UNIVERSITY OF CALIFORNIA, SAN DIEGO

A Block-Based Scalable Motion Model for Highly Scalable Video Coding

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical and Computer Engineering

by

Meng-Ping Kao

Committee in charge:

Professor Truong Nguyen, Chair
Professor Serge Belongie
Professor Pamela Cosman
Professor William Hodgkiss
Professor Nuno Vasconcelos

2008

The dissertation of Meng-Ping Kao is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2008

DEDICATION

To my family

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Pursuing the Ph.D. degree has been a challenging and memorable journey in my life. Without so much timely help, support, and encouragement, I can hardly arrive at the destination.

First and foremost, I would like to offer my deep and sincere gratitude to my advisor, Prof. Truong Nguyen. Prof. Nguyen gave me the opportunity to join the Video Processing Laboratory, introduced me to the exciting research field of scalable video coding, inspired me with innovative ideas and careful verifications, and showed continuing support and encouragement to help me get through my most difficult times in research.

I would also like to thank my committee members, Prof. Serge Belongie, Prof. Pamela Cosman, Prof. William Hodgkiss, and Prof. Nuno Vasconcelos, for providing me with valuable advice and comments to accomplish this work.

It has been a great pleasure to work with many brilliant people in the Video Processing Laboratory in the ECE Dept., UCSD. I would like to especially thank Lorenzo Cappellari, Gokce Dane, Min Li, Koohyar Minoo, Cheolhong An, and Shay Har-noy for their academic help and companionship, which makes the laboratory my second home.

I owe a great deal of thanks to my parents, Koong-Lian Kao and Lo Hui-Chu Kao, and my brother, Meng-Kang Kao, for encouraging me to set higher career goals and for showing endless love and support. Finally, my most sincere gratitude goes to my girlfriend, Pei Jen Lee, for her love, trust, and great sacrifice.

Portions of Chapter 2 appear in "Motion Vector Field Manipulation for Complexity Reduction in Scalable Video Coding", Meng-Ping Kao and Truong Nguyen, in Proceedings of the *IEEE Asilomar Conference on Signals, Systems and Computers*, Oct. 2006. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

Portions of Chapter 3 appear in "Motion Vector Field Manipulation for Complexity Reduction in Scalable Video Coding", Meng-Ping Kao and Truong Nguyen, in Proceedings of the *IEEE Asilomar Conference on Signals, Systems*

and Computers, Oct. 2006. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

Portions of Chapter 4 appear in "A Fully Scalable Motion Model for Scalable Video Coding", Meng-Ping Kao and Truong Nguyen, in the June 2008 issue of the *IEEE Transactions on Image Processing*; and also in "A Fully Scalable Motion Model for Scalable Video Coding", Meng-Ping Kao and Truong Nguyen, in Proceedings of the *IEEE International Conference on Image Processing*, Sep. 2007. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

Portions of Chapter 5 appear in "A Fully Scalable Motion Model for Scalable Video Coding", Meng-Ping Kao and Truong Nguyen, in the June 2008 issue of the *IEEE Transactions on Image Processing*; "A Fully Scalable Motion Model for Scalable Video Coding", Meng-Ping Kao and Truong Nguyen, in Proceedings of the *IEEE International Conference on Image Processing*, Sep. 2007; and also in "Coding and Optimization of a Fully Scalable Motion Model", Meng-Ping Kao and Truong Nguyen, in Proceedings of the *SPIE Applications of Digital Image Processing*, Oct. 2007. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

I am very grateful for the financial support of Conexant Systems, Inc. and the University of California Discovery Grant. I hope I have represented their contributions well.

VITA

2000	B. S., Electrical Engineering, National Taiwan University, Taipei, Taiwan
2004	M. S., Communication Engineering, National Taiwan University, Taipei, Taiwan
2008	Ph. D, Electrical and Computer Engineering, University of California, San Diego

PUBLICATIONS

Meng-Ping Kao and Truong Nguyen, "A Fully Scalable Motion Model for Scalable Video Coding", *IEEE Transactions on Image Processing*, Vol. 17, Pages 908-923, Jun. 2008.

Meng-Ping Kao and Truong Nguyen, "Coding and Optimization of a Fully Scalable Motion Model", in Proceedings of the *SPIE Applications of Digital Image Processing*, Vol. 6696, Oct. 2007.

Meng-Ping Kao and Truong Nguyen, "A Fully Scalable Motion Model for Scalable Video Coding", in Proceedings of the *IEEE International Conference on Image Processing*, Vol. 2, Pages 313-316, Sept. 2007.

Meng-Ping Kao and Truong Nguyen, "Motion Vector Field Manipulation for Complexity Reduction in Scalable Video Coding", in Proceedings of the *IEEE Asilomar Conference on Signals, Systems and Computers*, Pages 1095-1098, Oct. 2006.

Soo-Chang Pei and Meng-Ping Kao, "A Two-Channel Nonuniform Perfect Reconstruction Filter Bank with Irrational Down-Sampling Factors", *IEEE Signal Processing Letters*, Vol. 12, Pages 116-119, 2005.

Soo-Chang Pei and Meng-Ping Kao, "Direct N-Point DCT Computation from Three Adjacent N/3-Point DCT Coefficients", *IEEE Signal Processing Letters*, Vol. 12, Pages 89-92, 2005.

Soo-Chang Pei, Meng-Ping Kao and J. J. Ding, "A Perfect Reconstruction Filter Bank with Irrational Down-Sampling Factors", in Proceedings of the *IEEE International Symposium on Circuits and Systems*, Vol. 3, Pages 2036-2039, May 2005.

Soo-Chang Pei and Meng-Ping Kao, "Two Dimensional Nonuniform Perfect Reconstruction Filter Bank with Irrational Down-Sampling Matrices", in Proceedings of the *IEEE International Symposium on Circuits and Systems*, Vol. 2, Pages 1086-1089, May 2005.

Soo-Chang Pei and Meng-Ping Kao, "Direct N-Point DCT Computation from Three Adjacent N/3-Point DCT Coefficients", in Proceedings of the *IEEE International Conference on Image Processing*, Vol. 2, Pages 1113-1116, Oct. 2004.

Meng-Ping Kao and Truong Nguyen, "Scalable Motion Vector Field Manipulation for Complexity Reduction in Scalable Video Coding", *IEEE Transactions on Image Processing*, In preparation.

Meng-Ping Kao and Truong Nguyen, "Optimal Bitstream Extractor for Motion Scalability in Scalable Video Coding", *IEEE Transactions on Image Processing*, In preparation.

Meng-Ping Kao and Truong Nguyen, "Optimal Distortion Metric for Motion Estimation in Critically Sampled Wavelet-Based Scalable Video Coding", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, In preparation.

ABSTRACT OF THE DISSERTATION

A Block-Based Scalable Motion Model for Highly Scalable Video Coding

by

Meng-Ping Kao
Doctor of Philosophy in Electrical and Computer Engineering
University of California San Diego, 2008
Professor Truong Nguyen, Chair

Scalable video coding has gained considerable attention during the past decade, due to its attractive features that efficiently support flexible transmission over heterogeneous networks and adaptive display on a wide range of devices. As coding efficiency is predominantly the governing principle of most video coding algorithms, scalable video coding thrives in incessantly improving efficiency through incorporating newly emerged technologies while preserving the scalable features. Motion scalability, being the main topic of this dissertation, is one of these contributive technologies.

Motion scalability is based on the simple concept that different decoding scenarios require different motion prediction qualities in the optimized rate distortion sense. For example, lower decoding resolutions or bit rates usually demand lower motion prediction qualities in order to maintain a better balance between motion and texture coding. This concept, although simple, is not easily realizable in a practical scalable video codec. The error drifting effect introduced from quantized motion is the first problem to face, followed by the interactive issue with other scalabilities, the embedded coding of scalable motion, and the rate distortion optimized estimation algorithm for motion parameters.

In this dissertation, we deal with these challenges and propose a block-based scalable motion model, which provides both motion structure and accuracy

scalabilities in order to adapt to various decoding scenarios. Through the proposed model, rate distortion performance can be improved in the middle to low bit rate range. This accomplishment is jointly achieved by applying the proposed rate distortion optimized motion estimation algorithm at the encoder and the optimal motion quality selection algorithm at the bitstream extractor.

Extensive simulations will be demonstrated based on a wavelet-based scalable video codec. These results verify the superiority of the proposed scalable motion solution over non-scalable ones.

1 Introduction

Video coding standards have steadily evolved during the past twenty years due to the need for efficient transmission and storage of digital video media. While the first standard, H.120 [1], set a crucial milestone for digital video coding in 1984, H.261 [6] introduced the hybrid video coding infrastructure, which became the basis of all modern video coding standards. A simplified chronology of video coding standards is shown in Fig. 1.1.

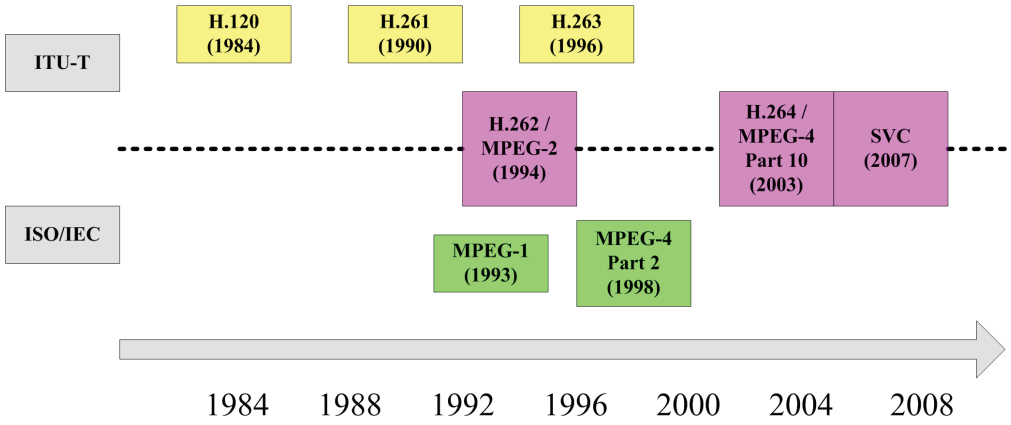


Figure 1.1: Chronology of international video coding standards.

Since H.261 was finalized in 1990, significant research has emerged to improve the coding efficiency of hybrid video compression. By 2003 when the state-of-the-art single layer video coding standard H.264/AVC [81] was completed, the performance of hybrid video coding reached an unprecedented summit. This achievement can be attributed to a comprehensive elaboration of the components, among which closed-loop motion prediction and rate-distortion optimized estimation play an important role.

On the other hand, due to the increasing need for video delivery through heterogeneous channels and display on various end-user devices, multi-layer video coding has gained lots of attention since the Moving Picture Experts Group (MPEG) issued the "Call for Proposals on Scalable Video Coding Technology" in 2003 [35]. The term "multi-layer" or "scalability" refers interchangeably to the removal of parts of a video bitstream in order to adapt it to various needs or preferences of end users, as well as to varying terminal capabilities or network conditions. As far as practical applications are concerned, Scalable Video Coding (SVC) finds its superiority mostly in video broadcasting, error protection, and surveillance systems. Fig. 1.2 shows basic building blocks of an SVC system.

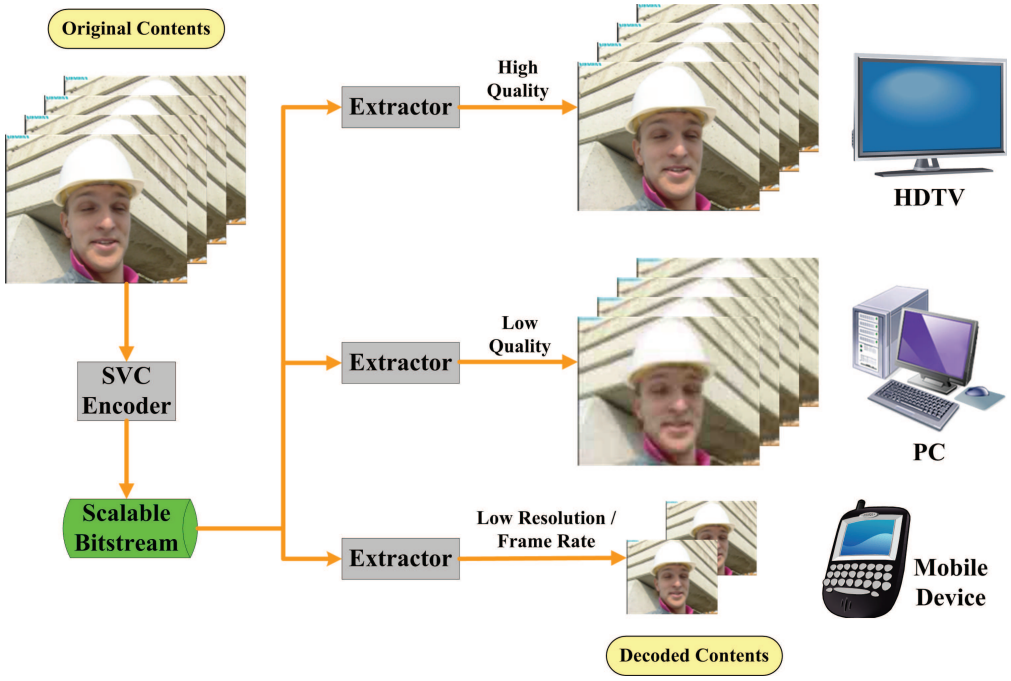


Figure 1.2: Scalable video coding.

In Nov. 2007, the Joint Video Team (JVT) standardized an SVC extension of the H.264/AVC standard [60]. The SVC extension not only positions itself as another video coding standard but also opens a new research field where the asynchronous motion prediction loop is extensively accepted, and the control of error drift becomes the key for achieving global optimality among various decod-

ing scenarios. Recall that in H.264/AVC, high coding efficiency comes from a synchronized motion prediction loop and rate distortion optimization for a single operating point. This is not necessarily true for SVC, since multiple operating points have to be considered simultaneously and a tradeoff on the asynchronous motion prediction loop might sometimes be helpful. In short, the synchronized motion prediction loop is no longer a requirement in SVC as long as better overall rate distortion performances can be achieved with an asynchronous solution.

Inspired by the above idea, researchers have searched for better solutions using asynchronous motion prediction. Scalable motion, for example, has been developed recently and proved to be beneficial to SVC in the range of middle to low decoding bit rates. The term "scalable motion" refers to an embedded bitstream containing motion information that can be truncated to provide different quality of motion predictions.

Recall that in conventional hybrid video coding, the encoded bitstream consists of two main categories of information, i.e. motion and texture. While texture information can be scalable, motion information is in general non-scalable. By introducing scalable motion, better distribution of budget bit rates between motion and texture can be determined, as opposed to the fixed motion bit rate scenario. The increased decoding options provide the potential to attain a better rate distortion performance.

In this dissertation, we will devote most of our efforts to elaborate the rationale of scalable motion and to provide a practical solution that can be incorporated into a generic SVC codec. With the proposed solution, a.k.a. the scalable motion model, the coding efficiency of an SVC codec can be greatly improved in the middle to low bit rate range, with little loss in high rate scenarios.

In order to have a better understanding of scalable motion and the platform it will be operated on, we start the dissertation with an introduction to scalable video coding in Chapter 2. The SVC extension of H.264/AVC is first discussed with illustrations on those techniques that facilitate the joint temporal, spatial, and quality scalabilities. The wavelet-based SVC, although not being chosen as the basic framework in the standard, provides an alternative to conventional closed-

loop solutions. We will study the state-of-the-art wavelet-based SVC, VidWav [9], that utilizes open-loop motion prediction, yet still has a comparable performance to the SVC standard. In the end, we propose our own wavelet-based SVC [41], which is similar to VidWav, but with a closed-loop motion prediction structure. Our proposed SVC will be the standard platform for all experimental setups throughout this dissertation.

In Chapter 3, the motion redundancy in video sequences with different spatial resolutions is first investigated. We show an example of motion manipulation within the proposed SVC framework that reduces the encoding complexity and at the same time improves the coding efficiency. This example illustrates how SVC can benefit from scalable motion.

To build a better foundation for the following research on scalable motion, the historical and theoretical backgrounds are introduced in Chapter 4. Scalable motion is a newly emerged research topic, as part of the advanced algorithms in the development of SVC. The earliest literature can be traced back to 2001 when Bottreau [17] proposed an SVC scheme using scalable motion. Thereafter, several methods have been proposed targeting the realization of scalable motion, which can be split into two main approaches: 1) motion vector precision scalability, 2) motion field structure scalability. In this chapter, the functionalities of scalable motion that cope with the joint scalabilities in SVC will be enumerated and analyzed. Moreover, the two main approaches for scalable motion will be evaluated according to the required functionalities. This evaluation process will serve as the core design criterion for our proposed scalable motion model detailed in Chapter 5. Finally, we conclude this chapter with a theoretical justification of the performance improvement that scalable motion offers. This justification is important for the design of rate distortion optimized motion estimation, as well as the optimal bitstream extractor.

Chapter 5 is the main body of this dissertation where we focus exclusively on the proposed scalable motion model. Our model is based on the aforementioned two approaches with a novel integration that preserves the advantages of both methods. Compatibility to the joint scalabilities of a generic SVC codec is also

carefully considered. As opposed to most of the post-estimation embedded coding methods in previous works, our model integrates flawlessly with the estimation process to provide scalability in an optimal manner. The optimality is achieved via the proposed rate distortion optimized estimation algorithm. In order to further compress the estimated parameters in our model, a modified Set Partitioning in Hierarchical Trees (SPIHT) [59] algorithm is used to encode the motion structure, while a model-based method is applied to compress the motion precision. All these advanced coding techniques help to reduce, as much as possible, the overhead of providing scalability. Meanwhile, an optimal bitstream extractor is proposed to help the decoder determine the best distribution of bits between motion and texture. Finally, extensive experimental results will be presented to verify the superiority of the proposed scalable motion model over non-scalable ones.

Finally, the conclusions will be summarized in Chapter 6.

2 Scalable Video Coding

In general, video coding can be evaluated by rate distortion performance [68, 55]. A codec that generates the encoded bitstream with less distortion under the same bit rate budget is considered better. Guided by this RD rule, video coding research devoted great efforts over the past thirty years to improving the RD curve. H.264/AVC [81] is so far the state-of-the-art international video coding standard that gives excellent compression performance under reasonable complexity. It belongs to the category of traditional (or non-scalable) video coding.

Scalable Video Coding (SVC), on the other hand, considers not only the RD characteristic but also the ability to produce a highly scalable, easily adaptable, and fully accessible bitstream. By the term "scalable" we mean the capability of easily removing parts of the video bitstream in order to adapt it to the various needs or preferences of end users, as well as to varying terminal capabilities or network conditions. In SVC, multiple RD curves [60], corresponding to various decoding scenarios, can be derived from a single encoded bitstream. These RD curves should be jointly evaluated in the comparison of different SVC codecs. In general, a balanced performance is preferable to a skewed one when no additional information on the decoding situation is provided.

As far as applications are concerned, SVC is tailored for the transmission of video contents through heterogeneous networks to end users with various display devices. The source content has to be encoded only once with the highest required resolution and bit rate. Requests for lower resolution and/or quality contents can be easily fulfilled by discarding corresponding parts of the bitstream, which is done by the so-called bitstream extractor. These requests may result from poor channel

conditions or restricted display sizes. In this case, SVC helps the video server to save tremendous storage space and computational power by avoiding lower quality duplicates of the same content. It can also help to replace the costly transcoder with a cheaper bitstream extractor if the same bitstream is broadcast.

Other desirable applications include unequal error protection for transmission over error prone channels and surveillance systems. The natural heritage of a well segmented and packed bitstream is highly preferable for applying unequal error protection mechanisms. Those packets which comprise the base layer decoding essentials should be protected with stronger channel coding algorithms. In this way, graceful degradation can be achieved under volatile channel conditions. As for surveillance systems, clients from different places with various displays might access the same real time recording. This is covered by the aforementioned video transmission scenario. In addition, huge numbers of recordings need to be stored and archived for possible future retrievals. SVC offers an easy solution to maintain the storage by discarding enhancement layer data whose importance decreases as time passes. The base layer contents can be preserved for long term archiving purposes.

Considering these functional requirements of the above target applications, the three fundamental scalabilities that a modern SVC has to offer include temporal, spatial, and quality scalabilities. Temporal scalability refers to the ability to flexibly adjust the frame rate, either diadic or non-diadic. Similarly, spatial scalability deals with the change of decoding picture sizes. Quality scalability, a.k.a. SNR scalability, is a tool to trade off distortion with bit rate. In the following sections of this chapter, we will introduce some SVC frameworks; each of them will be discussed through the basic building blocks that realize the aforementioned three scalabilities. In addition to these three popular ones, more rarely required scalabilities include region-of-interest (ROI) and object-base scalabilities, in which part of the original picture is enhanced through enhancement layers.

Before we get to the details of the various SVC designs, it is beneficial to know the history of the SVC development. Over the past twenty years, SVC has been an active research topic. Scalability features can be found in prior standards

including H.262/MPEG-2 [2], H.263 [7], and MPEG-4 [4]. However, significant loss in coding efficiency and increased decoding complexity have prevented them from widespread adoption. In October 2003, MPEG issued a call for proposals for efficient SVC technology with the intention to develop a new SVC standard. At that time, a wavelet-based temporal decomposition approach was extensively studied and widely believed to be a possible solution to SVC, due to its drift free property inherited from the open loop structure. In fact, twelve out of fourteen submitted proposals were based on a wavelet approach. Nevertheless, SVC based on an extension of H.264/AVC was chosen as the starting draft one year later because of its superior coding efficiency over wavelet-based ones. No major changes on the framework have occurred since then, and in November 2007, the Joint Video Team (JVT) finalized the first version of the SVC standard as an extension of H.264/AVC in Annex G. From now on, the term "SVC" will be used interchangeably for both the acronym of scalable video coding and for the particular design in the extension of the H.264/AVC standard, depending on the actual context.

Although the success of SVC requires further embrace from the industry, it does surpass those scalable profiles in prior standards in the following aspects: higher encoding efficiency, simple bitstream adaptation, lower decoding complexity, and full support of joint temporal, spatial, and quality scalabilities.

As a consequence, it is worthwhile to illustrate the state-of-the-art SVC frameworks, both H.264/AVC-based and wavelet-based, in the following sections along with their constituents to achieve different scalabilities. In addition, we conclude this chapter by proposing a low-complexity wavelet-based SVC framework. Our following work on scalable motion, which is the main part of this dissertation, will be performed on this platform.

2.1 SVC Extension of H.264/AVC

The SVC extension of H.264/AVC was originally developed by the Heinrich Hertz Institute (HHI) in Berlin, Germany, with the design principle of fully utilizing the mature core coding tools inherited from H.264/AVC [5]. New tools

should only be added if necessary for efficiently supporting the imposed scalabilities. An example encoder diagram supporting three levels of spatial scalability is shown in Fig. 2.1. In this example, spatial scalability is demonstrated by a Laplacian pyramid [18] which generates lower resolution sequences from the original video. Motion prediction is then applied on these sequences independently with a referencing scheme called hierarchical B-picture decomposition which is designed for supporting temporal scalability. Both motion and texture information are predictively coded from the base layer, using the so called inter-layer prediction mechanism. Finally, transform and entropy coding are applied on the texture information in a scalable manner.

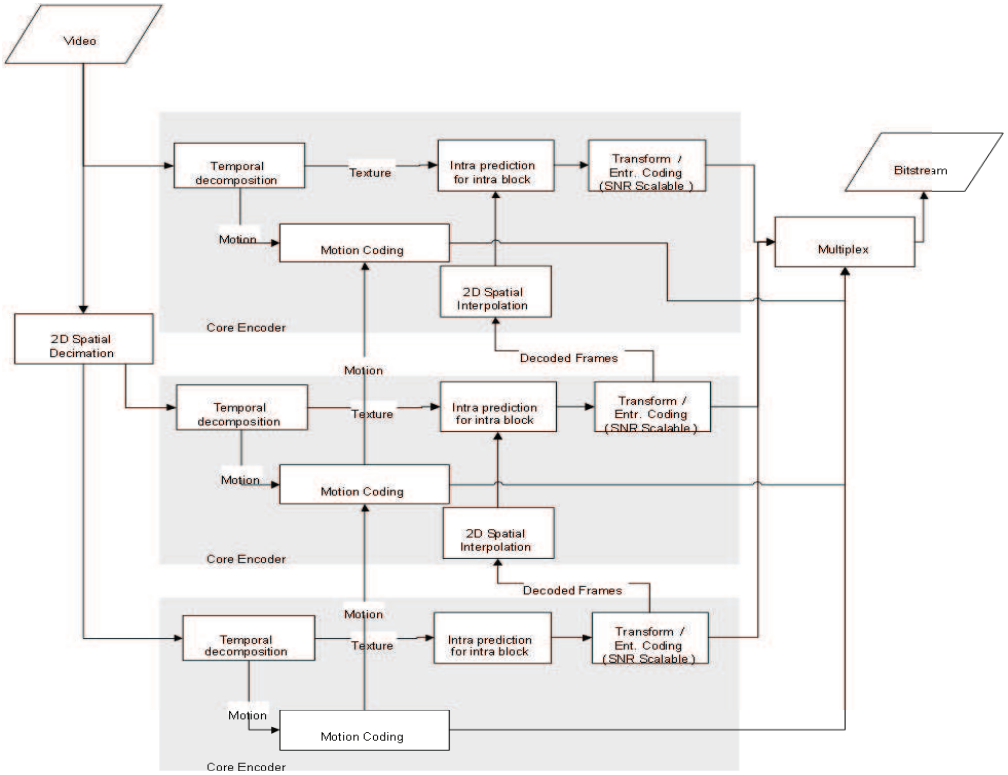


Figure 2.1: SVC encoder with three levels of spatial scalability.

Temporal scalability of SVC is realized by dividing the whole sequence into different temporal layers, starting from $T = 0$ for the base layer and increasing by 1 from one layer to the next. The decoded sequence with temporal layer k is obtained by discarding those pictures with $T > k$. In order to cooperate with motion

prediction, the referencing scheme must be designed in a way that pictures with $T = k$ can only be predicted from those reference pictures with $T \leq k$. In this way, reference pictures always exist no matter which temporal layer is requested. Note that the existence of reference pictures does not guarantee coherence between the encoder and the decoder, which is also known as the drift problem. An example of dyadic decomposition using hierarchical B-pictures is shown in Fig. 2.2. According to [60], the coding efficiency of dyadic hierarchical prediction structures in a high delay setting, i.e. group of pictures (GOP) equals 32, can outperform H.264/AVC (IBBP) by 1 dB in the medium bit rate range.

