the two neighboring pictures of a lower temporal layer as references.

(ii) **Nondyadic hierarchical prediction structure:**

Hierarchical prediction structures are not restricted to the dyadic case. Figure 3.12(b) illustrates a nondyadic hierarchical prediction structure, which provides 2 independently decodable sub-sequences with $\frac{1}{9}th$ and $\frac{1}{3}rd$ of the full frame rate. It should further be noted that it is possible to modify the prediction structure of the temporal base layer arbitrarily, e.g., in order to increase the coding efficiency [29].

(iii) **Prediction structure with a structural encoding/decoding delay of zero:**

Figure 3.12(c) shows a hierarchical prediction structure, which does not employ motion-compensated prediction from pictures in the future. Although this structure provides the same degree of temporal scalability as the prediction structure of Figure 3.12(a), its structural delay is equal to zero compared to 7 pictures for the prediction structure in Figure 3.12(a). However, such low-delay structures typically decrease coding efficiency.

SVC is based on a layered representation with multiple dependencies. To achieve temporal scalability, the construction of frame hierarchies is essential. In coding performance comparison with the motion compensated temporal filtering (MCTF), a hierarchical B frames scheme can provide higher coding efficiency for temporal scalability in SVC [92]. Therefore, in the following section, we are going to elaborate more on hierarchical B prediction structure.

3.4.1 Prediction Structure with Hierarchical B-pictures

Consider Figure 3.13, which shows the temporal decomposition of a group of 16 pictures. According to Equation 3.5, there are 4 decomposition stages. The first picture is independently coded as an instantaneous decoding refresh (IDR) picture, and all remaining pictures are coded in "B..BI" groups of pictures using the concept of hierarchical B pictures. The difference between this scheme and MCTF is that with hierarchical B pictures, the temporal update step is omitted, leading to a purely predictive coding structure [91].

$$\text{TemporalResolution} = \log_2 \text{GOPSize} \quad (3.5)$$

Figure 3.12: Hierarchical prediction structures for enabling temporal scalability. (a) Coding with hierarchical B-pictures. (b) Nondyadic hierarchical prediction structure. (c) Hierarchical prediction structure with a structural encoding/decoding delay of zero.

Figure 3.13: Temporal scalability with GOP size is 16

As shown in Figure 3.13, the B pictures of the lowest temporal level (labelled as B^1 and frame index is 8) use only the surrounding anchor frames A for motion-compensated prediction. For a group of 16 frames as shown in Figure 3.13, the frame indexes (fi) and temporal level t have the following relationship:

$$fi(t) = \begin{cases} 2^{(\text{TemporalResolution}-1)} & t = 0 \\ \{\frac{1}{2}fi(t-1), \frac{3}{2}fi(t-1)\} & t = 1, 2, 3 \end{cases} \quad (3.6)$$

The B pictures of highest temporal level B^4 are predicted from anchor frames A, B pictures B^1 , B pictures B^2 and B pictures B^3 , and they are transmitted as non-referencing pictures. If only pictures B^1 and anchor frames A are transmitted, the reconstructed sequence at the decoder side has one eighth of the temporal resolution of the input sequence. By additionally transmitting pictures B^2 , the decoder can reconstruct an approximation of the picture sequence that has one

fourth of the temporal resolution of the input sequence. Finally, if the remaining B pictures are transmitted, a reconstructed version of the original input sequence with the full temporal resolution is obtained. In general, by using n decomposition stages, the decomposition structure can be designed in a way that n levels of temporal scalability are provided. In addition, the coding order for hierarchical prediction structures has to be chosen in a way that reference pictures are coded before they are employed for motion-compensated prediction.

3.4.2 Correlation Between the Base Layer and Enhancement Layers

For Inter frame coding, the MBs are classified into coarse partitioned MBs (such as *MODE_SKIP*, *MODE* $_{16 \times 16}$, etc.) and fine partitioned MBs (such as *MODE* $_{8 \times 8}$, *INTRA* $_{4 \times 4}$, etc.). The number of fine partitioned MBs depends on the temporal distance of the current frame and reference frames [101]. Experiments on various video sequences are designed to investigate the statistical relationships of temporal distances and MB mode distributions. Figure 3.14 shows the experimental results for *MOBILE* sequence with GOP size of 16. It can be seen that as the temporal distance between certain motion compensation pair increases, the average percentage of fine partitioned MBs increases as well. When QP is fixed, at low temporal level, the hierarchical B frames are generated with large temporal distance. The temporal correlation between the current frame and its referencing frames is low; consequently, the percentage of fine partitioned MBs is high. Therefore, in temporal scalability, the partition of each MB in the low temporal level frames is finer than that of the corresponding MBs in the high temporal level frames and this shows that correlation also exists between the base layer and its enhancement layers in temporal scalability.

Figure 3.14: Average percentage of fine partitioned MBs for *MOBILE* sequence

3.4.3 Proposed Fast Mode Decision Scheme for Temporal Scalability

Our fast mode decision scheme is based on the correlation among low and high temporal level frames. A picture coded in a higher temporal level can use other pictures in the lower temporal levels as references. Meanwhile, a picture in a lower temporal levels cannot use picture from a higher temporal level as a reference. According to our intensive experiments, the best prediction mode of each MB in the current frame is most similar to the optimal mode of the corresponding MBs in its reference frames [99].

For the frames in temporal level 0, only the anchor intra-coded frames can

be used for motion-compensated prediction. Therefore, the original exhaustively block matching method is used in our scheme to search for the best mode for each MB in the frames at temporal level 0. In order to illustrate our idea, we take Figure 3.13 as an example. Frame 8 is estimated by using the block matching method without any fast mode decision scheme. Frame 4 and frame 12 are at temporal level 1. The best mode in frame 8 is the candidate mode for frame 4 and frame 12. Similarly, at temporal level 2, the best modes in frame 4 and frame 8 are the candidate modes for frame 6. Therefore, each frame has one or two candidate modes which are generated from the backward and/or forward reference frames. Our proposed fast mode decision algorithm for temporal scalability is discussed as follows.

3.4.3.1 Determination of Low- and High-Motion MBs

We divide the MB into two classes, low motion MB and high motion MB. In natural video sequences, many MBs, especially the MBs in the background area, exhibit similar motion even if not still, and are thus considered low motion MBs. In this section, we propose a method to distinguish the low and high motion MBs by estimating the motion energy for each MB. After exhaustively search for the best mode for each MBs in the frames at temporal level 0, each possible MB partition (16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 , 4×4) has independent motion vectors (mv_{x0}, mv_{y0}) and (mv_{x1}, mv_{y1}) for bi-direction prediction. The motion energy for each possible partition, denoted as MEb_0 and MEb_1 , can be calculated as follows:

$$MEb_i = |mv_{xi}| + |mv_{yi}|, i = 0, 1 \quad (3.7)$$

Note that the energy computed from equation (3.7) is the l_1 -norm of vectors and it is equivalent to that defined in [111] where l_2 -norm is used. We now use MEB_0 and MEB_1 to denote the average motion energy of an MB with respect to the backward and forward reference frames. They can be formulated as follows:

$$MEB_i = \frac{1}{N} \sum_{n=1}^N MEb_{ni}, i = 0, 1 \quad (3.8)$$

where N represents the total number of motion vectors of concerned MB. In our scheme, a threshold τ is set to distinguish high motion and low motion MBs as follows:

$$\begin{cases} \text{High Motion MB} & \text{if } MEB_0 > \tau \text{ or } MEB_1 > \tau \\ \text{Low Motion MB} & \text{otherwise} \end{cases} \quad (3.9)$$

In our experiments, high motion MB is defined as an MB that shows at least 10 pixel movement relative to reference frames. Motion vector resolution in our experiments is 1/4 pixel, therefore, the resulted motion vector value is 40. Therefore, the threshold τ is set to 40. Based on the experimental results on many test sequences, we found that setting achieves good and consistent results in most of the sequences.

3.4.3.2 Candidate Modes Assignment for Low- and High- Motion MBs

As shown in Figure 3.15, every square represents a 16×16 MB in the reference frame [112]. For each MB in the current frame, the co-located MB locates at the same position in the reference frame. The arrow represents an average motion vector of all the partitions in the current MB. The dotted MB is the motion compensated one which most closely matches the current block. The dominant MB has the

largest overlapping with the motion compensated one.

Figure 3.15: Definition of the dominant block from the motion estimated block

It is believed that motion is usually continuous, i.e. a directional feature of the current MB is similar to that of the motion compensated MB. Therefore, in our scheme we need to examine whether the dominant MB is at the same position of the co-located one. For low motion MBs, the dominant MB tends to have the same position of the co-located one. As a result, the MB mode of the co-located MB in the reference frame will be the candidate mode for the corresponding MB in the current frame. On the other hand, for high motion MBs, the dominant one tends to locate at the neighborhoods of the co-located MB. Consequently, the best modes of the co-located MB as well as its neighboring ones are composed of the candidate mode set for the corresponding MB in the current frames.

3.4.3.3 Overall Algorithm in Temporal Scalability

The flowchart of our scheme for temporal scalability is shown in Figure 3.16. If the MB mode at forward and/or backward reference frames (denoted by $MODE_{ref(i)}$) are intra coded, the rate distortion cost for $MODE_{8 \times 8}$ (RDcost8) and $INTRA_{4 \times 4}$ (RDcost4) are estimated. If RDcost4 is less than RDcost8, we assume that the

Figure 3.16: Flowchart of proposed scheme for temporal scalability

probability of intra coding is high, and the best mode is set to be $INTRA_4 \times 4$. On the other hand, if $RDcost8$ is less than $RDcost4$, we can regard that the best mode would be inter coded. Then $MODE_{16 \times 8}$ and $MODE_{8 \times 16}$ will be members of candidate mode set. On the other hand, if the MB mode at forward and/or backward reference frames is not intra coded, we need to examine whether the MB is in low temporal level frames. As discussed previously, low temporal level frames tend to have finer mode partition size than high temporal level frames. Moreover, the generated large distortion in the low temporal level frames coding will propagate and affect the coding efficiency of high temporal level frames. Therefore, it is

important to increase the motion estimation accuracy in low temporal level frames. For high motion MBs in low temporal level frames, it is difficult to find a temporal correlation from its reference frames. Instead, we need to consider the spatial correlation among MBs in the current frame. This is due to the fact that there is usually a high correlation between pixels that close to each other. Therefore, for high motion MB at low temporal level frames, the best modes of above and left MBs, denoted by $MODE_{above}$ and $MODE_{left}$, are considered as the candidate modes for current MB. On the other hand, for MBs at high temporal level frames or low motion MB at low temporal level frames, the best modes of MBs in the forward and backward reference frames, denoted by $MODE_{ref0}$ and $MODE_{ref1}$, are considered as the candidate modes for current MB.

3.5 Experimental Results

The performance of our proposed fast mode decision schemes for spatial, SNR, temporal and combined scalable video coding is evaluated through simulation studies. Our newly proposed schemes are implemented in JSVM encoder [91]. The test platform used is Intel Pentium IV, 1.83GHz CPU, 2.0GB RAM with Windows XP professional operating system. Experiments make use of video sequences that are common in the video coding community. Specifically, sequences consist of *FOREMAN*, *FOOTBALL*, *BUS*, *HARBOUR*, *CITY*, *CREW*, *AKIYO* and *NEWS*, covering slow motion, the moderate and high motion activity respectively. We only consider the two-layer case. All the test sequences are intra coded for the first frame (I frame) and followed with sub-sequent bi-directional inter-coded frames (B frame). Therefore, the encoding sequence of test video is IBBBB. The GOP size is set to be 16 for temporal scalable video coding. For other kind of salabilities, the

GOP size is set to be 8. The detailed encoder parameters are given in Table 3.10.

Table 3.10: Simulation condition

All Tested Video Sequences	<i>FOREMAN, FOOTBALL, BUS, CITY HARBOUR, CREW, AKIYO and NEWS</i>
Encoder Configurations and Platform	Software: JSVM encoder M.E. search range: ± 32 pixels MV resolution: $\frac{1}{4}$ pixel accuracy RDO: enabled GOP size: 8 and 16 (temporal scalability) Reference frame number: 1 Encoding sequence: IBBBB Entropy coding mode: CABAC Macroblock adaptive inter-layer prediction: enabled Machine: Intel Pentium IV, 1.82GHz CPU, 2.0GB RAM with Windows XP.

The analysis of the complexity of SVC is presented in term of encoding time. The time comparison between JSVM encoder and our proposed encoder with low complexity is showed. The JSVM encoder with high complexity showed the long encoding time than our proposed encoder. Not only the encoding time, but also the video quality and bit rate will be compared. Performance evaluation of the encoded video is based on subjective survey and objective evaluation. Subjective survey is based on the personal opinion and objective evaluation is based on the calculation of the average of the peak signal-to-noise ratio (PSNR) values of the frames in a video sequence. In our experiments, the results of fast mode decision methods for luminance blocks are shown, therefore, PSNR of the luminance component Y, known as Y-PSNR is used as an objective metric of video quality. For a given frame with $N_x \times N_y$ pixels with 8-bits per pixel, the PSNR is calculated from the

mean square error (MSE):

$$MSE = \frac{1}{N_x \times N_y} \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} [F(x, y) - R(x, y)]^2 \quad (3.10)$$

$$PSNR = 10 \cdot \log_{10} \frac{255^2}{MSE} \quad (3.11)$$

The testing parameters in our experiments include the average time saving (TS), Y-PSNR difference ($\Delta PSNR$) and the bit rate difference (ΔBR) which are defined as follows:

$$TS = \frac{T_{JSVM} - T_{proposed}}{T_{JSVM}} \times 100\% \quad (3.12)$$

$$\Delta PSNR = PSNR_{proposed} - PSNR_{JSVM} \quad (3.13)$$

$$\Delta BR = \frac{BitRate_{proposed} - BitRate_{JSVM}}{BitRate_{proposed}} \times 100\% \quad (3.14)$$

where T_{JSVM} , $PSNR_{JSVM}$ and $BitRate_{JSVM}$ are the encoding time, Y-PSNR and bit rate of JSVM encoder, and $T_{proposed}$, $PSNR_{proposed}$ and $BitRate_{proposed}$ represent the encoding time, Y-PSNR and bit rate of the modified encoder with our proposed fast mode decision scheme, respectively. In addition, Bjontegaard delta PSNR (BDPSNR) and Bjontegaard delta bit rate (BDBR) are also used as testing parameters in some of our experiments [113]. BDPSNR and BDBR are used to represent the average PSNR and bit rate differences between the rate distortion curves derived from JSVM encoder and the proposed fast algorithm, respectively.

3.5.1 Spatial Scalability

The performance of our proposed fast mode decision scheme for spatial scalable video coding is evaluated in this section. The test condition is shown in Table 3.11. In our experiments, QP value ranges from 10 to 40, and the same QP value is used for both base layer and enhancement layer. The spatial resolution for the base layer is 176×144 luma samples per picture (QCIF), while the spatial resolution for enhancement layer is 352×288 luma samples per picture (CIF). All image frames used 4:2:0 color sampling.

Table 3.11: Simulation condition for spatial scalability

Test Sequences		<i>FOREMAN, FOOTBALL, BUS, CITY AKIYO, CREW and NEWS</i>
QP Setting	Base	10, 20, 30 and 40
	Enhancement	10, 20, 30 and 40
Resolution	Base	QCIF
	Enhancement	CIF
Frame Rate	Base	15Hz
	Enhancement	15Hz

Table 3.12: Simulation results for spatial scalability with QP value of 10

Sequence	Δ PSNR [dB]	Δ BR [%]	TS [%]
<i>FOREMAN</i>	-0.047	0.020	41.28
<i>FOOTBALL</i>	-0.051	0.229	49.70
<i>BUS</i>	-0.068	0.291	37.27
<i>CITY</i>	-0.012	0.430	38.62
<i>AKIYO</i>	-0.019	0.029	69.70
<i>CREW</i>	-0.167	0.567	45.96
<i>NEWS</i>	-0.082	-0.630	62.82
Average	-0.064	0.134	49.34

Table 3.12, 3.13, 3.14 and Table 3.15 present the enhancement layer Y-PSNR, bit rate and time savings of the proposed scheme in comparison with JSVM encoder at different QP settings. It can be seen that enabling fast mode decision mechanism

Table 3.13: Simulation results for spatial scalability with QP value of 20

Sequence	ΔPSNR [dB]	ΔBR [%]	TS [%]
<i>FOREMAN</i>	0.006	1.010	45.47
<i>FOOTBALL</i>	-0.011	0.121	46.41
<i>BUS</i>	-0.001	0.691	40.54
<i>CITY</i>	-0.031	1.220	49.62
<i>AKIYO</i>	-0.033	0.225	75.49
<i>CREW</i>	-0.061	0.368	47.18
<i>NEWS</i>	-0.030	0.240	69.37
Average	-0.023	0.554	53.44

Table 3.14: Simulation results for spatial scalability with QP value of 30

Sequence	ΔPSNR [dB]	ΔBR [%]	TS [%]
<i>FOREMAN</i>	-0.094	0.883	56.03
<i>FOOTBALL</i>	-0.022	0.379	46.88
<i>BUS</i>	-0.006	0.886	51.72
<i>CITY</i>	-0.085	1.040	64.55
<i>AKIYO</i>	-0.046	0.005	77.54
<i>CREW</i>	-0.074	0.248	56.32
<i>NEWS</i>	-0.056	0.100	73.94
Average	-0.055	0.506	70.00

can reduce the overall running time by 70%, which is equivalent to a speedup of about 4 times. Moreover, our scheme achieves consistent time saving over a large bit rate range for a wide variety of video sequences with slow motion, moderate

Table 3.15: Simulation results for spatial scalability with QP value of 40

Sequence	ΔPSNR [dB]	ΔBR [%]	TS [%]
<i>FOREMAN</i>	-0.134	-0.864	61.65
<i>FOOTBALL</i>	-0.077	0.434	54.67
<i>BUS</i>	-0.139	0.118	62.47
<i>CITY</i>	-0.116	0.158	72.86
<i>AKIYO</i>	-0.011	0.120	79.13
<i>CREW</i>	-0.162	-1.291	68.80
<i>NEWS</i>	-0.043	-0.154	77.71
Average	-0.097	-0.211	68.18

Figure 3.17: Rate distortion curve for *FOREMAN* in spatial scalable video coding

motion and high motion activities. The improvement is achieved with a negligible loss in both bit rate and Y-PSNR, as confirmed by the bit rate and Y-PSNR values in the tables and the rate distortion (RD) curves in Figure 3.18, Figure 3.17 and Figure 3.19. RD curves describe the rate distortion relationship for *FOREMAN*, *FOOTBALL* and *BUS* sequences. From these figures, we can see that there is little difference between the two RD curves in each figure. Figure 3.20 shows the comparison of reconstructed video quality of our scheme and JSVM encoder.

Figure 3.18: Rate distortion curve for *FOOTBALL* in spatial scalable video coding

3.5.2 SNR Scalability

The performance of our proposed fast mode decision scheme for SNR scalable video coding is evaluated through simulation studies.

3.5.2.1 Fast Mode Decision Scheme 1

Table 3.17 and Table 3.18 show the coding results of JSVM and our proposed scheme with different QP settings at the enhancement layer. The total encoding time is reduced up to 48%. Figure 3.21 and Figure 3.22 present the rate distortion curves for *CITY* and *HARBOUR*. From these figures, we can conclude that this scheme can achieve consistent time saving over a large bit rate range with negligible loss in Y-PSNR and minor increments in bit rate.