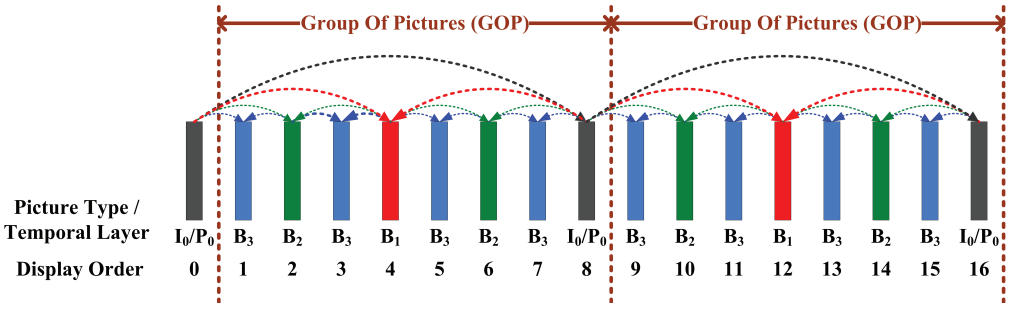


Figure 2.2: Hierarchical B-pictures with dyadic decomposition.

For supporting spatial scalability [63], SVC adopts the multi-layer coding scheme with each layer corresponding to a supported spatial resolution. As mentioned above, a Laplacian pyramid is used to generate lower resolution sequences. In order to improve the coding efficiency, inter-layer prediction mechanisms are incorporated as shown in Fig. 2.3 with solid arrowheaded lines, in contrast to simulcasting multiple layers. Note that spatial scalability is widely believed to be the most difficult scalability to design in terms of the balance between efficiency and complexity.

Inter-layer prediction can be divided into three categories, i.e. inter-layer motion prediction, inter-layer residual prediction, and inter-layer intra prediction. Inter-layer motion prediction is used to predict the motion information from lower spatial layers, including partition information, reference indices, and motion vectors. Inter-layer residual prediction utilizes the up-sampled version of lower spatial layer residuals as a prediction for current layer residuals. Inter-layer intra

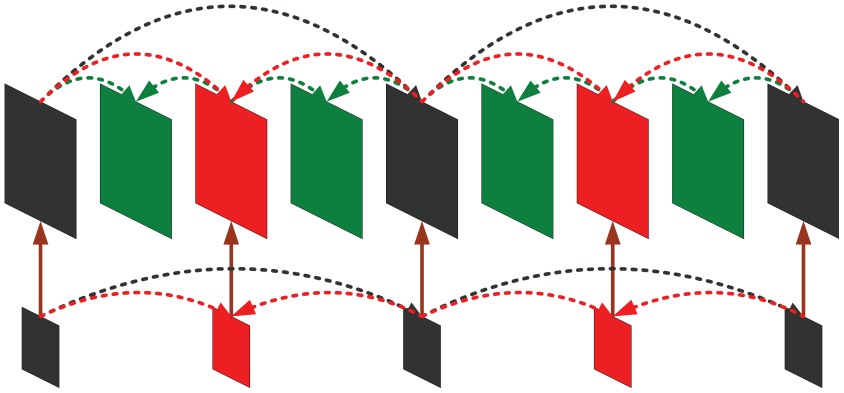


Figure 2.3: Spatial scalable coding using a multi-layer structure with inter-layer prediction (indicated by solid arrowheaded lines).

prediction is used when the co-located block in a lower spatial layer is coded using intra mode. Note that all three mechanisms exclude using predictions from reconstructed pixels that are inter coded. The idea is to avoid multiple motion compensation loops required for decoding a single spatial layer, a.k.a. single-loop decoding criterion. As reported in [60], the coding efficiency using these inter-layer prediction techniques, in comparison to H.264/AVC, is about 10% worse in the required bit rate under the same quality. This deficit is the price of scalability, and further note that this 10% can only be achieved with advanced encoder control algorithms.

Quality scalability in SVC can be supported via two modes, i.e. coarse-grain quality scalability (CGS) and medium-grain quality scalability (MGS). CGS can be viewed as a special case of spatial scalability where the picture sizes for different layers remain the same. Inter-layer prediction mechanisms, as mentioned above, can be employed. The limitations of CGS are relatively few supported bit rates and decreasing efficiency as the rate difference between successive CGS layers decreases. MGS, on the other hand, supports finer quality scalability than CGS in a way that quality layers can be flexibly adjusted on a picture-based level. Enhanced transform coefficients can be distributed among several slices to facilitate packet-based quality scalability.

One common problem that arises when quality scalability meets motion

compensated prediction is the tradeoff between efficiency and drift. Drift occurs when motion-compensated prediction loops at the encoder and the decoder are not synchronized, which might be caused by discarding enhancement packets in the bitstream. This can be prevented by using base quality layers as reference pictures. However, these lower quality predictions will in turn reduce the coding efficiency. SVC deals with this dilemma by using a so-called key picture concept, which is shown in Fig. 2.4. Key pictures are pictures labeled with the lowest temporal layers in the hierarchical B-Picture decomposition. These pictures are coded with the base layer prediction and thus guaranteed with no drift. Other pictures are coded with the highest enhancement layer prediction to achieve the best coding efficiency. In this way, a balanced performance can be expected under variant decoding situations. Again, as reported in [60], with the optimized encoder control it is possible to limit the bit rate increase, compared to single-layer coding at the same fidelity, to about 10% over the entire supported bit rate range.

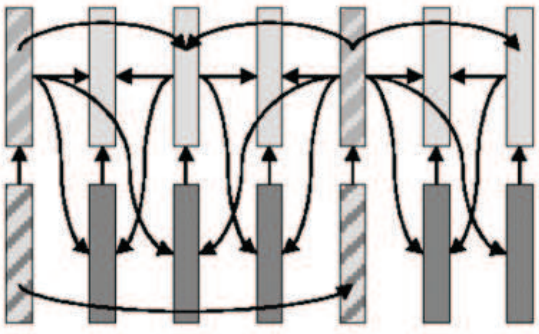


Figure 2.4: Referencing scheme using the key picture concept (key pictures are marked by the hatched boxes).

In summary, the H.264/AVC extension for SVC provides various tools for reducing the loss in coding efficiency relative to single layer coding. These new features provide SVC with a competitive rate-distortion performance, as compared to the scalable profiles in prior standards, while supporting joint temporal, spatial, and quality scalabilities. For more details, the readers can refer to [60].

2.2 Wavelet-Based SVC

Wavelet-based SVC (WSVC) [53] generally refers to the scalable coding scheme using motion-compensated 3D spatio-temporal decomposition by wavelets. Wavelets and their tailored bit plane coding techniques have been proven to provide excellent compression performance with inherited spatial and quality scalabilities in image coding. For example, the image compression standard, JPEG 2000 [3], has successfully demonstrated the effectiveness of combining the discrete wavelet transform and the bit plane coding algorithm, Embedded Block Coding with Optimal Truncation (EBCOT) [71]. Over the years, researchers have tried to duplicate the success of wavelets on image coding to video coding. Despite the discrepancy in statistical characteristics between image and video, the performance of WSVC has steadily improved, and more attention has focused on this prospective solution to SVC. In Dec. 2001, MPEG issued the "Ad Hoc Group on Exploration of Interframe Wavelet Technology in Video Coding", with the intention to emphasize the importance of WSVC in possible future international standards.

Although not selected in the SVC standard finalized in Nov. 2007, WSVC, which demonstrates outstanding coding efficiency with innate scalabilities, is still an alternative candidate for future standards. Therefore, it is still worthwhile to illustrate those different tools that WSVC utilizes toward supporting full scalabilities. Specifically, in the following paragraphs we will focus on the state-of-the-art WSVC based on STP-tool [8], a.k.a. VidWav [36], which gives the best performance among other wavelet-based variants so far. The system framework of VidWav is shown in Fig. 2.5.

The first problem encountered while applying a 3D wavelet transform to video sequences is the inefficiency of temporal filtering. Past experiences have shown that huge temporal redundancies exist in a generic video sequence and without motion compensated prediction, a video encoder can hardly achieve comparable performance to modern standards. As a consequence, directly applying the temporal filtering operation without considering the motion often fails to meet the efficiency requirement. The first work that combines motion compensated predic-

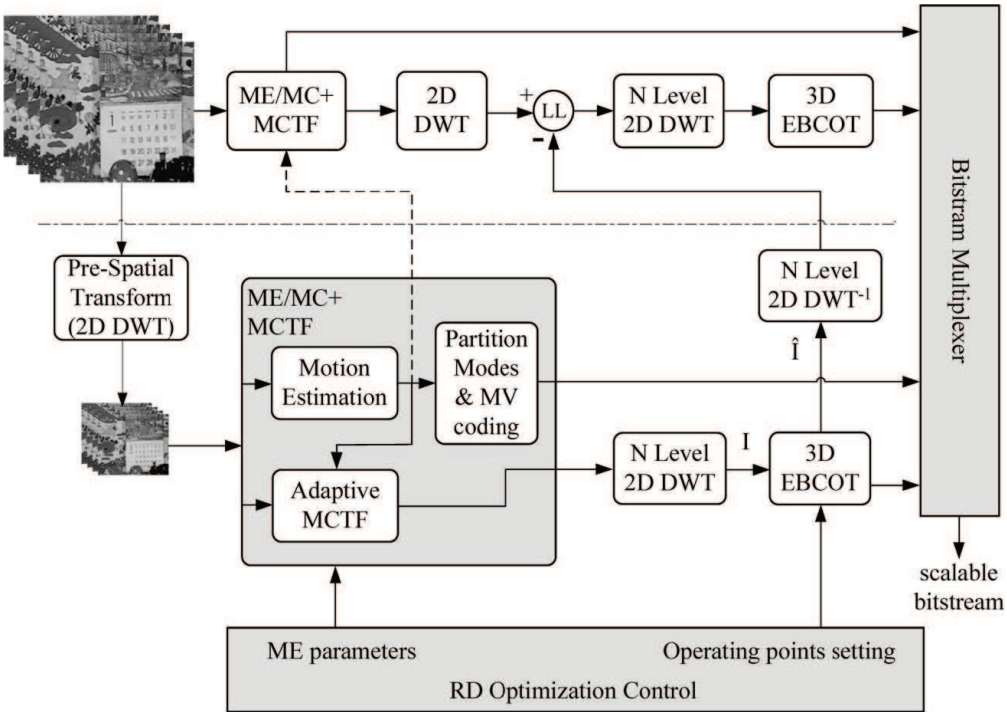


Figure 2.5: VidWav framework showing two levels of spatial scalability.

tion with temporal filtering was proposed by Ohm [52] in 1994, which is also known as the Motion Compensated Temporal Filtering (MCTF) framework. Since MCTF was proposed, significant research has focused on improving the performance and solving practical implementation issues. Up to now, the lifting structure [70, 24] implementation of MCTF, as shown in Fig. 2.6, using 5/3 analysis filters is the most widely adopted one [85, 22, 61, 57].

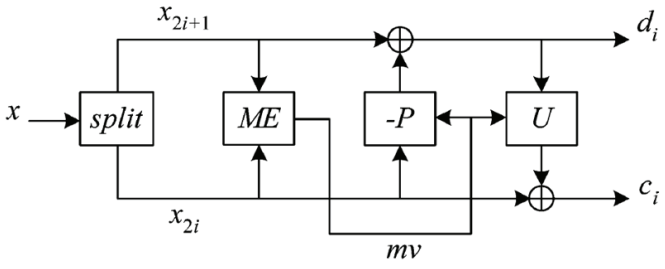


Figure 2.6: Prediction and update lifting steps using motion information.

The merit of the lifting structure is the property of guaranteed perfect

reconstruction even when motion prediction fails, e.g. those areas with motion occlusion and uncovering. The perfect reconstruction property also facilitates sub-pixel motion, which requires interpolation. 5/3 filters make use of bidirectional prediction which is popular in modern video codecs. Another variant of MCTF is called Unconstrained MCTF (UMCTF) [77], where the update step is omitted in order to eliminate the artifacts from failed motion prediction, or simply to reduce the complexity. In UMCTF, temporal low pass frames are the original frames without further processing. Finally, we note the difference between MCTF and hierarchical B-picture decomposition in SVC. MCTF takes the original pictures as the motion reference while hierarchical B-picture uses decoded pictures. This is the reason why MCTF is referred to as an open-loop structure while the other one is a closed-loop structure.

As for spatial scalability, WSVC offers an alternative to the redundant Laplacian pyramid method used in SVC, i.e. the critically sampled discrete wavelet transform (DWT). This concept is directly borrowed from image compression where DWT works efficiently to provide both spatial and quality scalabilities. However, DWT is not completely compatible with MCTF. In the first case where DWT is performed first and followed by MCTF, a.k.a. 2D+t, the shift-variant nature of DWT makes motion compensated prediction harder for high pass bands. On the other hand, with the reverse order, i.e. t+2D, the smaller sized sequence suffers from incoherent motion that is estimated based on the original sized sequence. Therefore, as we observe in the design of VidWav, another structure, 2D+t+2D, is adopted to overcome the problem. In the 2D+t+2D structure, a Laplacian pyramid is applied to generate redundant representations of various sized sequences, which is similar to the design in SVC. An Inter-Scale Prediction (ISP) technique is applied in the DWT domain to minimize the redundancy, which is named STP-tool. An example depicting three levels of spatial scalability in the STP-tool framework is shown in Fig. 2.7. Predictions of the DWT coefficients can be obtained from a lower resolution sequence in either an open-loop or closed-loop form.

Quality scalability in WSVC can be achieved by a generalization of 2D

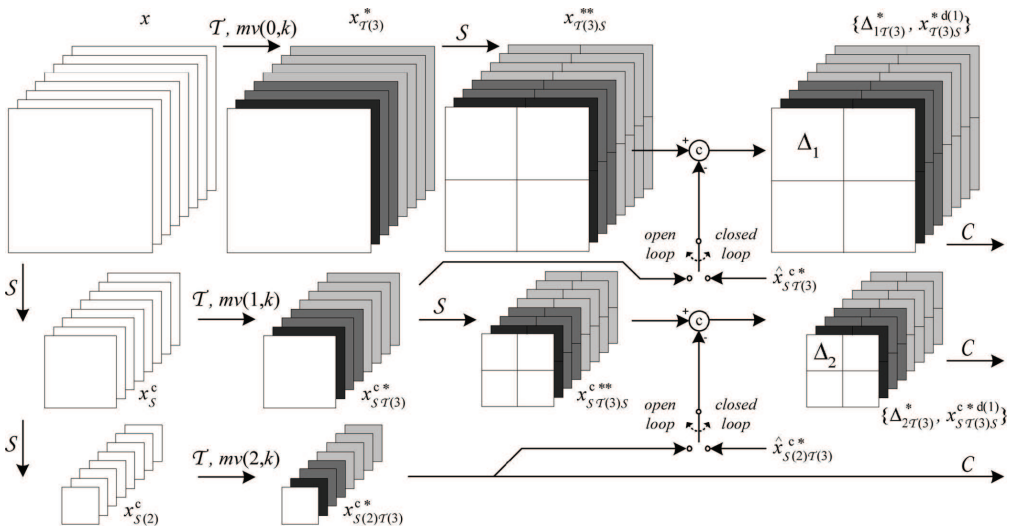


Figure 2.7: STP-tool framework in a signal representation perspective.

wavelet bit plane coding algorithms, including Embedded Zerotree Wavelet (EZW) [64], Set Partitioning In Hierarchical Trees (SPIHT) [59], Embedded Zero-Block Coding (EZBC) [32], and embedded block coding with optimal truncation (EBCOT) [71]. In VidWav, a 3-D spatio-temporal extension of EBCOT [86], named 3-D EBCOT, is adopted as the entropy coding algorithm. Each spatio-temporal sub-band is divided into 3-D blocks which are coded independently. For each block, fractional bit plane coding and spatio-temporal context-based arithmetic coding are used.

As a final note on WSVC, VidWav demonstrates a small efficiency loss as compared to SVC, according to [9]. The possible reasons include, but are not limited to, the inefficient motion model for DWT. Currently, a block-based motion model is addressed in VidWav, which is not fully suitable for the following frame-based DWT. The lack of efficient intra mode coding for the areas where motion prediction fails also results in some performance loss in VidWav. Unlike SVC, in which plenty of mature coding tools are inherited from conventional hybrid coding development, WSVC needs to build its own. Therefore, improvements can be expected in the future of wavelet video compression.

2.3 Proposed Wavelet-Based SVC

As mentioned earlier, since MPEG issued the call for proposals on scalable video coding technology in 2003, several interesting proposals have been submitted and carefully evaluated. The core building blocks targeting different scalabilities in these proposals are quite varied. Some of them were adopted as part of the draft standard, or evolved to be included in later amendments. The others were simply excluded. In this section, a detailed description of the proposed wavelet-based SVC will be illustrated. Some building blocks in our WSVC are borrowed from these proposals. By combining and modifying these tools properly, we are able to construct a low-complexity and fully scalable WSVC.

As before, the best way to understand an SVC system is to start with the individual components that support spatial, temporal, and quality scalabilities.

2.3.1 Temporal Scalability – Successive Temporal Approximation and Referencing

In the proposed WSVC, temporal scalability is supported in a similar manner to the hierarchical B-picture structure. Each picture is assigned a temporal level label, reflecting the actual frame rates at different levels. Scalability can be achieved by discarding irrelevant pictures according to their labels. Theoretically, this approach will introduce temporal aliasing due to the direct down-sampling operation without proper filtering. However, the advantages of being simple, effective, and ghost effect free, especially where motion prediction fails, have made it a success in the SVC standard. As a matter of fact, UMCTF is also based on the same concept. Moreover, as will be clear later, the end-to-end delay can be made very small in this case, enabling almost real time encoding/decoding of video sequences.

The key to this temporal scalability implementation, which discards irrelevant pictures, is a good referencing scheme for motion prediction. The scheme we utilize in our WSVC is called Successive Temporal Approximation and Referencing (STAR) proposed by Han [29]. The general framework of STAR allows

several scalability levels with different frame rate reduction ratios between them. The details are listed as follows.

1. The sequence with the highest frame rate has temporal level 1 and frame rate f_1 .

2. All the sequences with temporal level $i = 1, 2, \ldots, T$ are coded jointly in the bitstream.

3. The sequence at temporal level i can be obtained by keeping an average n out of m $(m > n \geq 1)$ of the frames of the sequence at temporal level $i-1$, and hence has $f_i = \frac{n}{m} f_{i-1} = K_T^i f_{i-1}$ as the frame rate, for $i = 2, 3, \ldots, T$.

4. All the frames $C^{(j)}$, where j is the (integer) temporal index of the frame, that belong to the sequence at temporal level i but do not belong to the sequence at temporal level $i+1$ have temporal level $T_j = i$, for $i = 1, 2, \ldots, T-1$.

5. All the frames $C^{(j)}$ belonging to the sequence at temporal level T have temporal level $T_j = T$.

To guarantee the existence of reference pictures at all temporal levels, the set of frames that can be referenced when decoding the inter frame $C^{(j)}$ should be a subset of

$$R^{(j)} = \{C^{(k)} | (T_k > T_j) \cup (T_k = T_j \cap k < j)\}. \tag{2.1}$$

While the delay concern is completely eliminated within the same temporal level by the constraint $k < j$, additional delays might be imposed from higher temporal levels. To limit the delay, we can introduce a delay parameter D that limits the set of possible references to

$$R_D^{(j)} = \{C^{(k)} | (T_k > T_j \cap k - j \leq D) \cup (T_k = T_j \cap k < j)\}. \tag{2.2}$$

An example of the reference scheme according to the STAR algorithm is shown in Fig. 2.8. There are four temporal levels in this example and in each of them the frame rate is halved, i.e. dyadic decomposition. All the arrows pointing to one frame start in a frame which will be referenced during the decoding of the same

frame. Note that many possibilities of referencing are possible with the STAR algorithm. For a complete list of them, please refer to [29].

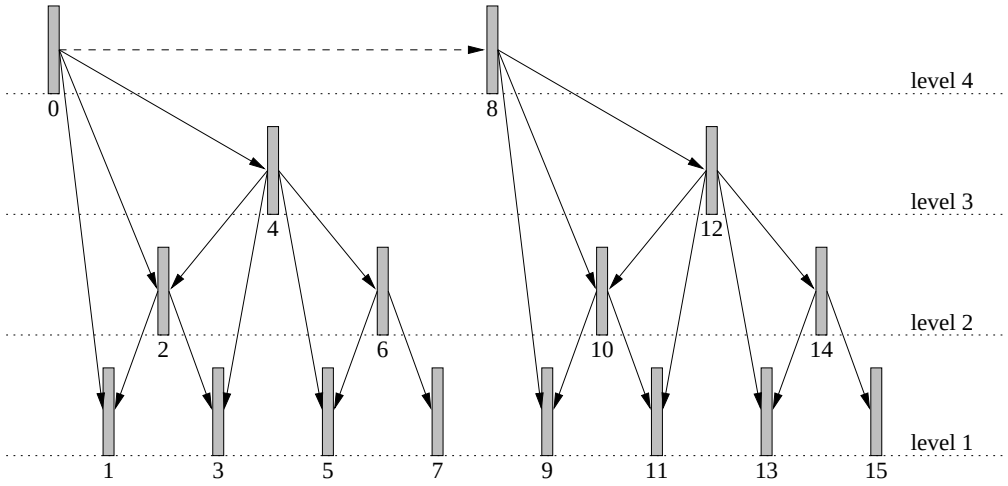


Figure 2.8: Example of the STAR referencing scheme.

2.3.2 Spatial Scalability – Low Band Correction

Spatial scalability in the proposed WSVC is realized using a Laplacian pyramid in the 2D+t+2D scheme with an ISP technique named Low Band Correction (LBC), which was proposed by Han [29]. We will denote with $\mathbf{R}_i$ the space of all frames at resolution level i, with the following assumptions:

1. $\mathbf{R}_0$ is the largest resolution encoded in the bitstream.
2. Any resolution $\mathbf{R}_i$, $i = 0, 1, \ldots, R-1$ is embedded into the bitstream.
3. The ratio between any two adjacent resolutions is a constant rational number.

Denoting with $C_i^{(j)}$ the representation of the j-th frame of the video sequence at the i-th resolution, we introduce the following operators:

1. $\mathcal{I} : \mathbf{R}_i \to \mathbf{R}_i$ is the identity operator, i.e. $\mathcal{I}(C_i^{(j)}) = C_i^{(j)}$.
2. $\mathcal{D} : \mathbf{R}_i \to \mathbf{R}_{i+1}$ is the down-sampling operator, i.e. $\mathcal{D}(C_i^{(j)}) = C_{i+1}^{(j)}$.

3. $\mathcal{U} : \mathbf{R}_i \rightarrow \mathbf{R}_{i-1}$ is the up-sampling operator and once combined with $\mathcal{D}$, it becomes the identity operator, i.e.

$$\mathcal{D} \circ \mathcal{U} = \mathcal{I}. \tag{2.3}$$

   Note that $\mathcal{U} \circ \mathcal{D} \neq \mathcal{I}$.

4. $\mathcal{L} : \mathbf{R}_i \rightarrow \mathbf{R}_i$ is the low band operator defined by $\mathcal{L} = \mathcal{U} \circ \mathcal{D}$.

5. $\mathcal{H} : \mathbf{R}_i \rightarrow \mathbf{R}_i$ is the high band operator defined by $\mathcal{H} = \mathcal{I} - \mathcal{L}$.

While $\mathcal{D}$ can be any linear operator, it is very important that $\mathcal{U}$ satisfies (2.3), since from that we can derive

$$\mathcal{D} \circ \mathcal{L} = \mathcal{D}, \tag{2.4}$$

$$\mathcal{D} \circ \mathcal{H} = 0, \tag{2.5}$$

$$\mathcal{L} \circ \mathcal{U} = \mathcal{U}, \tag{2.6}$$

$$\mathcal{H} \circ \mathcal{U} = 0. \tag{2.7}$$

and show that both L and H are idempotent, i.e. they satisfy

$$\mathcal{L} \circ \mathcal{L} = \mathcal{L}, \quad \mathcal{H} \circ \mathcal{H} = \mathcal{H}. \tag{2.8}$$

The idea of LBC is to merge all error frames from different resolutions, i.e. $E_i^{(j)},\ i = 0, 1, \ldots, R-1$, into a single error frame, $E^{(j)}$, which has the same dimension as the largest error frame. In this way, no more overhead will be introduced when implementing spatial scalability while coding the error frame. The examples for the LBC encoder and decoder are shown in Fig. 2.9 and Fig. 2.10 respectively. As observed in the merged error frame $E^{(j)}$, which we actually sent, the only information associated with resolution i is the highpass band signal $\mathcal{H}(E_i^{(j)})$. However, we can recover the original signal $E_i^{(j)}$ as long as we know $E_{i+1}^{(j)}$. The procedure is shown as follows,

$$\begin{aligned} E_i^{(j)} &= \mathcal{H}(E_i^{(j)}) + \mathcal{L}(E_i^{(j)}) \\ &= \mathcal{H}(E_i^{(j)}) + \mathcal{L}(C_i^{(j)} - R_i^{(j)}) \\ &= \mathcal{H}(E_i^{(j)}) - \mathcal{L}(R_i^{(j)}) + \mathcal{U}(C_{i+1}^{(j)}) \\ &= \mathcal{H}(E_i^{(j)}) - \mathcal{L}(R_i^{(j)}) + \mathcal{U}(R_{i+1}^{(j)} + E_{i+1}^{(j)}) \\ &= \{\mathcal{H}(E_i^{(j)}) - \mathcal{L}(R_i^{(j)}) + \mathcal{U}(R_{i+1}^{(j)})\} + \mathcal{U}(E_{i+1}^{(j)}). \end{aligned} \tag{2.9}$$

In general, we have

$$E^{(j)} = \mathcal{U}^{R-1}(E_{R-1}^{(j)}) + \sum_{i=0}^{R-2} \mathcal{U}^{i}\{\mathcal{H}(E_{i}^{(j)})\}. \tag{2.10}$$

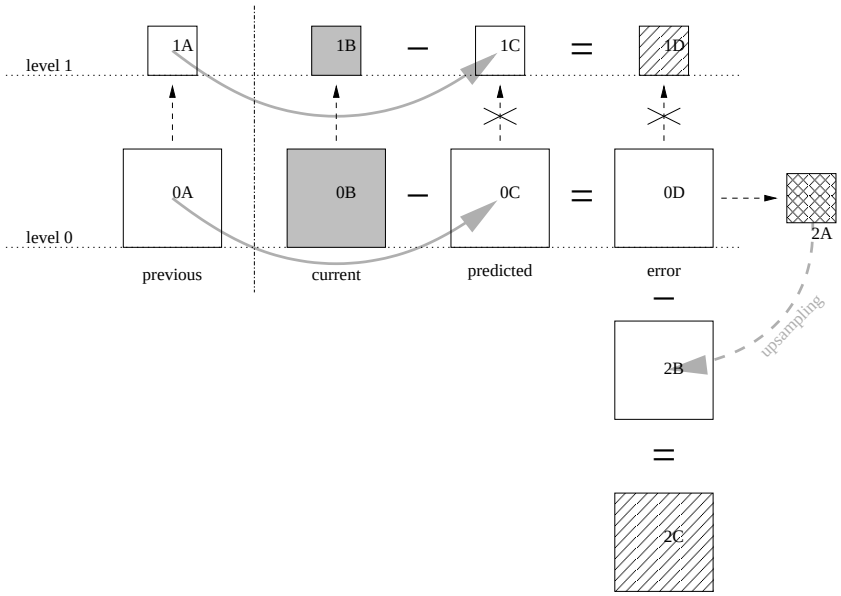


Figure 2.9: Two spatial-layer encoding of inter frames using LBC algorithm. The dashed arrows represent down-sampling using the $\mathcal{D}$ operator and the diagonal pattern represents the information to be sent.