Figure 3.19: Rate distortion curve for *BUS* in spatial scalable video coding

Table 3.16: Simulation condition for fast mode decision scheme 1 in SNR scalability

Test Sequences		<i>FOREMAN, FOOTBALL, BUS, AKIYO CITY, NEWS and CREW</i>
QP Setting	Base	40
	Enhancement	10 and 20
Resolution	Base	QCIF
	Enhancement	QCIF
Frame Rate	Base	15Hz
	Enhancement	15Hz

Figure 3.20: The comparison of reconstructed video quality, the 2nd, 4th and 6th frames of *FOOTBALL*. (a) reconstructed with our scheme for spatial scalability, (b) reconstructed with JSVM encoder.

Table 3.17: Simulation results in SNR scalability with QP value of 10

Sequence	ΔPSNR [dB]	ΔBR [%]	AVTS [%]
FOREMAN	-0.050	0.374	41.98
FOOTBALL	0.064	1.022	48.95
CITY	-0.092	0.132	41.24
HARBOUR	-0.056	0.285	44.07
AKIYO	-0.038	0.054	51.32
NEWS	-0.048	0.097	49.48
Average	-0.036	0.327	46.17

3.5.2.2 Enhanced Fast Mode Decision Scheme 2

The test condition is shown in Table 3.19. In our experiments, five standard test sequences including *FOREMAN*, *FOOTBALL*, *BUS*, *CITY* and *CREW* have been tested. The total number of frames is 100 for each sequence.

Two experiments were set up to examine the performance of our proposed

Table 3.18: Simulation results in SNR scalability with QP value of 20

Sequence	ΔPSNR [dB]	ΔBR [%]	AVTS [%]
FOREMAN	-0.080	0.574	41.27
FOOTBALL	0.013	0.848	47.48
CITY	-0.115	0.208	38.74
HARBOUR	-0.096	0.280	42.64
AKIYO	-0.055	0.132	49.57
NEWS	-0.078	0.198	47.99
Average	-0.068	0.373	44.62

Figure 3.21: Rate distortion curve for *CITY* in SNR scalable video coding

scheme. In the first experiment, the QP values for base layer and enhancement layer are set to be 40 and 30, respectively. In the second experiments, the QP values for base layer and enhancement layer are set to be 30 and 20, respectively. The QPD for both experiments are 10, therefore, the base layer and its enhancement

Figure 3.22: Rate distortion curve for *HARBOUR* in SNR scalable video coding

Table 3.19: Simulation condition for enhanced fast mode decision scheme 2 in SNR scalability

Test Sequences		<i>FOREMAN, FOOTBALL, BUS, CITY and CREW</i>
QP Setting	Base	20 and 30
	Enhancement	30 and 40
Resolution	Base	QCIF
	Enhancement	QCIF
Frame Rate	Base	15Hz
	Enhancement	15Hz

layer has strong motion correlation. The experimental results are given in Table 3.20 and 3.21. QP_b and QP_e are used to denote the QP values at the base layer and its enhancement layers. Note that in the table, positive values mean increments and negative values mean decrements. It can be seen that our scheme achieves encoding time saving without visual quality degradation. The enhanced fast mode decision scheme 2 provides a saving of 59-66% in encoding time. However, the previously proposed scheme 1 provided improvements of only 39-49%. Figure 3.23 shows the comparison of reconstructed video quality of JSVM encoder and our scheme. High quality encoded video is still achieved while the encoding time is significantly faster. The result is attributed to the fact that the computation complexity is reduced, and the encoding time saving is achieved as well as the video quality is maintained.

Table 3.20: Simulation results with $QP_b = 40$ and $QP_e = 30$ for SNR scalable video coding

Sequence		ΔBR [%]	$\Delta PSNR$ [dB]	TS [%]
FOREMAN	Scheme 1	1.866	0.0629	41.10
	Scheme 2	0.094	-0.1636	64.22
FOOTBALL	Scheme 1	3.420	0.1463	45.99
	Scheme 2	1.295	-0.0403	62.02
BUS	Scheme 1	1.001	0.0169	41.68
	Scheme 2	-0.097	-0.1735	61.64
CITY	Scheme 1	0.833	0.0151	39.32
	Scheme 2	-0.524	-0.1515	65.84
CREW	Scheme 1	2.401	0.0668	42.64
	Scheme 2	-0.008	-0.1594	66.01
Average	Scheme 1	1.904	0.062	42.15
	Scheme 2	0.152	-0.138	63.95

Table 3.21: Simulation results with $QP_b = 30$ and $QP_e = 20$ for SNR scalable video coding

Sequence		ΔBR [%]	ΔPSNR [dB]	TS [%]
FOREMAN	Scheme 1	2.073	0.0324	44.12
	Scheme 2	0.197	-0.1792	61.34
FOOTBALL	Scheme 1	3.146	0.1257	49.90
	Scheme 2	0.912	-0.0150	59.54
BUS	Scheme 1	1.837	0.0263	44.21
	Scheme 2	0.998	-0.0767	58.25
CITY	Scheme 1	0.622	0.0035	40.88
	Scheme 2	0.039	-0.1374	64.14
CREW	Scheme 1	3.431	0.1323	45.67
	Scheme 2	1.033	-0.0612	60.39
Average	Scheme 1	2.222	0.064	44.96
	Scheme 2	0.636	-0.094	60.73

Figure 3.23: The comparison of reconstructed video quality, the 20th, 21st, 22nd and 23rd frames of *FOREMAN*. (a) reconstructed with our scheme for SNR scalability, (b) reconstructed with JSVM encoder.

3.5.3 Temporal Scalability

The performance of our proposed fast mode decision scheme for temporal scalable video coding is evaluated in this section. The test condition is shown in Table 3.22. In this experiment, all the six test sequences have been tested. Only one enhancement layer case is considered. The total number of frames is 97 for each sequence, and the group of picture size is 16. Enhancement layer frames are the output frames that have full temporal resolution of input frames. Base layer frames are the output frames that have half temporal resolution of input frames. In our case, the frame rate of enhancement layer frames and base layer frames are 15 and 7.5 fps, respectively.

Table 3.22: Simulation condition for temporal scalable video coding

Test Sequences		<i>FOREMAN, FOOTBALL, BUS, HARBOUR, CITY and CREW</i>
QP Setting	Base	15, 25 and 35
	Enhancement	15, 25 and 35
Resolution	Base	QCIF
	Enhancement	QCIF
Frame Rate	Base	7.5Hz
	Enhancement	15Hz

Table 3.23: Simulation results in temporal scalability with QP value of 15

Sequence	Δ PSNR [dB]	Δ BR [%]	TS [%]
FOREMAN	-0.039	2.637	32.41
FOOTBALL	-0.046	0.915	27.84
BUS	-0.005	1.483	27.98
HARBOUR	0.002	1.026	34.14
CITY	0.001	2.095	32.11
CREW	-0.025	1.894	27.64
Average	-0.019	1.675	29.85

The average bit rate and Y-PSNR differences in terms of Δ BR and Δ PSNR and the average time saving in this experiment are shown in Table 3.23, Table 3.24 and

Table 3.24: Simulation results in temporal scalability with QP value of 25

Sequence	Δ PSNR [dB]	Δ BR [%]	TS [%]
FOREMAN	-0.043	3.185	39.13
FOOTBALL	0.006	1.963	31.13
BUS	-0.002	2.362	32.46
HARBOUR	-0.002	1.797	36.75
CITY	0.018	3.965	41.51
CREW	-0.021	3.758	31.09
Average	-0.007	2.838	35.35

Table 3.25: Simulation results in temporal scalability with QP value of 35

Sequence	Δ PSNR [dB]	Δ BR [%]	TS [%]
FOREMAN	0.008	9.61	51.01
FOOTBALL	0.024	5.790	37.43
BUS	0.007	7.588	44.59
HARBOUR	-0.003	5.216	53.72
CITY	0.079	8.852	59.66
CREW	0.007	9.318	45.57
Average	0.021	7.729	48.59

Table 3.25 for different QP settings. The results show that the proposed method is very effective in reducing the encoding time, especially for the sequence with high motion and fine detail. The total encoding time is reduced up to 29.85% for low QP value (high bit rates) and then increases to around 50% for high QP value (low bit rates). Figure 3.24 presents the rate distortion curves for the output frames that have full temporal resolution for *FOOTBALL*. High quality encoded video is still achieved, and the encoding time is significantly reduced for low QP value. From this figure, we can conclude that our scheme can achieve consistent time saving over a large bit rate range with negligible loss in Y-PSNR and increments in bit rate.

However, as shown in Table 3.25, the bit rate increase is not negligible, some increment is as high as 9%. This is due to the fact that our fast mode decision

Figure 3.24: Rate distortion curve for *FOOTBALL* in temporal scalable video coding

scheme is optimized to work perfectly for low QP value (high bit rates). For high QP value (low bit rates), most MBs are coarse partitioned MBs (such as $MODE_{16 \times 16}$, $MODE_{SKIP}$, etc) at enhancement layer regardless of long temporal distance and high motion MBs. Therefore, we modified the scheme and the flowchart of our new scheme is shown in Figure 3.25. Our modified scheme are tested and testing condition is similar to Table 3.22 except that the quantization parameters for base layer and enhancement layer are set to be 40, 30 and 20. The results are shown in Table 3.26, 3.27 and 3.28. Figure 3.26 shows the comparison of reconstructed video quality of JSVM encoder and our scheme. The results attributed to the fact that the computation complexity is reduced, and the encoding

time saving is achieved as well as the video quality is maintained.

Figure 3.25: Flowchart of modified scheme for temporal scalability for high QP value

3.5.4 Spatial and SNR Scalability

The performance of our proposed fast mode decision scheme for combined spatial and SNR scalability is evaluated in this section. The test condition is shown in Table 3.29.

The experimental results are given in Tables 3.30 and Figure 3.27 and 3.28. It can be seen that our scheme achieves encoding time saving significantly with minor

Table 3.26: Simulation results in temporal scalability with QP value of 40

Sequences	Base Layer		Enhancement Layer		TS[%]
	Δ PSNR[dB]	Δ BR[%]	Δ PSNR[dB]	Δ BR[%]	
FOREMAN	-0.111	0.079	-0.131	0.149	59.27
FOOTBALL	-0.002	2.573	-0.024	3.784	42.03
BUS	-0.029	1.230	-0.080	1.559	49.79
HARBOUR	-0.034	0.629	-0.044	0.282	63.09
CITY	-0.030	0.070	-0.037	-0.076	66.53
CREW	-0.059	1.233	-0.099	0.659	54.91
Average	-0.044	0.969	-0.053	1.059	55.94

Table 3.27: Simulation results in temporal scalability with QP value of 30

Sequences	Base Layer		Enhancement Layer		TS[%]
	Δ PSNR[dB]	Δ BR[%]	Δ PSNR[dB]	Δ BR[%]	
FOREMAN	-0.064	1.689	-0.083	2.442	43.23
FOOTBALL	0.066	1.533	0.052	2.301	32.12
BUS	-0.014	0.718	-0.031	1.393	37.29
HARBOUR	-0.009	0.508	-0.012	0.667	48.72
CITY	-0.040	0.275	-0.043	0.898	54.19
CREW	-0.030	1.642	-0.058	0.277	33.91
Average	-0.015	1.061	-0.030	1.663	41.58

coding quality degradation. By comparing Figure 3.27 and 3.28, the difference between two RD curves at high bit rate is larger for *FOOTBALL* sequence. This is because *FOOTBALL* represents a sequence with high motion and fine details.

Table 3.28: Simulation results in temporal scalability with QP value of 20

Sequences	Base Layer		Enhancement Layer		TS[%]
	Δ PSNR[dB]	Δ BR[%]	Δ PSNR[dB]	Δ BR[%]	
FOREMAN	-0.050	1.162	-0.053	1.687	30.29
FOOTBALL	0.055	0.813	0.037	1.165	26.85
BUS	-0.006	0.378	-0.009	1.027	26.71
HARBOUR	-0.013	0.283	-0.009	0.594	37.74
CITY	-0.008	0.636	-0.023	0.931	35.72
CREW	-0.005	0.913	-0.020	1.578	26.81
Average	-0.004	0.698	-0.013	1.164	30.69

Figure 3.26: The comparison of reconstructed video quality, the 60th, 61st, 62nd and 63rd frames of *BUS*. (a) reconstructed with our scheme for temporal scalability, (b) reconstructed with JSVM encoder.

Table 3.29: Simulation condition for combined spatial and SNR scalability

Test Sequences		<i>FOREMAN, FOOTBALL, BUS, CITY and CREW</i>
QP Setting	Base	20 and 30
	Enhancement	30 and 40
Resolution	Base	QCIF
	Enhancement	CIF
Frame Rate	Base	15Hz
	Enhancement	15Hz

The motion correlation between the base layer and the enhancement layer is lower compared to low motion sequence.

Figure 3.27: Rate distortion curve for *FOREMAN* in combined spatial and CGS scalable video coding

Table 3.30: Simulation results for combined spatial and CGS scalability

Sequences	BDPSNR [dB]	BDBR [%]	AVTS [%]
FOREMAN	-0.037	0.698	51.56
FOOTBALL	-0.128	0.921	61.02
BUS	-0.035	0.049	53.07
HARBOUR	-0.032	0.209	51.46
CITY	-0.039	0.384	49.27
CREW	-0.065	1.102	52.14
Average	-0.056	0.561	53.09

Figure 3.28: Rate distortion curve for *FOOTBALL* in combined spatial and CGS scalable video coding

3.5.5 Spatial and Temporal Scalability

The performance of our proposed fast mode decision scheme for combined spatial and temporal scalable video coding is evaluated in this section. The test condition is shown in Table 3.31.

Table 3.31: Simulation condition for combined spatial and temporal scalability

Test Sequences		<i>FOREMAN, FOOTBALL, BUS</i>
QP Setting	Base	15, 25 and 35
	Enhancement	15, 25 and 35
Resolution	Base	QCIF
	Enhancement	CIF
Frame Rate	Base	7.5Hz
	Enhancement	15Hz

Table 3.32 to Table 3.33 show the coding results of JSVM and our proposed scheme. T&S frames are the frame that both temporally and spatially enhanced. S frames are the frames that only spatially enhanced. There are two enhancement layers corresponding to one base layer in spatial and temporal scalable video coding. One enhancement layer is spatially enhanced, and the other is spatially and temporally enhanced. Therefore, for each test, two kinds of reconstructed frames are generated. The results show that the proposed method is very effective in reducing the encoding time, especially for the sequence with high motion and fine detail. The total encoding time is reduced up to 63%. Figure 3.29 presents the rate distortion curves for *FOREMAN*. From these figures, we can conclude that our scheme can achieve consistent time saving over a large bit rate range with negligible loss in Y-PSNR and increments in bit rate consumption.

Table 3.32: Simulation results for combined spatial and temporal scalability with QP value is 25

Sequence	T & S Frame		S Frame		TS
	Δ PSNR [dB]	Δ BR [%]	Δ PSNR [dB]	Δ BR [%]	
<i>FOREMAN</i>	-0.059	1.752	-0.041	1.397	57.92
<i>FOOTBALL</i>	-0.100	1.293	-0.091	1.312	52.02
<i>BUS</i>	-0.037	1.046	-0.041	0.681	52.84
Average	-0.065	1.364	-0.058	1.130	54.26

Table 3.33: Simulation results for combined spatial and temporal scalability with QP value is 35

Sequence	T & S Frame		S Frame		TS
	Δ PSNR [dB]	Δ BR [%]	Δ PSNR [dB]	Δ BR [%]	
<i>FOREMAN</i>	-0.095	1.357	-0.063	1.252	69.52
<i>FOOTBALL</i>	-0.121	1.554	-0.053	1.702	56.53
<i>BUS</i>	-0.085	0.062	-0.061	0.028	62.95
Average	-0.100	0.991	-0.059	0.994	63.00

Figure 3.29: Rate distortion curve for *FOREMAN* in spatial and temporal scalable video coding

3.5.6 Experimental Comparisons of Fast Mode Decision Algorithms

Recently, several fast mode decision algorithms [114–119] were proposed to reduce encoder complexity with a focus on scalable video coding. Many of them cited our work and presented improved performance as follows.

- An extended fast mode decision algorithm for coarse-grain scalability was proposed in [114]. Similar to our proposed algorithm, a reduced number of candidate modes is obtained by making use of the mode distribution correlations between layers. Given the best mode of the base layer and the quantization parameter, the candidate modes are explicitly determined by a lookup

table. In particular, more base layer information such as the reference frame indices and motion vectors is adaptively reused in the enhancement layers.

- In our proposed algorithm, the modes of a co-located macroblock at the base layer are used to predict the modes of current macroblock at the enhancement layer. However, in [115], Kim et al. noticed there is a high correlation in mode distribution in spatially neighboring macroblocks between layers. Enlightened by that, they developed an algorithm by predicting the mode of each macroblock at the enhancement layer using the modes of a co-located macroblock and its adjacent macroblocks at the base layer.
- The key idea of our proposed algorithm is to select the coding mode to achieve the best rate distortion performance without searching all coding options, thus leading to a reduction of computational complexity. In order to achieve that, the mode correlation between layers is examined. In [116], the candidate modes were optimally selected based on two factors: (i) the expected increase of the rate distortion cost caused by skipping each mode in the mode decision, and (ii) the expected increase of the rate distortion cost caused by skipping each mode in the mode decision is statistically derived using the pre-calculated rate distortion cost of the base layer. In addition, similar to our proposed algorithm, inter layer residual prediction was selectively applied to the coding modes depending on the rate distortion costs.
- Based on our proposed algorithm, Jung et al. [117] made an additional simplification by detecting all-zero blocks at CGS and spatial enhancement layers. An all-zero block detection algorithm is used for the early termination of motion estimation and the early termination of mode decision process. By checking the selected modes of the current macroblock in the enhancement

layer and its corresponding macroblock in the base layer, the possibility of the current macroblock to have all-zero coefficients is determined. By combining all-zero block detection with our proposed fast mode decision algorithm, an average computing time saving of 3% to 12% for CGS scalability and 16% to 35% for spatial scalability had been reported at the cost of bit rate increase for typical test sequences.

- Similar to our proposed algorithm, Sebastiaan et al. explored the correlation between the macroblock mode in the enhancement layer and the co-located macroblock in the reference layer in [118]. This observation was used in conjunction with the quantization of the enhancement layer to design a fast mode decision model for dyadic and non-dyadic spatial enhancement layers.
- Instead of classifying the macroblock into high and low motion block in our proposed algorithm for temporal scalability, [119] classified the macroblocks into four categories based on encoding macroblocks quantization coefficients, the coding mode information of temporal spatial neighboring macroblocks and characteristics of temporal layers. Similar to our proposed algorithm, the mode correlations between high and low temporal level frames were explored, and the candidate mode set was reduced accordingly.

Through a careful literature review, we noticed that many reported algorithms, including [115–117, 119], were compared with Lin’s method [114] and presented slightly improved performances. As such, we have used Lin’s method as a benchmark for CGS scalability in order to have a common basis for performance comparison. Nevertheless, we also paid attention to recently-published methods, for example, Sebastiaan’s method [118], which attracted much attention due to its effective complexity reduction and easy implementation. Therefore, we compared

our proposed algorithm with Sebastiaan’s method for spatial scalability.