In the common case where dyadic scalability is considered, the operators $\mathcal{D}$, $\mathcal{U}$, $\mathcal{L}$, and $\mathcal{H}$ can be implemented using a two-channel perfect reconstruction filter bank [67], as shown in Fig. 2.11.

2.3.3 Quality Scalability – Resolution Scalable Wavelet Difference Reduction

An algorithm that permits coding the merged representation of error frames $E^{(j)}$ in the subband domain and making different quality layers is Wavelet Difference Reduction (WDR) [78]. WDR is a subband bit plane coding algorithm, which in general consists of a significance pass and a refinement pass during the encoding of each bit plane [58]. In WDR, a specific scanning order for both passes is shown in Fig. 2.12. The low-pass band is first scanned and followed by a sequence of

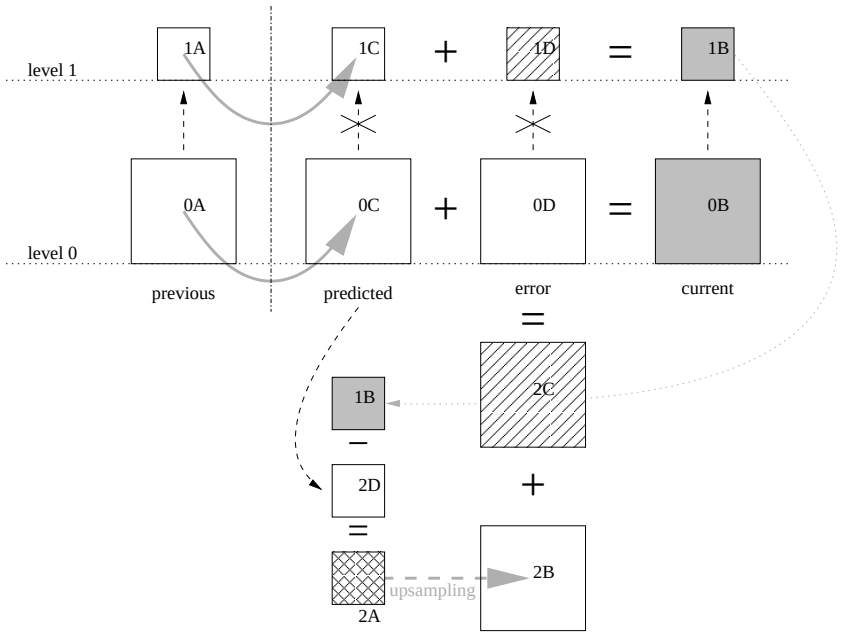


Figure 2.10: Two spatial-layer decoding of inter frames using LBC algorithm. The dashed arrows represent down-sampling using the $\mathcal{D}$ operator and the diagonal pattern represents the received information.

high-pass bands. Within each high-pass band, the scanning order tries to exploit, as much as possible, the correlation between the coefficients. Hence, the order is vertical for the bands which contain the high-pass band content of the rows only (since they represent vertical edges), and the order is horizontal for the bands which contain the high-pass band content of the columns only (horizontal edges), whereas for bands with the high-pass band content of both columns and rows, the order is a zig-zag scanning (diagonal edges). The term "difference reduction" refers to the way in which WDR encodes the locations of significant coefficients efficiently in the significance pass. For more details, please refer to [58].

The nature of WDR, where inter band correlation can be easily decoupled, makes it friendly to cooperate with spatial scalability. We can simply break the inter band links between different resolutions and apply WDR separately, as shown in Fig. 2.13. We call this Resolution Scalable WDR (RSWDR). In RSWDR, a target bit rate (or many of them) can be chosen for the base resolution and a different target bit rate can be chosen for higher resolutions. Note that while

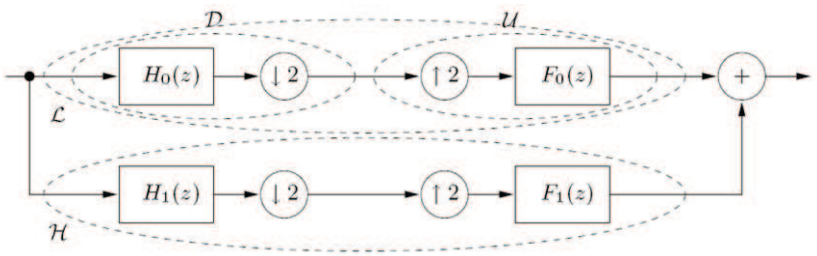


Figure 2.11: Two channel perfect reconstruction filter bank and its relation to $\mathcal{D}$, $\mathcal{U}$, $\mathcal{L}$, and $\mathcal{H}$ operators.

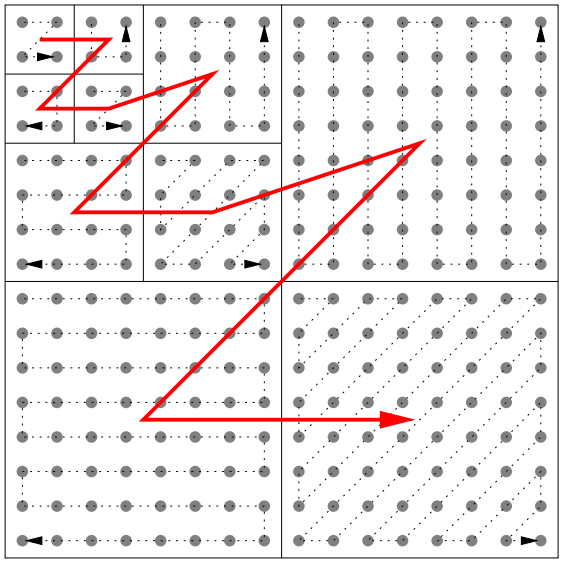
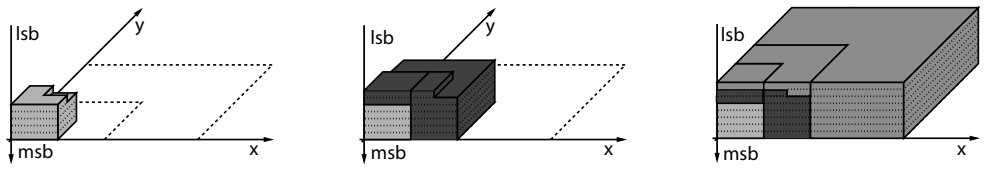


Figure 2.12: Scanning order of WDR using three levels of decomposition. Inter subband order is shown in bold solid line and intra subband orders are shown in dotted lines.

coding $\mathcal{H}(E_i^{(j)})$, it is possible to spend some bits back on refining $E_{i+1}^{(j)}$, since the decoding accuracy of $E_{i+1}^{(j)}$ has a direct impact on the decoding accuracy of $E_i^{(j)}$, as can be observed from (2.9).

The system diagram of the proposed WSVC, integrating the aforementioned techniques, is shown in Fig. 2.14.



(a) (b) (c)

Figure 2.13: Resolution scalable WDR. (a) $E_2^{(j)}$ coding until the bit budget for the smallest resolution sequence is reached. (b) $\mathcal{H}(E_1^{(j)})$ coding until the bit budget for the middle resolution sequence is reached. (c) $\mathcal{H}(E_0^{(j)})$ coding until the bit budget for the whole sequence is reached.

2.4 Acknowledgement

Portions of this chapter appear in "Motion Vector Field Manipulation for Complexity Reduction in Scalable Video Coding", Meng-Ping Kao and Truong Nguyen, in Proceedings of the *IEEE Asilomar Conference on Signals, Systems and Computers*, Oct. 2006. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

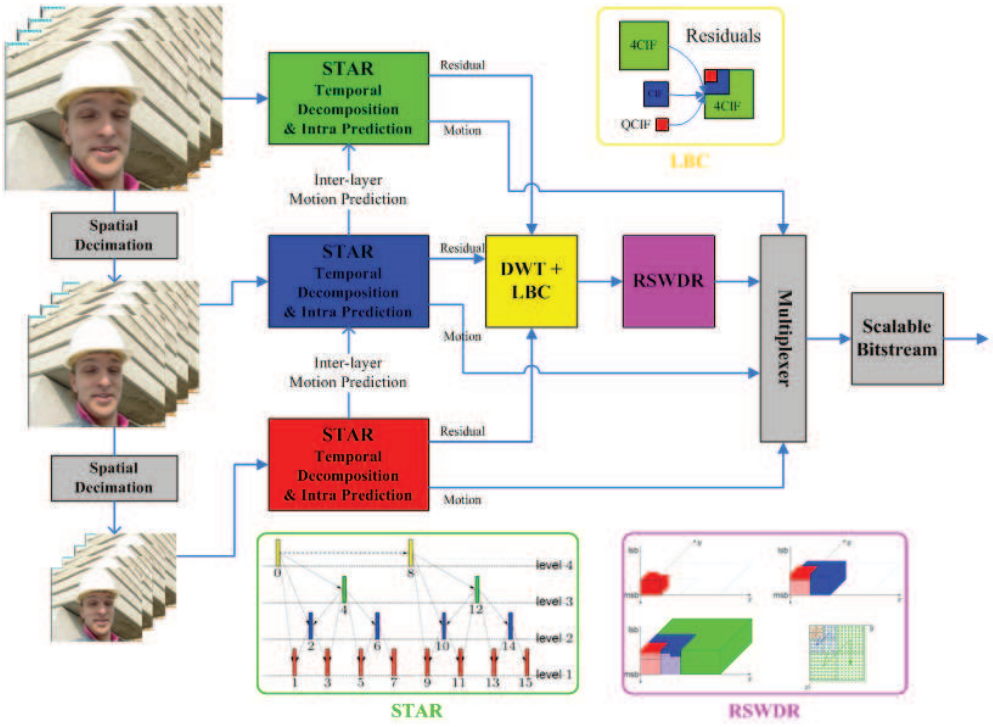


Figure 2.14: System diagram of the proposed WSVC consisting of STAR, LBC, and RSWDR.

3 Complexity Reduction via Scalable Motion Manipulation

In general, the most computationally intensive part of a modern video codec is Motion Estimation (ME) and the associated rate distortion optimization (RDO) selection [80]. For example, H.264/AVC provides many inter and intra modes to be chosen from. For each inter mode, reference indices and motion vectors are further determined from an enormous candidate set. The complexity issue will only get worse if RDO is turned on. As far as the SVC standard is concerned, this complexity problem becomes more severe since not only ME has to be performed on all spatial resolution layers, but also the number of possible modes is increased due to newly introduced inter-layer predictions.

A straightforward remedy to the complexity problem is to utilize fast ME [87] and RDO algorithms [38]. Experimental results have shown that advanced fast ME and RDO algorithms can efficiently reduce the encoder complexity while introducing only slight performance losses. However, when it comes to SVC, we can expect more savings.

As observed in Fig. 2.1, the ME operation is embedded in the "temporal decomposition" block using a hierarchical B-picture structure. Obviously, separate ME needs to be applied on each spatial layer. Although further coding redundancy could be explored by inter-layer motion prediction, it does not help to reduce the complexity. This gives us a motivation for the proposed work in this chapter, aiming at complexity reduction in SVC. The idea is to design a scalable motion vector field (MVF) model and the associated MVF scaling algorithms, such that

the entire MVF information required for encoding could be accurately predicted from a single MVF at a certain spatial layer. In this way, the encoding complexity will be reduced to a level comparable to a single layer video codec.

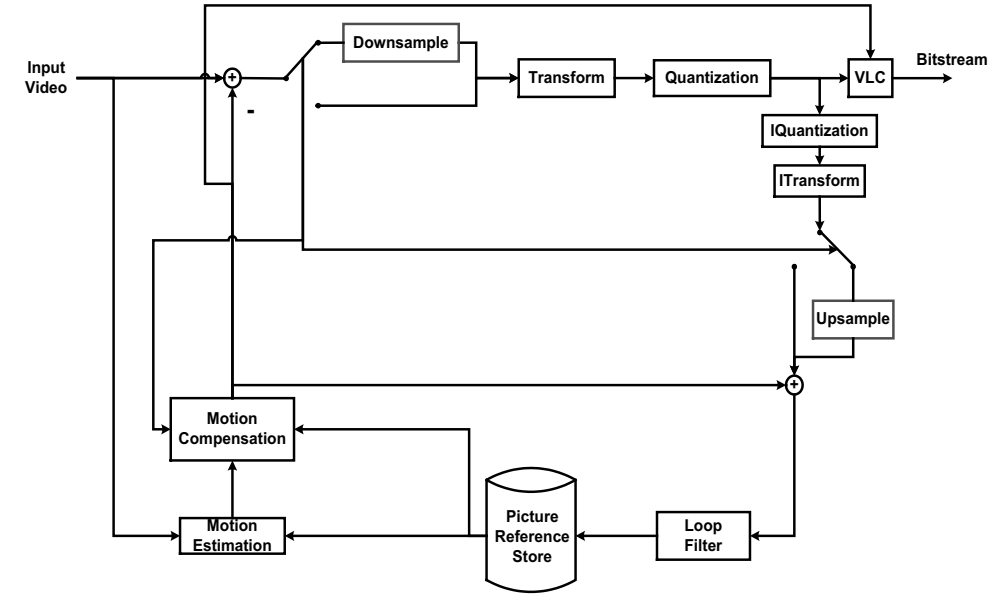
As far as coding efficiency is concerned, since the RDO in the SVC standard is performed from one spatial layer to the next, only local optimization given the previous encoded layer is guaranteed. On the other hand, RDO needs to be performed only once on the specified ME layer based on the proposed scalable MVF model. Therefore, we are able to obtain better total efficiency due to the global optimization process. As far as applications are concerned, a good example is video broadcasting, where total coding efficiency of the combined layers is an important requirement, instead of that of a single layer.

In this chapter, previous works on complexity reduction in SVC will first be reviewed. We then propose our scalable MVF model and the associated MVF scaling operations. The complexity analysis will be provided for various coding schemes. Note that simulation results in this chapter are based on the proposed low complexity WSVC framework introduced in the previous chapter.

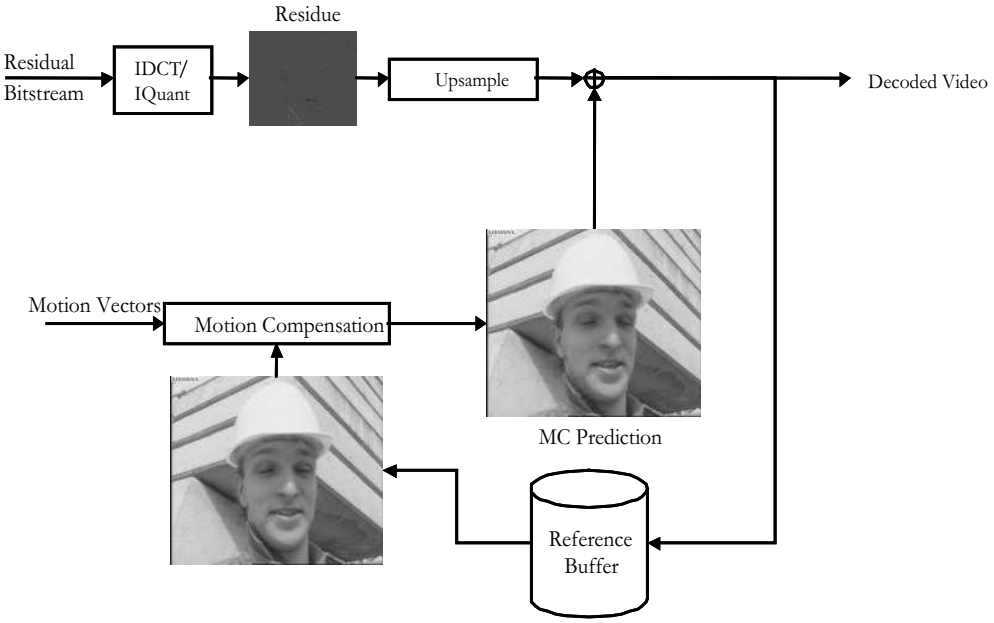
3.1 Reduced Resolution Update Mode

The Reduced Resolution Update (RRU) mode [73] was proposed by Tourapis and Boyce to Video Coding Experts Group (VCEG) as a potential tool for achieving improved encoding performance within H.264/AVC and its future extensions. The basic idea of RRU is to encode an image at a reduced resolution, while performing motion prediction using a full resolution reference, which also allows the final image to be reconstructed at full resolution. The size of a macroblock is doubled to 32x32 in the RRU mode and the associated macro/sub block partitions are also doubled. The prototype encoder and decoder of the RRU mode are shown in Fig. 3.1 (a) and (b) respectively.

The concept of the RRU mode can be used to support spatial scalability in SVC with reduced complexity that is similar to a single layer codec. Thomson's proposal to the MPEG meeting in April 2005 [37] clearly demonstrated the idea.



(a)



(b)

Figure 3.1: Reduced resolution update mode. (a) Encoder. (b) Decoder.

In their proposal, RRU is used for inter picture (P and B) coding and the traditional spatial scalability approach is retained for intra picture (I) coding. For inter pictures, a single bitstream (including MVF and residuals) could be decoded into both low and high resolutions. The RRU mode is directly applied for decoding high resolution sequences as it was originally designed for. A down-sampled MVF, along with the encoded residuals (already down-sampled in RRU), are used to decode the low resolution sequence. The decoder system diagrams for low and high resolutions are shown in Fig. 3.2 (a) and (b) respectively.

We also summarize the operations of Thomson's decoder in Table 3.1. In short, the key concept introduced in Thomson's proposal that enables RRU with spatial scalability is the MVF down-sampling algorithm. It is also the key factor to the reconstruction quality of low resolution sequences.

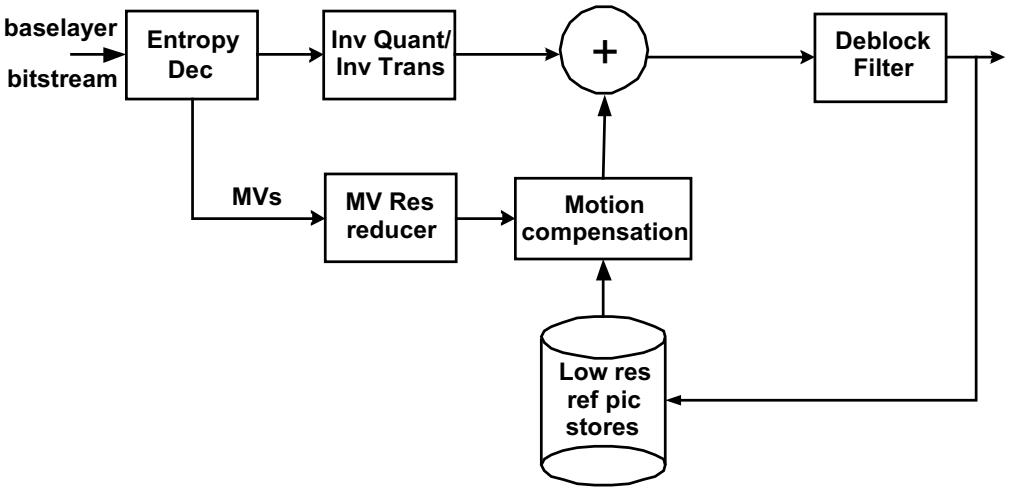
Table 3.1: Thomson's decoder operations

decoder	MVF	residuals	MC	decoded buffer
low resolution	down-sample	-	low resolution	low resolution
full resolution	-	up-sample	full resolution	full resolution

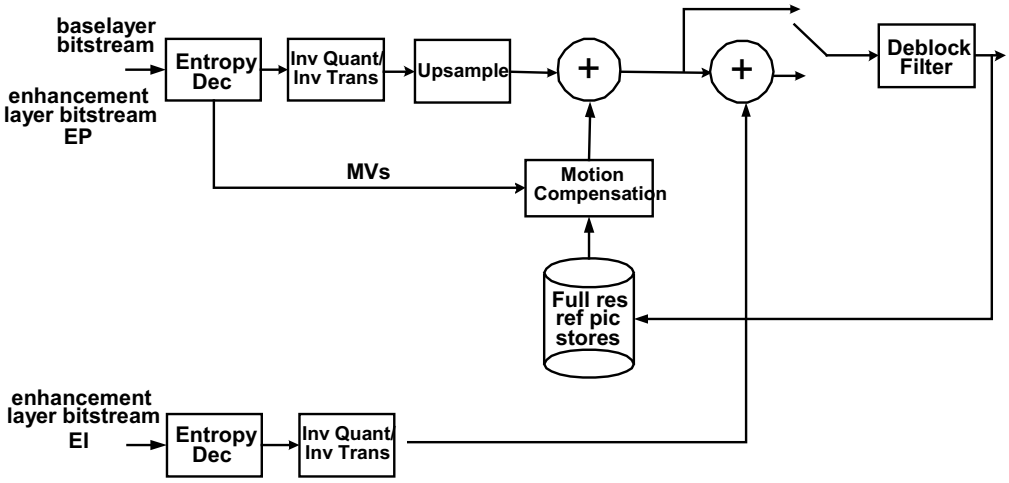
The advantages and disadvantages of the RRU mode are summarized below.

Advantages

- Low complexity (both encoder and decoder).
- High resolution and low resolution sequences are decoded independently (for inter frames).
- High coding efficiency for full resolution sequences.
- Traditional intra-picture coding prevents error drifting in low resolution sequences and provides better visual quality in high resolution sequences.



(a)



(b)

Figure 3.2: System diagrams of Thomson's decoder for reduced complexity SVC. (a) Low resolution sequence. (b) High resolution sequence.

Disadvantages

- Error drifting problem in low resolution sequences due to incoherent reference pictures.
- Visual artifacts in high resolution sequences due to non-invertibility of the down-sampling and up-sampling processes.
- Spatial scalability is constrained to only 2 levels, i.e. high and low resolutions.
- Inaccurate motion estimation in high resolution sequences due to doubled block sizes.

The simulation results from [56] show that for full resolution sequences, Thomson's decoder performs better in low bit rate scenarios and worse in high bit rate scenarios. As for decoding low resolution sequences, it is always worse than JSVM [40]. However, as far as complexity is concerned, Thomson's encoder acts like a single layer encoder which is more efficient compared to the multiple ME operations in JSVM.

3.2 Scalable MVF Model

The idea of a scalable MVF model is to generate all the MVFs for different spatial layers from a single MVF that we have at hand. Since the given MVF could be at any arbitrary layer, the model must provide the capabilities for both up-sampling and down-sampling operations of the given MVF.

In order to maintain a low complexity SVC encoder, a single ME operation (for a pre-specified spatial layer) throughout the entire encoding process is strictly enforced. In other words, motion refinement using inter-layer motion prediction in the SVC standard will not be included in our work. However, a few MC operations are allowed when deciding the best mode, or when picking the best MV, as we will illustrate later.

3.2.1 Up-Sampling

Since the up-sampled MVFs need not be encoded or transmitted (they can be reproduced identically at the decoder in the same manner as at the encoder), we do not have to worry about the bit rate required for encoding them. Therefore, it is straightforward to use smaller MB sizes, and thus more motion vectors, to increase the prediction accuracy. In our case, we keep the same macroblock size for higher resolution layers. Since the resolution increases by 4 times from layer to layer, the MVF size also increases by 4 times, i.e. for each motion vector at the current resolution, we should be able to generate 4 corresponding motion vectors for the above resolution, as shown in Fig. 3.3. There are two candidate methods, and the final up-sampled MVF will be the one which gives lower RD costs.

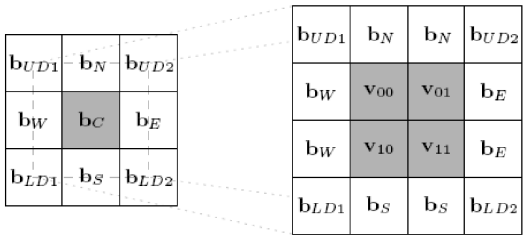


Figure 3.3: Motion vector interpolation from a lower resolution.

Repeat Mode

In the repeat mode, we have the same MVs for the 4 up-sampled macroblocks.

$$v_{00} = v_{01} = v_{10} = v_{11} = 2b_C. \tag{3.1}$$

It is equivalent to having a larger macroblock in the higher resolution sequence and the associated MV is duplicated, except for scaling by a factor of 2.

Smooth Mode

In the smooth mode [23], we have a cost function which measures the smoothness along various directions, i.e. vertical, horizontal, diagonal, and etc.

$$\Psi(\mathbf{v}) = \Psi_N(\mathbf{v}) + \Psi_S(\mathbf{v}) + \Psi_E(\mathbf{v}) + \Psi_W(\mathbf{v}) + \Psi_{UD}(\mathbf{v}) + \Psi_{LD}(\mathbf{v}) + \Psi_C(\mathbf{v}), \quad (3.2)$$

where

$$\Psi_N(\mathbf{v}) = \sum_{i,j=0,1} (v_{i,j} - v_{i-1,j})^2 \quad (3.3)$$

is the vertical smoothness measurement from top to bottom. The other measurements can be defined in a similar way (please refer to [23] for details). Eq. (3.3) can be rewritten in a matrix form,

$$\begin{aligned} \Psi_N(\mathbf{v}) &= ||\mathbf{A}_N\mathbf{v} - \mathbf{b}_N||^2 \\ &= \mathbf{v}^T(\mathbf{A}_N^T\mathbf{A}_N)\mathbf{v} - (2\mathbf{b}_N^T\mathbf{A}_N)\mathbf{v} + \mathbf{b}_N^T\mathbf{b}_N. \end{aligned} \quad (3.4)$$

Thus, the total cost function becomes

$$\begin{aligned} \Psi(\mathbf{v}) &= \sum_x \Psi_x(\mathbf{v}) \\ &= \mathbf{v}^T(\sum_x \mathbf{A}_x^T\mathbf{A}_x)\mathbf{v} - (2\sum_x \mathbf{b}_x^T\mathbf{A}_x)\mathbf{v} + \sum_x \mathbf{b}_x^T\mathbf{b}_x. \end{aligned} \quad (3.5)$$

Consequently, the smoothest solution is as follows:

$$\mathbf{v} = \arg\min_{\mathbf{v}} \Psi(\mathbf{v}) = (\sum_x \mathbf{A}_x^T\mathbf{A}_x)^{-1} \sum_x \mathbf{A}_x^T\mathbf{b}_x. \quad (3.6)$$

In summary, the smooth mode solution minimizes the difference between neighboring MVs and produces a smoother MVF, which tends to resemble the true motions in a natural video sequence.