- Fast mode decision in CGS scalability

Our proposed algorithm distinguishes from Lin’s method in the following ways. Firstly, in Lin’s method, the correlation of Intra 4×4 prediction modes between coding layers was examined. The candidate prediction modes for enhancement layer intra macroblock could be reduced based on the prediction modes of the co-located macroblock at the base layer. Secondly, in Lin’s method, based on the statistical data, the coding modes smaller than 8×8 did not contribute much to the coding efficiency. Hence they are neglected. Thirdly, Lin’s method paid attention to the selection of temporal prediction types (forward, backward, and bi-directional), which increased more than triple of the total motion search calculations. By making use of the correlations of temporal prediction of each macroblock at enhancement layer and its base layer counterpart, the temporal prediction set at enhancement layer can be reduced. Finally, in Lin’s method, due to the strong correlation between layers in motion vectors, base layer motion vector was used as initial search point in order to reduce the number of search points in motion estimation in the enhancement layer. However, in our proposed algorithm, only the mode distribution correlation between the base layer and its enhancement layer is explored and this correlation is used to reduce the candidate mode set at the enhancement layer.

In order to compare the performance of our proposed algorithm versus Lin’s method, we implemented both algorithms in the JSVM encoder. Table 3.34 shows the test platform and the detailed encoder parameters. In this experiment, CGS scalability is considered. One base layer and one enhancement layer with different quantization parameter settings are used. Table 3.35

compares Lin’s method [114] and our proposed algorithm over a set of CIF-format test sequences. Further, the rate distortion performances of the two algorithms are compared in Figure 3.30 to Figure 3.32 for different video sequences. The results reveal that both algorithms are effective in reducing the encoding time without severely influencing the bit rate and PSNR. In our experiments, Lin’s method outperforms than our proposed algorithm in terms of time saving. Table 3.35 shows that Lin’s method achieves the time saving by 63.00% - 79.60%, while our proposed method achieves 53.02% - 59.78%. This is because Lin’s method made use of much more base layer information, such as temporal prediction references, motion vectors, intra prediction modes. In addition, Lin’s method did not perform the inter-layer residual prediction when the base layer quantization parameter is higher than 33, whereas our proposed algorithm selectively performed this. However, Lin’s method achieved higher computational complexity reduction at the expense of visual quality loss and bit rate increment. As shown in Figure 3.31 and Figure 3.32, our proposed algorithm outperforms Lin’s method in terms of video performances. In addition, the experimental results are justifiable due to the fact that the Lin’s method extended our work, and it was proposed four years later than our algorithm.

- Fast mode decision in spatial scalability

Sebastiaan’s method [118] distinguishes from our proposed algorithms in the following ways. Firstly, in Sebastiaan’s method, the intra coding mode $INTRA_{16 \times 16}$ was neglected at the enhancement layer because of the proven insignificant occurrence of this type. However, in our proposed algorithm, if the selected mode of co-located macroblock is $INTRA_{16 \times 16}$, then this mode will be one of the candidate modes in the enhancement layer.

Table 3.34: Simulation condition

All Tested Video Sequences	<i>AKIYO, FOREMAN, FOOTBALL</i> <i>MOBILE</i> and <i>STEFAN</i>
Encoder Configurations and Test Platform	Software: JSVM encoder M. E. search range: ± 32 pixels MV resolution: $\frac{1}{4}$ pixel accuracy RDO: enabled GOP size: 8 Reference frame number: 1 Entropy coding mode: CABAC Macroblock adaptive inter-layer prediction: enabled 8×8 transform: enabled Machine: Intel Xeon, 2.53GHz CPU, 4GB RAM Number of frames encoded: 1 intra frame + 4 GOPs Video resolution: CIF

Secondly, in Sebastiaans method, a modified version of $MODE_{8 \times 8}$ was evaluated at the enhancement layer. Small sub-partition sizes including 4×4 , 4×8 and 8×4 were hardly selected in the enhancement layer, and they contributed high computational complexity. Hence they were neglected in the rate distortion optimization process for the modified $MODE_{8 \times 8}$. However, in our proposed algorithm, small sub-partition sizes such as 4×4 , 4×8 and 8×4 were used in the rate distortion optimization process. Thirdly, Sebastiaan's method could be easily extended to reduce computational complexity for dyadic and non-dyadic spatial scalability for both P and B frames. However, our proposed algorithm can only be used in dyadic spatial scalability.

For performance comparison, we implemented Sebastiaan's method and our proposed algorithm in the JSVM reference software. The test platform and the detailed encoder parameters are given in Table 3.36. The video sequences with various texture and motion characteristics are used for the experiments.

Table 3.35: Comparison of performance on CGS

Sequences	(QP_B, QP_E)	Method	$\Delta PSNR$ [dB]	ΔBR [%]	TS [%]
AKIYO	(40,30)	Lin's	-0.009	-0.017	79.52
		Proposed	-0.002	0.622	53.78
	(40,20)	Lin's	-0.010	0.554	79.60
		Proposed	-0.002	0.253	53.38
	(40,10)	Lin's	-0.002	0.314	78.55
		Proposed	0.002	0.194	53.02
FOREMAN	(40,30)	Lin's	-0.026	0.342	77.25
		Proposed	0.042	1.489	56.11
	(40,20)	Lin's	-0.045	1.971	77.19
		Proposed	0.021	0.469	56.92
	(40,10)	Lin's	0.015	1.693	79.25
		Proposed	0.016	0.199	59.78
FOOTBALL	(40,30)	Lin's	-0.054	0.789	68.99
		Proposed	0.266	2.603	58.91
	(40,20)	Lin's	-0.009	1.538	69.00
		Proposed	0.207	1.829	58.61
	(40,10)	Lin's	0.020	1.128	71.38
		Proposed	0.152	1.821	59.63
MOBILE	(40,30)	Lin's	-0.018	0.506	67.38
		Proposed	0.011	0.348	55.31
	(40,20)	Lin's	-0.014	1.105	67.40
		Proposed	-0.005	0.025	55.30
	(40,10)	Lin's	0.001	0.696	63.00
		Proposed	0.002	0.098	51.29
STEFAN	(40,30)	Lin's	-0.016	0.452	77.58
		Proposed	0.028	0.923	55.67
	(40,20)	Lin's	0.006	0.800	79.11
		Proposed	0.008	0.357	55.45
	(40,10)	Lin's	-0.001	0.461	79.08
		Proposed	-0.001	0.089	57.68

Figure 3.30: Rate distortion curves for *AKIYO* in CGS scalable video coding

Table 3.37 compares the proposed fast mode decision method with Sebastiaan's method in [118]. All methods achieved significant encoding time reduction and maintained rate distortion performance with negligible amounts of degradation regardless of quantization parameter value settings for spatial layers. In the experiments, our proposed method achieved significant time savings, especially for static video sequences such as *AKIYO*. For *AKIYO*, Sebastiaan's method achieved the time saving by 66.66% on average, while our proposed algorithm achieved 76.85% on average. For sequences with high spatial detail and high motion such as *FOOTBALL* and *MOBILE*, Sebastiaan's method managed to achieve better time saving performance. Take

Figure 3.31: Rate distortion curves for *FOOTBALL* in CGS scalable video coding

FOOTBALL as an example, the average time saving of Sebastiaan's method and our proposed algorithm is 64.26% and 50.14%, respectively. This is due to the fact that our proposed scheme mainly uses the mode information from the lower spatial layer. If the mode distribution of the lower layer is biased to specific modes, the resulting fast mode decision fails to reduce the encoding time effectively. Figure 3.33 to Figure 3.35 show the rate distortion curves of the *AKIYO*, *FOREMAN* and *MOBILE* sequences in spatial scalability, respectively. The rate distortion curves show that both of our proposed method and Sebastiaan's method are efficient to achieve more encoding time saving without significant loss of PSNR and drastic increase of bit rate.

Figure 3.32: Rate distortion curves for *FOREMAN* in CGS scalable video coding

3.6 Conclusions

In this chapter, we addressed the high computational complexity problem associated with the existing scalable video coding systems: exhaustive search technique is employed to select the best coding mode for each macroblock in each video frame. This technique achieves highest possible coding efficiency, but it results in higher computational complexity. We examined the macroblock mode distribution correlation between the base layer and its enhancement layers for spatial, SNR and temporal scalability. The following fast mode decision schemes have been newly proposed:

- (i) Fast mode decision scheme for spatial scalable video coding.

Table 3.36: Simulation condition

All Tested Video Sequences	<i>AKIYO, FOREMAN, FOOTBALL MOBILE and STEFAN</i>
Encoder Configurations and Test Platform	Software: JSVM encoder M. E. search range: ± 32 pixels MV resolution: $\frac{1}{4}$ pixel accuracy RDO: enabled GOP size: 8 Reference frame number: 1 Entropy coding mode: CABAC Macroblock adaptive inter-layer prediction: enabled 8×8 transform: enabled Machine: Intel Xeon, 2.53GHz CPU, 4GB RAM Number of frames encoded: 1 intra frame + 4 GOPs Video resolution: QCIF for base layer, CIF for enhancement layer

- (ii) Fast mode decision scheme for SNR scalable video coding.
- (iii) Fast mode decision scheme for temporal scalable video coding.
- (iv) Fast mode decision scheme for combined scalable video coding.

Using the coded base layer motion information, the number of candidate modes for luma and chroma blocks in an MB that takes part in RDO calculation has been reduced significantly at enhancement layers. As a result, 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, our proposed approach provides an average saving of 60% or higher in the overall encoding time. The penalty on rate distortion performance is negligible. For spatial scalability, the average bit rate increase is below 0.8%, and the average Y-PSNR loss is below 0.07dB. For SNR scalability, the average bit rate increase is

Table 3.37: Comparison of performance on spatial scalability

Sequences	QP	Method	Δ PSNR [dB]	Δ BR [%]	TS [%]
<i>AKIYO</i>	10	Sebastiaan's	-0.017	-0.129	66.71
		Proposed	-0.021	-0.056	71.48
	20	Sebastiaan's	-0.009	0.089	66.89
		Proposed	-0.024	0.100	77.26
	30	Sebastiaan's	-0.003	0.047	66.59
		Proposed	-0.039	-0.016	78.84
	40	Sebastiaan's	-0.005	-0.082	66.44
		Proposed	-0.010	-0.143	79.28
<i>FOREMAN</i>	10	Sebastiaan's	-0.011	0.943	67.19
		Proposed	-0.047	0.127	42.68
	20	Sebastiaan's	-0.049	1.192	64.28
		Proposed	-0.040	1.018	47.47
	30	Sebastiaan's	-0.138	-0.117	62.23
		Proposed	-0.139	0.569	57.31
	40	Sebastiaan's	-0.105	-0.358	61.22
		Proposed	-0.163	-0.076	68.13
<i>FOOTBALL</i>	10	Sebastiaan's	0.026	0.347	68.16
		Proposed	-0.027	0.122	54.28
	20	Sebastiaan's	-0.016	0.501	66.75
		Proposed	-0.019	0.213	46.96
	30	Sebastiaan's	-0.052	0.195	63.79
		Proposed	-0.020	0.261	46.00
	40	Sebastiaan's	-0.117	-0.711	58.34
		Proposed	-0.098	-0.192	53.31
<i>MOBILE</i>	10	Sebastiaan's	0.082	1.447	67.86
		Proposed	-0.007	0.290	43.62
	20	Sebastiaan's	0.021	1.101	66.46
		Proposed	-0.017	0.439	48.81
	30	Sebastiaan's	-0.031	0.418	66.67
		Proposed	-0.035	0.509	59.49
	40	Sebastiaan's	-0.039	0.031	66.16
		Proposed	-0.100	0.421	62.42
<i>STEFAN</i>	10	Sebastiaan's	-0.115	0.085	66.00
		Proposed	-0.083	0.112	40.20
	20	Sebastiaan's	0.076	1.853	66.01
		Proposed	-0.215	1.616	48.10
	30	Sebastiaan's	0.037	2.801	64.54
		Proposed	-0.124	2.116	56.95
	40	Sebastiaan's	-0.216	-0.388	62.29
		Proposed	-0.136	0.348	63.52

Figure 3.33: Rate distortion curves for *AKIYO* for spatial scalable video coding

below 0.5%, and the average Y-PSNR loss is below 0.05dB. For temporal scalability, our proposed scheme can achieve a reduction of 60% encoding time on average, with an acceptable average Y-PSNR loss of 0.007dB and 2.838% bit rate increase. For spatial and CGS scalability, our proposed scheme can achieve a reduction of 53.09% encoding time on average, with an acceptable average Y-PSNR loss of 0.056dB and 0.561% bit rate increase. For spatial and temporal scalability, our proposed scheme can achieve a reduction of 63% encoding time on average, with an acceptable average Y-PSNR loss of 0.1dB and 0.991% bit rate increase.

Figure 3.34: Rate distortion curves for *FOREMAN* for spatial scalable video coding

Figure 3.35: Rate distortion curves for *MOBILE* for spatial scalable video coding

Adaptive Decoder Complexity Reduction for Coarse Grain Scalability

4.1 Introduction

Among the scalability available in SVC, coarse granular scalability (CGS) is one of the most important. Current CGS scheme shows significant achievements in terms of coding efficiency by using fractional ($1/4$) pixel precision, adaptive block size motion compensation, loop filter, 4×4 integer transform and context-based adaptive binary arithmetic coding, etc [120]. However, the excellent performance of CGS is achieved at the expense of computational complexity at both encoder and decoder. The computational capacity of devices, especially handheld devices, is usually very limited. Thus, it is desirable to obtain a good tradeoff between complexity and compression for CGS encoding and decoding systems.

In Chapter 3, we have proposed fast mode decision schemes to reduce the SVC encoder complexity. Our results are attributed to the fact that the computational

complexity is greatly reduced, and the encoding time saving is achieved as well as the video quality is maintained. Nevertheless, computational complexity reduction of an SVC decoder is more important than that of an SVC encoder in many applications. For most Internet video content, as well as video server and broadcast application, video is encoded once and decoded many times. The end user must decode and display the content in real time. Thus decoder complexity is critical in these cases.

To address this problem, many works have been proposed for decoder complexity reduction. Most of these works focuses on either the encoder algorithm for generating low-complexity decoder-friendly bit streams or hardware design optimization. Ugur *et al.* proposed several encoding algorithms that can be used to generate fully H.264/AVC compliant low-complexity bit streams, which require less amount of half-pixel and quarter pixel interpolations at the decoder, as interpolation step consumes most of the decoding processor cycles [43]. An effort has also been made by Wang *et al.* to reduce the decoder size computational cost required at the H.264/AVC decoder effectively [44] by using a complexity adaptive motion estimation and mode decision scheme. Lossless platform-independent optimization methods are utilized in the H.26L video decoder implementation [45]. The optimization is performed in the module of image interpolation, deblocking filter, inverse transform and quantization. Zhou *et al.* implemented H.264/AVC decoders based on Intel's Single-Instruction-Multiple-Data (SIMD) architecture to reduce the decoding complexity of H.264/AVC [46]. In addition, decoder complexity models have been studied in the past. A MPEG-4 video complexity verifier (VCV) model was described in [47], where the numbers of boundary macroblocks and non-boundary macroblocks decoded per second are estimated, and the decoding complexity can be modeled as a function of these two numbers. A complexity

model for motion compensation in the H.264/AVC decoder is proposed in [48]. This model considers a rich set of inter prediction modes allowed by H.264/AVC as well as the relationship of motion vectors, which are related to cache miss performance. In [49], a memory efficient decoding ordering scheme is proposed. By adopting 1×4 column-by-column decoding ordering, the number of memory access can be reduced both in inter prediction unit and intra prediction unit in decoder. All these methods are efficient in reducing the computational complexity with acceptable quality degradation in H.264/AVC decoder. However, these methods are not applicable to all the layers in CGS scalable video coding systems. Moreover, they did not provide any formulation to choose the Lagrangian factor associated with the complexity issue.

The decoding process consists of interpreting the coded symbols of a compliant bit stream and processing this data according to the standard specification in order to generate a reconstructed video sequence. A generalized block diagram of a two-layer SVC decoder is given in Figure 4.1, the decoder data flow at the base layer is as follows [6, 121]:

- (i) A compressed bit stream is entropy decoded to extract coefficients, motion vector and header for each macroblock. A quantized, transformed macroblock X is produced.
- (ii) X is de-quantized and inverse transformed to produce a decoded residual D' .
- (iii) The decoded motion vector is used to locate a 16×16 region in the decoder's copy of the previous (reference) frame F'_{n-1} . This region becomes the motion compensated prediction P .
- (iv) P is added to D' to produce a reconstructed macroblock. The reconstructed macroblocks are saved to produce decoded frame F'_n .

- (v) In the final stage of the decoding process, a SVC decoder must apply the normative deblocking filtering process, which reduces blocking artifacts that are introduced by the coding process at block boundaries.
- (vi) After a complete frame is decoded, F'_n is ready to be displayed and may also be stored as a reference to produce decoded frame F'_{n+1} .

Figure 4.1: Block diagram of a SVC decoder

According to extensive experiments, image interpolation and deblocking filter consume the majority of the decoding time in SVC decoder [45]. They are described in detail in the following.

(i) Image Interpolation

As discussed in Section 2.5, the subsequent interpolation scheme is used in the encoder. SVC first uses a 6-tap Finite Impulse Response (FIR) filter for half-pixel interpolation. Once all the half-pixel data is available, SVC uses simple bilinear sampling to achieve quarter-pixel precision from the half-pixel data. In SVC decoder, direct interpolation scheme is adopted. A quarter-pixel sample can be computed directly from full-pixel samples. Three 6-tap

one-dimensional interpolation filters are used according to three possible sub-pixel positions, as shown in Table 4.1.

Table 4.1: Filter coefficients for different sub-pixel positions

Sub-pixel Position	Filter Coefficients
1/4	(1,-5,52,20,-5,1)/64
2/4	(2,-10,40,40,-10,2)/64
3/4	(1,-5,20,52,-5,1)/64

(ii) Deblocking Filter

In SVC video coding standard, an in-loop deblocking filter is employed to remove the artifacts introduced by the block-based motion compensation and discrete cosine transform (DCT) techniques. The filtering process is performed on the reconstructed signal on both the encoder and decoder side to improve the coding performance significantly. According to an analysis of run-time profiles of decoder sub-functions report in [122], this deblocking filter process is the most computationally intensive part that took as much as one-third of computational resources of the decoder. This is due to:

- The high adaptivity of the filter requires conditional processing on the block edge and sample levels.
- Residual coding in SVC coding algorithm makes use of small block size. With 4×4 blocks and a typical filter length of 2 samples in each direction, almost every sample in a picture must be loaded from memory, either to be modified or to determine if neighboring samples will be modified [72].

There are mainly two stages in the deblocking process. In the first stage, according to the pixel values along the normals to the edges, the edges are classified into different edge strengths. In the second stage, according to the

strengths obtained in stage one, different filtering schemes are applied. The computational operations during the filtering include addition, subtraction, multiplication, division, clipping, and absolute value calculations as well as bitwise shifting, conditional branching, and computing the maximum and minimum values [45].

In this chapter, we propose an adaptive encoding algorithm that would generate light weight decoder-friendly bit streams which require less amount of sub-pixel interpolations at the decoder. Moreover, our proposed algorithm is adaptive to all the layers in SVC. This is achieved by using new rate-distortion-complexity optimized cost functions for motion estimation and mode decision. The experimental results show that our proposed algorithms can reduce the decoding complexity effectively and flexibly with insignificant performance degradation.

The rest of this paper is organized as follows. In Section 4.2, a new framework of rate-distortion-complexity optimization for reducing the interpolation complexity of SVC is formulated. In Section 4.3, a complexity model for the interpolation is provided. Simulation results are presented in Section 4.4. Finally, Section 4.5 concludes the chapter.