3.2.2 Down-Sampling

In contrast to MVF up-sampling, MVF coding has to be taken into consideration in the down-sampling process. Since the MVF in lower resolution sequences will be derived directly from that in the pre-specified ME layer, the same

encoded MVF must be conveyed in order to decode lower resolution sequences. This is, however, very inefficient since some redundant motion information from higher resolution sequences is required while decoding any lower resolution ones. A possible solution is to encode the MVF in an embedded manner such that the down-sampled version can be easily extracted as a subset of the original bitstream.

Secker [62] proposed in 2004 a MVF down-sampling algorithm using DWT such that an embedded coding technique can be applied afterwards. However, this linear approach, also known as the averaging method, is shown to have poor performance on the reconstructed sequence [66]. Instead, some non-linear approaches such as Align to The Best (ATB), Align to The Worst (ATW), and Vector Median (VM) filter seem to be better choices, even if a little overhead due to mode map coding (in the VM case) is introduced [26].

Note that the methods mentioned above, either linear or nonlinear, are designed to downsize the MVF and thus retain the same macroblock size in lower resolution sequences. Although the motion info is further reduced through the down-sampling process, the MC performance is also sacrificed. Therefore, another method, which applies the given MVF to all lower resolution sequences without down-sampling, is proposed to increase the MC accuracy. The improved MC quality may sometimes benefit the final coding performance, depending on the actual RD characteristics. We will discuss the two MVF down-sampling modes below and again the final decision will lean toward the one with lower RD cost.

Merged Mode

In the merged mode, MVF is halved between two successive spatial layers. The four MVs (and the corresponding motion information) from the upper layer are merged into a single MV. Therefore, the macroblock size remains unchanged from layer to layer. This is the most compatible mode, since no undefined macroblock size will be introduced. On the other hand, motion information is reduced by four times, at the price of impaired MC quality.

There are many methods for merging neighboring MVs to obtain an approximation for lower resolution sequences. Most of them come from the video

transcoding literature [26, 20, 25]. In summary, the linearly averaging method, e.g. Align-to-Average Weighting (AAW), generally does not give satisfactory result. On the other hand, choosing the most suitable MV from the four upper layer candidates seems to be a better choice. In particular, the Align-to-Best Weighting (ABW) method picks the MV that gives the smallest MC residual energy in the upper layer. The Align-to-Worst Weighting (AWW) method takes the MV with the largest MC residual energy instead.

The VM filter is yet another choice, if no residual information is available besides the MVF itself [10].

$$\sum_{i=1}^{N} ||\mathbf{v}_{VM} - \mathbf{v}_i||_p \leq \sum_{i=1}^{N} ||\mathbf{v}_j - \mathbf{v}_i||_p \qquad j = 1, 2, \ldots, N. \tag{3.7}$$

The VM method tends to produce a smoother MVF which resembles the true motion characteristics. However, since less information is taken into consideration, VM is usually not as good as ABW or AWW.

On the other hand, if more information is available, such as the MC residuals or the Discrete Cosine Transform (DCT) coefficients, Adaptive Motion Vector Resampling (AMVR) can assign proper weightings adaptively to VM, and achieve better performance [66].

$$\sum_{i=1}^{N} w_i||\mathbf{v}_{WVM} - \mathbf{v}_i||_p \leq \sum_{i=1}^{N} w_i||\mathbf{v}_j - \mathbf{v}_i||_p \qquad j = 1, 2, \ldots, N. \tag{3.8}$$

More advanced methods based on AMVR which are optimized to different MVF characteristics are also available, e.g. AMVR-DIM [26].

As for applying MVF merging techniques in SVC, the situation is slightly different from transcoding. During the encoding phase for a lower resolution sequence, the residual information, or DCT coefficients for higher resolution layers, are not yet available. As a consequence, any method that relies on this unavailable information is not applicable, unless it uses proper modifications or approximations. However, we do have the current frame and reference frame for the current spatial layer, which allows us to evaluate all four possible MV candidates (in the dyadic case) from the higher resolution layer, and choose the best MV. In this

way, the resulting down-sampled MVF is guaranteed to outperform any VM-based method in the sense of MC quality.

Since the size of MVF shrinks by 4 times from one layer to the next, the number of bits for MVF coding can be reduced, which in turn benefits the residual coding under the same target bit rate. Note that the proposed merged mode requires an additional mode map that indicates where the current MV comes from in the higher resolution MVF, similar to any other VM-based methods. This is roughly half of the number of bits per block, before applying predictive or entropy codings. On the other hand, ABW and AWW do not require these additional bits.

Direct Mode

In order to achieve better MC accuracy than the merged mode, we can simply retain the original MVF, i.e. without any down-sampling process, and apply it to the lower resolution layer with halved block size. In other words, for every block, the co-located block size in the lower resolution sequence is down-sampled such that the original MV could be applied properly without modifications (except for down-scaling by 2 in the dyadic case). This is called the direct mode. As long as the down-sized block is still standard compliant, e.g. bigger or equal to 4x4 in H.264/AVC, the direct mode serves as a valid option for MVF down-sampling.

In order to provide a theoretical justification for the direct mode, we model it in a simplified scenario where only a 1D signal and global translational motion are considered. Assume that the current signal is $C(n)$ and the reference signal is $R(n)$. The motion compensated error is

$$E(n;d) = C(n) - R(n-d), \tag{3.9}$$

given d as the applied MV. The best MV under MMSE criterion is

$$d^* = arg\min_d \{\sum_n E^2(n;d)\}. \tag{3.10}$$

Suppose both $C(n)$ and $R(n)$ undergo the same down-sampling operation as shown in Fig. 3.4.

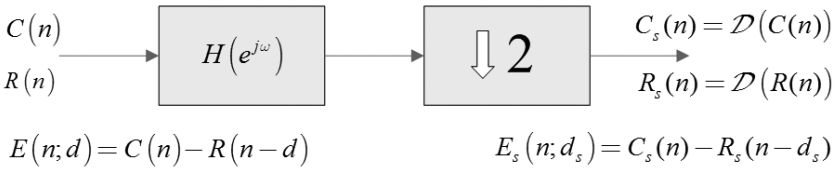


Figure 3.4: Relationships of down-sampling operation.

$$E_s(n; d_s) = \mathcal{D}\{E(n; 2d_s)\}. \tag{3.11}$$

By applying Discrete Time Fourier Transform (DTFT) [54],

$$E_s(e^{j\omega}; d_s) = \frac{1}{2}\{E(e^{j\frac{\omega}{2}}; 2d_s)H(e^{j\frac{\omega}{2}}) + E(e^{j\frac{\omega-2\pi}{2}}; 2d_s)H(e^{j\frac{\omega-2\pi}{2}})\}. \tag{3.12}$$

According to Parseval's Theorem [54],

$$\begin{aligned}\sum_n |E_s(n; d_s)|^2 &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |E_s(e^{j\omega}; d_s)|^2 d\omega \\ &= \frac{1}{4\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} |E(e^{j\omega}; 2d_s)H(e^{j\omega}) + E(e^{j(\omega-\pi)}; 2d_s)H(e^{j(\omega-\pi)})|^2 d\omega.\end{aligned} \tag{3.13}$$

In our case, $H(e^{j\omega})$ is an anti-aliasing pre-filter with cutoff frequency $\pi/2$. Therefore, it is reasonable to approximate it with an ideal lowpass filter with the same cutoff frequency. Eq. (3.13) then becomes

$$\begin{aligned}\sum_n |E_s(n; d_s)|^2 &\approx \frac{1}{4\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} |E(e^{j\omega}; 2d_s)H(e^{j\omega})|^2 d\omega \\ &\approx \frac{1}{4\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} |E(e^{j\omega}; 2d_s)|^2 d\omega.\end{aligned} \tag{3.14}$$

If we further assume that the error signal $E(n; d)$ is a lowpass signal, i.e. the Power Spectral Density (PSD) of the error signal is lowpass, we have

$$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} |E(e^{j\omega}; 2d_s)|^2 d\omega \approx \int_{-\pi}^{\pi} |E(e^{j\omega}; 2d_s)|^2 d\omega. \tag{3.15}$$

Note that the satisfaction of this assumption is highly dependent on the video content, and not necessarily true for all video sequences. In order to verify this assumption among various natural video sequences, the PSD of the first MC residual

frame is shown in Fig. 3.5 for the FOREMAN, BUS, and FOOTBALL sequences. As observed from Fig. 3.5, the major energy resides in the low pass band from $-\pi/2$ to $\pi/2$ for all three sequences, despite the fact that small non-zero energy appears in the high pass bands and the individual energy distribution varies from sequence to sequence.

Eq. (3.14) then becomes

$$\sum_n |E_s(n;d_s)|^2 \approx \frac{1}{2}\frac{1}{2\pi}\int_{-\pi}^{\pi} |E(e^{j\omega};2d_s)|^2 d\omega = \frac{1}{2}\sum_n |E(n;2d_s)|^2. \tag{3.16}$$

The best MV for the down-sampled signal under the MMSE criterion is therefore

$$d_s^* = arg\min_d\{\sum_n E_s^2(n;d)\} = arg\min_d\{\frac{1}{2}\sum_n E^2(n;2d)\} = \frac{d^*}{2}. \tag{3.17}$$

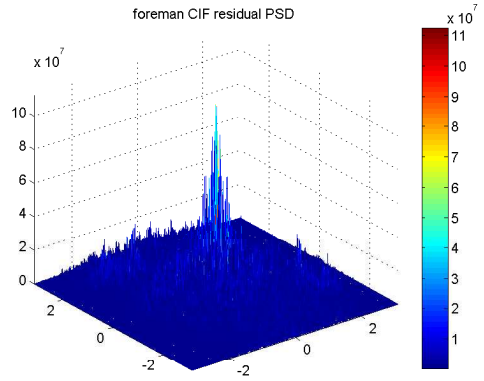
The above equation implies that the best MV for the lower resolution sequence is exactly half of its corresponding optimal MV for the higher resolution sequence, if the two assumptions are satisfied. This also justifies the proposed direct mode for MVF down-sampling.

As a final remark, there is always a tradeoff between MC accuracy and MVF coding bits. Therefore, the decision between the proposed MVF down-sampling modes (direct and merged) should be based on the actual RD cost in order to achieve the best performance.

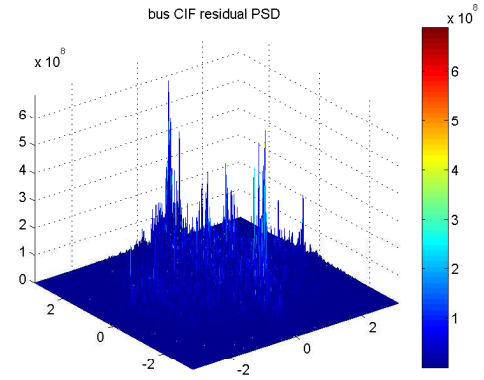
3.3 Proposed Low Complexity WSVC

Fig. 3.6 shows an example system diagram of the proposed low-complexity WSVC, where the middle spatial layer is chosen as the ME layer, and the MVF scaling algorithms illustrated above are utilized to generate the motion information for both the high and low spatial layers. The encoder complexity is reduced since only one ME operation and the associated RDO are required.

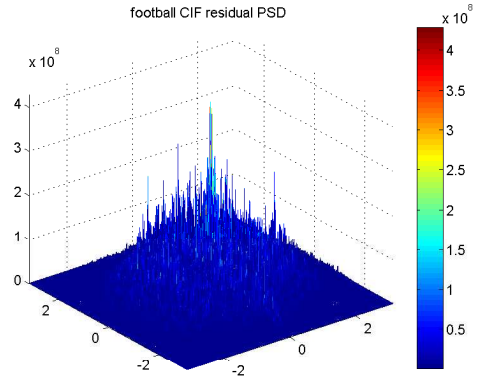
The main difference between our method and Thomson's proposal is that we have relieved many of Thomson's constraints. For example, by introducing the merged mode in the down-sampling operation, we are able to keep the same macroblock size for every layer. This feature makes it possible to have more than one



(a)



(b)



(c)

Figure 3.5: Power spectral density of motion compensated residual signals. (a) FOREMAN. (b) BUS. (c) FOOTBALL.

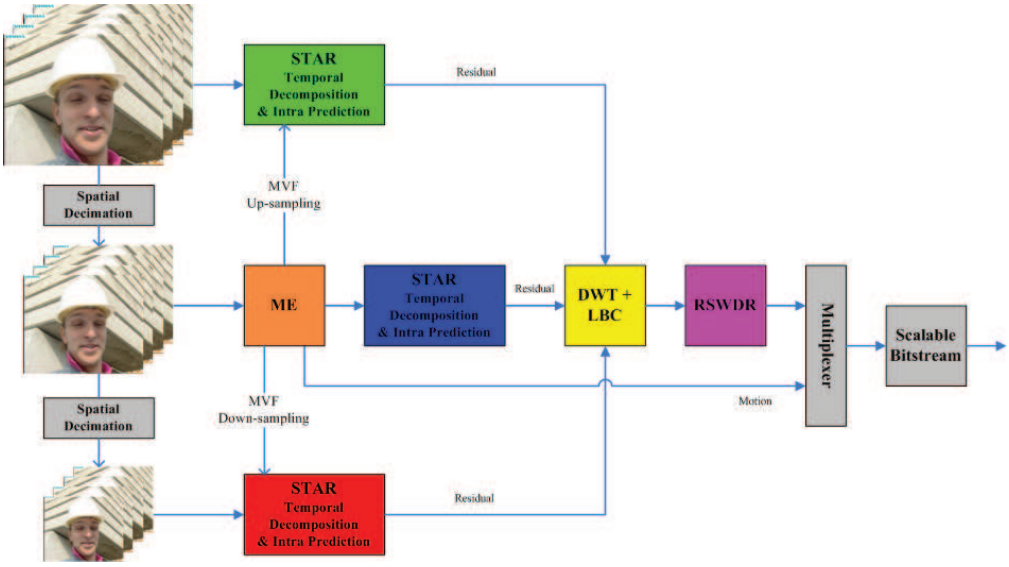


Figure 3.6: System diagram of proposed low complexity WSVC.

spatially down-sampled layer without introducing non standard-compliant block sizes. As for the residual sequence, we relieve the constraint that only lower resolution residuals are sent. The proposed WSVC codec is designed to encode all error pictures together using LBC. These error pictures are considered as a whole quantity during RDO to give the best total efficiency.

Since RDO is performed only once, we are able to achieve the global optimum that maximizes the total coding efficiency. We now elaborate the details. First, instead of performing ME independently for each layer, we now do it only once to a specified layer i. The choice of i depends on the preference between complexity and total coding efficiency. For example, if we want the complexity to be lower, we choose the lower spatial resolution layer (larger i).

After the ME layer is decided, we are now ready to perform ME for that chosen layer. Note that the criterion for RDO can be designed such that the global optimum is reached. As long as the MVF for the ME layer is ready, we can automatically generate the MVFs for the higher and lower layers, based on the proposed scalable MVF model.

We would like to point out an alternative approach for the merged mode. If the RDO is turned on for the ME layer, it is required to explore every possible

block size that gives the best coding efficiency. If we save the best MV for every block size, it will be useful for predicting the merged MV of the down-sampled block. For example, if the final mode of a certain 8x8 block is decided to be four 4x4 blocks, we choose the best MV for the 8x8 block (already saved after RDO) as the MV for the 4x4 down-sampled block in the lower resolution. In this way, we save the computation for checking the 4 original MVs. However, the tradeoff is that more bits are required to code a new MV with this alternative.

3.4 Complexity Analysis

In this section, we compare the complexities among the aforementioned scalable video coding schemes through the usage of the most computationally intensive building blocks. These operations, including motion estimation, spatial interpolation for subpixel motion estimation, discrete wavelet transform and inverse discrete wavelet transform, contribute more than 95% of the total computations in both the encoder and the decoder, where ME is not considered at the decoder.

In order to compare with Thomson's codec, we incorporate the RRU mode into our proposed wavelet codec, in which the LBC is replaced by DWT. The encoder block diagram is shown in Fig. 3.7. Note that the block denoted by "LL Band" performs the extraction of the low-pass band signal from the first level decomposition of wavelet transform, i.e. only one quarter of the transformed coefficients are retained after this operation. The LL Band operation is essentially the counterpart of down-sampling in the DCT-based codec, as previously shown in Fig. 3.1. We note that the RRU mode is similar to a conventional (spatially non-scalable) video codec. Spatial scalability is supported at the decoder by reusing the same non-scalable bitstream to produce different resolution sequences.

The complexity comparison chart for Full Search (FS), Down (DN), UP, and the RRU modes at the encoder, the QCIF decoder and the CIF decoder are listed in Tables 3.2, 3.3 and 3.4, respectively. Here we assume there are N pictures in a GOP, in which only the first picture is coded as intra frame. The motion estimation operation is implemented in C and called by MATLAB through the

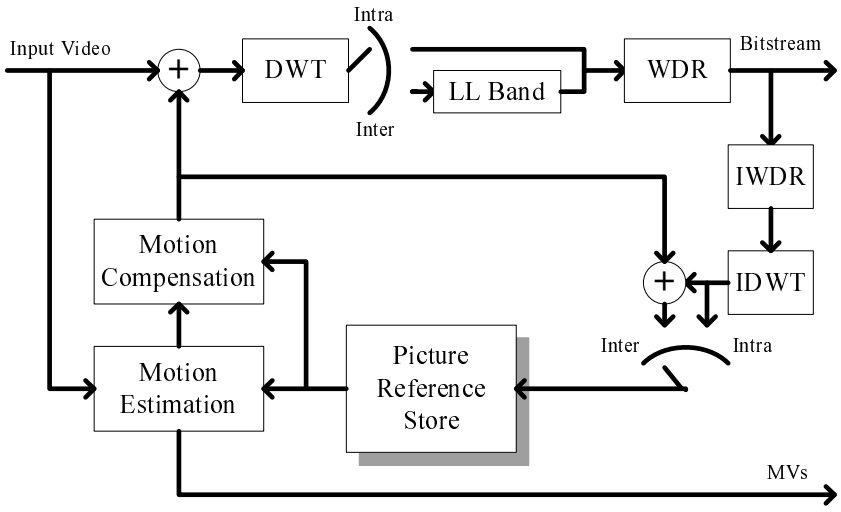


Figure 3.7: RRU mode encoder in WSVC.

mex function to achieve further acceleration.

Note that although each module listed here can be itself optimized to different simulation platforms, this is not our main interest. Instead, we provide the number of usages for each module from the system-level point of view, which can better describe the behavior and complexity of different coding schemes. Therefore, the figures in the "seconds per usage" column are listed for reference purpose only, and they are obtained directly from the MATLAB profiler using BUS as the testing sequence. These figures provide a practical measure of complexity savings in terms of execution time, which we list in the last row of each table using a GOP size of 8. Moreover, we also choose the most computationally intensive modules to be analyzed based on these figures.

According to Table 3.2, the proposed DN mode and the RRU mode both achieve similar savings to the conventional FS mode at the encoder. These savings come from the QCIF size ME for all inter pictures. The RRU mode gains an additional 5% savings from discarding the encoding and decoding processes of the QCIF size sequence. Our proposed UP mode can save up to 2/3 of encoder complexity by omitting the CIF size motion estimation. On the other hand, all these four modes have the same complexity in the QCIF size decoding process, although the coding efficiency is quite different as we will see in the next section.

Table 3.2: Complexity chart: SVC encoder

module	size		sec/use	FS	DN	RRU	UP
ME	CIF		2.28	N-1	N-1	N-1	0
	QCIF		0.57	N-1	0	0	N-1
Interpolation	CIF		0.4	N/2	N/2	N/2	N/2
	QCIF		0.1	N/2	N/2	0	N/2
DWT	CIF	HP Bands	0.07	N	N	1	N
		LL Band	0.03	2N-1	2N-1	N	2N-1
	QCIF		0.02	N	N	N	N
IDWT	CIF		0.07	N	N	N	N
	QCIF		0.015	2N	2N	N	2N
Others			0.13	N	N	N	N
Total		General		3.48N	2.91N	2.745N	1.2N
				-2.88	-2.31	-2.21	-0.6
		N=8		24.96	20.97	19.75	9
Savings				-	16%	21%	64%

For the CIF size decoding process, the RRU mode again benefits from not decoding the QCIF size sequence and thus leads by 19% savings.

3.5 Simulation Results

The following settings are applied in our simulations.

- Input sequences: CIF 30 *fps* (BUS, FOOTBALL, FOREMAN, and MOBILE).
- 3-level temporal scalability (30, 15, and 7.5 *fps*).
- 2-level spatial scalability (CIF and QCIF).
- FGS quality scalability using RSWDR.
- GOP of 8 pictures.

Table 3.3: Complexity chart: QCIF decoder

module	size	sec/use	FS	DN	RRU	UP
Interpolation	QCIF	0.1	N/2	N/2	N/2	N/2
IDWT	QCIF	0.015	N	N	N	N
Others		0.05	N	N	N	N
Total	General		0.115N	0.115N	0.115N	0.115N
	N=8		0.92	0.92	0.92	0.92
Savings			-	0%	0%	0%

Table 3.4: Complexity chart: CIF decoder

module	size	sec/use	FS	DN	RRU	UP
Interpolation	CIF	0.4	N/2	N/2	N/2	N/2
	QCIF	0.1	N/2	N/2	0	N/2
DWT	CIF	0.03	N-1	N-1	0	N-1
IDWT	CIF	0.07	N	N	N	N
	QCIF	0.015	2N	2N	N	2N
Others		0.1	N	N	N	N
Total	General		0.48N	0.48N	0.385N	0.48N
			-0.03	-0.03		-0.03
	N=8		3.81	3.81	3.08	3.81
Savings			-	0%	19%	0%

- Uni-directional (UNI) and bi-directional (BI) ME/MC.

Experiment 1

In this experiment, we compare our proposed MVF down-sampling modes, i.e. the merged mode (MG) and the direct mode (DR), with the VM filter, align-to-best weighting (ABW), and align-to-worst weighting (AWW) methods. The full search (FS) mode which applies full search on both the CIF and QCIF layers is also tested for reference. The bit allocation is shown in Table 3.5.

The PSNR comparison charts of reconstructed and MC sequences are shown

Table 3.5: Bit rate allocation in Experiments 1-4 ($kbps$)

	Exp 1, 2		Exp 3		Exp 4	
	CIF	QCIF	CIF	QCIF	CIF	QCIF
30 *fps*	1024	0	1024 - 1536	512	1024	256 - 512
15 *fps*	512	128	512 - 768	256	512	128 - 256
7.5 *fps*	0	64	0	128	0	64 - 128

in Table 3.6 and Table 3.7, respectively. From Fig. 3.8, it is clear that DR outperforms FS in almost every aspect except for the QCIF (low resolution) reconstructed sequence, as expected. In other words, our proposed SVC codec does achieve better total coding efficiency while maintaining a low encoding complexity, according to Table 3.2. In addition, MG is shown to be better than all other VM-based methods, among which we observe that AWW is better than ABW, which in turn surpasses VM.

Table 3.6: Comparison of different down-sampling modes: BUS reconstructed sequence (dB)

			FS	DR	MG	VM	ABW	AWW
UNI	CIF	PSNR	27.44	27.61	27.21	26.61	26.88	26.99
		△	-	***+0.16***	-0.24	-0.84	-0.56	-0.45
	QCIF	PSNR	25.18	24.47	24.13	23.86	24.05	24.10
		△	-	-0.72	-1.06	-2.81	-1.97	-1.60
BI	CIF	PSNR	27.66	27.85	27.46	26.90	27.23	27.29
		△	-	***+0.19***	-0.20	-0.77	-0.43	-0.38
	QCIF	PSNR	25.22	24.47	24.11	22.78	23.50	23.77
		△	-	-0.75	-1.11	-2.45	-1.72	-1.45

As for the MC sequences, with DR, both the CIF and QCIF MC sequences are better than FS. This implies that the MC accuracy of DR is very good. The QCIF reconstructed sequence is worse because we spend too many bits on MVF coding such that the bit budget is not sufficient for error coding.

On the other hand, MG is designed to reduce MVF coding bits in the QCIF sequence. Compared to DR, MG generates a MVF that is approximately

Table 3.7: Comparison of different down-sampling modes: BUS motion compensated sequence (dB)

			FS	DR	MG	VM	ABW	AWW
UNI	CIF	PSNR	24.41	24.49	24.24	23.86	24.05	24.10
		△	-	***+0.08***	-0.16	-0.55	-0.36	-0.31
	QCIF	PSNR	22.39	23.17	19.80	16.29	18.09	18.24
		△	-	***+0.77***	-2.60	-6.11	-4.30	-4.16
BI	CIF	PSNR	25.30	25.39	25.22	24.98	25.13	25.12
		△	-	***+0.08***	-0.08	-0.32	-0.18	-0.19
	QCIF	PSNR	22.61	23.33	20.03	17.09	18.71	18.85
		△	-	***+0.72***	-2.58	-5.53	-3.91	-3.76

four times smaller in coding bits. However, we do observe a huge performance drop in the MC QCIF sequence, indicating that the down-sampled MVF is no longer a good prediction. Even if more bits are reserved for residual coding, the gap is not easily compensated, as shown in the reconstructed QCIF sequence. Furthermore, due to the error propagation characteristic of LBC, a poor reconstructed QCIF sequence will result in a poor reconstructed CIF sequence. Therefore, MG is in general worse than FS. In terms of total coding efficiency, we conclude that DR > FS > MG.

Experiment 2

After exploring the various possible modes in MVF down-sampling, we will adopt the best one, i.e. the DR mode, as our MVF down-sampling (DN) mode, and compare it with the MVF up-sampling (UP) operation. Again, FS is listed for the purpose of comparison.

The test sequences include BUS, FOOTBALL, FOREMAN, and MOBILE, all of which are in CIF and 30 *fps*. The bit allocation is the same as in Experiment 1. Only uni-directional prediction is tested here.