4.2 Rate-Distortion-Complexity Optimization

4.2.1 Problem Formulation

In SVC, variable block-size matching motion estimation is used to reduce the temporal redundancy among frames in the same layer and the spatial redundancy among frames in different layers. SVC defines 7 macroblock modes for inter prediction (16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 and 4×4), 2 macroblock modes for intra prediction (4×4 and 16×16) and *SKIP*. For encoding the motion field of

an enhancement layer, ‘Base_layer_mode’ and ‘Qpel_refinement_mode’, are added to the modes applicable in the base layer. In order to select the best mode for each MB, the encoder exhausts all possible modes in the rate-distortion optimization (RDO) framework [91]. However, the excellent performance of RDO is achieved at the expense of computational complexity at the encoder and decoder. It is desirable to design a new framework to obtain a good tradeoff between complexity and compression. Here, such a framework called rate-distortion-complexity optimization (RDCO) [123, 124] is formulated as

$$\begin{aligned} & \min D, \\ & \text{subject to } R < R_c, C < C_c, \end{aligned} \quad (4.1)$$

where D , R and C stand for the distortion, rate and complexity. R_c and C_c stand for the rate and complexity constraints. This problem appears in both motion estimation and mode decision stages, and can be solved by the conventional Lagrangian optimization method as below.

1) In motion estimation stage, searching for the best motion vector can be viewed as the minimization of the Lagrangian cost function:

$$\begin{aligned} & J(MV|QP, \lambda_{SAD}, \lambda_{MOTION}) \\ & = SAD(MV|QP) + \lambda_{SAD}R(MV|QP) \\ & + \lambda_{MOTION}C_{MOTION} \end{aligned} \quad (4.2)$$

where C_{MOTION} is the complexity cost associated with the selected motion vector (MV). λ_{SAD} and λ_{MOTION} are two Lagrangian multipliers. SAD is the sum of absolute difference of Hadamard-transform coefficients. R represents the number

of bits that would be used to code the MV.

2) RDCO mode decision refers to the minimization of the following Lagrangian function:

$$\begin{aligned}
 & J(MODE|QP, \lambda_{SSD}, \lambda_{MODE}) \\
 & = SSD(MODE|QP) + \lambda_{SSD}R(MODE|QP) \\
 & + \lambda_{MODE}C_{MODE}
 \end{aligned} \tag{4.3}$$

where λ_{MOTION} and λ_{SSD} are two Lagrangian factors. SSD is the sum of the squared differences between the original MB and the reconstructed MB located in the reference frames. R denotes the bit cost for encoding the motion vectors, MB header and all the residual information. C_{MODE} is the complexity cost associated with all motion vectors involved in the candidate block mode, i.e.

$$C_{MODE} = \sum_{n=1}^N C_{MOTION}(n) \tag{4.4}$$

with N representing the total number of motion vectors of the concerned MB.

4.2.2 Adaptive Choices of λ_{MODE} and λ_{MOTION}

To apply (4.2) and (4.3) to an SVC encoder, it is required to determine the values of λ_{SAD} , λ_{SSD} , λ_{MODE} and λ_{MOTION} properly.

The Lagrangian multipliers λ_{MODE} and λ_{MOTION} are usually defined as constant values [43]. However, the optimal choice of λ_{MODE} and λ_{MOTION} should depend on the QP. The algorithm for the complexity-constrained motion estimation can be modified in order to incorporate macroblock quantization step-size changes. Hence, for a fixed value of λ_{MODE} , a particular setting of λ_{MOTION} and QP yields

a minimum Lagrangian cost function in equation 4.2. In SVC, the Lagrangian multiplier λ_{SAD} is given by

$$\lambda_{SAD} = 0.92 * 2^{QP/6-2} \quad (4.5)$$

Therefore, we need to examine the relationship between λ_{SAD} and λ_{MOTION} . Figure 4.2 shows the relationship between the λ_{SAD} and λ_{MOTION} for several Lagrange parameter settings. The Lagrange multiplier λ_{MOTION} is varied over seven values, 4, 8, 12, 16, 20, 24, 28. λ_{SAD} is the required Lagrangian multiplier which result in the best quality and complexity tradeoff for SNR scalable video coding. The data of λ_{SAD} are gathered while coding 145 frames of the *FOREMAN* sequence. The

Figure 4.2: λ_{SAD} versus λ_{MOTION} for *FOREMAN* in CGS scalable video coding

bold curve in Figure 4.2 depicts the function:

$$\lambda_{MOTION} = K_{MOTION} * \ln(\lambda_{SAD} + 1) \quad (4.6)$$

By adjusting the value of K , we can achieve complexity and quality tradeoff at motion estimation stage. Similarly, for mode decision, we have:

$$\lambda_{MODE} = K_{MODE} * \ln(\lambda_{SSD} + 1) \quad (4.7)$$

The choices of K_{MOTION} and K_{MODE} provide a tradeoff between the complexity and video quality in SNR scalable video coding. Suitable values can be determined based on extensive experiments on test sequences. For example, $K_{MOTION} = 45$ and $K_{MODE} = 1$ may be empirical settings for the experiments.

4.3 Interpolation Complexity Modelling

Sub-pixel motion estimation and compensation involve searching sub-sample interpolated positions as well as integer-sample positions, choosing the position that gives the best match and using the integer- or sub-sample values at this position for motion compensated prediction [6]. Figure 4.3 shows the concept of a quarter-pixel motion estimation. There are 15 possible sub-sample positions, including three half-pixel positions (at location 3, 9, and 11) and twelve quarter-pixel positions. Each of the sub-pixel position requires a different interpolation filter.

It is obvious that the location of a motion vector influences the decoder interpolation complexity directly. We can assign the interpolation complexity term to each MV location as follows [6]:

1) If the motion vector has integer values for both its horizontal and vertical directions, the decoder does not perform any interpolation at all. Hence the interpolation cost is 0.

2) If the motion vector has integer value for either horizontal or vertical direction, the decoder needs to perform at least one 6-tap interpolation. The interpolation cost is set to 1 in the proposed scheme.

3) If the motion vector is at location 6, 8, 14 and 16, the decoder needs to perform two 6-tap interpolation and one 2-tap interpolation. Hence, the interpolation cost is set to 2 in the proposed scheme.

4) If the motion vector is at location 7, 10, 11, 12 and 15, the decoder needs to perform at least seven 6-tap interpolation. Hence, the interpolation cost is set to 4 in the proposed scheme. With the interpolation complexity assignment for each

X	2	3	4	X
5	6	7	8	
9	10	11	12	
13	14	15	16	
X				X

Figure 4.3: Sub-pixel locations in SVC

motion vector, the rate distortion optimization function for motion estimation and mode decision can be modified by adding the corresponding interpolation cost.

4.4 Experimental Results

The proposed decoder complexity reduction algorithm is embedded in JSVM encoder [91]. The test platform used is Intel Pentium IV, 1.83GHz CPU, 1G RAM with Windows XP professional operating system. The test condition is shown in Table 4.2. In our experiments, five standard test sequences including *FOREMAN*, *FOOTBALL*, *BUS*, *CREW* and *CITY* have been tested. We only consider the two-layer case and the QP value settings for all the layers are shown in Table 4.2. The GOP size is set to be 8.

Table 4.2: Simulation condition

		Tested Video Sequences
Frame Rate		15 Hz
Resolution		QCIF
QP Setting	Base	40
	Enhance	30 to 10
Coding Option Used		MV search range is ± 32 pels. Reference frame number is 1. Full search is used in ME.
Codec		JSVM encoder

The testing parameters in our experiments include the complexity saving, Y-PSNR and bit rate for both base layer and enhancement layer. We use Complexity Saving (CS) to indicate the total interpolation computational cost saving in the decoding process:

$$CS = \frac{C_{JSVM} - C_{proposed}}{C_{JSVM}} \times 100\% \quad (4.8)$$

where C_{JSVM} and $C_{proposed}$ are the total computational cost of JSVM system and its modified system according to the proposed algorithm, respectively.

Let QP_e stand for QP values at the enhancement layer. Table 4.3 to 4.5 show the tabulated performance comparison of the proposed algorithm with JSVM for

different QP_e . CS refers the complexity saving due to interpolation computations. Note that in the table, positive values mean increments and negative values mean decrements. The results show that the proposed method is very effective in reducing the decoding complexity, especially for the sequence with high motion and fine detail. The average decoding complexity is reduced up to 61%. Figure 4.4 and 4.5 present the rate distortion curves for *FOREMAN* and *FOOTBALL*. From these figures, we can conclude that our scheme can achieve consistent decoding complexity saving over a large bit rate range with negligible loss in Y-PSNR and minor increments in bit rate.

Table 4.3: Simulation results with $QP_e = 10$

Sequence	Base		Enhancement		CS [%]
	Δ PSNR [dB]	Δ BR [%]	Δ PSNR [dB]	Δ BR [%]	
Foreman	-0.045	0.223	0.005	1.366	61.90
Football	0.006	0.514	0.060	0.796	60.80
Bus	-0.016	0.986	0.044	0.844	43.91
Crew	-0.033	0.755	0.067	1.379	67.86
City	-0.048	1.399	0.001	0.622	45.36
Average	-0.027	0.775	0.035	1.001	55.97

Table 4.4: Simulation results with $QP_e = 20$

Sequence	Base		Enhancement		CS [%]
	Δ PSNR [dB]	Δ BR [%]	Δ PSNR [dB]	Δ BR [%]	
Foreman	-0.045	0.223	-0.076	1.759	64.53
Football	0.006	0.514	-0.089	1.084	69.01
Bus	-0.016	0.986	-0.010	1.265	46.96
Crew	-0.033	0.755	-0.060	2.489	75.37
City	-0.048	1.399	-0.025	1.978	46.49
Average	-0.027	0.775	-0.052	1.715	60.47

Table 4.5: Simulation results with $QP_e = 30$

Sequence	Base		Enhancement		CS [%]
	Δ PSNR [dB]	Δ BR [%]	Δ PSNR [dB]	Δ BR [%]	
Foreman	-0.045	0.223	-0.097	0.574	65.10
Football	0.006	0.051	-0.091	0.071	72.29
Bus	-0.016	0.099	-0.035	1.008	48.09
Crew	-0.033	0.755	-0.125	0.396	75.95
City	-0.048	1.399	-0.078	1.174	44.41
Average	-0.027	0.775	-0.085	0.645	61.17

Figure 4.4: Rate distortion curve for *FOREMAN*

4.5 Conclusions

In this chapter, an adaptive decoder complexity reduction algorithm for SNR scalable video coding is presented. With this scheme, a light weight decoder-friendly

Figure 4.5: Rate distortion curve for *FOOTBALL*

bit stream can be generated at the encoder, and it requires less decoding source consumption. The results of the simulations in Section 4.4 demonstrate that the proposed algorithm can save up to 76% of the decoding complexity as compared to the original JSVM system. Moreover, it introduces insignificant picture degradation and minor bit rate increase.

Adaptive SNR Enhancement Layer Motion Estimation and Motion Compensation Schemes

5.1 Introduction

Coding efficiency of motion-compensated video coding is strongly dependent on the bit allocation between motion data and residual (texture) data. To optimize the performance of the current SNR scalable video coding systems, we need to consider the tradeoff between the motion data and residual data. Furthermore, the optimal tradeoff is dependent on the spatial-temporal resolution as well as the bit rate. It is usually achieved by a rate distortion optimization (RDO) that is formulated by minimization of the Lagrangian formulation of the function J [125], where J is given by:

$$J = D + \lambda R \quad (5.1)$$

Here, the distortion D is the average of the forward and backward SAD or SSD between the current macroblock and the motion compensated matching blocks. This distortion is used to evaluate the correctness of the forward and backward motion vectors. The R term denotes the bit cost for encoding the motion vectors, λ is a weight parameter used to control the contribution of the motion bits in the total cost function, which needs to be determined by the rate-distortion property of the compressed bit stream [125]. Each λ corresponds to a bit rate range, and each λ determines a tradeoff between motion information and residual information. However, since scalable video coder is an embedded coder, λ can only be a fixed value for all the bit rates.

Theoretically, a large λ works well at a low bit rate while a small λ works well at a high bit rate [108]. A small λ is optimized at a high bit rate, and it is usually chosen in the MCTF based SVC. At low bit rates, the truncated bit stream contains too much motion information and too less residual information. Consequently, the amount of residual information can not support motion information. As a result, the video coding efficiency is relatively poor. For the case where a large λ is selected, such as FGS schemes [52, 126, 127], it is thus optimized at a low bit rate. However, at high bit rates, only the residual information is refined. As a result, the enhanced bit stream contains too much residual information and too less motion information. Thus, the visual quality is relatively poor. As a result, there still exists a PSNR gap of about 1-3 dB when compared to the state-of-the-art single layer coding scheme at high bit rate [53]. Therefore, a scalable bit stream should contain a scalable representation of the motion data. With a single motion field, it is impossible to generate a scalable bit stream that provides a nearly rate-distortion-optimal coding efficiency over a large scale of spatial-temporal resolutions and bit rates.

In this chapter, two adaptive SNR scalable video coding refinement schemes based on motion information scalability are proposed. In these two schemes, multi-layer motion estimation is performed, and then one base layer and one or more enhancement layer motion fields are generated. In our schemes, the base layer is designed based on the existing video coding standards such as H.264/AVC. At a low bit rate, motion information and residual information are generated to form scalable video coding base layer. In order to achieve the coding efficiency at high bit rate, both the motion information as well as residual information are refined at the enhancement layers. Different SNR layer at a fixed spatial resolution might have different motion vector field. As a result, our proposed scheme outperforms the existing scheme without motion refinement in the sense that a good tradeoff between motion information and residual information can be obtained over a large bit rate range. Experimental results show that the proposed scalable motion coding increases the Y-PSNR values and the coding efficiency.

The rest of this chapter is organized as follows. We begin in Section 5.1 by introducing a rate control mechanism and detailing the importance of motion vector scalability in scalable coding. Then in Section 5.2, 5.3 and 5.4, we will present our proposed schemes for SNR scalability in detail by providing results and a study into some of the characteristics of the performance of SNR scalability. Finally, Section 5.5 will recap the approaches taken in this chapter.

Figure 5.1: Flowchart of the base layer motion estimation and mode decision process

5.2 Base Layer Motion Estimation and Mode Decision Process

This section describes a method for estimating the prediction data arrays M_{base} used in the base layer prediction steps. This algorithm employs Lagrangian optimization techniques with cost function given in equation 5.1 that are widely used to optimize the rate-distortion efficiency of video coders. Figure 5.1 shows the flowchart of the motion information M_{base} generation by using the basic motion estimation/compensation (ME/MC) scheme. In this flowchart, MV_0 and MV_1 are motion vector information in list 0 and list 1 direction, respectively. P and sub-macroblock mode P_{sub} if applicable denote certain macroblock partition of current

macroblock and P_i , where i is the sub-macroblock partition index, specifies certain sub-macroblock partitions of current macroblock partition P or P_{sub} . This method is described in detail as follows.

(i) **Determine List 0 and List 1 Motion Vectors for Each Possible Macroblock Partition**

Assume $A_{org}(x, y)$ is a predicted frame and $A_{ref,0/1}(x, y)$ are its referencing frames at list0 and list1 prediction directions at the l^{th} temporal level. Let R_0 and R_1 specify the set of active entries of the reference indices in list0 and list1 direction. The reference indices r_0 and r_1 satisfy $r_0 \in R_0$ and $r_1 \in R_1$. For each possible macroblock partition P (P_{sub} if applicable) and its associated P_i , motion vector $MV_0(dx, dy)$ and $MV_1(dx, dy)$ together with the associated reference indices r_0, r_1 are determined by minimizing the Lagrangian function.

$$MV_{0/1}(r_{0/1}) = \arg \min_{MV_{0/1} \in S} \{D_{SAD}(P_i, r_{0/1}, MV_{0/1}) + \lambda_{SAD}(R(r_{0/1}) + R(MV_{0/1}))\} \quad (5.2)$$

with the distortion term given as:

$$D_{SAD}(P_i, r_{0/1}, MV_{0/1}) = \sum_{(x,y) \in P} |A_{org}(x, y) - A_{ref,0/1}(x - dx_{0/1}, y - dy_{0/1})| \quad (5.3)$$

where S is the motion vector search range. All integer-sample accurate motion vectors in the search range are first proceeded in the motion search process. After finding the best integer motion vector, the eight surrounding half sampled accurate motion vectors are tested. Finally, the eight surrounding

quarter sampled accurate motion vectors are tested by giving the best half sampled accurate motion vector. The terms $R(r_{0/1})$ and $R(MV_{0/1})$ denote the amount of bits used to transmit the reference index $r_{0/1}$ and all components of the motion vector $MV_{0/1}$.

(ii) **Determine Block Prediction Mode for Each Possible Macroblock Partition**

Given the motion vector candidates for all sub-macroblock partitions P_i and the forward prediction and backward prediction reference indices $r_0 \in R_0$ and $r_1 \in R_1$, the indices r_0 , r_1 and the associated motion vectors MV_0 and MV_1 are selected as:

$$\begin{aligned} r_{0/1} = \arg \min_{r_{0/1} \in R_{0/1}} \{ & D_{SAD}(P_i, r_{0/1}, MV_{0/1}(r_{0/1}, i)) \\ & + \lambda_{SAD}(R(MV_{0/1}(r_{0/1}, i)) + R(r_{0/1})) \} \end{aligned} \quad (5.4)$$

Given the determined reference indices r_0 , r_1 and their associated motion vectors $MV_0(dx, dy)$, $MV_1(dx, dy)$, the Lagrangian costs for list0 and list1 prediction J_{L0} and J_{L1} are calculated by:

$$\begin{aligned} J_{L0/L1} = \sum_{i \in P} (& D_{SAD}(P_i, r_{0/1}, MV_{0/1}(r_{0/1}, i)) \\ & + \lambda_{SAD}(R(MV_{0/1}(r_{0/1}, i)) + R(r_{0/1}))) \end{aligned} \quad (5.5)$$

In a similar way to that in [77], the reference indices r_{B0} and r_{B1} and their associated motion vectors MV_{B0} and MV_{B1} for bi-prediction can be obtained.

The Lagrangian function for bi-prediction is as follows:

$$\begin{aligned}
 J_{BI} = & \sum_{i \in P} (D_{SAD}(P_i, r_{B0}, r_{B1}, MV_{B0}(r_{B0}, i), MV_{B1}(r_{B1}, i)) \\
 & + \lambda_{SAD}(R(MV_{B0}(r_{B0}, i)) \\
 & + R(MV_{B1}(r_{B1}, i)) + R(r_{B0}) + R(r_{B1})))
 \end{aligned} \tag{5.6}$$

with the distortion term given as:

$$\begin{aligned}
 D_{SAD} = & \sum_{(x,y) \in P} | A_{org}(x, y) - \frac{1}{2}(A_{ref,0}(x - dx_0, y - dy_0) \\
 & + A_{ref,1}(x - dx_1, y - dy_1) + 1) |
 \end{aligned} \tag{5.7}$$

Above steps are proceeded over all sub-macroblock partitions P_i of every possible macroblock partition P . After finding the reference indices r_0, r_1, r_{B0} and r_{B1} and the associated motion vectors MV_0, MV_1, MV_{B0} and MV_{B1} as well as the Lagrangian cost values J_{L0}, J_{L1} and J_{BI} , the prediction method p_{pred} can be determined by the following function.

$$p_{pred} = ((J_{B1} \leq J_{L0}) \& \& (J_{BI} \leq J_{L1}) ? BI : ((J_{L0} \leq J_{L1}) ? L0 : L1)) \tag{5.8}$$

(iii) **Determine Macroblock Partition p (or Sub-macroblock Partition if Applicable)**

Given the motion vector candidates and block prediction mode for all sub-macroblock partitions P_i , the macroblock partition type can be determined as:

$$p = \arg \min_{p \in S_{mb}} \{D_{SAD}(P_i, p) + \lambda_{SAD}R(p)\} \tag{5.9}$$

where $R(p)$ specifies the number of bits that is needed to transmit the information of each macroblock including the macroblock partition (sub-macroblock partition), block prediction mode and motion vector information. S_{mb} specifies various macroblock partition types. As a result, the motion information for base layer M_{base} can be generated. Finally, all motion information and residual image are coded and sent to the decoder.

Figure 5.2: Original (a) and base layer reconstructed (b) frame of *FOREMAN*

Figure 5.2 shows the original and base layer reconstructed frame of *FOREMAN* sequence. In base layer, the bit rate is relatively low, and only part of the motion information and residual information is transmitted to the decoder. Moreover, some distortion has been introduced due to some quantization process, for example, around detailed areas such as the face. Therefore, we can conclude that significant compression is achieved at the expense of relatively poor image quality by using this method.

5.3 Proposed Enhancement Layer Motion Estimation and Motion Compensation Schemes

As discussed previously, after the process of the base layer motion estimation and mode decision, all motion information M_{base} and a part of the residual image are coded and transmitted to the decoder. When the bit rate is higher than the bit rate of the base layer, the remaining information D_{br} should be entropy coded and transmitted to the decoder. Two possible methods are used to code D_{br} at higher bit rate [108]. Same as the FGS scheme, code it directly is one way. This is a kind of ‘intra’ coding mode of the remaining energy. The other one is a kind of ‘inter’ coding mode by further performing ME/MC in order to generate a new residual image and additional MVs. In order to achieve best coding efficiency at a higher bit rate, both motion and residual information needs to be refined and the enhancement layer. To achieve that, both the coded residual information and the motion information M_{base} at the base layer is used by the encoder to generate the motion information and the residual images at the enhancement layer. Consequently, a scalable bit stream containing a scalable representation of motion data and residual data can be obtained. Two schemes called scheme A and scheme B will be proposed to address this issue respectively in this section. In scheme A, the base layer and enhancement layer share the same macroblock partition and block prediction mode information for each macroblock. In the enhancement layer, the motion vectors will be refined based on the base layer motion vectors and enhancement layer quantization parameter (QP) value. In scheme B, the motion information of enhancement layer is generated based on the rate-distortion optimization function. The two schemes are described in detail as follows.

5.3.1 Proposed Enhancement Layer Refinement Scheme A

Figure 5.3: Flowchart of proposed SNR refinement scheme A

Our proposed SNR refinement scheme A, with flowchart shown in Figure 5.3, is now described as follows. With this scheme, macroblock partition information and block prediction mode information are shared by the ME/MC at both base layer and enhancement layer.

Figure 5.4 illustrates the general idea of this scheme. As shown in the figure, for the right upper macroblock, the macroblock partition (sub-macroblock partition

Figure 5.4: Motion information correlations in the base layer and enhancement layers

if applicable) and block prediction mode (list 0, list 1 or bidirectional prediction) are the same for both base layer and enhancement layer. Moreover, the motion vector information of enhancement layer is predicted based on the motion vector information of the base layer.

Let $(dx_{0/1,b}, dy_{0/1,b})$ denote an MV obtained by the basic ME/MC at the l^{th} temporal level and j^{th} spatial level. Define a set of MVs around $(dx_{0/1,b}, dy_{0/1,b})$ as:

$$\rho(dx_{0/1,b}, dy_{0/1,b}, \delta_x, \delta_y) = \{(dx_{0/1,e}, dy_{0/1,e}),$$

$$|dx_{0/1,e} - dx_{0/1,b}| < \delta_x, |dy_{0/1,e} - dy_{0/1,b}| < \delta_y\} \quad (5.10)$$

Suppose that $(dx_{0/1,e}, dy_{0/1,e})$ is in the set $\rho(dx_{0/1,b}, dy_{0/1,b}, \delta_x, \delta_y)$ and is a candidate for the refinement. An SNR refinement criterion is defined as:

$$D_{br} = \sum_{(x,y) \in P} |A_{org}(x, y) - A_{ref,0/1}(x - dx_{0/1,e}, y - dy_{0/1,e})$$

$$- IQ(Q(A_{org}(x, y) - A_{ref,0/1}(x - dx_{0/1,b}, y - dy_{0/1,b})))| \quad (5.11)$$

where $Q()$ denotes quantization operation and $IQ()$ denotes the inverse quantization operation. And the SNR refinement criterion for bi-direction prediction is defined as:

$$D_{br} = \sum_{(x,y) \in P} | A_{org}(x, y) - \frac{1}{2}(A_{ref,0}(x - dx_{0,e}, y - dy_{0,e}) + A_{ref,1}(x - dx_{1,e}, y - dy_{1,e})) - IQ(Q(A_{org}(x, y) - \frac{1}{2}(A_{ref,0}(x - dx_{0,b}, y - dy_{0,b}) + A_{ref,1}(x - dx_{1,b}, y - dy_{1,b})))) | \quad (5.12)$$

The weight parameters λ_{low} and λ_{high} are used to control the motion bits allocation in the total cost function for base layer and enhancement layer, which need to be predefined by user. In this report, $\lambda_{low}(l, j)$ corresponds to the low bit rate range at l^{th} temporal level and j^{th} spatial level. $\lambda_{high}(l, j)$ corresponds to high bit rate range. As discussed previously, the following equation should be satisfied.

$$\lambda_{low}(l, j) \geq \lambda_{high}(l, j) \quad (5.13)$$

A further ME/MC with an MV as (dx, dy) is performed if the following inequality is satisfied.

$$D_{br}(dx_{0/1,e}, dy_{0/1,e}, dx_{0/1,b}, dy_{0/1,b}) + \lambda_{high} R_{MV}(dx_{0/1,e} - dx_{0/1,b}, dy_{0/1,e} - dy_{0/1,b}) \leq D_{br}(dx_{0/1,b}, dy_{0/1,b}, dx_{0/1,b}, dy_{0/1,b}) \quad (5.14)$$

where $R_{MV}(dx_{0/1,e} - dx_{0/1,b}, dy_{0/1,e} - dy_{0/1,b})$ represents the number of bits required to code the $MV(dx_{0/1,e} - dx_{0/1,b}, dy_{0/1,e} - dy_{0/1,b})$. The final $MV(dx_{0/1,e}, dy_{0/1,e})$

can be obtained by minimizing the Lagrangian function, i.e.

$$\begin{aligned} (dx_{0/1,e}, dy_{0/1,e}) = \arg \min_{(dx,dy) \in S} \{ & D_{br}(dx_{0/1,e}, dy_{0/1,e}, dx_{0/1,b}, dy_{0/1,b}) \\ & + \lambda_{high} R_{MV}(dx_{0/1,e} - dx_{0/1,b}, dy_{0/1,e} - dy_{0/1,b}) \} \end{aligned} \quad (5.15)$$

As a result, the motion information for enhancement layer M_e can be obtained. This motion information includes only the refinement motion vector information based on the base layer for each macroblock. All motion information M_e and the residual information are coded and sent to the decoder.

Figure 5.5 shows the original, base layer reconstructed and enhancement layer reconstructed frame of *FOREMAN* sequence. In the enhancement layer, the bit rate is relatively high, more motion information and residual information has been transmitted to the decoder. Therefore, the quality of the receiving frame is improved.

Figure 5.5: Original (a), reconstructed base layer (b) and reconstructed enhancement layer (c) frames of *FOREMAN*

5.3.2 Proposed Enhancement Layer Refinement Scheme B

In the proposed scheme A, macroblock partition information and block prediction mode information are shared by the ME/MC at both base layer and enhancement

layer. Moreover, the motion vector information of the enhancement layer is predicted based on the motion vector information of the base layer. However, by sharing the same macroblock partition information and block prediction mode information, a relatively large cost may be resulted based on Lagrangian function due to relatively less motion correlation between reconstructed base layer and enhancement layer. In scheme B, the ME/MC technique for enhancement layer generation is highly similar to the base layer generation. The only difference is that the base layer motion vectors may be used for prediction in the enhancement layer. The motion information M_e is generated at the l^{th} level by using the following process.

(i) **Determine List 0 and List 1 Motion Vectors for Each Possible Macroblock Partition**

At this step, similar to the base layer motion estimation, motion vectors for every possible macroblock partition and every possible sub-macroblock prediction mode are generated. For each possible macroblock partition P and its sub-macroblock mode P_{sub} of current macroblock, the motion vector $MV_{e0}(dx, dy)$, $MV_{e1}(dx, dy)$ together with the associated reference indices r_0 and r_1 are determined by:

$$\begin{aligned} MV_{e0/1}(r_{0/1}) = \arg \min_{MV_{e0/1} \in S} \{ & D_{br}(P_i, r_{0/1}, MV_{b0/1}, MV_{e0/1}) \\ & + \lambda_{high}(R(r_{0/1}) + R_{MV}(dx_{0/1,e} - dx_{0/1,b}, dy_{0/1,e} - dy_{0/1,b})) \} \end{aligned} \quad (5.16)$$

with the distortion term given as:

$$\begin{aligned} D_{br} = \sum_{(x,y) \in P} | & A_{org}(x, y) - A_{ref,0/1}(x - dx_{0/1,e}, y - dy_{0/1,e}) \\ & - IQ(Q(A_{org}(x, y) - A_{ref,0/1}(x - dx_{0/1,b}, y - dy_{0/1,b}))) | \end{aligned} \quad (5.17)$$

(ii) Determine Block Prediction Mode for Each Macroblock Partition

Figure 5.6: Flowchart of scheme B for prediction mode determination for each P_i

Figure 5.6 shows the flowchart of motion estimation process of particular sub-macroblock partition. For certain macroblock partition P and all its sub-macroblock P_i , three kinds of block prediction (list0, list1 and bi-directional prediction) will be analyzed. Take list0 prediction mode as an example, if the associated reference index $r_{0,e}$ in the enhancement layer is equal to the correlated base layer sub-macroblock index $r_{0,b}$, then the corresponding base layer motion vector is used for the motion prediction in the enhancement layer. Otherwise, the enhancement layer will not use base layer motion vector for prediction. Another motion search will be performed, and the corresponding enhancement layer QP value will be used for the motion search and cost calculation. After the cost evaluation for three block prediction modes, the lowest cost and corresponding motion vector field will be generated. The

same process will be performed for every macroblock partition. Finally, the motion vector field with minimum cost for the macroblock will be generated. This process can be described in mathematical form as follows.

Given the motion vector candidates for all sub-macroblock partitions P_i and the forward prediction and backward prediction reference indices $r_{0,e} \in R_0$ and $r_{1,e} \in R_1$, the indices $r_{0,e}$, $r_{1,e}$ and the associated motion vectors $MV_{0,e}$, $MV_{1,e}$ are selected by minimizing the Lagrangian function:

$$r_{0/1,e} = \arg \min_{r_{0/1} \in R_{0/1}} \{D_{br}(P_i, r_{0/1,e}, MV_{b_{0/1}}(r_{0/1,b}, i), MV_{e_{0/1}}(r_{0/1,e}, i)) + \lambda_{high}(R(MV_{e_{0/1}}(r_{0/1,e}, i)) + R(r_{0/1,e}))\} \quad (5.18)$$

Given the determined reference indices $r_{0,e}$ and $r_{1,e}$ and the associated motion vectors $MV_{0,e}$ and $MV_{1,e}$, the Lagrangian costs for forward prediction and backward prediction J_{L0} and J_{L1} are calculated by:

$$J_{L0/L1} = \sum_{i \in P} (D_{br}(P_i, r_{0/1,e}, MV_{e_{0/1}}(r_{0/1}, i), MV_{b_{0/1}}(r_{0/1,b}, i)) + \lambda_{high}(R(MV_{e_{0/1}}(r_{0/1,e}, i)) + R(r_{0/1,e}))) \quad (5.19)$$

Similarly, the Lagrangian cost for bidirectional prediction is calculated. Above steps are proceeded over all sub-macroblock partitions P_i of every possible macroblock partition P. After finding the reference indices r_0 , r_1 , r_{B0} and r_{B1} and the associated motion vectors $MV_{0,e}$, $MV_{1,e}$, $MV_{B0,e}$ and $MV_{B1,e}$ as well as the Lagrangian cost values J_{L0} , J_{L1} and J_{BI} , the prediction mode can be determined using the following function.

$$p_{pred} = ((J_{BI} \leq J_{L0}) \& \& (J_{BI} \leq J_{L1}) ? BI : ((J_{L0} \leq J_{L1}) ? L0 : L1)) \quad (5.20)$$

- (iii) **Determine Macroblock Partition p (or Sub-block Partition if Applicable)**

Figure 5.7: Flowchart of proposed scheme B for macroblock partition determination

Given the motion vector candidates and block prediction mode for all sub-macroblock partitions P_i , the macroblock partition type can be determined by minimizing the Lagrangian cost function:

$$p = \arg \min_{p \in S_{mb}} D_{SAD}(P_i, p) + \lambda_{SAD} R(p) \quad (5.21)$$

where $R(p)$ specifies the number of bits required to transmit the information of each macroblock including macroblock partition (sub-macroblock partition), block prediction mode and motion vector information. Figure 5.7 shows the flowchart of macroblock partition type determination. As a result, the motion information for base layer M_e can be generated. Finally, all motion information M_e and the residual image are coded and sent to the decoder.

5.4 Experimental Results

In this section, we shall test our proposed SNR scalable video coding schemes experimentally. Our experiment is based on the software package from HHI [77]. We use the *MOBILE* (75 frames), *FOREMAN* (75 frames), and *BUS* (38 frames) sequences in our experiments. These three sequences have a temporal sampling of 7.5 frames/sec. All sequences have a spatial resolution of 176×144 pixels (QCIF). We compare scheme A and scheme B with the traditional SNR scalability scheme where the same motion field is used at all SNR layers [77]. For simplicity, we consider cases that there are one based layer and one or two enhancement layer(s). Only one motion vector field (MVF) is generated by using a large motion vector QP in the traditional SNR scalability scheme. Whereas, two MVFs are generated in our schemes. At the base layer, a relatively large motion vector QP is used to generate a MVF. At the enhancement layer, a relatively small motion vector QP is chosen to generate a refined MVF. The values of QP's are given in Table 5.1.

Table 5.1: Setting of QPs for different SNR scalability schemes

BUS FOREMAN MOBILE	HHI[[77]]	MV_QP = 38 for both layers
	Our Scheme A&B (One EL)	MV_QP = 38 for base layer
		MV_QP = 28 for enhancement layer
	Our Scheme A&B (Two ELs)	MV_QP = 38 for base layer
		MV_QP = 28 for enhancement layer 1
		MV_QP = 18 for enhancement layer 2

5.4.1 Comparison of SNR Scalability for 4 Schemes

The experimental results on the peak signal-to-noise ratio (PSNR) of the Y components for test sequence *FOREMAN*, *BUS* and *MOBILE* are shown in Table 5.2, Table 5.3 and Table 5.4 respectively. The PSNR gap between Scheme A and HHI is

around 0.4dB for video sequences with high motion and almost zero for sequences with low motion. The PSNR gap between scheme B and HHI is around 0.9 dB for high motion sequence and zero for low motion sequence, while, the PSNR gap between scheme $A + B$ and HHI is around 1dB for high motion sequence and zero for low motion sequence. Figure 5.8, Figure 5.9 and Figure 5.10 plot coded PSNR per frame of enhancement layer SNR scalability for the *FOREMAN*, *MOBILE* and *BUS* Sequences.

Figure 5.8: Coded PSNR per frame of enhancement layer SNR scalability for *FOREMAN*

Figure 5.9: Coded PSNR per frame of enhancement layer SNR scalability for *MOBILE*

Table 5.2: Comparison of the average PSNR for *FOREMAN* at different bit rates

Bit rate (Kbps)	Specification	Method			
		HHI[[77]]	Scheme A	Scheme B	A+B
20	Y-PSNR (dB)	31.044	31.044	31.044	31.044
	Actual Rate (Kbps)	20.030	20.030	20.030	20.030
110	Y-PSNR (dB)	38.895	39.295	39.809	39.876
	Actual Rate (Kbps)	110.04	109.96	109.90	110.22

Figure 5.10: Coded PSNR per frame of enhancement layer SNR scalability for *BUS*

Table 5.3: Comparison of the average PSNR for *BUS* at different bit rates

Bit rate (Kbps)	Specification	Method			
		HHI[[77]]	Scheme A	Scheme B	A+B
48	Y-PSNR (dB)	27.728	27.728	27.728	27.728
	Actual Rate (Kbps)	48.160	48.160	48.160	48.160
256	Y-PSNR (dB)	37.441	37.720	38.125	38.163
	Actual Rate (Kbps)	256.25	257.07	256.03	256.42

Table 5.4: Comparison of the average PSNR for *MOBILE* at different bit rates

Bit rate (Kbps)	Specification	Method			
		HHI[[77]]	Scheme A	Scheme B	A+B
48	Y-PSNR (dB)	28.126	28.126	28.126	28.126
	Actual Rate (Kbps)	48.340	48.340	48.340	48.340
256	Y-PSNR (dB)	36.241	36.566	36.870	36.889
	Actual Rate (Kbps)	257.25	256.76	256.85	257.32

5.4.2 SNR Scalable Coding vs Non-scalable Coding

Although we have compared the four schemes of SNR scalability, we have not shown its performance as compared to non-scalable compressed sequence. Certainly, if SNR scalability greatly under performs non-scalable coding, then the justification of SNR scalability can be undermined by the lack of performance. This comparison of the performance of SNR scalability with its non-scalable counterpart is presented in Figure 5.11. This figure shows the coded PSNR per frame of scalable SNR scalability coding (Scheme $A + B$) and non scalable coding for *FOREMAN* sequence. For two layer SNR scalable coding, the bit rate for base layer is 20Kbps and the bit rate for enhancement layer is 110Kbps. For non-scalable coding, the bit rate for the single base layer is 110Kbps. The QP value for the enhancement layer in scalable coding and base layer in non-scalable coding is 28. The upper graph shows the performance of enhancement layer of SNR scalable coder and the lower plot shows the base layer PSNR value of non-scalable coder. The results tell us that the SNR scalable coding is directly comparable to the non-scalable single layer coding. In the worst case, at 110Kbps the enhancement layer suffers by only 0.46dB. However, it outperforms in having two levels of quality readily available after only one encoding process.

Figure 5.11: Scalable coding versus non-scalable coding for *FOREMAN*

5.4.3 Layer Separation Characteristic

A fundamental question that needs to be considered is the behavior of SNR scalability as a function of the layer bit rates. By applying the scalable video codec at increasing layer separation, we can quantify the characteristics of the scalable video codec. Since SNR quality is a function of both the layer separation and the base layer quality, the tests were conducted at two different base layer quality. The first test was based on a low base layer quality starting from 20Kbps. Thus, starting from a 20Kbps base layer, if the layer separation is 10Kbps, then the first enhancement layer has a total bit rate of 30Kbps. Five layer separations ranging from 10 to 160 Kbps were tested and are presented in Table 5.5. This

table compares the SNR scalable characteristic of both HHI scheme and proposed scheme $A + B$. We see that for both schemes as the separation increase, the gains in the average PSNR for the enhancement layer are higher. This is because the base layer is at the same quality as a non-scalable video, and ME/MC has a chance to perform well. The gains for the enhancement layer of scheme $A + B$ become clearly more significant as the separation increases. As a result, we can conclude that our scheme is very efficient for video sequences with high motion when SNR range is wide. This test was repeated using a base layer of 64Kbps. At this rate, the base layer has presented a good encoding of a frame and less error will need to be encoded in the enhancement layer. These results are given in Table 5.6.