The PSNR comparison charts of the decoded and MC sequences are shown in Table 3.8 and Table 3.9, respectively. We observe that the reconstructed sequences using DN are better for CIF and worse for QCIF as expected (comparing

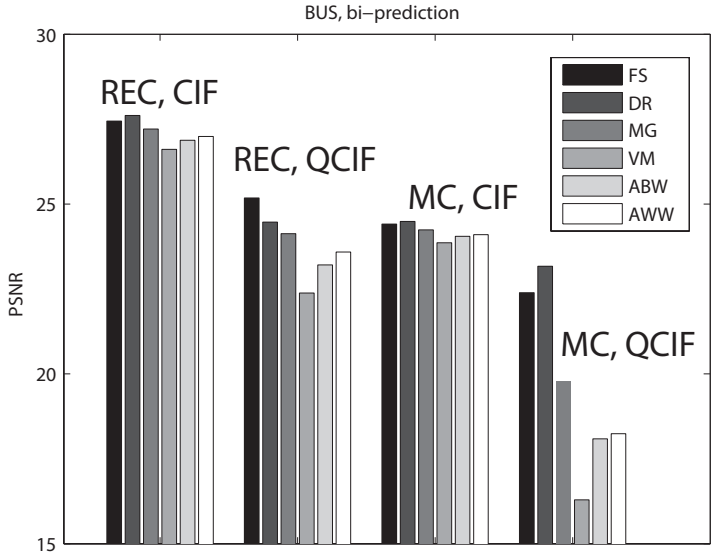


Figure 3.8: PSNR comparison chart of different MVF down-sampling modes.

to FS). As expected, the UP mode is the worst since it requires only 1/3 the computations according to Table 3.2. From this experiment, similar results are observed in different test sequences, which demonstrates the effectiveness of our proposed SVC codec. In particular, our codec works best for FOOTBALL and worst for MOBILE, under the same encoder settings.

Experiment 3

We now test the BUS sequence with varying bit rates to obtain the RD plots. The bit allocation is shown in Table 3.5, and the results are shown in Fig. 3.9.

From this experiment, the coding efficiency of the proposed DR mode is consistently better than FS, with an additional 19% computational saving at the encoder. On the other hand, the RRU mode achieves the best performance. However, as we have mentioned before, the RRU mode is essentially a non-scalable codec, which sacrifices the spatial scalability for CIF coding efficiency. As we will see in the next experiment, the performance of RRU in the QCIF decoding is so

Table 3.8: Comparison of the FS, DN, and UP modes using different test sequences: reconstructed sequences (dB)

		FS	DN	△	UP	△
BUS	CIF	27.58	27.73	***+0.15***	26.53	-1.05
	QCIF	25.38	24.75	-0.63	25.38	0
FOOTBALL	CIF	30.84	31.13	***+0.29***	30.85	+0.01
	QCIF	27.63	26.28	-1.35	27.63	0
FOREMAN	CIF	33.89	34.03	***+0.14***	33.28	-0.61
	QCIF	31.26	30.29	-0.97	31.26	0
MOBILE	CIF	24.37	24.39	***+0.02***	23.29	-1.1
	QCIF	22.77	22.47	-0.30	22.77	0

Table 3.9: Comparison of the FS, DN, and UP modes using different test sequences: motion compensated sequences (dB)

		FS	DN	△	UP	△
BUS	CIF	24.72	24.79	***+0.07***	21.37	-3.35
	QCIF	22.77	23.40	***+0.63***	22.77	0
FOOTBALL	CIF	25.09	25.23	***+0.14***	22.82	-2.27
	QCIF	22.83	24.30	***+1.47***	22.83	0
FOREMAN	CIF	30.70	30.79	***+0.09***	27.83	-2.87
	QCIF	28.35	29.04	***+0.69***	28.35	0
MOBILE	CIF	23.10	23.12	***+0.02***	20.72	-2.38
	QCIF	21.65	21.69	***+0.04***	21.65	0

bad that it can hardly be qualified as a scalable codec.

When the bit rate increases, both PSNR gaps from FS are gradually reduced. The advantage from saving the motion bits for the QCIF sequence diminishes when the bit rate goes up.

Experiment 4

In this simulation, we try different bit rates on the BUS sequence. We fix the bit rate for the CIF sequences and vary the ones for the QCIF sequences as shown in Table 3.5. The results are shown in Fig. 3.10.

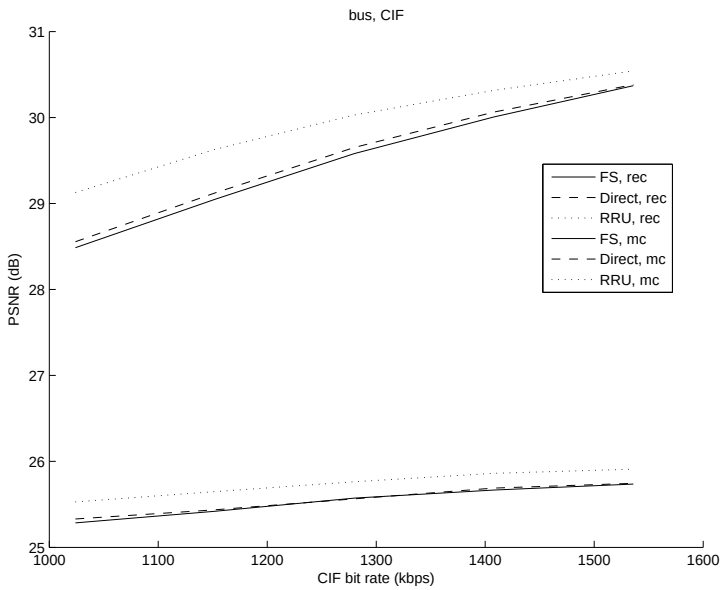


Figure 3.9: RD plots in Experiment 3: BUS, CIF.

Although the proposed DR mode is slightly worse than FS as expected, they both follow a similar rate distortion trend throughout a wide range of bit rates. Unlike the DR mode, the RRU mode is far off the trend and get even worse as the bit rate increases. Note that the total bit rate is fixed in this experiment. Again, the QCIF MC sequence using DR is still the best.

3.6 Summary

In this chapter, MVF scaling algorithms are proposed based on the observed relationships among different spatial resolution layers in an SVC system. These algorithms help to improve the SVC system in two aspects: 1) lower encoder complexity, and 2) increased total coding efficiency.

In the following chapters, a more advanced scalable motion model will be proposed. In addition to spatial scalability, quality scalability is also incorporated. The new scalable motion model is designed to improve the RD curves of a fully scalable SVC system under all possible decoding scenarios.

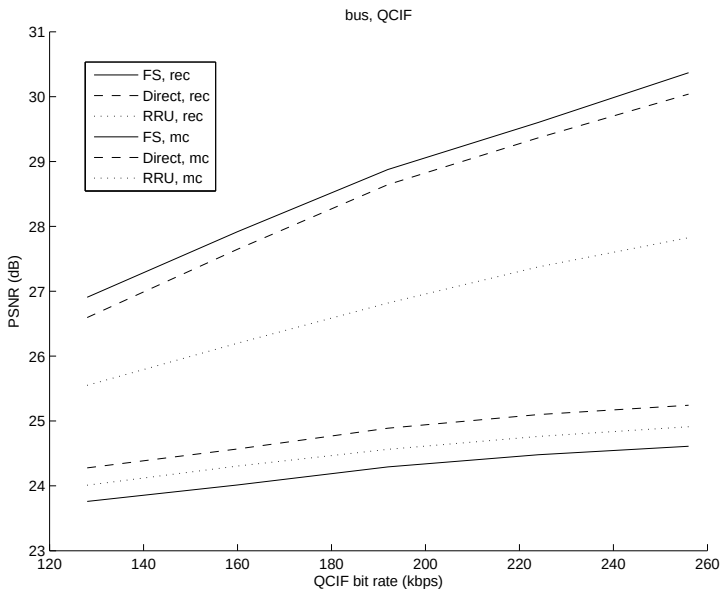


Figure 3.10: RD plots in Experiment 4: BUS, QCIF.

3.7 Acknowledgement

Portions of this chapter appear in "Motion Vector Field Manipulation for Complexity Reduction in Scalable Video Coding", Meng-Ping Kao and Truong Nguyen, in Proceedings of the *IEEE Asilomar Conference on Signals, Systems and Computers*, Oct. 2006. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

4 Scalable Motion

As far as a hybrid video coding scheme is concerned, the total bit rate can be divided into two main portions, motion and texture. Motion bits, together with the underlying motion model, provide a prediction of the current picture. Texture bits, on the other hand, compensate for the difference between the prediction and the current picture. In a lossy coding scenario, the imposed rate constraint prevents lossless description of the prediction difference, and thus distortion is introduced.

As mentioned earlier in the beginning of Chapter 2, a lossy video codec is evaluated by its rate distortion performance. Under a certain rate constraint, the smaller distortion a codec introduces, the better it is. For years, people have been seeking the optimal distribution strategy of the bit budget between motion and texture. The most widely accepted technique is the Lagrange multiplier based motion estimation and mode decision algorithm [55, 68].

A typical RD curve of an encoded video bitstream can be theoretically obtained by finding the convex hull of an infinite number of operating points resulting from different encoder settings, as shown in Fig. 4.1. The Lagrange multiplier for a certain bit rate can be chosen as the negative slope of the tangent line through that particular operating point on the RD curve. The physical meaning of the chosen Lagrange multiplier can be interpreted as a measurement for the efficiency of distortion reduction per bit rate. Any additional bit rate that contributes less than this measurement should be avoided in order to meet the rate constraint. The chosen Lagrange multiplier is applied in the RDO ME process to determine the optimal motion parameters for the target bit rate.

It is easily recognized from the above rationale that in order to achieve

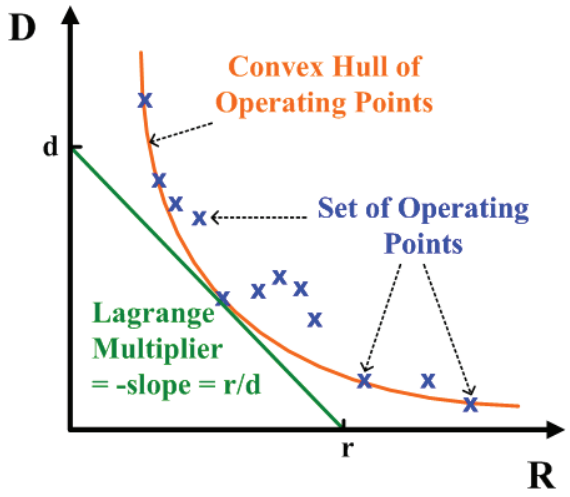


Figure 4.1: Rate distortion curve.

optimality in the RD sense, motion parameters should be optimized to a specific bit rate. In other words, the same set of motion parameters can not be equally efficient for various decoding bit rates. Therefore, in an SVC system where decoding quality can be adjustable, a non-scalable motion bitstream will inevitably sacrifice the coding efficiency for any decoding bit rates away from the presumed one. Moreover, the more the offset is, the worse the performance will be.

Scalable motion, as the main topic of this chapter, is designed to improve the coding efficiency for an SVC framework. By providing different motion quality layers, different versions of motion prediction, each optimized to a presumed decoding bit rate, can be supported within the embedded motion bitstream. In this way, efficiency over a wide range of bit rates can be sustained. As a matter of fact, strong experimental evidence has shown that scalable motion can effectively improve the coding efficiency in the low bit rate range and for low resolution sequences, regardless of the particular SVC system design. Meanwhile, a feasible decoding bit rate can be extended toward its lower end with the help of scalable motion.

Scalable motion is a relatively new concept that has arisen with the development of SVC. In order to provide a solid and comprehensive background of this newly emerged research field, the first section will be devoted to a literature review

of scalable motion. Although the idea of scalable motion was initially inspired by quality scalable coding, issues such as temporal and spatial scalability should also be considered, when applied to an SVC system. We will discuss the required scalable functionalities of a scalable motion design in the second section. In the final section, the assumptions and theories behind the concept of scalable motion will be illustrated. One should often check back with these premises while designing a scalable motion system.

4.1 History

As mentioned earlier, the study of scalable motion began with the development of SVC. Researchers notice that the lossless coding of motion information becomes inefficient when decoding the bitstream at lower resolutions. In particular, a down-scaled motion vector for a low resolution sequence is, in general, obtained by a divide-by-2 operation followed by a rounding operation. Redundant accuracy, which is lost during the rounding phase, is useless for low resolution decoding. By simply applying the progressive coding algorithm, the redundant information can be removed from the bitstream before decoding the low resolution sequence. The saved bits yield improved coding efficiency, with almost no additional overhead.

Based on this concept, Bottreau [17] proposed the first scalable motion framework in 2001, which is the earliest literature in this field to this author's knowledge. In his work, scalable motion is embedded into a fully scalable 3D subband SVC system for improved efficiency on temporally and spatially down-sampled sequences. As will be explained later, temporal scalability for scalable motion is trivial. On the other hand, spatial scalability for scalable motion is supported by a combination of predictive and progressive codings. The lowest spatial resolution motion vectors are obtained by down-scaling and rounding the original MVs, and encoded with a DPCM technique followed by entropy coding using VLC tables. Higher resolution motion vectors are progressively refined and encoded using contextual arithmetic encoding. No specific technique is addressed in his work to target the quality scalability for scalable motion.

In 2003, several schemes for scalable motion coding were proposed. Taubman [72] applied the wavelet transform and bit plane coding techniques to the motion vector field to provide scalability. His following work in 2004 [62], which analyzed the quantization effect on motion vectors, became the theoretical foundation of scalable motion, which we will illustrate later in this chapter.

Hang [31] proposed the first scalable motion framework based on the MC-EZBC [33] codec. Elaborations on scalable motion in MC-EZBC can be found in [34, 74, 82, 83]. Recently, Wu [83] demonstrated an integrated scalable motion vector coding method for MC-EZBC. The new method improves the coding efficiency of both spatial and quality scalabilities.

During the same year 2003, Valentin [76] proposed a vector refinement coding scheme for scalable motion. Turaga [75] demonstrated a special prediction scheme for motion coding in SVC. Barbarien [13], on the other hand, proposed another motion prediction and coding algorithm for in-band motion estimation. His following works can be found in [14, 15], where rate distortion optimization for scalable motion vector coding is explored.

In the following year, 2004, more than 20 publications addressed motion scalability in the context of different requirements and different SVC architectures. Maestroni [47], for example, used the quadtree structure bit plane coding technique to encode the variable block size (VBS) [45, 42, 16, 69, 19] MVF. Xiong [84] proposed an estimation algorithm along with a layered scalable motion structure. In this work, an optimal bitstream extractor for the decoder is also provided. Similar work can also be found in the Microsoft MVC codec [85]. Mrak from the University of London, Queen Mary has done extensive research on scalable motion, including both motion vector accuracy coding and layered motion modeling algorithms. Her research can be found in [49, 50, 51].

In summary, early developments of scalable motion focused mainly on improving the performance of low spatial resolution decoding within an SVC framework. Various motion prediction and coding algorithms were proposed in this stage. More advanced scalable motion approaches began to deal with improved quality scalability in SVC, and layered coding on motion structure has gained

more attention. Optimal bitstream extractor design [11] is also another research topic in this stage. Modern scalable motion design should focus on combined functionalities that support joint scalabilities in an SVC framework. Moreover, due to the non-stationary characteristics of video sequences, rate distortion optimized motion estimation together with scalable motion coding is expected to give better performance to individual contents.

Inspired by prior developments, we realize how important scalable motion is to SVC. Furthermore, the need for a systematic and complete solution to this problem is urgent and requires more study. The contribution of our work is to systematically formulate the scalable motion problem, and to provide a feasible and efficient solution. The particular solution, which will be discussed in the next chapter, not only passively implements the motion scalability, but also actively works seamlessly with the encoder to achieve optimal coding efficiency for scalable video. Compared to previous works, this is the first scalable motion solution that offers joint optimization on both MV accuracy and variable block size scalabilities. The optimization can be achieved via tailored coding techniques and RDO motion estimation.

Before introducing our scalable motion solution, we will first formulate the problem by illustrating those required functionalities.

4.2 Functionalities

Scalable motion is defined as a single and progressively encoded motion bitstream such that applicable MVFs can be efficiently extracted and decoded for any spatial, temporal and quality specification demanded from the SVC decoder. In other words, the scalable motion bitstream should be able to provide all suitable MVFs, which cover all the decoding possibilities that the decoder targets. For example, any combination of sequence sizes ranging from 4CIF to QCIF, frame rates ranging from 30 *fps* to 7.5 *fps*, and bit rates ranging from 2000 *kbps* to 50 *kbps* should find its corresponding MVFs from the unique scalable motion bitstream, and most important of all, in a RD-efficient manner. By RD-efficient we mean the

extracted motion information should provide comparable RD performance as if it were optimally estimated for that non-scalable decoding specification.

A scalable motion model (SMM) is the correct tool for achieving scalable motion within a specific SVC codec. It comprises two major constituents: model structure and RDO ME algorithm. Model structure defines how the encoded motion parameters can be interpreted to the actual MVFs. Various motion quality layers should be structurally supported and the corresponding progressive coding algorithm should also be clearly defined. The RDO ME algorithm, on the other hand, obtains the optimal motion parameters for the chosen model structure, which optimizes the coding efficiency for a wider range of operating bit rates and various decoding scenarios.

From the above definition, SMM is usually tailored for a specific SVC framework, i.e. it is highly codec dependent. However, some common properties do exist among various SVC designs. For example, the supporting mechanisms for temporal and quality scalabilities are similar despite different SVC designs. As far as temporal scalability is concerned, JSVM-9 uses the hierarchical B-picture structure, which is similar to the STAR algorithm in our WSVC codec, while MC-EZBC uses MCTF. In either case, MVFs associated with those irrelevant frames can be simply discarded with no harm to the remaining low frame rate sequence. The saved bits can lead to improved coding efficiency. As a consequence, temporal scalability of scalable motion is in general very trivial and directly inherited from the reference picture selection scheme utilized in the codec, e.g. hierarchical B-picture structure, STAR, or MCTF.

Quality scalability of scalable motion, on the other hand, is another common problem for all architectures. Specifically, given a certain target bit rate for motion, SMM should be able to provide the best MVF among all possible candidates without exceeding that target bit rate. By best we mean under a predefined distortion measurement, e.g. the sum of absolute motion compensated difference (SAD). The operating target bit rates could be fine or coarse grain scalable (FGS or CGS). FGS supports finer motion quality layers, while more scalability overhead might be imposed.

In general, there are two ways to achieve quality scalability. The first way is MV accuracy scalability, which is independent of any underlying motion model. By coding the MV accuracy progressively, e.g. integer MV with half and quarter pixel refinements, the FGS can be roughly achieved.

The second way is motion structure scalability, which is motion model dependent. For simplicity, we focus only on the block-based motion model in this work. In a block-based motion model, the VBS tree structure [65] is a common tool to efficiently describe motions in various scenes. By changing the block size, motions of different objects can be better described. From the VBS point of view, MVF can be decoded at a high bit rate with more refined block sizes, or at a low bit rate with larger block sizes, and thus fulfills the quality scalability requirement. A smart combination of both MV accuracy and motion structure scalability would be ideal for a quality scalable SMM.

In contrast to temporal and quality scalability, spatial scalability for scalable motion depends on the SVC architecture employed. For example, in WSVC, which uses the 2D+t approach, motion information obtained from in-band ME [46, 12, 48] can be predictively coded across increasing resolution bands. As for t+2D, progressive coding of the full resolution motion information can help to easily remove the redundancy for low resolution decoding. In this case, techniques from quality scalable SMM can be used. In our proposed WSVC, the 2D+t+2D scheme resembles the t+2D scheme in that no in-band ME is performed. All ME/MC are applied on low pass pictures, which produces similar MVFs across resolutions and leaves much room for exploring the inter-scale motion redundancy.

Recall from Chapter 3, it has been shown that under the two following assumptions, the optimal MV for one resolution will also be optimal for other resolutions (after proper scaling) in the mean squared error sense. The first assumption requires the resemblance between the down-sampling filter and an ideal low pass filter with cutoff frequency $\pi/2$. This is generally satisfied for commonly used filters such as Daubechies 9/7 filters. The second assumption is that the motion compensated residual signal has a low pass power spectrum density (PSD). This is also supported by experimental results on several testing sequences. As a

consequence, the optimal MV for one resolution should be shared by other resolutions in order to maintain high efficiency, with minimum sacrifice on quality. In other words, the spatial scalability of scalable motion can be easily carried out by down-scaling the highest fidelity MVF to meet the desired resolution. For example, the MVs for the QCIF sequence are the halved versions of those from the CIF sequence.

Note that the down-scaling process could cause possible problems on MV accuracy, as well as the block size for a block-based motion model. For example, a quarter pixel MV will result in a one eighth pixel MV when halving, which might be either not supported by the codec or redundant for the current resolution. On the other hand, a 4x4 block will become a 2x2 block on smaller resolution pictures, which might again either exceed the decoder's capabilities or simply not be worth coding in the RD sense. These two well-known problems on spatial scalability of scalable motion were used to be solved by scalable MV coding and layered modeling, respectively, after the non-scalable MVF is estimated [83].

In our proposed SMM, a complete scalable MV coding algorithm is provided. Given the coding algorithm, MVF is estimated and optimized in a scalable manner via the proposed RDO ME algorithm. At the same time, the function of layered modeling is also inheritably provided after MVF is estimated. This approach provides an alternative to conventional post-estimation coding methods [62], where motion scalability is enabled at the decoder from the non-scalable motion estimated at the encoder. Moreover, in order to guarantee that all down-scaled MVFs are decodable, spatial scalability for scalable motion should be formulated as a constraint problem in our SMM rather than a scalability problem. The details will be explained in the next chapter.

4.3 Theoretical Justification

Motion parameters have traditionally been coded losslessly due to the drift problem inherited from the closed-loop structure. Considering the scenario where a low quality sequence is decoded, it is intuitively a lot easier to discard the extra

texture bits than to quantize the motion parameters. One reason is that the quantization effect of texture bits is easily predicted and quantified. In contrast, quantization of motion bits might result in a non-predictable behavior that is highly correlated to the individual video content. Another reason is that, even though motion information can be quantized, the corresponding texture that reflects the actual difference between the current picture and the new motion-predicted picture can not be re-transmitted. Only the unique version of texture that corresponds to the highest fidelity motion prediction is available for decoding. The mismatch between the quantized motion parameters and the original texture information prevents further investigation of the feasibility of scalable motion.

In this section, theoretical supports for the effectiveness of scalable motion will be discussed. For simplicity, we first consider the case where block motion prediction and texture coding is applied. In addition, uni-directional motion prediction is adopted and thus the motion compensated block can be expressed as a function of the underlying motion vector $\mathbf{v}$, i.e. $M(\mathbf{v})$. The texture block is then

$$T = c - M(\mathbf{v}^*), \tag{4.1}$$

where c is the current block and $\mathbf{v}^*$ is the highest fidelity motion vector. The reconstructed block can be expressed as

$$r = M(\mathbf{v}) + Q(T), \tag{4.2}$$

where Q denotes the quantization function. Consider two scenarios where the sum of motion and texture bits remains the same, i.e. $R(\mathbf{v}_1) + R(Q_1(T)) = R(\mathbf{v}_2) + R(Q_2(T))$. In the first case we assume the motion is coded losslessly, i.e. $\mathbf{v}_1 = \mathbf{v}^*$.

$$r_1 = M(\mathbf{v}_1) + Q_1(T) = M(\mathbf{v}^*) + Q_1(T) \tag{4.3}$$

In the second case $\mathbf{v}_2$ is a quantized version of $\mathbf{v}^*$.

$$r_2 = M(\mathbf{v}_2) + Q_2(T) \tag{4.4}$$

Comparing with the original block, $c = M(\mathbf{v}^*) + T$, if

$$||c - r_2||^2 < ||c - r_1||^2, \tag{4.5}$$

the reconstruction quality in case 2 will outperform the losslessly coded motion in case 1. By plugging in, we get

$$||(M(\mathbf{v}^*) - M(\mathbf{v}_2)) + (T - Q_2(T))||^2 < ||T - Q_1(T)||^2. \tag{4.6}$$

By applying the additive distortion model [62], we have

$$||(M(\mathbf{v}^*) - M(\mathbf{v}_2))||^2 + ||(T - Q_2(T))||^2 < ||T - Q_1(T)||^2 \tag{4.7}$$

Note that the additive distortion model assumes that the motion distortion is orthogonal to the texture distortion. By moving the motion distortion to one side and the texture distortion to the other, we have

$$||(M(\mathbf{v}^*) - M(\mathbf{v}_2))||^2 < ||T - Q_1(T)||^2 - ||(T - Q_2(T))||^2. \tag{4.8}$$

In other words, if the distortion introduced by quantization of motion information is less than the texture distortion difference resulting from using different quantization parameters, rate-scalable motion can achieve better decoding quality. While the distortion from texture quantization can be easily formulated [30, 21], the complicated interaction between motion parameter quantization and the resulting motion prediction quality is not trivial. In [62], Secker analyzed this relationship via the power spectrum density characteristics of the reference picture. In order to make this dissertation more self-contained, we will summarize his work in the rest of this section.

Motion Vector Quantization

The relationship between motion vector mean-squared error (MSE) and the resulting video distortion depends primarily on the power spectral properties of the video data.

Consider the reference picture, $x[\mathbf{n}] \equiv x[n_1, n_2]$, and the motion warping operation, $\mathcal{W}$, which corresponds to a certain non-quantized motion. The motion compensated picture can be expressed as $y[\mathbf{n}] = \mathcal{W}(x)[\mathbf{n}]$. Suppose now $\mathcal{W}$ is affected by quantization of its motion parameters, resulting in the quantized operator $\mathcal{W}'$, which produces $y'[\mathbf{n}] = \mathcal{W}'(x)[\mathbf{n}]$.