Table 5.5: Performance of SNR scalability as a function of layer separation at base layer rates of 20Kbps for HHI scheme and scheme $A + B$

Layer Separation (Kbps)	HHI Y-PSNR(dB)		Scheme $A + B$ Y-PSNR(dB)	
	Base Layer (20Kbps)	Enhancement Layer	Base Layer (20Kbps)	Enhancement Layer
10	28.751	30.506	28.751	28.599
20	28.751	32.155	28.751	30.914
40	28.751	34.399	28.751	34.206
80	28.751	37.210	28.751	37.592
160	28.751	40.749	28.751	41.450

By comparing Table 5.5 and Table 5.6, when starting from the higher quality base layer, we see that the gains in both of these steps are roughly the same. We can conclude that for higher quality base layers, it may not be feasible to have more enhancement layers. Moreover, this result also shows that in order to gain the most benefit from SNR scalability, it is better to have a lower quality base layer than a high quality one. This was demonstrated in the layer separation experiments using the 20Kbps and 64Kbps base layer. Figure 5.12 shows the PSNR response of enhancement layers as a function of layer separation for base layer bit

Table 5.6: Performance of SNR scalability as a function of layer separation at base layer rates of 64Kbps for HHI scheme and scheme $A + B$

Layer Separation (Kbps)	HHI Y-PSNR(dB)		Scheme $A + B$ Y-PSNR(dB)	
	Base Layer (20Kbps)	Enhancement Layer	Base Layer (20Kbps)	Enhancement Layer
10	35.207	35.221	35.207	35.293
20	35.207	35.814	35.207	35.918
40	35.207	37.080	35.207	37.299
80	35.207	39.081	35.207	39.551
160	35.207	41.981	35.207	42.415

rate at 20Kbps and 64Kbps by using proposed scheme $A + B$. For base layer rate at 64Kbps, it is shown that it may be wiser to spend 80Kbps on other forms of scalability rather than to spend it on SNR to gain only 2.864 dB PSNR increases going from 144Kbps to 224Kbps.

5.4.4 Actual Bit Rate

The second performance measure of SNR scalability, the encoded bit rate, has been consistently very good. As shown in Table 5.2, over the two layers 20-110 Kbps the greatest deviation from the desired encoded rate is 0.63Kbps when the desired 110Kbps enhancement layer of Scheme B is encoded using 110.63Kbps. At the base layer and enhancement layers, the deviations are less than 0.5Kbps. Therefore, we can conclude that over all three layers adhere to their bit rates effectively.

5.4.5 Three Layers SNR Scalable Coder

In this section, we consider the case that there are one base layer and two enhancement layers for SNR coding. The QP to generate motion information for each layer is shown in Table 5.1. *FOREMAN* sequence is used in this testing. The base layer

Figure 5.12: PSNR responses of enhancement layers as a function of layer separation for base layer bit rate at 20Kbps and 64Kbps by using proposed scheme $A + B$

is encoded at 20Kbps, the first enhancement layer plus the base layer will provide a total of 110Kbps, and the base layer plus the first and the second enhancement layers will provide a total of 180Kbps. Table 5.7 shows the performance of different schemes for three layers SNR Scalable Coder. It can be seen that the SNR scalable codecs as defined in Scheme A, Scheme B and Scheme A+B perform better than the original HHI scheme, especially for two enhancement layers. Moreover, the quality of the second enhancement layer is comparable. And in case of Scheme $A + B$, it outperforms HHI at the second enhancement layer at about 1.4dB. Figure 5.13 compares the coding efficiency of different schemes for three layers. As the number of enhancement layer increases, the proposed schemes become more

efficiency compared with original HHI scheme. Consequently, the visual quality of the frame increases as more layers are added.

Table 5.7: Performance of three layers SNR scalable coder for various schemes

Layer	Y-PSNR (dB)			
	HHI	Scheme A	Scheme B	A+B
Base Layer (20Kbps)	31.044	31.044	31.044	31.044
Enhancement Layer 1 (110Kbps)	38.995	39.295	39.809	39.876
Enhancement Layer 2 (180Kbps)	40.949	41.555	42.278	42.386

Figure 5.13: Comparison of the coding efficiency of various of three layer SNR scalable coders

5.5 Conclusions

Two novel enhancement layer ME/MC refinement schemes for SNR scalability of scalable video coding have been proposed in this chapter. The improvement in scalable coding scheme is achieved by refining both motion information and residual information at the enhancement layers by our proposed schemes. A good tradeoff between motion information and residual information can be obtained at all bit rates. The coding efficiency is improved by up to 1dB at high bit rate. It can be argued that the proposed SNR scalability requires more computational power than non-scalable encoding, this is due to the fact that for an n layer scalable video codec, n re-quantizations are necessary. In order to do this n forward and inverse transform will have to be taken for each layer for every frame. This indeed can be a drawback if a large number of layers are present. However, in the case of two or three layers, the encoding time is minimal and the performance gained by the proposed schemes can easily justify this type of multi-pass encoding process.

Enhanced Intra Coding in Scalable Video Coding

6.1 Introduction

In intra coding, various lossless and/or lossy compression techniques are performed relative to information this is contained only within the current frame rather than any other frame in a video sequence. Therefore, no temporal processing is performed outside the current frame in intra coding. It includes intra frame coding, which all the macroblocks (MB) in a frame are intra coded as well as intra MB coding in inter frames. Intra coding plays an important role in scalable video coding in the following way [6]. Intra frames are very important for use as anchor points so that the frames in the video can be accessed randomly without requiring one to decode all previous frames. In order to decode any other frame, we need only find its nearest previous intra frame. This is important for allowing reverse playback, skip-ahead, or error-recovery. Moreover, intra frames serve as the reference frames for previous and following prediction frames, therefore, the coding performance of

intra frames will affect the coding performance of other types of frames [6]. However, compared with other types of frames, compression ratios for intra frames are low. Therefore, intra frames contain the most amount of bits and take up more bandwidth during the transmission.

In recent years, a number of efforts have been made to improve the coding performance and bit rate saving of intra coding in H.264/AVC. An enhanced rate distortion cost function is proposed in [54], which combines the sum of absolute integer-transformed differences and a rate predictor for H.264/AVC intra 4×4 mode decision. In addition, a fast computation algorithm which uses the property of linear transform and the fixed spatial relationship of predicted pixels in intra modes is proposed in [54]. Intensity gradient filters are used for intra prediction in H.264/AVC encoding system in [55], which enhance the performance and efficiency of previous intra coding algorithms. Hideaki Kimata *et al.* proposed a recursively weighting pixel domain intra prediction in H.264/AVC [56]. The resulted bit rate saving for their proposed scheme is 1 to 2% in average with increasing encoder complexity. An effort has also been made by Ik-Hwan Cho *et al.* to use collocated weighted chroma pixel value to perform the intra luma prediction in H.264/AVC [57]. An adaptive intra mode bit skip method is proposed by Kim [58], in which intra 4×4 prediction is split into single prediction or multiple prediction modes depending on the smoothness of neighbor blocks boundary pixels. The mode bits are skipped in single prediction, which is used in low-detailed picture areas. Recently, a distance-based weighted prediction method is proposed in [59] for replacing the DC mode in intra 4×4 prediction. All these methods achieved some bit rate saving compared to the scheme used in H.264/AVC, but the results are not significant and at a cost of increasing encoder complexity. Consequently, an effective and simplified intra coding scheme is required for current SVC.

In SVC, intra coding is implemented by using directional prediction scheme. This prediction scheme shows superior for predicting sharp edge images. However, the performance of this scheme is not effective for the blocks containing blur luminance information [56]. In this chapter, we propose a novel intra prediction algorithm that each pixel is predicted not only from the reconstructed pixels in the neighboring blocks but also from the previously predicted pixels in the current block. Moreover, a novel prediction mode coding algorithm is proposed to increase the bit rate saving effectively. Experimental results show that our proposed scheme can improve the bit rate saving as well as the coding performance effectively over a large bit rate range. The rest of this chapter is organized as follows. In Section 6.2, intra coding scheme in scalable video coding is studied. In Section 6.3, a new intra coding scheme is proposed to achieve more efficient coding performance and more bit rate savings. Experimental results are presented in Section 6.4. Finally, Section 6.5 concludes this chapter.

6.2 Overview of Intra Coding in SVC

In SVC, intra coding is achieved by referring to neighboring pixels of previously coded blocks and it is performed in the spatial domain. The difference between the current block and their prediction is then calculated and entropy coded. SVC defines several advanced directional prediction modes, as a result, the performance of intra coding in SVC is comparable to that of the recent still image compression standard JPEG 2000 [128]. Figure 6.1 shows a block diagram of a basic video encoder for intra frames only.

Figure 6.1: Video encoder for intra frame coding in SVC

6.2.1 Intra Prediction Methods in SVC

SVC defines several spatial prediction block sizes including 16×16 and 4×4 for luma and 8×8 for chroma. Here, *INTRA* $_{16 \times 16}$ and *INTRA* $_{4 \times 4}$ are used to indicate these two kinds of MB size for luma component and intra coding for chroma component is not considered here. In addition to provide the ability for the encoder to select among the various macroblock sizes, the encoder can also select among the various prediction directions for each macroblock size. As discussed in Chapter 2, SVC defines 4 prediction modes for *INTRA* $_{16 \times 16}$ MB, and 9 prediction modes for *INTRA* $_{4 \times 4}$ MB. Figure 6.2 illustrates 12 prediction modes defined in SVC. Mode 2 is not shown in the figure, and it is the DC averaging mode. All modes shown in Figure 6.2 are directional predictions and the directions are indicated by the arrows. The encoder typically selects the prediction mode for each block that

minimizes the difference between previously encoded, reconstructed block and the block to be encoded.

Figure 6.2: (a) Prediction modes for $INTRA_{4 \times 4}$, and (b) prediction modes for $INTRA_{16 \times 16}$ defined in SVC

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Figure 6.3: Labelling of samples for 4×4 intra luma prediction

Figure 6.3 shows a 4×4 luma block that needs to be predicted. The samples from A to M, which are located above and to the left of the current sample have been encoded and reconstructed previously. Therefore, they are available in the encoder and decoder to form a prediction reference. Take mode 0 for 4×4 block as an example, pixels a, e, i, and m are predicted based on the neighboring pixel

A and so on and the resulted prediction block P is given by:

$$\mathbf{P} = \begin{bmatrix} A & B & C & D \\ A & B & C & D \\ A & B & C & D \\ A & B & C & D \end{bmatrix} \quad (6.1)$$

6.2.2 Entropy Coding of Prediction Modes in SVC

Entropy coding plays an important role in video compression. By using entropy coding, a series of symbols representing various elements of the video sequence can be converted into a compressed bit stream that is suitable for transmission or storage. Those input symbols may include quantized transform coefficients (run-level or zero tree encoded coefficients), motion vectors (an x-direction and y-direction displacement vector for each motion-compensated block, with integer or sub-pixel resolution), headers (macroblock headers, picture headers, sequence headers, etc.), etc [6]. In this section, the choice of prediction modes for each 4×4 intra-coded block and prediction modes for each 16×16 intra-coded macroblock must be signalled to the decoder, and this could require a large number of bits. In SVC, the encoder makes use of the prediction modes correlation among neighboring blocks to reduce the bit cost greatly.

In local regions of an image, some symbols are highly correlated. For example, the average or DC value of neighbouring intra-coded blocks of pixels may be very similar. Therefore, coding efficiency may be improved by predicting elements of the current block or macroblock from previously-encoded blocks and encoding the difference between the actual value and the prediction. Similarly, intra modes for neighbouring 4×4 blocks are often correlated. As shown in Figure 6.4, A, B and E are the left, upper and current 4×4 blocks respectively. If previously-encoded 4×4

blocks A and B are predicted using mode 1, it is probable that the best prediction mode for block E is also mode 1. To take advantage of this correlation, predictive coding is used to signal 4×4 intra modes in SVC.

Figure 6.4: Current and neighboring 4×4 blocks

For each current block E, the encoder and decoder calculate the most probable prediction mode, the minimum of the prediction modes of the above block and the left block [6]. If either of these neighboring blocks is not available, which is outside the current frame or not coded in $INTRA_{4 \times 4}$, the corresponding value of the above block or the left block is set to DC prediction mode.

In the coding stage, the encoder sends a flag for each 4×4 block, *prev_intra4 × 4_pred_mode*. If the flag has a value of ‘1’, the most probable prediction mode is used. If the flag has a value of ‘0’, another parameter *rem_intra4 × 4_pred_mode* (*rem_mode*) is sent to indicate a change of mode [6]. If *rem_mode* is smaller than the current most probable mode, then the prediction mode is set to *rem_mode*. Otherwise the prediction mode is set to (*rem_mode* + 1). In this way, only eight values of *rem_mode* are required (0 to 7) to signal the current intra mode (0 to 8) [6]. Table 6.1 shows the choice of *rem_mode* corresponding to each prediction mode when the most probable prediction mode is 2. The prediction mode for macroblock coded in $INTRA_{16 \times 16}$ is signalled in the macroblock header and predictive coding of the mode is not used in this case.

Table 6.1: Choice of prediction mode

rem_mode	Prediction Mode	Bit Pattern	Bit Pattern for
	for Block E	for Flag	rem_mode
0	0	0	000
1	1	0	100
2	3	0	010
3	4	0	110
4	5	0	001
5	6	0	101
6	7	0	011
7	8	0	111
-	2	1	-

6.3 Our Proposed Intra Coding Scheme

6.3.1 Proposed Intra Prediction Algorithm

If an MB is encoded in $INTRA_4 \times 4$, each 4×4 block of the luma component utilizes one of nine prediction modes. If current prediction mode is one of eight directional prediction beside DC mode, then it shows an edge in that direction across the block. If the current block contains sharp edge contents, then eight directional prediction modes used in SVC can perform the prediction effectively. If current block contains very smooth contents, then DC mode used in SVC can perform the prediction effectively. However, for some cases, there is an edge contained in the block, but the edge is not very sharp and obvious. For such cases, both the DC prediction and directional prediction cannot perform effectively, and our proposed algorithm can be used to solve this problem. In addition to 9 prediction modes proposed in SVC, we propose 9 more prediction modes (mode a - mode i) for each 4×4 block and 3 more prediction modes (mode a - mode c) for each 16×16 block in our scheme as shown in Figure 6.5. The details of each newly proposed mode are shown in the following section.

As an alternative to the prediction modes shown in equation (6.1), we suggest

Figure 6.5: (a) Newly proposed prediction modes for $INTRA_{4 \times 4}$, and (b) newly proposed prediction modes for $INTRA_{16 \times 16}$

that each pixel in a current block is predicted from its surrounding pixels. The surrounding pixels can be the reconstructed pixels in neighboring blocks or the predicted pixels in the current block. As shown in Figure 6.3, the prediction pixels of each sample a, b, e, and f for different prediction modes are calculated by pixel-by-pixel as follows. Similarly, the prediction pixel of other samples in the 4×4 block can be generated. For DC prediction (mode c), the pixels in the current block are predicted by average the above, left, above to the left and above to the right pixels. Moreover, the similar prediction method for mode a, mode b and mode c for 4×4 blocks can be extended to 16×16 block.

- Proposed vertical prediction (mode a)

$$\begin{aligned}
 Pred(a) &= (2 * A + M + I + 2) >> 2 \\
 Pred(b) &= (2 * B + A + Pred(a) + 2) >> 2 \\
 Pred(e) &= (2 * Pred(a) + I + J + 2) >> 2 \\
 Pred(f) &= (2 * Pred(b) + Pred(a) + Pred(e) + 2) >> 2 \quad (6.2)
 \end{aligned}$$

- Proposed horizontal prediction (mode b)

$$\begin{aligned}
Pred(a) &= (2 * I + M + A + 2) >> 2 \\
Pred(b) &= (2 * Pred(a) + A + B + 2) >> 2 \\
Pred(e) &= (2 * J + I + Pred(a) + 2) >> 2 \\
Pred(f) &= (2 * Pred(e) + Pred(a) + Pred(b) + 2) >> 2 \quad (6.3)
\end{aligned}$$

- Proposed diagonal-down-left prediction (mode d)

$$\begin{aligned}
Pred(a) &= (2 * B + Pred(b) + A + 2) >> 2 \\
Pred(b) &= (2 * C + Pred(c) + B + 2) >> 2 \\
Pred(e) &= (2 * Pred(b) + Pred(a) + Pred(f) + 2) >> 2 \\
Pred(f) &= (2 * Pred(c) + Pred(g) + Pred(b) + 2) >> 2 \quad (6.4)
\end{aligned}$$

- Proposed diagonal-down-right prediction (mode e)

$$\begin{aligned}
Pred(a) &= (2 * M + I + A + 2) >> 2 \\
Pred(b) &= (2 * A + Pred(a) + B + 2) >> 2 \\
Pred(e) &= (2 * I + Pred(a) + J + 2) >> 2 \\
Pred(f) &= (2 * Pred(a) + Pred(b) + Pred(e) + 2) >> 2 \quad (6.5)
\end{aligned}$$

- Proposed vertical-right prediction (mode f)

$$\begin{aligned}
Pred(a) &= (M + A + 1) >> 1 \\
Pred(b) &= (A + B + 1) >> 1
\end{aligned}$$

$$\begin{aligned}
Pred(e) &= (I + Pred(a) + 1) >> 1 \\
Pred(f) &= (Pred(a) + Pred(b) + 1) >> 1
\end{aligned} \tag{6.6}$$

- Proposed horizontal-down prediction (mode g)

$$\begin{aligned}
Pred(a) &= (M + I + 1) >> 1 \\
Pred(b) &= (A + Pred(a) + 1) >> 1 \\
Pred(e) &= (I + J + 1) >> 1 \\
Pred(f) &= (Pred(a) + Pred(e) + 1) >> 1
\end{aligned} \tag{6.7}$$

- Proposed vertical-left prediction (mode h)

$$\begin{aligned}
Pred(a) &= (A + B + 1) >> 1 \\
Pred(b) &= (B + C + 1) >> 1 \\
Pred(e) &= (Pred(a) + Pred(b) + 1) >> 1 \\
Pred(f) &= (Pred(b) + Pred(g) + 1) >> 1
\end{aligned} \tag{6.8}$$

- Proposed horizontal-up prediction (mode i)

$$\begin{aligned}
Pred(a) &= (I + J + 1) >> 1 \\
Pred(b) &= (Pred(a) + Pred(e) + 1) >> 1 \\
Pred(e) &= (J + K + 1) >> 1 \\
Pred(f) &= (Pred(e) + Pred(i) + 1) >> 1
\end{aligned} \tag{6.9}$$

In order to illustrate our intra prediction algorithm, we take a 4×4 block as

an example, and the algorithm is shown as follows. The same steps can be applied to a 16×16 block.

Step 1, the nine prediction modes defined in SVC is used, and rate distortion cost function is calculated for each possible prediction mode. After that, the best prediction mode, that results the lowest rate distortion cost (`rdcost_best`), is known for the current block.

Step 2, according to the edge direction resulted from step 1, one of the nine newly proposed prediction modes (mode a to mode i) will be selected, and its rate distortion cost (`rdcost_new`) will be calculated.

Step 3, we compare `rdcost_best` and `rdcost_new` in this step. If `rdcost_best` is lower, then the best prediction mode from step 1 will be used. Otherwise, the best prediction mode will be changed to the prediction mode we get from step 2 and the `rdcost_best` is also changed accordingly.

In order to signal the difference of mode a to mode i from mode 0 to mode 8 to the decoder, a flag is used in the coding stage. If the best prediction mode is from mode 0 to mode 8, then '0' will be coded. Otherwise, '1' will be coded in the coding stage. In addition, the encoding complexity of our proposed algorithm is slightly increased. In our intra prediction algorithm, five prediction modes are preformed instead of four in SVC for each 16×16 block; and ten prediction modes are performed instead of nine in SVC for each 4×4 block.

6.3.2 Proposed Entropy Coding of Prediction Modes

As discussed in the previous section, SVC uses 1 flag and 3 bits to code the prediction modes for *INTRA*- 4×4 blocks. For the current block, if the prediction mode for the current block is the same as the most probable prediction mode, then flag

'1' is coded, and only one bit is used for the coding process. However, if the prediction mode for the current block is different from the most probable prediction mode, then flag '0' is coded, and another 3 bits are used to code the change of the prediction mode. As a result, four bits are used to code the prediction mode in this case.

If the most probable prediction mode can be effectively predicted from its neighboring blocks, then the bit rate will be reduced. From experiments, we observed that if a 4×4 block contains an edge, then most likely the edge will be extended to its neighboring 4×4 blocks. Therefore, due to the inherent spatial correlation within a single frame, the direction of the edge in the current block can be predicted by that of the neighboring blocks. As illustrated in Figure 6.6(a), if the edge crosses through the upper block B in the vertical direction, the direction of the edge in current block E is likely to be vertical too. Therefore, if the prediction mode for B is in the vertical direction, it is likely that the prediction mode for E is in the vertical direction. Similarly, as shown in Fig. 6.6(b), if the prediction mode for A is in the horizontal direction, it is likely that the prediction mode for E is in the horizontal direction.

Figure 6.6: Direction correlations for neighboring 4×4 blocks

Considering the above observations, the most probable prediction mode is generated based on the following rules in our proposed scheme:

- 1) If the above block B has prediction mode of 0, 7 or 5, then there is a high

probability that the current block E has prediction mode of 0, 7 or 5, therefore, the most probable mode for E is set to 0, 7 or 5.

2) Otherwise, if the left block A has prediction mode of 1 or 8, then there is a high probability that the current block has prediction mode of 1 or 8, the most probable mode for E is set to 1 or 8.

3) Otherwise, if the above to the left block C has prediction mode of 4, then there is a high probability that the current block E has prediction mode of 4, the most probable mode for E is set to 4.

4) Otherwise, if the above to the right block D has prediction mode of 3, then there is a high probability that the current block has prediction mode of 3, the most probable mode for E is set to 3.

5) Otherwise, the comparison of the prediction modes of the above block B and the left block A are performed and the most probable mode for E takes the smaller value.

By using our proposed coding method, the chance of predicting the most probable mode correctly is increased, and the bit rate is reduced. The result can be shown in the following section.

6.4 Experimental Results

The proposed intra coding scheme is embedded in JSVM encoder [91]. The test platform used is Intel Pentium IV, 1.83GHz CPU, 2G RAM with Windows XP professional operating system. In our experiments, test sequences *FOREMAN*, *FOOTBALL*, *HARBOUR*, *CITY* with the resolution of QCIF are used as shown in Table 6.2. Moreover, we only consider the one layer case and all the frames are intra coded in our experiments. The quantization parameter (QP) is varied from

40 to 20 in our experiments. The testing parameters in our experiments include Y-PSNR difference (ΔPSNR) and bit rate difference (ΔBR).

Table 6.2: Test sequences in our experiments

Video Sequence	Frames
FOREMAN (QCIF)	100 (15Hz)
FOOTBALL (QCIF)	100 (15Hz)
HARBOUR (QCIF)	100 (15Hz)
CITY (QCIF)	100 (15Hz)

Table 6.3: Simulation results for proposed intra prediction algorithm with QP value of 40

Sequence	ΔPSNR [dB]	ΔBR [%]
<i>FOREMAN</i>	0.056	-6.439
<i>FOOTBALL</i>	0.142	-2.038
<i>HARBOUR</i>	0.112	-1.152
<i>CITY</i>	0.111	-2.037
Average	0.105	-2.917

Table 6.4: Simulation results for proposed intra prediction algorithm with QP value of 30

Sequence	ΔPSNR [dB]	ΔBR [%]
<i>FOREMAN</i>	0.172	-3.626
<i>FOOTBALL</i>	0.159	-1.515
<i>HARBOUR</i>	0.105	-1.275
<i>CITY</i>	0.162	-1.234
Average	0.150	-1.913

Table 6.3, 6.4 and Table 6.5 present our proposed Y-PSNR and bit rate difference compared to JSVM encoder at different QP value settings. It can be seen that our proposed intra prediction scheme achieve consistent video quality improvement and bit rate reduction. In addition, Figure 6.7 to 6.8 show the rate distortion curves for *FOREMAN* and *CITY* video sequences of our proposed scheme and the original scheme used in SVC. From these figures, we can conclude that our

Table 6.5: Simulation results for proposed intra prediction algorithm with QP value of 20

Sequence	Δ PSNR [dB]	Δ BR [%]
<i>FOREMAN</i>	0.244	-1.591
<i>FOOTBALL</i>	0.261	-0.439
<i>HARBOUR</i>	0.195	-0.782
<i>CITY</i>	0.267	-0.407
Average	0.242	-0.805

Figure 6.7: Rate distortion curve for *FOREMAN*

scheme can achieve consistent and improved coding performance over a large bit rate range. However, compared with original JSVM intra prediction method, each 4×4 block of luma component utilize one of ten prediction modes instead of nine prediction modes and each 16×16 macroblock utilize one of five prediction modes instead of four. This will result in higher computational complexity. As shown in

Figure 6.8: Rate distortion curve for *CITY*

Table 6.6, the encoding time is increased by 11% on average.

Table 6.6: Encoding time comparison

Sequence	Encoding Time Increment [%]
<i>FOREMAN</i>	8.05
<i>FOOTBALL</i>	14.29
<i>HARBOUR</i>	16.67
<i>CITY</i>	6.25
Average	11.32

6.5 Conclusions

In this chapter, an enhanced intra coding scheme for scalable video coding is presented. With this scheme, more effective prediction modes for 4×4 and 16×16 blocks are proposed, respectively. Moreover, the algorithm for predicting the most probable prediction mode is improved. The results of the simulations in Section 6.4 demonstrate that the proposed algorithm could achieve the bit rate saving as compared to the original JSVM system, without a substantial increase in the complexity of the encoding and decoding processes.

Conclusions and Future Research

Directions

7.1 Conclusions

In this section, the original contributions of the thesis and the suggested research directions are described. Digital video processing is a difficult task because of the high bandwidth and low latency requirements. When coupled with making the algorithms scalable, it becomes an interesting and useful problem to analyze. The aim of this thesis is to provide solutions for two major problems the scalable video coding standard are facing: relatively high computational complexity and a lack of coding efficiency in comparison with single layer coding. The primary contributions of this thesis can be summarized in four major areas.

- (i) Fast mode decision schemes are proposed for spatial, SNR, temporal and combined scalable video coding system, respectively. The macroblock mode distribution relationship between the base layer and its enhancement layers is explored. The number of candidate modes in a macroblock that take part

in rate distortion optimization calculation at the enhancement layers has been reduced significantly. It is observed that the computation complexity is reduced, and the encoding time saving is achieved as well as the video quality is maintained.

- (ii) Decoder computational complexity issue for coarse grain SNR scalability is studied. An adaptive encoding scheme that would generate light weight bit streams which require less amount of computational power at the decoder is proposed. This is achieved by using a new rate-distortion-complexity optimized cost function for motion estimation and mode decision as compared to traditional scalable video encoding algorithms. In this way, our proposed scheme could reduce the decoding computational complexity effectively and flexibly with insignificant performance degradation.
- (iii) Two novel enhancement layer motion estimation and motion compensation schemes are proposed for SNR scalable video coding. The improvement is achieved by refining both motion information and residual information at the enhancement layers. Hence, a good tradeoff between these two kinds of information is achieved at all bit rates. Consequently, the coding efficiency of the video encoder with our proposed schemes is improved.
- (iv) A novel intra coding scheme is proposed for scalable video coding system. More effective prediction modes for 4×4 and 16×16 blocks are proposed, respectively. Moreover, an algorithm for predicting the most probable prediction modes is improved. As a result, the coding efficiency is improved without a substantial increase in the complexity of the encoding and decoding process.

In comparison to the existing scalable video coding scheme, the video coder with our proposed fast mode decision algorithms achieves significant computational gain for spatial, SNR, temporal and combined scalability. In addition, light weight bit streams, which require significantly less decoding complexity, could be generated at the SNR scalable video encoder with our proposed adaptive encoding scheme. Furthermore, the video coder with our proposed rate control scheme for SNR scalability and intra coding scheme achieves better subjective and objective visual quality.

7.2 Future Research Directions

In this section, directions for further investigation into some of the aspects of scalable video coding are proposed.

- Motion Vector Entropy Coding

The goal of digital video compression algorithms is to remove the temporal redundancy. In order to achieve temporal redundancy reduction, object motions in a frame are modelled by motion vectors (MVs). A motion vector is the important element in the motion estimation and motion compensation process in video compression. Therefore, the bit rate used to represent MVs can become a non-trivial portion of the total bit rate used to transmit video data. In order to reduce the bit rate consumption and remove the redundancy of the MVs, each MV is predicted from neighboring MVs, thus generating MVP. After that, the difference between the MV and MVP, motion vector difference (MVD), is calculated and entropy coded. In the state-of-the-art H.264/AVC coding system, context-based adaptive binary arithmetic coding (CABAC) is used. Context modelling in data compression has become a

significant topic of research, being considered an attractive feature for highly efficient video coding. The basic idea behind context modelling in video compression is that the probability of a currently encoded symbol may depend on a number of prior encoded symbols or parts of them, called context parameters. Thus, the challenge is to design an algorithm for selection of an appropriate subset of symbols from the observed data for the estimation of conditional probabilities such that the code length of the data sequence will be minimized [129]. Moreover, probability models are adaptively modified in adaptive arithmetic coding. The CABAC coding process for MVD_x (motion vector difference in the x-direction, coded for each partition or sub-macroblock partition in an inter macroblock) is as follows.

- (i) Binarise and convert the value MVD_x to a sequence of bins as described in [73].
- (ii) A context model is chosen for each binarised bins. As shown in Table 7.1, for each vector component, all together 7 different models are used. One of three models is selected for the first bin (Table 7.2), based on the norm of two previously coded MVD_x (as shown in Figure 7.2) values, e_k :

$$e_k = |MVD_{x,A}| + |MVD_{x,B}| \quad (7.1)$$

where A is the block immediately to the left of the current block and B is the block immediately to the above of the current block. If e_k is small, then the probability that the current MVD will have a small magnitude is high. Conversely, if e_k is large, then it is more likely that the current MVD will have a large magnitude. After that, a probability table (context model) is selected accordingly. One of four further context

models are used to code the remaining bits [73].

(iii) Encode each bin and update the context models.

In current scalable video coding, the above process is used for entropy coding of motion information in both base layer and enhancement layer. However, there are some correlation between the base layer information and enhancement layer motion information for scalable coding. The enhancement layer motion vector is predicted from the base layer motion vector for certain macroblock. Moreover, the enhancement layer and the base layer share the same macroblock partition type and block prediction mode in proposed scheme A. Therefore, different kinds of modified context models should be designed based on this correlation to improve the probability estimation in scheme A. The accurate probability estimation can thus improve the efficiency of context-based arithmetic coder. Therefore, in proposed entropy coding scheme, the proper probability model among three models is selected using the following equation 7.2.

$$e_k = 2 \times \lfloor \text{MVD}_{x,C,base} \rfloor \quad (7.2)$$

where the value of e_k determines the context index by the method shown in Table 7.2. $\text{MVD}_{x,C,base}$ is the motion vector difference in x direction of C macroblock in base layer. Compared to the method used in scalable extension of H.264/AVC scheme, the distribution of the motion vector difference can be estimated more precisely by using our proposed method. The information of the same block instead of neighboring blocks which may belong to different objects is used in our model. Especially for blocks with larger block size (such as 16×16 or more), the neighboring MVs have less correlation since they

are spatially more apart from each other. Therefore, our proposed method is independent of the selected block size and efficiently and accurately models the MVD context in all MB division modes in enhancement layers.

Figure 7.1: Illustration of two neighboring macroblock A and B of current macroblock C

Table 7.1: Context models	
Bin	Context Model
1	0,1 or 2 depending on e_k
2	3
3	4
4	5
5	6
6	6

Table 7.2: Context models for bin 1	
e_k	Context Model for Bin 1
$0 \leq e_k < 3$	Model 0
$3 \leq e_k < 32$	Model 1
$32 \leq e_k$	Model 2

- Detection of Object of Human Interest for Video Coding Applications

In order to transmit enormous amounts of video data via an error prone and bandwidth limited network, the data must be compressed to meet the bits budget. However, too strong compression may cause quality degradation due to data loss. Furthermore, not all lost data induce the same impact on human perception [130]. The perceived quality is usually dominated by the quality in human interest regions. Therefore, spotting the region with the greatest human interest and allocating more bits to this region is an important task to improve the compression efficiency perceptually of current video coding systems.

Table 7.3 shows human interest regions for various video sequences. One of the main characteristics of any video sequences listed in the table is the presence of diverse types of motion. Such motion can be produced by static objects with a moving background due to camera motion, objects moving in different trajectories, or simply random motion objects [131]. In these scenarios, a viewer gives priority to face (such as face in *NEWS* and *MOTHER & DAUGHTER*) and moving objects (such as moving ship in *COASTGUARD* and moving human in *STEFAN*). Consequently, the subjective picture quality can be improved by spending more bits on objects of human interest such as faces and moving objects, at the cost of lower image quality in the visually less important background [132].

Following steps could be adopted for detection of objects of human interest and region of interest (ROI) coding by using scalable video coding.

- (i) Detection of objects of human interest

Moving object detection is to separate the objects from the background.

The combination of adaptive background subtraction and three frame

Table 7.3: Human interest regions for various video sequences

Video Sequence	Human Interest Regions
AKIYO	Face
BUS	Moving bus
COASTGUARD	Moving ship
CONTAINER	Moving ship
DANCER	Moving human
HALL_MONITOR	Moving human
MOTHER & DAUGHTER	Faces
MOBILE	Moving toy
SILENT	Face
STEFAN	Moving human
CREW	Moving human group
FOREMAN	Face
FOOTBALL	Moving human group
HARBOUR	Moving ship
ICE	Moving human group
NEWS	Face
SOCCKER	Moving human group

differencing could be used to detect moving objects in a video sequence.

The major drawback of adaptive background subtraction is that it makes no allowance for stationary objects in the scene that start to move since the stationary objects are learned into the background. When the stationary objects start to move, “holes” are left behind where the newly exposed background image differs from the learned background model. Eventually, these “holes” are learned into the background model gradually depending on the background modelling speed, however, they generate false alarms for a short period of time. Frame differencing is not subject to this phenomenon; however, it is generally not an effective method for extracting the entire shape of a moving object especially when the object moving speed is slow. To overcome these problems,

these two methods could be combined. First, we can use three frame differencing method to determine regions of legitimate motion. Then, adaptive background subtraction method is used to extract the entire moving region.

For face detector, skin color and shape information can be utilized [133]. In addition, a window-sliding technique could be employed. That is, the classifier (such as Eigenfaces [134]) is used to classify the portions of an image, at all locations and scales, as either faces or non-faces.

(ii) Video coding applications

The face area or the moving object area extracted by the detection and tracking algorithm is the focus of attention in the video communication application and has to be coded with high quality than the less relevant background. For region of interest (ROI) coding different quantization parameters are used for macroblocks that belong to the face or moving object region and macroblocks belonging to the background. An encoder with a smaller quantization parameter usually could generate more encoding bits and achieve better visual quality.

Author's Publications

Journal Paper:

- He Li, Z. G. Li, and C. Y. Wen, “Fast mode decision algorithm for inter-frame coding in fully scalable video coding,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 889–895, July 2006 (SCI Times Cited: 36 (without self-citation)).

Conference Paper:

- He Li, Z. G. Li, C. Y. Wen, and L.-P. Chau, “Fast mode decision for spatial scalable video coding,” *IEEE International Symposium on Circuits and Systems*, pp. 3005–3008, May 2006.
- He Li, Z. G. Li, and C. Y. Wen, “Fast mode decision for coarse grain SNR scalable video coding,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 1793–1796, May 2006.
- He Li, Z. G. Li, C. Y. Wen, and S. Xie, “Fast mode decision for coarse granular scalability via switched candidate mode set,” *IEEE International Conference on Multimedia and Expo*, pp. 1323–1326, 2007.

-
- He Li, Z. G. Li, and C. Y. Wen, “Fast mode decision for temporal scalable video coding,” *Proceedings Picture Coding Symposium*, 2006.
 - He Li, Z. G. Li, and C. Y. Wen, “Adaptive Decoder Complexity Reduction for Coarse Granular Scalability,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1145-1148, 2007.

Bibliography

- [1] R. Schafer, T. Wiegand, and H. Schwarz, “The emerging H.264/AVC standard,” *EBU Technical Review*, vol. 1, pp. 253–255, 2003.
- [2] Z.-Y. Chen, J.-W. Syu, and P.-C. Chang, “Fast inter-layer motion estimation algorithm on spatial scalability in H.264/AVC scalable extension,” *IEEE International Conference on Multimedia & Expo*, pp. 442–446, 2010.
- [3] C. A. Segall and G. J. Sullivan, “Spatial scalability within the H.264/AVC scalable video coding extension,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, 2007.
- [4] A. C. Bovik, *Handbook of Image and Video Processing*. Elsevier Academic Press, 2005.
- [5] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Prentice Hall, September 2001.
- [6] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia Subject Digital Video*. Wiley and Sons, December 2003.

-
- [7] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. New York, Academic, 1990.
 - [8] L. Debnath, *Wavelet Transform and Their Applications*. Springer, Hardcover, 2000.
 - [9] T. Sikora, "MPEG digital video-coding standards," *IEEE Trans. on Signal Processing*, vol. 14, no. 5, pp. 82–100, September 1997.
 - [10] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. L. Gall, *MPEG Video Compression Standard*. Chapman and Hall, 1996.
 - [11] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An introduction to MPEG-2*. Chapam Hall, 1997.
 - [12] J. R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Trans. on Image Processing*, vol. 3, no. 9, pp. 559–571, September 1994.
 - [13] H. Schwarz, D. Marpe, and T. Wieg, "MCTF and scalability extension of H.264/AVC," in *Proc. of PCS 2004*, 2004.
 - [14] R. Schafer, H. Schwarz, D. Marpe, T. Schierl, and T. Wiegand, "MCTF and scalability extension of H.264/AVC and its application to video transmission, storage, and surveillance," in *Proceedings of SPIE Symposium on Visual Communications and Image Processing (VCIP'05)*, Beijing, China, July 2005.
 - [15] D. Salomon, *Data Compression: the Complete Reference*. Springer-Verlag London Limited, 2007.