Assume that the quantization introduces a constant displacement error, $\boldsymbol{\delta}$, and the ideal motion compensated interpolation for subpixel motion is supported. The displacement in spatial domain, $y'[\mathbf{n}] = y[\mathbf{n}-\boldsymbol{\delta}]$, results in a linear phase shift in the frequency domain, $y'(\boldsymbol{\omega}) = y(\boldsymbol{\omega})e^{-j\boldsymbol{\omega}^t\boldsymbol{\delta}}$. By Parseval's theorem, the total squared error D_y is given by

$$\begin{aligned} D_y &= \sum_{n_1}\sum_{n_2} |y[n_1,n_2] - y'[n_1,n_2]|^2 \\ &= \frac{1}{(2\pi)^2}\int_{\omega_1}\int_{\omega_2} S_y(\boldsymbol{\omega})|(1-e^{-j\boldsymbol{\omega}^t\boldsymbol{\delta}})|^2 d\omega_1 d\omega_2, \end{aligned} \tag{4.9}$$

where $S_y(\boldsymbol{\omega}) = |y(\boldsymbol{\omega})|^2$ is the energy spectral density of $y[\mathbf{n}]$. Applying the Taylor series expansion, we get

$$\begin{aligned} |(1-e^{-j\boldsymbol{\omega}^t\boldsymbol{\delta}})|^2 &= 2 - 2\cos(\boldsymbol{\omega}^t\boldsymbol{\delta}) \\ &= 2\left[\frac{(\boldsymbol{\omega}^t\boldsymbol{\delta})^2}{2!} - \frac{(\boldsymbol{\omega}^t\boldsymbol{\delta})^4}{4!} + \frac{(\boldsymbol{\omega}^t\boldsymbol{\delta})^6}{6!} - \cdots\right]. \end{aligned} \tag{4.10}$$

For small $\boldsymbol{\delta}$, $\boldsymbol{\omega}^t\boldsymbol{\delta}$ is also small, which leaves the actual approximation with the second order term only.

$$D_y \approx \frac{1}{(2\pi)^2}\int_{\omega_1}\int_{\omega_2} S_y(\boldsymbol{\omega})(\boldsymbol{\omega}^t\boldsymbol{\delta})^2 d\omega_1 d\omega_2 \tag{4.11}$$

By further assuming $S_y(\boldsymbol{\omega}) \approx S_x(\boldsymbol{\omega})$, we have

$$D_y \approx \frac{1}{(2\pi)^2}\int_{\omega_1}\int_{\omega_2} S_x(\boldsymbol{\omega})(\boldsymbol{\omega}^t\boldsymbol{\delta})^2 d\omega_1 d\omega_2. \tag{4.12}$$

Note that this assumption is true as long as the motion warping operator $\mathcal{W}$ is not excessively expansive or contractive ,and is implemented using sufficiently high order interpolators. Eq. (4.12) can be expressed in another form as follows.

$$D_y \approx \Psi_{1,x}\delta_1^2 + \Psi_{2,x}\delta_2^2 + \Psi_{3,x}\delta_1\delta_2 \tag{4.13}$$

where

$$\Psi_{1,x} = \frac{1}{(2\pi)^2}\int_{\omega_1}\int_{\omega_2} S_x(\boldsymbol{\omega})\omega_1^2 d\omega_1 d\omega_2 \tag{4.14}$$

$$\Psi_{2,x} = \frac{1}{(2\pi)^2}\int_{\omega_1}\int_{\omega_2} S_x(\boldsymbol{\omega})\omega_2^2 d\omega_1 d\omega_2 \tag{4.15}$$

$$\Psi_{3,x} = \frac{1}{(2\pi)^2}\int_{\omega_1}\int_{\omega_2} S_x(\boldsymbol{\omega})\omega_1\omega_2 d\omega_1 d\omega_2 \tag{4.16}$$

represent three motion sensitivity factors. One property can be observed from the formulation of Ψ. The squared terms of the frequency variables, ω, inside the integrals reflect the strong dependency between motion sensitivity factors and high frequency energy contents in the reference picture. Intuitively, spatial edges are affected by motion errors much more than smooth spatial regions.

Quantization can also affect the power spectrum of the reference picture. In hybrid video coding, the reference picture is chosen from the reconstructed picture buffer. At low bit rates, high frequency components are highly quantized, which results in a smaller Ψ. In other words, low bit rate sequences suffer less from the motion error than the high bit rate sequences.

By expressing $\boldsymbol{\delta}$ in polar form, we can rewrite D_y in (4.13) as a function of $||\boldsymbol{\delta}||^2$ and $\theta_{\boldsymbol{\delta}}$.

$$\begin{aligned} D_y &\approx \left(\Psi_{1,x}\cos^2(\theta_{\boldsymbol{\delta}}) + \Psi_{2,x}\sin^2(\theta_{\boldsymbol{\delta}}) + \Psi_{3,x}\cos(\theta_{\boldsymbol{\delta}})\sin(\theta_{\boldsymbol{\delta}})\right)||\boldsymbol{\delta}||^2 \\ &\triangleq \Psi_x(\theta_{\boldsymbol{\delta}})||\boldsymbol{\delta}||^2 \end{aligned} \tag{4.17}$$

Since natural images often exhibit roughly isotropic power spectra, the average motion sensitivity, Ψ_x, which is obtained by averaging $\Psi_x(\theta_{\boldsymbol{\delta}})$ over all $\theta_{\boldsymbol{\delta}}$, should be adequate to approximate the original $\Psi_x(\theta_{\boldsymbol{\delta}})$.

$$\Psi_x = \frac{1}{2\pi}\int_{-\pi}^{\pi}\Psi_x(\theta_{\boldsymbol{\delta}})d\theta_{\boldsymbol{\delta}} = \frac{\Psi_{1,x}+\Psi_{2,x}}{2} \tag{4.18}$$

To summarize, under the assumption of small and constant motion error, $\boldsymbol{\delta}$, we have a linear distortion model as follows.

$$D_y \approx \Psi_x||\boldsymbol{\delta}||^2, \tag{4.19}$$

where Ψ_x is an isotropic motion sensitivity factor, which depends on the actual energy spectral density of the reference picture. In order to improve the accuracy of the linear model, the higher order terms in (4.10) can be taken into consideration. For example, the quadratic distortion model derived by including two terms in (4.10) is shown below.

$$D_y \approx \Psi_x||\boldsymbol{\delta}||^2 - \left(\frac{\Phi_{1,x}+\Phi_{2,x}}{24}\right)||\boldsymbol{\delta}||^4, \tag{4.20}$$

where

$$\Phi_{1,x} = \frac{1}{(2\pi)^2}\int_{\omega_1}\int_{\omega_2} S_x(\boldsymbol{\omega})\omega_1^4 d\omega_1 d\omega_2 \tag{4.21}$$

$$\Phi_{2,x} = \frac{1}{(2\pi)^2}\int_{\omega_1}\int_{\omega_2} S_x(\boldsymbol{\omega})\omega_2^4 d\omega_1 d\omega_2 \tag{4.22}$$

The comparison of experimental results with both the linear and the quadratic distortion models, using the first frame of the MOBILE sequence, is shown in Fig. 4.2. Two observations on the effect of increased quantization can be made. First, the accuracy of the linear distortion model, as an approximation to the actual experimental results, increases for heavier quantization. Second, the motion sensitivity factor decreases as quantization increases. Both of these results are as expected. In addition, the quadratic model fits the actual results better than the linear model, although the improvement is fairly limited.

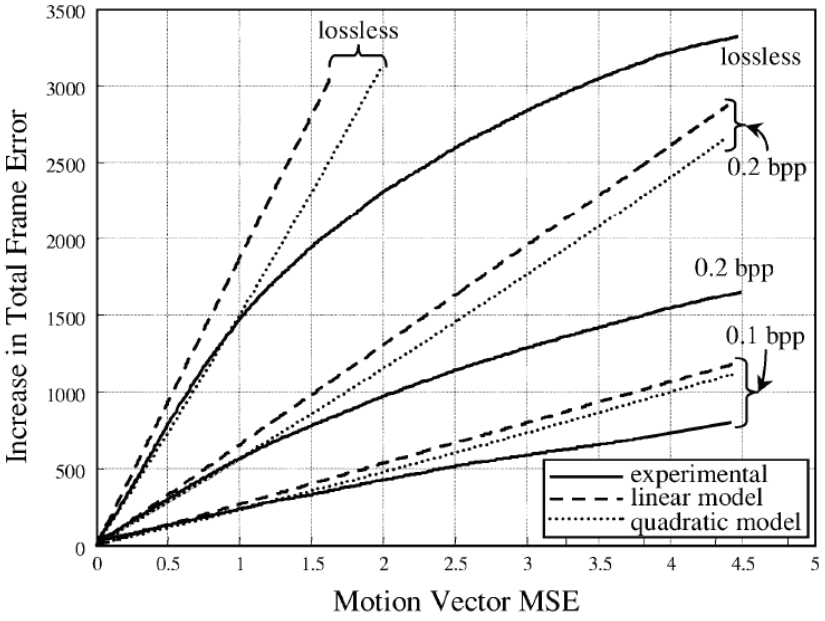


Figure 4.2: Distortion as a function of motion errors. Linear and quadratic models are compared with experimental results.

With the help of the linear distortion model, it is easier to evaluate the left hand side of (4.8). Specifically, the criterion for scalable motion to be beneficial

can be simplified as follows.

$$\Psi_x ||\boldsymbol{\delta}||^2 < D_{t,1} - D_{t,2} \tag{4.23}$$

where $D_{t,i} \triangleq ||T - Q_i(T)||^2$ denotes the texture distortion introduced by quantizer i.

4.4 Acknowledgement

Portions of this chapter appear in "A Fully Scalable Motion Model for Scalable Video Coding", Meng-Ping Kao and Truong Nguyen, in the June 2008 issue of the *IEEE Transactions on Image Processing*; and also in "A Fully Scalable Motion Model for Scalable Video Coding", Meng-Ping Kao and Truong Nguyen, in Proceedings of the *IEEE International Conference on Image Processing*, Sep. 2007. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.

5 Proposed Block-Based Scalable Motion Model

Considering the required functionalities of scalable motion as illustrated in Section 4.2, we propose a novel and fully scalable motion model in this chapter. The proposed SMM is based on block motion and is best suitable for SVC frameworks with Laplacian pyramid realization for spatial scalability, e.g. 2D+t+2D WSVC or the SVC standard.

The basic cell of the proposed SMM is a macroblock. Fig. 5.1 shows the structure of this basic cell. It is clear that we explicitly implement both refining methods for motion scalability in our model, i.e. the MV accuracy dimension and the VBS dimension along horizontal and vertical axes, respectively.

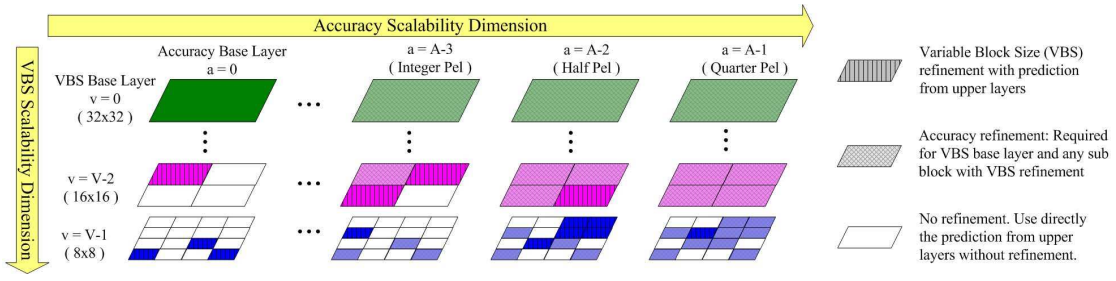


Figure 5.1: Proposed fully scalable motion model.

In the following sections, we begin with illustrating the algorithm that adapts the motion parameters in our SMM to different decoding scenarios, i.e. different combinations of temporal, spatial, and quality specifications. This is enabled by further exploring the properties of both the MV accuracy and the VBS

dimensions.

The RDO ME algorithm will then be discussed in Section 5.2, which provides the tool for the encoder to select the most suitable motion parameters for various target operating bit rates, as well as different resolutions.

Accompanied by the RDO ME algorithm is the coding method for the parameters used in our SMM. This is crucial for the complete RDO ME process since without the coding algorithm, the rate cost for encoding the motion parameters can not be determined. Moreover, the coding algorithm requires further elaborations to minimize the overhead resulting from carrying additional scalabilities. Detailed descriptions will be given on both VBS structural coding and MV accuracy refinement coding in Section 5.3.

Due to the coarse grain quality scalability of scalable motion, decoding bit rates might reside between two predefined operating points adopted by the encoder. In this case, optimal decoding involves determination of the motion quality layer that gives the best decoding quality. This is the main task of the bitstream extractor that is enabled with scalable motion. This topic will be covered in Section 5.4.

In the end of this chapter, we demonstrate the rate distortion efficiency of the proposed SMM with extensive experiments. Various testing sequences, as well as miscellaneous decoding setups, will be taken into account.

5.1 Dimension of Scalable Motion

According to the design shown in Fig. 5.1, quality scalability can be easily realized by choosing the decoding levels on each dimension. Lower decoding levels correspond to lower quality MVFs.

Even though the proposed model provides two dimensions to fine tune the motion quality, the best motion for a target bit rate remains unique. This is ideally determined by trying all possible refinements (based on the previous motion quality layer) on each dimension and finding the combination with the lowest residual energy under the target bit rate constraint. This method is highly intractable due

to the enormous number of combinations to be estimated.

On the other hand, by introducing the Lagrange multiplier, λ, associated with the target bit rate, any refinement with a higher slope (distortion reduction to rate ratio) than λ should be included in the current motion quality layer. Note that this approach is only a suboptimal solution due to the greedy algorithm along a specific search order. A different search order might lead to a different solution. Also note that the rate measurement plays an important role in the estimation process. Since the rate overhead occurs on both refined and non-refined information (to be illustrated in Section 5.3), the freedom to choose refinements on both the accuracy and the VBS dimensions might also hurt the total efficiency if a better coding method dealing with those overheads is not available.

One possible way to further simplify the estimation problem is to associate either the MV accuracy or the VBS dimension with the motion quality layers, and let the rate distortion motion estimation determine the best parameters for the other dimension. For example, if the accuracy dimension is chosen as the motion quality layers, each accuracy level is assigned a λ according to its target bit rate. No refinements at higher accuracy levels are allowed in the current motion quality layer with λ. On the other hand, the refinements on the VBS dimension are purely determined by the RD motion estimation. A refinement at the highest VBS level might be included in the lowest motion quality layer if a better RD curve is observed. This simplification reduces the estimation problem from 2D to 1D and saves some rate overhead on the reduced dimension.

The next question is which dimension, as a motion quality layer, is better for the coding efficiency. To answer this question, an experiment is performed to compare the motion compensation performances of these two candidates. Fig. 5.2 shows the PSNR of the motion compensated frame versus the bits required to describe the motion parameters for various video sequences. A curve towards the top-left corner indicates a better coding performance. As observed from the figure, the accuracy dimension outperforms the VBS dimension throughout all testing sequences. Moreover, with the aid of the proposed SMM structure coding algorithm (to be described in Section 5.3), the motion bits can be further reduced

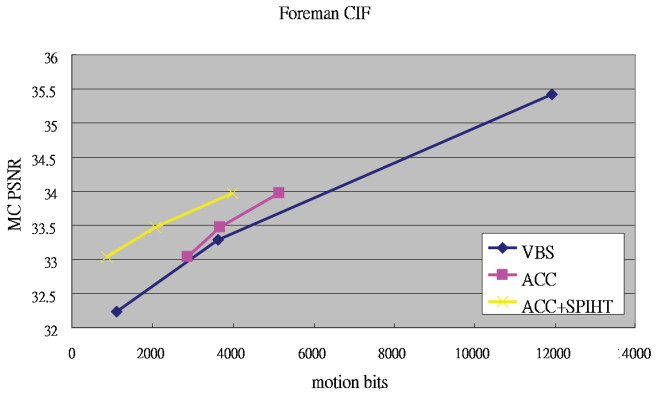
for the same MC PSNR. Note that the dynamic range of the motion bits can be adjusted by changing λ and the number of motion quality layers is set to three in this experiment.

The idea behind choosing the accuracy dimension as quality layers as opposed to the VBS dimension is that the VBS structure is more content dependent and it should be optimized to the underlying motion via RDO ME. It is, however, not to say that the MV accuracy is not content dependent. The MV accuracy suffers less from not being optimized to underlying motion than the VBS structure. If only one can be chosen so as to reduce estimation complexity and save rate overhead, the MV accuracy would be the better choice.

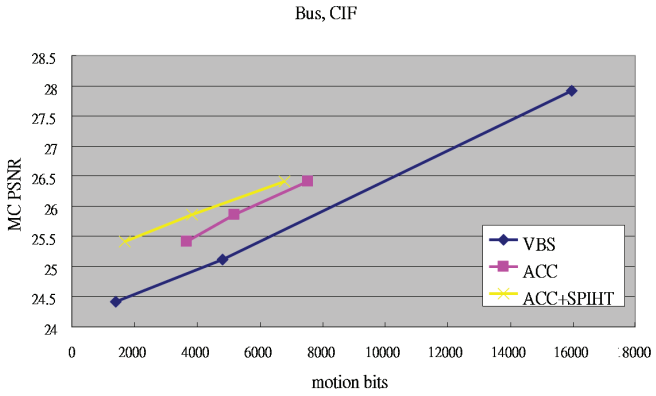
As far as VBS scalability is concerned, all the internal nodes in the tree structure should be determined and encoded for possible decoding purposes, as well as the leaf nodes. An example for quadtree structure can be referred to in [42]. To further increase the coding efficiency, an incomplete quadtree structure is adopted in our SMM as shown in Fig. 5.3. A considerable amount of bits can be saved when some of the leaf nodes have similar MVs to that of their parent node. The decision process is rate distortion optimized as we will discuss in Section 5.2.

Some notations must be clarified before further description of our SMM can proceed.

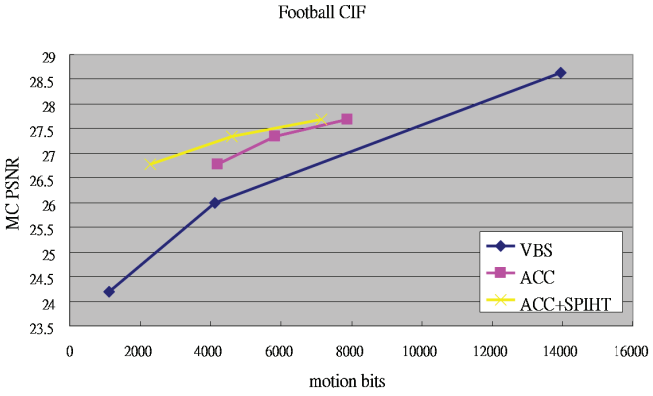
- A: Number of MV accuracy levels. $A \geq 1, A \in \mathbb{Z}$.
- a: Index of a MV accuracy level. $0 \leq a < A, a \in \mathbb{Z}$. The MV accuracy base level is denoted by $a = 0$.
- λ_a: Rate multipliers for accuracy level a. In general, $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{A-1}$.
- V: Number of VBS levels. $V \geq 1, V \in \mathbb{Z}$.
- v: Index of a VBS level. $0 \leq v < V, v \in \mathbb{Z}$. The VBS base level is denoted by $v = 0$, which has the largest block size.
- R: Number of resolution levels. $R \geq 1, R \in \mathbb{Z}$.



(a)



(b)



(c)

Figure 5.2: Motion rate distortion curve. (a) FOREMAN. (b) BUS. (c) FOOTBALL.

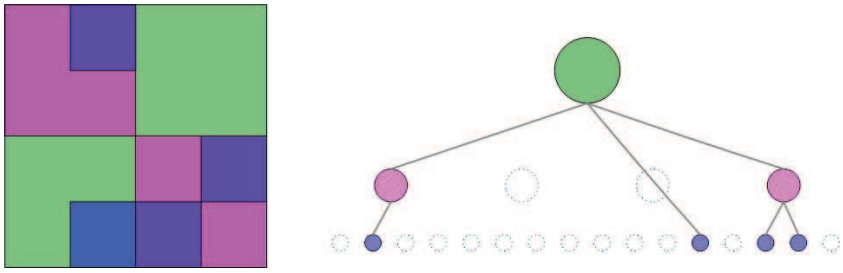


Figure 5.3: Incomplete quadtree structure.

- r: Index of a resolution level. $0 \leq r < R, r \in \mathbb{Z}$. The largest picture is denoted by $r = 0$.

- w_r: Distortion multipliers for resolution level r.

- I: Actual MV accuracy offset. $I \in \mathbb{Z}$. The actual MV accuracy (in pixels) for resolution r and accuracy level a is denoted by $2^{-(I+a+r)}$. For example, $I = 0$ is equivalent to integer pixel accuracy and $I = 1$ is equivalent to half pixel accuracy for $a = r = 0$. This parameter provides an absolute scale for the MV accuracy. In general, $I = 0$, i.e. integer pixel accuracy for the MV accuracy base level in the largest sized picture, is assumed in the sequel.

- (v, i, j): Coordinate of the quadtree node (subblock) at VBS level v and indexed by (i, j), where $0 \leq i, j < 2^v, i, j \in \mathbb{Z}$ denote the vertical and horizontal indices respectively. Note that the root node is denoted by $(0, 0, 0)$. Each subblock has its own unique coordinate within a MB.

The proposed SMM shown in Fig. 5.1 is for resolution $r = 0$ only, i.e. it contains ready-to-use motion information for the largest resolution sequence. The descriptions in parenthesis are examples for practical applications. In this example, the SMM supports up to quarter pixel accuracy MV and the block size may range from 32x32 to 8x8. If MVFs for smaller size sequences are requested, i.e. $r > 0$, a proper down-scaling process is required. For example, "Integer Pel" means that it is the integer pixel accuracy level for $r = 0$, which should be the half pixel accuracy layer for $r = 1$, and so on. Similarly, "32x32" means

that the block size is 32x32 for $r = 0$, which should be 16x16 for $r = 1$ and, so on. Therefore, when decoding smaller sized pictures, irrelevant/unsupported information should be discarded for maintaining high coding efficiency. This is accomplished by applying two constraints on the accuracy and the VBS dimensions as follows. For resolution r, the highest accuracy level a_r and the highest VBS level v_r are

$$a_r = A - 1 - r \tag{5.1}$$

$$v_r = V - 1 - r \tag{5.2}$$

respectively. By highest levels a_r and v_r we mean the codec does not support more refined levels for the given resolution r. In other words, quality scalability of the SMM can only operate within the range $a = 0, \cdots, a_r$ and $v = 0, \cdots, v_r$, i.e. part of the top-left corner in the SMM structure.

Knowing the constraints posed by spatial scalability, we are ready to move on to quality scalability, which is the most important part of the SMM. In our SMM, every accuracy level a is associated with a target motion bit rate and is optimized to that bit rate through the RDO ME process to be discussed in the next section. By increasing the total number of accuracy levels A, we can achieve FGS gradually. For example, given resolution r, the decodable MVF quality can be lowest by choosing $a = 0$, and can be progressively improved up to $a = a_r$.

As mentioned earlier, the motion quality is determined by a, not v. The VBS structure is optimized to a certain motion rate implied by a. Reducing the motion rate by choosing a smaller v would result in a non-optimal MVF. Instead, optimal MVFs can always be acquired to meet the motion rate requirement by changing a. The only exception is when the decoder asks for a smaller bit rate than the base accuracy level $a = 0$ can provide. Only in this case shall $v < v_r$ be allowed.

However, this is not to say VBS scalability has no control on motion quality. Instead, it comes in great effect in a more implicit way. An increasing bit budget for the motion model could result in a more refined motion structure. This is the reason why the incomplete quadtree structure keeps growing as a increases, which can be observed from our example in Fig. 5.1.

The final note on our proposed block-based SMM is that it can be easily applied to multiple hypothesis motion models. The information indicating whether the current MB is intra or inter coded (if inter coded, whether it is P or B MB) is encoded in the MB header. For inter MB, the corresponding reference index should be provided. In the case where B MB is present, two reference indices along with two jointly estimated SMM's are encoded. For decoding, predictions from each SMM (with different reference pictures) with the assigned motion quality layer are first derived. A weighted average of these two predictions is adopted as the final prediction of the current B MB. An obvious limitation of the above scheme is that all information in a MB-based SMM has to come from a single reference picture. No multiple references are allowed within a P MB.

5.2 Rate Distortion Optimized Motion Estimation

A motion model without the corresponding rate distortion optimization algorithm can not achieve the best coding efficiency. It is the encoder that has full access to the original video sequence and thus should be responsible for providing the best motion information that suits the decoder's requirements. Therefore, given all possible decoding scenarios, a good RDO strategy at the encoder can usually outperform a good standalone bitstream extractor at the decoder.

In our proposed SMM, the RDO is performed in the basis of subblocks and the scanning order is shown in Fig. 5.4 for an example of a structure with three VBS layers. As observed from Fig. 5.4, the scanning order is from the top layer, $v = 0$, to the bottom layer, $v = V - 1$, with a raster scan in group of four subblocks within the same VBS layer.

For each subblock indexed by (v, i, j), our goal is to determine the best scalable motion vector (SMV) in a RD sense. The SMV is predictively coded and thus composed of two parts: motion vector prediction (MVP) and motion vector

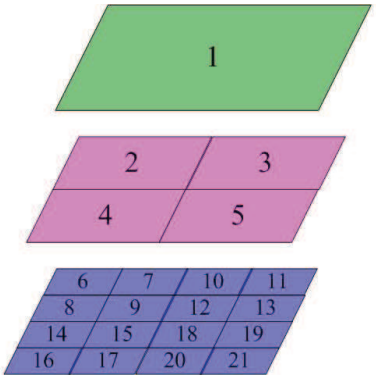


Figure 5.4: RDO ME scanning order.

difference (MVD).

$$SMV(v,i,j,r,a) = MVP(v,i,j,r) + MVD(v,i,j,r,a) \tag{5.3}$$

Note that both SMV and MVD are functions of subblock position (v,i,j) as well as spatial resolution r and accuracy level a, while MVP is independent of a. The reason is that all MVPs are obtained only from the accuracy base level $a = 0$ to ensure SMVs can be decoded under all situations.

The root node of each MB is predicted from its available neighboring MBs to its left, top, and top-right. The prediction is obtained by taking the median value independently on both x and y components.

$$\begin{aligned} &MVP(0,0,0,r) \\ &= median(SMV_L(0,0,0,r,0), SMV_U(0,0,0,r,0), SMV_{UR}(0,0,0,r,0)) \end{aligned} \tag{5.4}$$

In the case one or more of these three neighboring MBs is unavailable, the following rule applies.

$$\begin{aligned} &MVP(0,0,0,r) \\ &= \begin{cases} \mathbf{0} & \text{if none is available,} \\ SMV_i(0,0,0,r,0) & \text{if one is available,} \\ \left(SMV_i(0,0,0,r,0) + SMV_j(0,0,0,r,0)\right)/2 & \text{if two are available.} \end{cases} \end{aligned} \tag{5.5}$$

For nodes other than the root node, i.e. $v > 0$, a prediction is obtained

directly from their parent nodes.

$$\begin{aligned} MVP(v,i,j,r) &= SMV(v-1, i \gg 1, j \gg 1, r, 0) \\ &= MVP(0,0,0,r) + \sum_{v'=0}^{v-1} MVD(v', i \gg (v-v'), j \gg (v-v'), r, 0) \end{aligned} \tag{5.6}$$

where $\gg$ is the bitwise right shift operation.