- [16] I. Witten, R. Neal, and J. Cleary, “Arithmetic coding for data compression,” *Communications of the ACM*, vol. 30, pp. 520–540, 1997.
- [17] M. Nelson and J.-L. Gailly, *The Data Compression Book*. Ed. M&T Books, 1996.
- [18] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS,” *IEEE Trans. on Image Processing*, vol. 9, pp. 1309–1324, August 2000.
- [19] T. Wiegand, G. J. 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, July 2003.
- [20] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, “Video coding with H.264/AVC: Tools, performances, and complexity,” *IEEE Circuits and Systems Magazine*, pp. 7–28, 2004.
- [21] “Video codec for audiovisual services at $p \times 64$ kbit/s,” CCITT Recommendation H.261, CCITT doc COM XV-R 37-E, 1995.
- [22] “Video coding for low bit rate communication,” ITU-T, Recommendation H.263, Version 2, January 1998.
- [23] “H.261 test model long term number 9 (tml-9) draft 0,” ITU-T, Video Coding Expert Group (VCEG), December 2001.
- [24] “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 Coding of Moving Pictures and

- Associated Audio MPEG97/N1796 Doc. JVT-G050r1, Geneva, Switzerland, May 2003.
- [25] “Coding of moving pictures and associated audio for digital storage media at up to 1.5 mbps - Part2:Video,” ISO/IEC 11172-2, 1993.
- [26] “MPEG-2 test model 5,” Doc. ISO/IEC JTC1/SC29 WG11/93-400, April 1993.
- [27] “Overview of the MPEG-4 standard,” ISO/IEC JTC1/SC29/WG11/N4668, March 2002.
- [28] P. Amon, H. Li, A. Hutter, D. Renzi, and S. Battista, “Scalable video coding and transcoding,” *IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 336–341, May 2008.
- [29] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, pp. 1103–1120, September 2007.
- [30] “Applications and requirements for scalable video coding,” ISO/IEC JTC1/SC29/WG11/N5880, Trondheim, Norway, July 2003.
- [31] X. Q. Yi, J. Zhang, N. Ling, and W. J. Shang, “Improved and simplified fast motion estimation for JM,” JVT-P021, 16th Meeting of JVT of ISO/IEC MPEG & ITU-T VCEG, Poznan, Poland, 24 - 29, July 2005.
- [32] X. Q. Yi and N. Ling, “Improved normalized partial distortion search with dual-halfway-stop for rapid block motion estimation,” *IEEE Transactions on Multimedia*, vol. 9, no. 5, pp. 995–1003, August 2007.

- [33] F. Pan, X. Lin, R. Susanto, K. P. Lim, Z. G. Li, G. N. Feng, D. J. Wu, and S. Wu, "Fast mode decision algorithm for intraprediction in H.264/AVC video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 7, pp. 813–822, July 2005.
- [34] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, S. Rahardja, and C. C. Ko, "Fast intermode decision in H.264/AVC video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 7, pp. 953–958, July 2005.
- [35] A. C. Yu, "Efficient block-size selection algorithm for inter-frame coding in H.264/MPEG-4 AVC," *IEEE International Conference on Acoustics, Speech, Signal Processing*, pp. 169–172, May 2004.
- [36] A. C. Yu and G. R. Martin, "Advanced block size selection algorithm for inter frame coding in H.264/MPEG-4 AVC," *IEEE International Conference on Image Processing*, pp. 95–98, 2004.
- [37] C. Grecos and M. Y. Yang, "Fast inter mode prediction for p slices in the H.264 video coding standard," *IEEE Trans. Broadcasting*, vol. 51, no. 2, pp. 256–263, June 2005.
- [38] 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. on Circuits and Systems for Video Technology*, vol. 15, no. 3, pp. 378–401, March 2005.
- [39] 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. on Circuits and Systems for Video Technology*, vol. 16, no. 12, pp. 1557–1561, December 2006.
- [40] P. Yin, H. Tourapis, A. Tourapis, , and J. Boyce, “Fast mode decision and motion estimation for JVT/H.264,” in *Proc. IEEE International Conference on Image Processing*, September 2003.
- [41] X. Jing and L.-P. Chau, “Fast approach for H.264 inter mode decision,” *Electronics Letters*, vol. 40, no. 17, August 2004.
- [42] A. Ahmad, N. Khan, S. Masud, and M. A. Maud, “Selection of variable block sizes in H.264,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 173–176, May 2004.
- [43] K. Ugur, J. Lainema, A. Hallapuro, and M. Gabbouj, “Generating H.264/AVC compliant bitstreams for lightweight decoding operation suitable for mobile multimedia systems,” *IEEE International Conference on Acoustics, Speech, Signal Processing*, May 2006.
- [44] Y. Wang and S.-F. Chang, “Complexity adaptive H.264 encoding for light weight streams,” *IEEE International Conference on Acoustics, Speech, Signal Processing*, May 2006.
- [45] V. Lappalainen, A. Hallapuro, and T. D. Hamalainen, “Complexity of optimized H.26L video decoder implementation,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 717–725, July 2003.
- [46] X. Zhou, E. Li, and Y. K. Chen, “Implementation of H.264 decoder on general-purpose processors with media instructions,” *Proc. VCIP*, January 2003.

-
- [47] J. Valentim, P. Nunes, and F. Pereira, "Evaluating MPEG-4 video decoding complexity for an alternative video complexity verifier model," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 11, pp. 1034–1044, December 2002.
- [48] S.-W. Lee and C.-C. Kuo, "Complexity modeling for motion compensation in H.264/AVC decoder," *IEEE International Conference on Image Processing*, vol. 5, 2007.
- [49] T.-A. Lin, T.-M. Liu, and C.-Y. Lee, "A low-power H.264/AVC decoder," *IEEE VLSI-TSA International Symposium on VLSI Design, Automation and Test*, pp. 283–286, August 2005.
- [50] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301–317, March 2001.
- [51] A. Secker and D. Taubman, "Highly scalable video compression with scalable motion coding," *IEEE Trans. on Image Processing*, vol. 13, no. 8, August 2004.
- [52] H. Schwarz, D. Marpe, and T. Wiegand, "Scalable extension of H.264/AVC," ISO/IEC JTC1/SC29 WG11 MPEG2004/M10569/S03, Munchen, Germany, March 2004.
- [53] Y. Wu, A. Golwelkar, and J. W. Woods, "Information technology-coding of audio-visual objects-part 10: Advance video coding," ISO/IEC FDIS 14496-10.

- [54] C.-H. Tseng, H.-M. Wang, and J.-F. Yang, "Enhanced intra- 4×4 mode decision for H.264/AVC coders," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 8, pp. 1027–1032, August 2006.
- [55] 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. on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 694–698, May 2008.
- [56] H. Kimata, M. Kitahara, and Y. Yashima, "Recursively weighting pixel domain intra prediction on H.264," *Proceedings of SPIE*, vol. 5150, pp. 2035–2042, 2003.
- [57] I.-H. Cho, J.-H. Lee, W.-H. Lee, and D.-S. Jeong, "New intra luma prediction mode in H.264/AVC using collocated weighted chroma pixel value," *Proceedings of SPIE*, vol. 6077, pp. 1–8, 2006.
- [58] D. Y. Kim, K. H. Han, and Y. L. Lee, "Adaptive intra mode bit skip in intra coding," *IEEE Asia Pacific Conference on Circuits and Systems 2008*, pp. 446–449, November 2008.
- [59] S. S. Yu, Y. Gao, J. Z. Chen, and J. L. Zhou, "Distance-based weighted prediction for H.264 intra coding," *IEEE IET International Conference on Audio, Language and Image Processing*, pp. 1477–1480, July 2008.
- [60] "Information technology - generic coding of moving pictures and associated audio information: Video," ISO/IEC 13818-2, MPEG-2 Video Coding Standard, 1995.
- [61] S. Battista, F. Casalino, and C. Lande, "A multimedia standard for the third millennium," *IEEE Multimedia*, vol. 6, no. 4, pp. 74–83, 1997.

-
- [62] N. Brady, “MPEG-4 standardized methods for the compression of arbitrarily shaped video objects,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1170–1189, 1999.
- [63] “H.26L Test Model Long Term Number 9 (TML-9) Draft 0,” ITU-T, Video Coding Expert Group (VCEG), December 2001.
- [64] “Report of the formal verification tests on AVC,” ISO/IEC 14496–10|ITU-T Rec. H. 264 MPEG2003/N6231, December 2003.
- [65] W. Thomas, W. Steffen, P. Torsten, S. Bernhard, and K. Florian, “Advances in hybrid video coding,” *Proceedings of the SPIE, Visual Communications and Image Processing*, vol. 6508, 2007.
- [66] M. Ezhilarasan and P. Thambidurai, “A hybrid transform coding for video codec,” *International Conference on Information Technology*, 2006.
- [67] N. Ling, “Expectations and challenges for next generation video compression,” *5th IEEE Conference on Industrial Electronics and Applications*, pp. 2339–2344, 2010.
- [68] J. Chen, U.-V. Koc, and K. J. R. Liu, *Design of Digital Video Coding Systems*. Marcel Dekker, 2001.
- [69] “H.264 joint video surveillance group compression research data:2008,” Joint Video Team, ISO/IEC JTC1/SC29/WG11, Document N6716, October 2008.
- [70] S. Ma, W. Gao, and Y. Lu, “Rate-distortion analysis for H.264/AVC video coding and its application to rate control,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, pp. 1533–1544, December 2005.

-
- [71] “H.264 advanced video coding: A whirlwind tour,” PixelTools.
- [72] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, “Adaptive deblocking filter,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, 2003.
- [73] D. Marpe, H. Schwarz, and T. Wiegand, “Context based adaptive binary arithmetic coding in the H.264/AVC video compression standard,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, July 2003.
- [74] G. J. Sullivan and T. Wiegand, “Video compression - from concepts to the H.264/AVC standard,” *Proceedings of the IEEE*, vol. 93, no. 1, January 2005.
- [75] “Advanced video coding for generic audio-visual services,” ITU-T Recommendation H.264 Amendment 3, ISO/IEC 14496-10/2005:Amd 3 - Scalable Extension of H.264 (SVC), July 2007.
- [76] J. V. Team, “Technical description of the HHI proposal for SVC CE1,” October 2004.
- [77] R. M. W. and H. S., “Scalable video model 3.0,” October 2004.
- [78] H. Schwarz, D. Marpe, and T. Wiegand, “Advanced video coding for generic audiovisual services,” ITU-T Recommendation H.264 & ISO/IEC 14496-10 AVC,(version 1: 2003, versions 2: 2004) version 3: 2005, 2005.
- [79] J. V. Team, “Scalable video coding working draft 1,” Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, Document JVT-N020, Januray 2005.

- [80] “Applications and requirements for scalable video coding,” ISO/IEC JTC1/SC 29/WG 11, Doc. N6052, October 2003.
- [81] J.-R. Ohm, “Advances in scalable video coding,” *Proceedings of IEEE*, vol. 93, pp. 42–56, January 2005.
- [82] Z. G. Li, Y. C. Soh, and C. Y. Wen, “Switched and impulsive systems: analysis, design and applications,” *Springer-Verlag*, pp. 197–219, November 2004.
- [83] H.-C. Huang, W.-H. Peng, T. Chiang, and H.-M. Hang, “Advances in the scalable amendment of H.264/AVC,” *IEEE Communications Magazine*, vol. 45, 2007.
- [84] M. van der Schaar and H. Radha, “A hybrid temporal-SNR fine-granular scalability for internet video,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, 2001.
- [85] G. J. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, November 1998.
- [86] T. Berger, *Rate Distortion Theory*. Englewood Cliffs, NJ: Prentice Hall, 1971.
- [87] M. Jiang and N. Ling, “On lagrange multiplier and quantizer adjustment for H.264 frame-layer video rate control,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 5, May 2006.

-
- [88] C. Tsai, C. Tang, C. Chen, and Y. Yu, "Adaptive rate-distortion optimization using perceptual hints," *Proceeding of International Congress on Mathematical Education*, pp. 667–670, 2004.
- [89] J. Chen, U.-V. Koc, and K. J. R. Liu, *Design of Digital Video Systems*. Marcel Dekker, 2001.
- [90] M. Karczewicz and A. Hallapuro, "Interpolation solution with low encoder memory requirements and low decoder complexity," VCEG-N31, 24-27 September 2001.
- [91] J. Reichel, H. Schwarz, and M. Wien, "Joint scalable video model (JSVM) 4.0 reference encoding algorithm description," ISO/IEC JTC1/SC29/WG11/N7556, Nice, France, October 2005.
- [92] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," *IEEE International Conference on Multimedia and Expo*, pp. 1929–1932, 2006.
- [93] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *Proc. 1997 Int. Conf. Information, Communications and Signal Processing (ICICS)*, vol. 1, pp. 292–296, September 1997.
- [94] —, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Processing*, vol. 9, no. 2, pp. 287–290, February 2000.
- [95] Y. Nie and K. K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," *IEEE Trans. Image Processing*, vol. 11, no. 12, pp. 1442–1448, December 2002.

- [96] H. Li, Z. G. Li, C. Y. Wen, and L.-P. Chau, "Fast mode decision for spatial scalable video coding," *IEEE International Symposium on Circuits and Systems*, pp. 3005–3008, May 2006.
- [97] H. Li, Z. G. Li, and C. Y. Wen, "Fast mode decision for coarse grain SNR scalable video coding," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 1793–1796, May 2006.
- [98] H. Li, Z. G. Li, C. Y. Wen, and S. L. Xie, "Fast mode decision for coarse granular scalability via switched candidate mode set," *IEEE International Conference on Multimedia and Expo*, pp. 1323–1326, 2007.
- [99] H. Li, Z. G. Li, and C. Y. Wen, "Fast mode decision for temporal scalable video coding," *Proceedings Picture Coding Symposium*, 2006.
- [100] I. Sodagar, H.-J. Lee, P. Hatrack, and Y.-Q. Zhang, "Scalable wavelet coding for synthetic/natural hybrid images," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3, pp. 244–254, 1999.
- [101] S.-J. Choi and J. Woods, "Motion compensated 3-D subband coding of video," *IEEE Transactions on Image Processing*, vol. 8, no. 2, February 1999.
- [102] G. Karlsson and M. Vetterli, "Three dimensional subband coding of video," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1988, pp. 1100–1103.
- [103] S.-T. Hsiang and J. W. Woods, "Embedded image coding using zeroblocks of subband/wavelet coefficients and context modelling," MPEG-4 Workshop and Exhibition at ISCAS 2000, Geneva, Switzerland, May 2000.

-
- [104] I. Shin and H. W. Park, "Adaptive up-sampling method using DCT for spatail scalability for scalable video coding," vol. 19, no. 2, February 2009.
- [105] B. Jeon and J. Lee, "Fast mode decision for H.264," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q6/J033, Hawaii, USA, December 2003.
- [106] H. Li, Z. G. Li, and C. Y. Wen, "Fast mode decision algorithm for inter-frame coding in fully scalable video coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 889–895, July 2006.
- [107] H. M. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP," *IEEE Transactions on Multimedia*, vol. 3, 2001.
- [108] Z. G. Li, X. K. Yang, K. P. Lim, X. Lin, S. Rahardaj, and F. Pan, "Customer oriented scalable video coding," ISO/IEC JTC1/SC29 WG11 MPEG2004/M11187, Spain, October 2004.
- [109] L. Yang, Y. Chen, J. F. Zhai, and F. Zhang, "Low complexity intra prediction for enhancement layer," ISO/IEC JTC1/WG11, JVT-Q084, Nice, France, October 2005.
- [110] H. Schwarz, D. Marpe, and T. Wiegand, "Hierarchical B pictures," Joint Video Team, Doc. JVT-P014, Poznan, Poland, July 2005.
- [111] H. Zhu, C. K. Wu, Y. L. Wang, and Y. Fang, "Fast mode decision for H.264/AVC based on macroblock correlation," *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, 2005.

-
- [112] M. C. Hwang, J. K. Cho, J. H. Kim, and S. J. Ko, "A fast intra prediction mode decision algorithm based on temporal correlation for H.264," *Proc. of ITC-CSCC 2005*, vol. 4, pp. 1573–1574, July 2005.
- [113] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITUT-T Q6/SG16, Doc. VCEG_M33, April 2001.
- [114] 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. on Circuits and Systems for Video Technology*, vol. 20, no. 5, pp. 732–748, May 2010.
- [115] S.-T. Kim, K. R. Konda, C.-S. Park, C.-S. Cho, and S.-J. Ko, "Fast mode decision algorithm for inter-layer coding in scalable video coding," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, pp. 1572–1580, August 2009.
- [116] C.-S. Park, B.-K. Dan, H. Choi, and S.-J. Ko, "A statistical approach for fast mode decision in scalable video coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 19, no. 12, pp. 1915–1920, December 2009.
- [117] S.-W. Jung, S.-J. Baek, C.-S. Park, and S.-J. Ko, "Fast mode decision using all-zero block detection for fidelity and spatial scalable video coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 20, no. 2, pp. 201–206, February 2011.
- [118] S. V. Leuven, G. V. Wallendaal, K. D. Wolf, J. D. Cock, P. Lambert, and R. V. de Walle, "An enhanced fast mode decision model for spatial

- enhancement layers in scalable video coding,” *Multimedia Tools and Applications*, vol. 58, no. 1, pp. 215–237, May 2012.
- [119] Z. T. Song, W. H. Li, K. Sam, C.-C. Kuo, and W. A. Halang, “Hierarchical B-picture mode decision in H.264/SVC,” *Journal of Visual Communication and Image Representation*, vol. 22, no. 7, pp. 627–633, October 2011.
- [120] P. Chen, K. Hanke, T. Ruster, and J. W. Woods, “Improvements to the MC-EZBC scalable video coder,” in *Proc. IEEE International Conference on Image Processing*, Barcelona, Spain, September 2003, pp. 81–84.
- [121] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, “H.264/AVC baseline profile decoder complexity analysis,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 704–716, July 2003.
- [122] K.-H. Lam, “Reduced complexity deblocking filter for H.264 video coding,” *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, pp. 1372–1374, November 2005.
- [123] K. Lengwehasatit and A. Ortega, “Rate-complexity-distortion optimization for quadtree-based DCT coding,” *International Conference on Image Processing*, pp. 821–824, 2000.
- [124] X. Gao, K. M. Lam, L. Zhuo, and L. Shen, “Complexity scalable control for H.264 motion estimation and mode decision under energy constraints,” *Signal Processing, Special Section on Processing and Analysis of High-Dimensional Masses of Image and Signal Data*, vol. 90, pp. 2468–2479, August 2010.

- [125] D. T. Hoang, P. M. Long, and J. Vitter, "Rate-distortion optimizations for motion estimation in low-bitrate video coding," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 488–500, August 1998.
- [126] F. Wu, S. P. Li, and Y. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, pp. 332–344, 2001.
- [127] H. C. Huang, W. H. Peng, Y. C. Lin, C. N. Wang, T. Chiang, and H. M. Hang, "Response of CFP on scalable video coding technology: Proposal S07-A robust scalable video coding," ISO/IEC JTC1/SC29 WG11 MPEG2004/M10569/S07, Munchen, Germany, March 2004.
- [128] T. Halbach, "Performance comparison: H.26L intra coding vs. JPEG2000," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT 4th Meeting, Klagenfurt, Austria, July 2002.
- [129] M. Mrak, D. Marpe, and T. Wiegand, "A context modeling algorithm and its application in video compression," *Proceedings of the 2003 International Conference on Image Processing (ICIP)*, pp. 845–848, September 2003.
- [130] K.-C. Yang, C. C. Guest, and P. K. Das, "Human visual attention map for compressed video," *Proceedings of the Eighth IEEE International Symposium on Multimedia*, 2006.
- [131] M. Bosch, F. Zhu, and E. J. Delp, "Video coding using motion classification," *IEEE International Conference on Image Processing*, 2008.
- [132] Meier, T. Ngan, and K. N., "Segmentation and tracking of moving objects for content-based video coding," *IEE Proceedings of Vision, Image and Singal Processing*, vol. 146, June 1999.

-
- [133] B. Menser and M. Brunig, “Face detection and tracking for video coding applications,” *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 49–53, 2000.
- [134] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 586–591, June 1991.