For simplicity, the common parameter (v, i, j) will be omitted from now on. Different $MVD(r, a)$ are obtained from a unique and progressively encoded scalable motion vector difference $SMVD$ via the parameters r and a.

$$SMVD = (s, \mathit{ref}(0), \mathit{ref}(1), \cdots, \mathit{ref}(A-1)), \tag{5.7}$$

where $-1 \leq s < A$ denotes the starting/minimal accuracy level for the current $SMVD$ to be decoded into $MVD(r, a)$. For any $a < s$, $MVD(r, a) = \mathbf{0}$ where $\mathbf{0} \triangleq (0, 0)$. Moreover, $s = -1$ is reserved for the case where no $SMVD$ is provided. In this case, $MVD(r, a) = \mathbf{0}$ for all r, a.

$\mathit{ref}(a), a = 0, \cdots, A-1$ are the refinement vectors for level a. The possible range of $\mathit{ref}(a)$ is defined as follows.

$$\mathit{ref}(a) \in \begin{cases} \{(x,y) | x, y \in \mathbb{Z}\} & \text{if } a = 0, \\ \{(x,y) | x, y \in \{-1, 0, 1\}\} & \text{if } a > 0. \end{cases} \tag{5.8}$$

$\mathit{ref}(0)$ is the first SMVD refinement, which in general has no boundary limitation but is practically bounded by the ME search range. Considering the distribution of a MVD, an Exp-Golomb encoder is usually used to encode $\mathit{ref}(0)$.

$\mathit{ref}(a), a > 0$ is the following SMVD refinement for level a. It is limited within the nearest eight neighbors, or simply no refinement, i.e. a total of nine candidates as indicated in (5.8).

These refinement vectors all have integer entries such that an encoding codebook can be easily designed to achieve further compression. They can be mapped back to the actual 2D MVs through the following function.

$$T(\mathit{ref}(a)) = 2^{-(I+a)} \mathit{ref}(a) \tag{5.9}$$

As a final remark on $T(\cdot)$, any zero *ref*(a) will also result in a zero MV, i.e.

$$T(\mathbf{0}) = \mathbf{0}. \tag{5.10}$$

Knowing the actual 2D MVs, $T($*ref*$(a))$, the MVD for resolution r up to accuracy level a can be obtained in the following manner.

$$MVD(r,a) = \begin{cases} \mathbf{0} & \text{if } a_r < s \text{ or } s = -1, \\ 2^{-r} \sum_{i=0}^{\min(a,a_r)} T(\textit{ref}(i)) & \text{otherwise.} \end{cases} \tag{5.11}$$

While (5.11) provides a clear formula for decoding an SMVD to a desired $MVD(r,a)$, a simple example might further help us to understand the practical meaning on each entry of the SMVD. Suppose we have three spatial resolutions (R=3), i.e. 4CIF, CIF, and QCIF, three MV accuracy levels (A=3), i.e. integer, half, and quarter pixel, and an SMVD expressed explicitly as $(1,(5,3),(1,0),(-1,1))$. The full accuracy MVD for 4CIF resolution is $MVD(0,2) = (5,3) + (0.5,0) + (-0.25,0.25) = (5.25,3.25)$. On the other hand, the full accuracy MVD for CIF resolution is $MVD(1,2) = ((5,3) + (0.5,0))/2 = (2.75,1.5)$. Note that when a corresponding scaling operation for smaller resolution is applied, the last entry of SMVD, i.e. (-1,1), has virtually no effect on CIF resolution. In fact, we have $MVD(1,2) = MVD(1,1)$ in this particular situation. To further explore the impact of the first entry s, we now decode the least accurate MVD for QCIF resolution, i.e. $MVD(2,0) = (0,0)$. Note that although the integer pixel entry is supposed to be valid for QCIF resolution, which should have resulted in a MVD of $(5,3)/4 = (1.25,0.75)$, the first accuracy level entry $s = 1$ prevents this from happening. As a matter of fact, any resolution smaller than CIF, i.e. $\{r|a_r < 1\}$, gets virtually zero information from this particular SMVD with $s = 1$. The benefit of a larger s is to force the SMVD to be invisible for smaller a_r, in return for the possibility to encode this SMVD in a lower penalty zone. The details will be elaborated in Section 5.3.

In order to obtain the optimal SMVD, a new cost function will be introduced to serve as the minimization criterion in the RDO ME process. This new cost function consists of distortion and rate measures, which are similar to the

conventional RDO ME process [68]. However, both measures are modified to accommodate the scalable features of SVC, including various operating bit rates and decoding resolutions. From the notation mentioned earlier, we have a set of rate multipliers $\{\lambda_a | a = 0, \cdots, A-1\}$ such that each accuracy level a will be weighted by the penalty multiplier λ_a. In general, a smaller a corresponds to a lower decoding bit rate, which in turn requires a larger penalty multiplier λ_a. Therefore, we usually have the relationship $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{A-1}$.

Note that the actual determination process for optimal $\{\lambda_a\}$ has to go through a similar training process as in [80]. The ideal outcome is expected to be a regression function with a rate allocation table, e.g. Table 5.5, as the input and $\{\lambda_a\}$ as the output. However, the details are beyond the scope of this paper. The capability of the proposed SMM so far is to provide a tool for SVC encoder optimization. There are still some parameters, e.g. $\{\lambda_a\}$, to be fine tuned to achieve true optimization. In our simulations, $\{\lambda_a\}$ are experimentally and optimally determined for each testing sequence under a certain bit rate allocation table.

The new rate function, assuming $\{\lambda_a\}$ is given, is defined as,

$$\begin{aligned} RF(SMVD) &= RF((s, ref(0), ref(1), \cdots, ref(A-1))) \\ &= \begin{cases} \lambda_0 R(s) + \lambda_s R(ref(0)) + \sum_{a=1}^{A-1} \lambda_{\max(a,s)} R(ref(a)) & \text{if } s \neq -1, \\ \lambda_0 R(s) & \text{if } s = -1. \end{cases} \end{aligned} \tag{5.12}$$

where $R(\cdot)$ is the function that returns the actual coding bits of each component in $SMVD$. $R(\cdot)$ is determined by the $SMVD$ codebook designing method to be discussed in the next section.

Moreover, since the single SMVD would provide MVs for all possible resolutions, the total distortion function should also be a combination from all possible resolutions. The individual distortion function from each resolution is weighted by $\{w_r | r = 0, \cdots, R-1\}$. Note that unlike λ_a, w_r has no conventional restrictions on its relative values. Instead, w_r is determined according to the decoder's preference. If the decoder prefers better coding efficiency for a specific resolution r, w_r should

be chosen relatively larger than the others, and vice versa. Given the set $\{w_r\}$, a new distortion function is defined as

$$\begin{aligned} DF(SMVD) &= DF((s, ref(0), ref(1), \cdots, ref(A-1))) \\ &= \sum_{r=0}^{R-1} w_r D_r(SMV(r, A-1)) \end{aligned} \tag{5.13}$$

where $D_r(\cdot)$ is the function that returns the distortion measurement when $SMV(r, a)$ is used in the MC process for resolution r. The distortion can be chosen as either the reconstruction error for optimality or the MC error for a better tradeoff between complexity and performance [68]. The possible measurements include sum of absolute difference (SAD), sum of square difference (SSD), and reconstruction square error (RSE) [51] for MCTF-based SVC. Note that in our following experiments, the SAD metric of the MC error is adopted for reduced complexity.

Combining (5.12) and (5.13), the new cost function can be defined as follows:

$$\begin{aligned} CF\,(SMVD) =& RF\,(SMVD) + DF\,(SMVD) \\ =& \lambda_0 R\,(s) + \lambda_s R\,(Ref\,(0)) + \sum_{a=1}^{A-1} \lambda_{\max(a,s)} R\,(Ref\,(a)) \\ &+ \sum_{r=0}^{R-1} w_r D_r\,(SMV\,(r, A-1)) \end{aligned} \tag{5.14}$$

The RDO process for finding the best SMVD begins with determining the optimal pair $(s, ref(0))$. s and $ref(0)$ should be jointly optimized to minimize (5.14). The detailed pseudo-code is listed in Algorithm 1.

If the best s turns out to be -1, all refinements $\{ref(a)\}$ are automatically set to $\mathbf{0}$ and this completes the RDO ME process. If not, since the optimal pair $(s, ref(0))$ is already determined, the RDO ME process proceeds by tracing through $a = 1, \cdots, A-1$ to find the remaining $ref(a)$. $ref(a), a = 1, \cdots, A-1$ are sequentially determined according to the following rule.

$$ref(a) = \arg\min_{\mathbf{x}}(CF(s, ref(0), \cdots, ref(a-1), \mathbf{x}, \mathbf{0}, \cdots, \mathbf{0})) \tag{5.15}$$

Algorithm 1 Finding $(s, ref(0))$

for $s = 0$ to $A - 1$ **do**
  $ref(0) = \arg\min_{\mathbf{x}} CF(s, \mathbf{x}, \mathbf{0}, \cdots, \mathbf{0})$
  if $ref(0) \neq \mathbf{0}$ **then**
    break for loop
  else if $s = A - 1$ **then**
    $(s, ref(0)) = (-1, \mathbf{0})$
  end if
end for

5.3 Coding of Scalable Motion Model

The objective of the proposed SMM is to progressively optimize the coding efficiency along the rate distortion curve. So far, the RDO ME algorithm for finding the best $SMVD$ has been illustrated in the previous section. However, the function $R(\cdot)$ which depends on the $SMVD$ codebook design is still undetermined.

In this section, we will investigate the coding techniques for the proposed SMM. To further illustrate the individual properties for improved coding efficiency, three subtopics will be discussed: 1) SMM structure, 2) first SMVD refinement $ref(0)$, and 3) following SMVD refinements $ref(a), a = 1, \cdots, A - 1$. For each topic, the properties that contain compression potentials will be examined and optimal/sub-optimal coding methods will be proposed accordingly.

5.3.1 Structure Coding

In our SMM, VBS scalability is realized via the incomplete quadtree structure as shown previously in Fig. 5.3. At the same time, the VBS structure may also evolve as the bit rate increases. For example, a block that does not need a motion vector in low bit rate regimes might be benefited from one at a higher bit rate. One straightforward way to explicitly describe this relationship might be a label associated with each SMVD that indicates from which accuracy level this SMVD will be decodable. This is exactly the first component, s, in our SMVD rep-

resentation shown in (5.7). For example, $s = 0$ indicates that the current SMVD is decodable from the very first accuracy level, while $s = 1$ is only decodable from the second level. There will be no refinement at all for a subblock with $s = 1$ if the decoder decides to stop decoding after the first accuracy level. However, note that $s = 1$ does not mean there will be no $ref(0)$; $ref(0)$ will be encoded at level $a = 1$ together with $ref(1)$. As far as compression is concerned, if all the s within a MB are encoded without further processing, a considerable amount of bits will be consumed. Assuming there are three accuracy levels, a two-bit codeword is inevitable for each s (four possible values for $-1 \leq s < 3$). This would cause a huge burden especially for low bit rate scenarios.

To solve the problem, we introduce the zero-tree bit plane coding strategy to encode the SMM structure. Specifically, our accuracy levels, a, can be viewed as the bit planes in zero-tree coding. Starting from the first accuracy level $a = 0$, we trace through all subblocks to find their significance. As usual, once an SMVD is found significant, it will be put into the significant list and remain significant for the remaining sorting passes.

A deviation from the original zero-tree bit plane coding happens immediately after an SMVD is found significant. Instead of coding the sign bit of the current coefficient, $\{ref(i)|0 \leq i \leq a\}$ will be coded. However, this operation, named the outset process, will wait until the entire sorting pass for level a is completed. The idea behind delaying the outset process is to separate the SMVD structure bits from the remaining SMM refinement bits. The isolation of the SMM structure bits can lead to a more efficient design for coding the SMM structure itself.

The specific sorting method we adopt here is borrowed from Set Partitioning in Hierarchical Trees (SPIHT) proposed by Said and Pearlman [59]. This method takes advantage of the sparsity of significance bits to be coded. By introducing the zero-tree structure and the set partition rule, the significance of a tree structure can be represented using as few as one bit. This property suits our need very well since RDO motion estimation at low bit rates gives a sparse MVF, which in turn requires very few bits to encode according to SPIHT. As the bit budget

increases, the resulting MVF becomes less sparse, which reduces the efficiency of SPIHT. However, the rate concern is not as important as distortion in the high rate regime.

In the sorting pass, the list of insignificant vectors (LIV) is first traced through, followed by the list of insignificant sets (LIS). Since the operating point on the RD curve shifts slowly from one quality level to another, only a few elements in the list become significant for each pass. Knowing this property, we apply an additional run length coding (RLC) on the significance bits from LIV. This will help further compress the zero dominant significance bitstream. We list the RLC decoding details in Table 5.1. Note that since the length of LIS is known, the bits required for coding the position of "1" are either $floor(\log_2 N)$ or $ceil(\log_2 N)$.

Table 5.1: RLC decoding rules for LIV

LIV length (N)	First bit	Second bit	Decoding operations
0, 1, 2			No RLC. Read following N bits.
3	0		Padding with all zeros.
	1		Read following N-1 bits. If all zeros, append one at the end. If not, read one more bit.
4 or more	0		Padding with all zeros.
	1		Read up to $ceil(\log_2 N)$ following bits to determine the position of one.
	1	1	Read following N-2 bits. If all zeros, append two ones at the end. If not, read one more bit. If total number of ones so far is one, append one at the end. If not, read one more bit.

Before introducing the complete SMM structure coding algorithm, let us first define the notation below.

- $O(v, i, j)$: Set of coordinates of all offspring of node (v, i, j).
- $D(v, i, j)$: Set of coordinates of all descendants of node (v, i, j).

- $L(v,i,j)$: $D(v,i,j) - O(v,i,j)$.
- $s(v,i,j)$: s component of $SMVD(v,i,j)$.
- $S_a(v,i,j)$: Significance of node (v,i,j) at level a.
- *ref*(v,i,j,a): *ref*(a) component of $SMVD(v,i,j)$.
- LIV: List of insignificant vectors.
- LIS: List of insignificant sets.
- LSV: List of significant vectors.
- SigLIV: Bit string recording the significance bits from scanning through LIV.
- SigLIS: Bit string recording the significance bits from scanning through LIS.

Note that the significance is defined as,

$$S_a(v,i,j) = \begin{cases} 1 & \text{if } 0 \leq s(v,i,j) \leq a, \\ 0 & \text{otherwise.} \end{cases} \tag{5.16}$$

The pseudo-code for the SMM structure coding algorithm is listed in Algorithm 2.

In summary, the original SPIHT algorithm is modified to encode the SMM structure in a progressive manner. The extraction of significance bits helps to further improve the coding efficiency of the SMM structure.

5.3.2 Precision Coding

The first refinement *ref*(0), as mentioned earlier in (5.8), is encoded using the Exp-Golomb code. The horizontal and vertical components will be encoded separately by a signed Exp-Golomb encoder and the resulting codewords are concatenated to give a single output. Note that the same MVD encoder is also used in H.264/AVC [81]. It is statistically optimal when the magnitude of MVD is exponentially distributed. Note also that every $SMVD(v,i,j)$ with $s(v,i,j) \neq -1$ will have a non-zero *ref*(0), which is a necessary consequence from Algorithm 1.

Algorithm 2 SMM Structure Bit Plane Coding

♦ ***Initialization***

Set LIV, LIS, and LSV as empty lists. Set $a = 0$.

Add $(0, 0, 0)$ to LSV. Add $(0, 0, 0)$ to LIS as type A.

Output *ref*$(0, 0, 0, 0)$.

♦ ***Sorting Pass***

Clear SigLIV.

for all (v, i, j) in LIV **do**

  Save $S_a(v, i, j)$ to SigLIV.

  if $S_a(v, i, j) = 1$ **then**

    move (v, i, j) to LSV.

  end if

end for

Output SigLIV using Table 5.1.

for all newly added (v, i, j) in LSV **do**

  Output *ref*$(v, i, j, a'), a' = 0, \cdots, a$.

end for

Clear SigLIS.

for all (v, i, j) in LIS **do**

  if type A **then**

    Save $S_a(D(v, i, j))$ to SigLIS.

    if $S_a(D(v, i, j)) = 1$ **then**

      for all $(l, m, n) \in O(v, i, j)$ **do**

        Save $S_a(l, m, n)$ to SigLIS.

        if $S_a(l, m, n)$=1 **then**

          Add (l, m, n) to LSV.

        end if

        if $S_a(l, m, n)$=0 **then**

          Add (l, m, n) to LIV.

        end if

---

end for
if $L(v,i,j) \neq \emptyset$ **then**
move (v,i,j) to the end of LIS as type B.
else
remove (v,i,j) from LIS.
end if
end if
end if
if type B **then**
Save $S_a(L(v,i,j))$ to SigLIS.
if $S_a(L(v,i,j))$=1 **then**
Add all $(l,m,n) \in O(v,i,j)$ to the end of LIS as type A.
Remove (v,i,j) from LIS.
end if
end if
end for
Output SigLIS.
for all newly added (v,i,j) in LSV **do**
Output *ref*$(v,i,j,a'), a' = 0, \cdots, a$.
end for

♦ ***Refinement Pass***
for all (v,i,j) in LSV except for those newly added in the last sorting pass **do**
Output *ref*(v,i,j,a).
end for

♦ ***Accuracy Level Update***
if $a < A-1$ **then**
$a = a+1$. Go to ***Sorting Pass***.
end if

The remaining refinements $ref(a), a = 1, \cdots, A-1$, usually the subpixel refinements ($I = 0$), are the last part of SMVD. These refinements contain a series of vectors representing the progressive improvements of SMVD accuracy, i.e. $a = 1, \cdots, A-1$. Practically speaking, we have half-pixel, quarter-pixel, and so on.

1D Case – Optimal Codebook Design Without RDO

Our main purpose here is to develop an optimal coding algorithm for these refinements. For simplicity, we first consider the case where RDO is turned off during ME, i.e. the best $ref(a)$ is chosen to minimize the distortion function only. Moreover, only one component of MV is considered here to further reduce the dimension of this problem. In this regard, an example can be depicted in Fig. 5.5, where the red star denotes the best MV one can ever achieve with infinite accuracy.

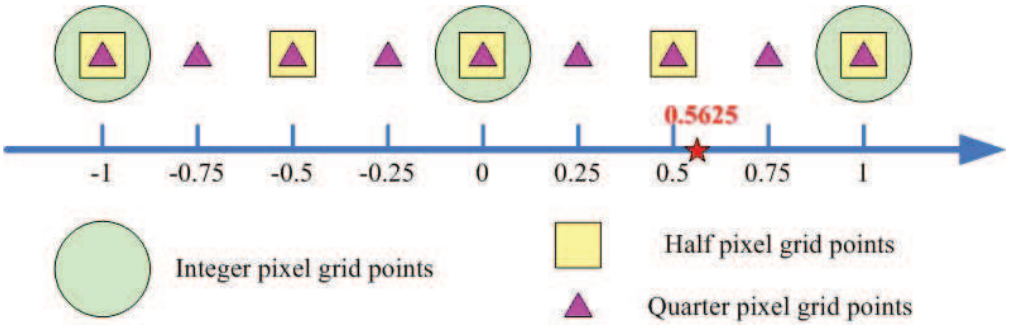


Figure 5.5: 1D SMV refinement constellation.

Assuming $r = 0$, the best motion vector with only the first accuracy refinement is estimated as,

$$SMV_x(0,0) = \arg \min_{2^I d \in \mathbb{Z}} \left\{ \sum_n E^2(n;d) \right\} \tag{5.17}$$

where $E(n;d)$ is defined as (3.9). Similarly, the best MV with refinements up to accuracy level a is

$$SMV_x(0,a) = \arg \min_{2^{(I+a)} d \in \mathbb{Z}} \left\{ \sum_n E^2(n;d) \right\} \tag{5.18}$$

Therefore, the refinements to be encoded can be represented as

$$ref_x(a) = T^{-1}\left(SMV_x\left(0,a\right) - SMV_x\left(0,a-1\right)\right), 1 \leq a < A-1 \tag{5.19}$$

$ref_x(a)$ in (5.19) might seem too arbitrary to be compressed efficiently. Therefore, an assumption is introduced to regulate the possible distribution of $ref_x(a)$.

Assumption 1 (Linearity Between MV Quantization Error and MC Error).

$$\sum_n E^2\left(n;d\right) = k\left(d-d^*\right)^2 \tag{5.20}$$

where d^* and d are the best and currently estimated MVs respectively. k is a positive constant.

This assumption simply assumes that the MC error given d is linearly proportional to the squared distance between d^* and d. It is a common assumption which is also implicitly used in state of the art H.264/AVC sub-pixel motion estimation [39]. Basically, it justifies the optimality for hierarchical motion estimation from integer to subpixels.

Note also that Assumption 1 resembles the 1D counterpart of the linear distortion model shown in (4.19), except that the additive distortion introduced by the motion error is replaced by the actual motion compensated distortion. Recall that one assumption to the linear distortion model is that $\boldsymbol{\delta}$ should be small. It is also satisfied here, i.e. $|d-d^*|$ is small, since only subpixel refinements are considered. As a matter of fact, if we assume the perfect motion compensation, Assumption 1 becomes the same as (4.19) in 1D case.

By applying Assumption 1, (5.18) becomes

$$\begin{aligned} SMV_x\left(0,a\right) = \{d_i | \left(d_i-d^*\right)^2 \leq \left(d_j-d^*\right)^2, \forall j \neq i, \\ 2^{(I+a)}d_i \in \mathbb{Z}, 2^{(I+a)}d_j \in \mathbb{Z}\} \end{aligned} \tag{5.21}$$

Equivalently,

$$d^* \in \left[SMV_x\left(0,a\right) - 2^{-I-a-1}, SMV_x\left(0,a\right) + 2^{-I-a-1}\right] \triangleq range\left(a\right) \tag{5.22}$$

For example, both (5.21) and (5.22) are satisfied after plugging in the parameters in Fig. 5.5, i.e. $d^* = 0.5625$, $SMV_x(0,0) = 1$, and $SMV_x(0,1) =$

$SMV_x(0,2) = 0.5$. Eq. (5.22) is important in limiting the number of $SMV_x(0,a)$ to no more than three candidates, which makes efficient compression possible. Specifically, the possible range of d^* before estimating $SMV_x(0,a)$ is listed in (5.23). The possible alphabets which $ref_x(a)$ belongs to are listed in (5.24). Back to our example in Fig. 5.5, we have $ref_x(1) = -1 \in \{-1,0,1\}$ and $ref_x(2) = 0 \in \{0,1\}$.

$$
\begin{aligned}
d^* &\in range(0) \cap \cdots \cap range(a-1) \\
&\triangleq PRange(a) \\
&= \begin{cases}
\left[SMV_x(0,a-1) - 2^{-I-a}, SMV_x(0,a-1) + 2^{-I-a}\right] \\
\qquad \text{if } SMV_x(0,0) = \cdots = SMV_x(0,a-1), \\
\left[SMV_x(0,a-1) - 2^{-I-a}, SMV_x(0,a-1)\right] \\
\qquad \text{if } \exists k, 0 \le k < a-1, SMV_x(0,k) < SMV_x(0,k+1) = \cdots \\
\qquad = SMV_x(0,a-1), \\
\left[SMV_x(0,a-1), SMV_x(0,a-1) + 2^{-I-a}\right] \\
\qquad \text{if } \exists k, 0 \le k < a-1, SMV_x(0,k) > SMV_x(0,k+1) = \cdots \\
\qquad = SMV_x(0,a-1).
\end{cases}
\end{aligned}
\tag{5.23}
$$

$$
ref_x(a) \in \begin{cases}
\{-1,0,1\} & \text{if } a = 1 \text{ or } ref_x(k) = 0, \forall k, 1 \le k < a, \\
\{-1,0\} & \text{if } \exists k', ref_x(k') = 1, 1 \le k' < a, ref_x(k) = 0, \forall k, k' < k < a, \\
\{0,1\} & \text{if } \exists k', ref_x(k') = -1, 1 \le k' < a, ref_x(k) = 0, \forall k, k' < k < a.
\end{cases}
\tag{5.24}
$$

In summary, Assumption 1 leads to the compact formula in (5.24) for efficiently describing $ref_x(a), 1 \le a < A$. For most of the cases, one bit is sufficient to encode $ref_x(a)$.

In order to design the optimal codebook for $ref_x(a)$, estimation of its distribution is required. From (5.21), finding the best $SMV_x(0,a)$ is equivalent to an optimal scalar quantizer problem which quantizes the best MV d^* to the possible $SMV_x(0,a)$ candidates. According to the theory of the Lloyd-Max quantizer [27], regardless of the distribution of d^*, the optimal decision boundary is always the

middle point of two adjacent reconstruction points. In our case, the two possible decision boundaries will be $SMV_x(0,a-1)-2^{-I-a-1}$ and/or $SMV_x(0,a-1)+2^{-I-a-1}$. Therefore, given the decision boundaries, the distribution of $ref_x(a)$ can be easily obtained as long as the distribution $p(d^*|ref_x(k), k=0,\cdots,a-1)$ is known.

$PRange(a)$ is the only possible interval that d^* might reside in, given $ref_x(k), k=0,\cdots,a-1$. Since the best MV d^* might appear anywhere in $PRange(a)$ without any preference, it is reasonable to assume the underlying distribution is uniform.

Assumption 2 (Uniform Distribution for d^* within $PRange(a)$).

$$p\left(d^*|ref_x\left(k\right), k=0,\cdots,a-1\right) = U\left(PRange\left(a\right)\right) \tag{5.25}$$

Given Assumption 2, the distribution of $ref_x(a)$ can be easily derived and shown in (5.26). The expected coding bits for $ref_x(a)$ are thus 1.5, 1 and 1 for the three different situations, after Assumption 2 is applied.

$$\begin{aligned} &P(ref_x(a)|ref_x(k), k=0,\cdots,a-1), 1\le a<A \\ &= \begin{cases} \left(\frac{1}{4},\frac{1}{2},\frac{1}{4}\right) & \text{if } a=1 \text{ or } ref_x(k)=0, \forall k, 1\le k<a, \\ \left(\frac{1}{2},\frac{1}{2}\right) & \text{if } \exists k', ref_x(k')=1, 1\le k'<a, ref_x(k)=0, \forall k, k'<k<a, \\ \left(\frac{1}{2},\frac{1}{2}\right) & \text{if } \exists k', ref_x(k')=-1, 1\le k'<a, ref_x(k)=0, \forall k, k'<k<a. \end{cases} \end{aligned} \tag{5.26}$$

1D Case – Optimal Codebook Design with RDO

In a video codec, RDO is commonly used to achieve the best quality under some rate constraint. Therefore, RDO ME chooses the best MV that minimizes the cost instead of simply the distortion. In this regard, (5.18) becomes

$$SMV_x\left(0,a\right) = \arg\min_{2^{I+a}d\in\mathbb{Z}} \left\{ \sum_n E^2\left(n;d\right) + \lambda_a R\left(d\right) \right\} \tag{5.27}$$

where $R(d)$ denotes the actual bits for encoding d, and λ_a is the Lagrange multiplier.

Introducing RDO in the ME process will inevitably shift the optimal decision boundary towards the reconstruction level with smaller possibility when the distribution of $ref_x(a)$ is non-uniform. Therefore, only the first situation in (5.26), where $ref_x(a)$ has the distribution $[1/4, 1/2, 1/4]$, will be affected. The problem then becomes how to find the optimal decision boundary b_a^* that minimizes the cost given λ_a. Without loss of generality, assume $SMV_x(0, a-1) = 0$. This problem can be visualized in Fig. 5.6.

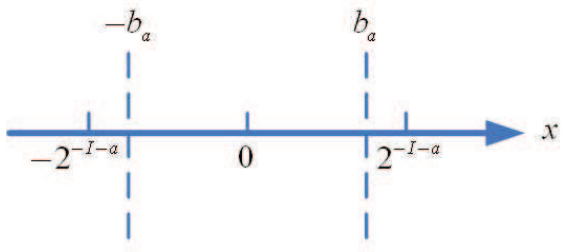


Figure 5.6: 1D SMV decision boundary for accuracy level a.

Given Assumption 2, the distribution of $ref_x(a)$ is determined by b_a^* as follows.

$$\begin{aligned} &P(ref_x(a)|ref_x(0), ref_x(1) = \cdots = ref_x(a-1) = 0) \\ &= \frac{1}{2^{-I-a+1}}[2^{-I-a} - b_a, 2b_a, 2^{-I-a} - b_a], 1 \leq a < A \end{aligned} \tag{5.28}$$

The optimal rate allocation is thus

$$R(d) = \begin{cases} (1-I-a) - \log_2(2b_a) & \text{if } d = 0, \\ (1-I-a) - \log_2\left(2^{-I-a} - b_a\right) & \text{if } d = \pm 2^{-I-a}. \end{cases} \tag{5.29}$$

The expected cost for all d^* is

$$\begin{aligned}
&E_X\left(Cost\right) = E_X\left(k\left(d-x\right)^2 + \lambda_a R\left(d\right)\right) \\
=&\frac{1}{2^{-I-a+1}}\int_{-2^{-I-a}}^{-b_a}\left(x+2^{-I-a}\right)^2 + \lambda_a\left[\left(1-I-a\right) - \log_2\left(2^{-I-a}-b_a\right)\right]dx \\
&+\frac{1}{2^{-I-a+1}}\int_{-b_a}^{b_a}\left(x-0\right)^2 + \lambda_a\left[\left(1-I-a\right) - \log_2\left(2b_a\right)\right]dx \\
&+\frac{1}{2^{-I-a+1}}\int_{b_a}^{2^{-I-a}}\left(x-2^{-I-a}\right)^2 + \lambda_a\left[\left(1-I-a\right) - \log_2\left(2^{-I-a}-b_a\right)\right]dx \\
=&\left(b_a^2 - 2^{-I-a}b_a + \frac{1}{3}2^{-2(I+a)}\right) \\
&+\lambda_a\left[1-I-a-2^{I+a}b_a - 2^{I+a}b_a\log_2\left(b_a\right) - \left(1-2^{I+a}b_a\right)\log_2\left(2^{-I-a}-b_a\right)\right]
\end{aligned} \tag{5.30}$$

By taking the derivative with respect to b_a, we have

$$\frac{d}{db_a}E_X\left(Cost\right) = 2b_a + 2^{I+a}\lambda_a\log_2\left(\frac{2^{-I-a}}{b_a}-1\right) - \left(2^{-I-a}+2^{I+a}\lambda_a\right) \tag{5.31}$$

The optimal decision boundary b_a^* satisfies the following equation

$$2b_a^* + 2^{I+a}\lambda_a\log_2\left(\frac{2^{-I-a}}{b_a^*}-1\right) - \left(2^{-I-a}+2^{I+a}\lambda_a\right) = 0 \tag{5.32}$$

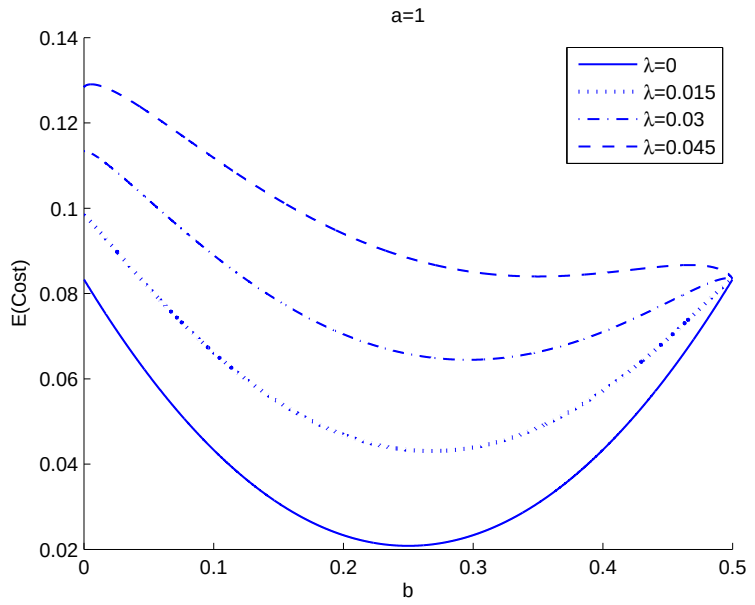
which has no closed-form solution. The expected value of cost as a function of decision boundary b_a is shown in Fig. 5.7, assuming $I = 0$. The optimal decision boundary b_a^* as a function of λ_a is shown in Fig. 5.8, again assuming $I = 0$. We observe that b_a^* is a monotonically increasing function of λ_a, as expected. The optimal decision boundary and the associated expected coding bits of $ref_x(a)$ for the RDO ME case are

$$b_a^* = \left\{b_a \middle| 2b_a + 2^{I+a}\lambda_a\log_2\left(\frac{2^{-I-a}}{b_a}-1\right) - \left(2^{-I-a}+2^{I+a}\lambda_a\right) = 0\right\} \tag{5.33}$$

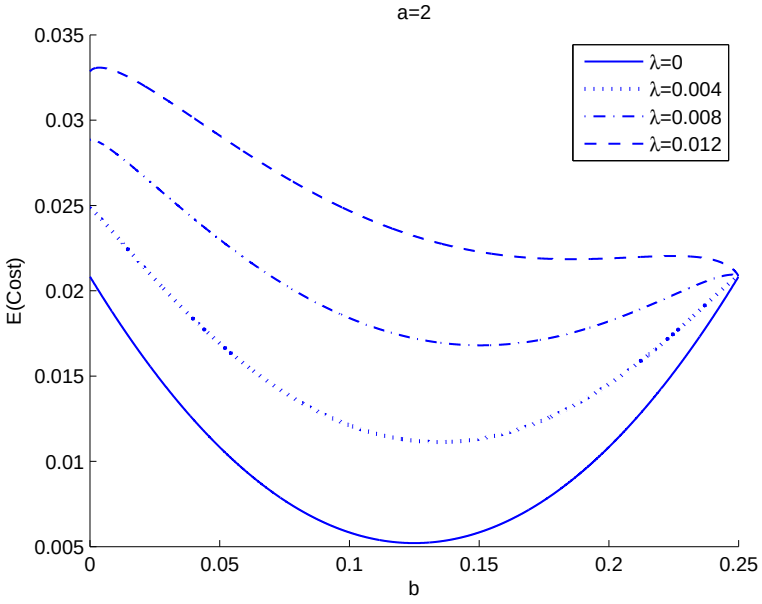
and

$$\begin{aligned}
&E\left[\text{bits for coding } ref_x\left(a\right)\right] \\
=&\frac{b_a^*}{2^{-I-a}}\left[\left(1-I-a\right) - \log_2\left(2b_a^*\right)\right] + \left(1-\frac{b_a^*}{2^{-I-a}}\right)\left[\left(1-I-a\right) - \log_2\left(2^{-I-a}-b_a^*\right)\right]
\end{aligned} \tag{5.34}$$

respectively.

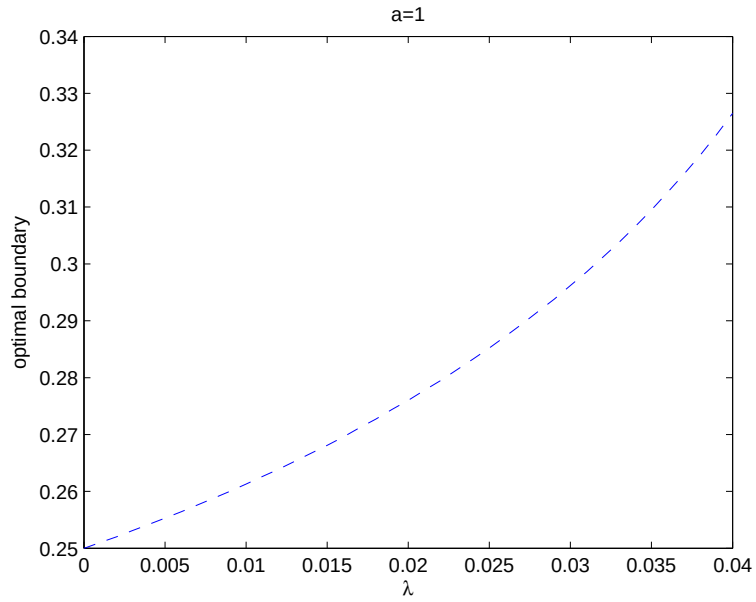


(a)

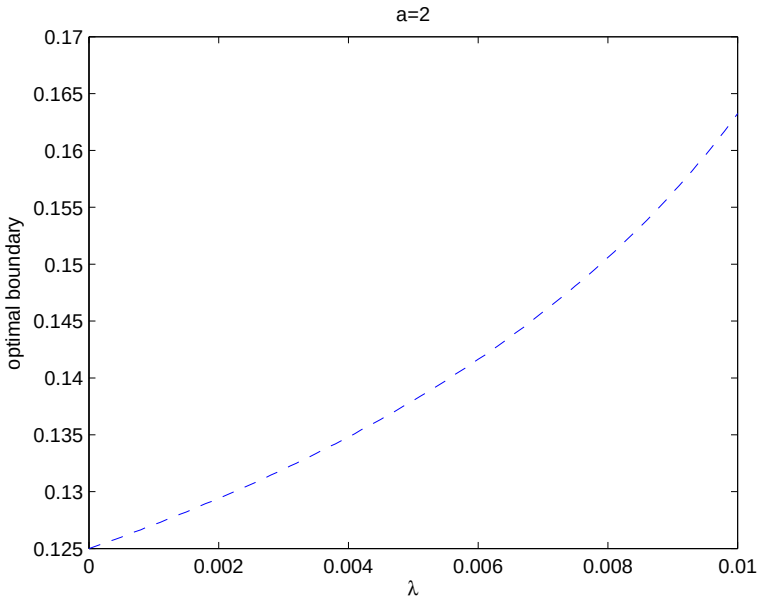


(b)

Figure 5.7: Expected cost as a function of decision boundary. (a) $a = 1$. (b) $a = 2$.



(a)



(b)

Figure 5.8: Optimal decision boundary as a function of λ_a. (a) $a = 1$. (b) $a = 2$.

2D Case – Direct Generalization from 1D Counterpart

Now that we know the optimal codebook designing method for subpixel refinements in the 1D case, the generalization to the 2D case is straightforward.

$$ref(a) = (ref_x(a), ref_y(a)) \tag{5.35}$$

$$\begin{aligned}
&P(ref(a)|ref(k), k = 0, \cdots, a-1), 1 \le a < A \\
=&P(ref_x(a)|ref_x(k), k = 0, \cdots, a-1) \times P(ref_y(a)|ref_y(k), k = 0, \cdots, a-1) \\
=&\begin{cases}
\left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right) \times \left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right) & \text{if } a = 1 \text{ or } ref_i(k) = 0, \forall k, 1 \le k < a, \forall i, i \in \{x, y\}, \\
\left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right) \times \left(\frac{1}{2}, \frac{1}{2}\right) & \text{if } \exists ref_i(k) \neq 0, ref_j(k) = 0, i, j \in \{x, y\}, i \neq j, \\
\left(\frac{1}{2}, \frac{1}{2}\right) \times \left(\frac{1}{2}, \frac{1}{2}\right) & \text{if } \forall i, \exists ref_i(k) \neq 0, i \in \{x, y\}.
\end{cases}
\end{aligned} \tag{5.36}$$

The distribution of $ref(a)$ is shown in (5.36) for three different situations. The expected coding bits for these three situations are 3, 2.5 and 2. The visualization of three examples (one in each case) is shown in Fig. 5.9. The areas with textures are the possible regions that the best MV might locate in. The triangular grid points covered by the textured areas are the candidates for $ref(a)$, each of which has its own partition of the whole textured area and can be differentiated by various patterns. According to Assumption 2, $P(ref(a)|ref(k), k = 0, \cdots, a-1)$ is proportional to the size of area that belongs to $ref(a)$.

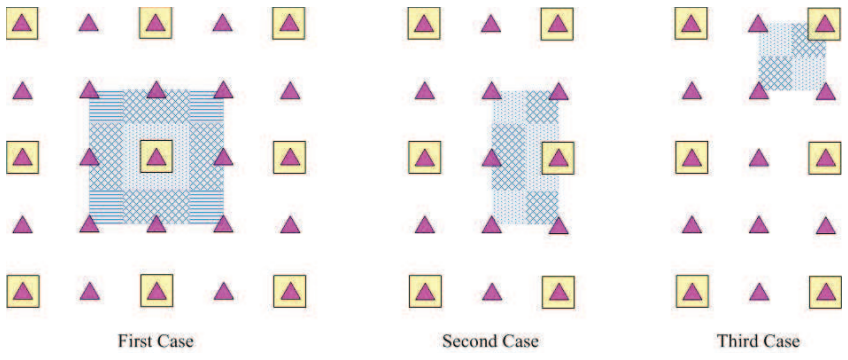


Figure 5.9: Examples of possible refinement candidates (covered by the textured area) and their own territories (with different patterns) in 2D constellation.

2D Case – Non-Separable Refinements for Very Low Bit Rate Scheme

In a very low bit rate scenario, each subpixel refinement consuming 2 to 3 bits is still very inefficient. By eliminating less relevant grid point candidates, we should be able to reduce the average bits to 1 and 1.5. An example for two levels refinements, i.e. half and quarter pixel refinements, is shown in Fig. 5.10. In this example, *ref*(1) requires 1.5 bits while *ref*(2) requires only 1.

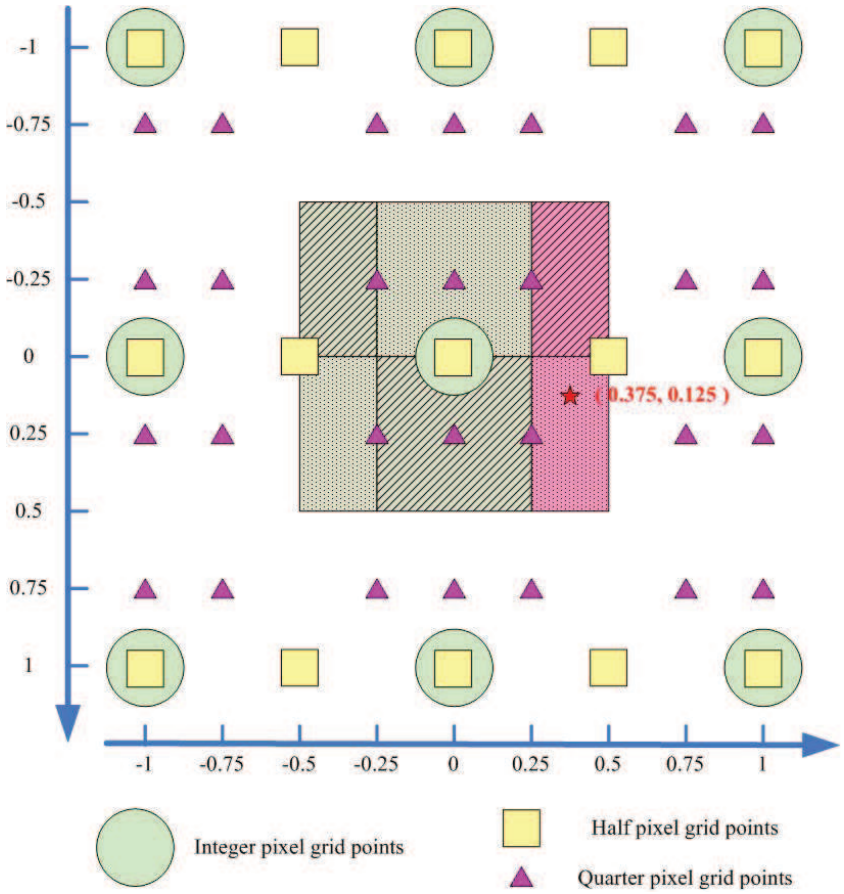


Figure 5.10: 2D SMV refinement constellation for low bit rate scenario.

In Fig. 5.10, only those possible candidate grid points are marked. Assuming that the best MV $d^* = (0.375, 0.125)$, d^* will be encoded as $SMV(0,2) = (0.25, 0.25)$ or equivalently $ref(0) = (0,0)$, $ref(1) = (1,0)$, and $ref(2) = (-1,-1)$ where $ref(1) \in \{(-1,0),(0,0),(1,0)\}$ and $ref(2) \in \{(-1,-1),(-1,1)\}$.

5.4 Extractor for Optimized Decoder

The main task of the bitstream extractor in SVC is to truncate the scalable bitstream according to the scaling parameters demanded by the decoder. The adapted bitstream remains scalable itself and can be fed back to the extractor again for further truncations. The decoder should then be able to decode all adapted bitstreams that are legitimately generated by the extractor.

The designing criterion for a generic SVC bitstream extractor can be rather trivial. For example, in the case where all scaling parameters are explicitly specified, the extractor can simply select those packets with required labels and discard others. This can be done by properly assigning high level syntax descriptions [89]. On the other hand, when only partial constraints are imposed on the scaling parameters, the extractor has more freedom in selecting its output packets. For example, in some situations where the total bit rate is restricted by the applicable channel conditions, only the decoding bit rate will be specified, as opposed to frame rate or spatial resolution. In those cases, extractor can decide the optimal combination of frame rate and spatial resolution under the total rate constraint, which may be optimal in the sense of visual quality or decoding complexity [44, 43, 79].

Taking the scalable motion into account, the SVC bitstream extractor now has extra work to do, i.e. optimal bit allocation among motion and texture [15, 11]. In this section, we consider the case where decoding frame rate and spatial resolution are pre-specified and fixed. At a certain decoding bit rate, one of the motion quality layers, combining with the corresponding texture information, will provide the best reconstructed quality. As the bit rate varies, the optimal motion quality layer also changes accordingly. The optimal motion quality layer as a function of decoding bit rate, if not provided by the encoder, will be determined by the extractor. Based on this function, the adapted bitstream is guaranteed with the best decoding quality throughout all possible rates, for the specified frame rate and spatial resolution.

We propose three approaches for optimal bit rate allocation among motion and texture in this section, i.e. brute force, model-assisted, and model-based. The

brute force method determines the best motion quality layer for a certain bit rate by exhaustively searching among all layers. A finite set of possible decoding bit rates can be tested in this approach.

The model-assisted method is based on properties resulting directly from the exponential rate-distortion model from general source coding theory. These properties include the monotonically non-decreasing property of the optimal motion quality layer as a function of rate, and the unimodal property of decoded quality as a function of motion quality layer. By applying these properties, irrelevant testing scenarios can be omitted without sacrificing the extractor performance. Moreover, the monotonically non-decreasing property can further simplify the description of optimal motion quality layers by recording only the critical rates at which the change in optimal motion quality layer occurs. For example, only two critical rates are required for a three-layer motion model. The determination of these critical rates can be realized using the bisection method, which is computationally efficient.

Now that the output of the extractor can be simplified to a series of critical rates, the question becomes whether there exists a more efficient testing algorithm to approach the critical rates than the bisection method. By explicitly applying the rate-distortion model, a.k.a. the model-based method, a more accurate estimation of the critical rate can be predicted. This estimation is in general better than the middle point of the possible rate range, which is blindly predicted from the bisection method. Through parameter estimation, the resulting model can adapt to the actual video contents. In general, if the adapted model fits well with the encoded bitstream, a reduced number of trial and errors can be expected before the actual critical rate is reached.

5.4.1 Brute Force Method

In the brute force method, the same video bitstream is decoded multiple times, each time a different motion quality layer is applied. The same process is repeated several times for all those decoding bit rates of interest, resulting in a table as shown in Table 5.2 for example (note that the entry marked with "-"

indicates non-decodable). Those entries marked with bold face numbers reflect the best motion quality layer for each decoding rate. A simplified table with one entry for each motion quality layer identifying its effective range of decoding bit rates should be stored for each frame and resolution. An example that corresponds to Table 5.2 is shown in Table 5.3. These tables are not large and could be efficiently compressed.

Table 5.2: Extractor RD table for the brute force method (BUS @ CIF 30 *fps*)

	Decoding bit rate (*kbps*)						
Motion quality layer	256	384	512	640	768	896	1024
0	**23.3**	24.27	24.96	25.55	25.86	26.13	26.29
1	-	**24.63**	**25.58**	**26.54**	27.01	27.42	27.74
2	-	-	25.2	26.48	**27.2**	**27.63**	**28.21**

Table 5.3: Extractor information for the brute force method (BUS @ CIF 30 *fps*)

Motion quality layer	Effective rate range (*kbps*)
0	0 - 320
1	320 - 704
2	704 - 1024

5.4.2 Model-Assisted Method

Note that for the rest of the current section, the notation "R" is used purely for the "rate" only. This should not be confused with the "resolution" notation that we have seen in previous sections, since the problem domain of the extractor is limited within a single resolution and frame rate scenario. The "a" and "A", which are used for describing motion quality layers, remain the same as before.

As observed from Table 5.2, the optimal motion quality layer is a monotonically non-decreasing function of the decoding bit rate. This is neither a coincidence nor a surprising result. The underlying rate distortion model for texture coding, along with the motion distortion model mentioned in Section 4.3, can explain this property well.

Although the true distortion-rate model is data dependent and complicated, a simpler model has been derived and used for video texture coding [88, 28, 30].

$$D_t(R_t) = \sigma_t^2 \exp\left(-\frac{R_t}{a_t}\right), \tag{5.37}$$

where σ_t and a_t are content dependent parameters. Note that this exponential model is derived by applying the high-rate assumption, i.e. the quantization step size is small enough such that the quantization noise is almost uniformly distributed. This model provides an explicit way to quantify the texture distortion $D_{t,i}$ in (4.23) using R_t instead of the individual quantizer i.

Similarly, by plugging (5.37) into (4.19), the motion distortion-rate model can be expressed as follows.

$$D_m(R_m) = \Psi\sigma_m^2 \exp\left(-\frac{R_m}{a_m}\right) \tag{5.38}$$

The additive distortion model [62], which first appeared in (4.7), can be restated formally as follows.

$$D(R) = D_m(R_m) + D_t(R_t) \tag{5.39}$$

where

$$R = R_m + R_t \tag{5.40}$$

Note that the distortion-rate models discussed so far are limited to single frame encoding, given fixed reference pictures. Generalizing to a group of pictures is not straightforward. The quality of reference pictures will be affected by the encoding bit rate as well. The extra dependency prevents a single model for a group of pictures using the above derivations.

A typical distortion-rate plot, depicting contributions from both motion and texture, is shown in Fig. 5.11. Note that the plot is obtained using the texture associated with the highest motion quality layer. Clearly, the rate efficiency, i.e. negative slope of the distortion-rate plot, is monotonically decreasing for both motion and texture.

$$-D_t'(R_t) \triangleq \lambda_t = \frac{\sigma_t^2}{a_t} \exp\left(-\frac{R_t}{a_t}\right) \tag{5.41}$$

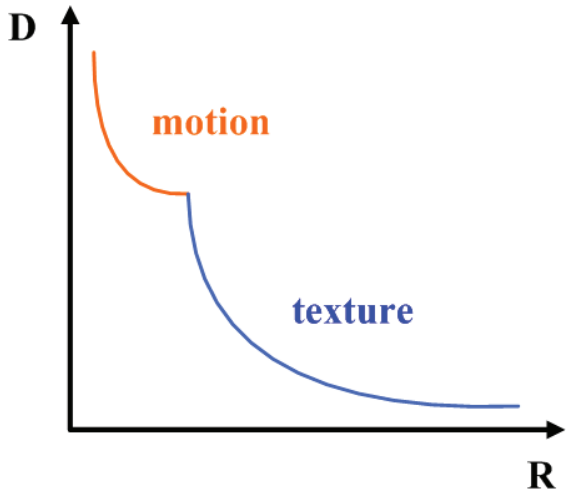


Figure 5.11: Ideal distortion-rate plot showing contributions from both motion and texture.

$$-D'_m(R_m) \triangleq \lambda_m = \Psi \frac{\sigma_m^2}{a_m} \exp\left(-\frac{R_m}{a_m}\right) \tag{5.42}$$

Monotonically Non-decreasing Property

Knowing the aforementioned distortion-rate models, we are now able to prove the monotonically non-decreasing property. Suppose at a certain decoding rate R_0, the minimal distortion is achieved with motion quality layer i.

$$\begin{aligned} &\Psi\sigma_m^2 \exp\left(-\frac{R_m^i}{a_m}\right) + \sigma_t^2 \exp\left(-\frac{R_0 - R_m^i}{a_t}\right) \leq \\ &\Psi\sigma_m^2 \exp\left(-\frac{R_m^j}{a_m}\right) + \sigma_t^2 \exp\left(-\frac{R_0 - R_m^j}{a_t}\right), \forall j \neq i \end{aligned} \tag{5.43}$$

$$\begin{aligned} &\Psi\sigma_m^2 \left(\exp\left(-\frac{R_m^i}{a_m}\right) - \exp\left(-\frac{R_m^j}{a_m}\right)\right) \leq \\ &-\sigma_t^2 \exp\left(-\frac{R_0}{a_t}\right)\left(\exp\left(\frac{R_m^i}{a_t}\right) - \exp\left(\frac{R_m^j}{a_t}\right)\right), \forall j \neq i \end{aligned} \tag{5.44}$$

Given an extra bit rate $\triangle R$, the total distortion difference between motion quality layer i and j becomes,

$$\begin{aligned} D^i - D^j =& \Psi\sigma_m^2 \left(\exp\left(-\frac{R_m^i}{a_m}\right) - \exp\left(-\frac{R_m^j}{a_m}\right)\right) \\ &+ \sigma_t^2 \exp\left(-\frac{R_0 + \triangle R}{a_t}\right)\left(\exp\left(\frac{R_m^i}{a_t}\right) - \exp\left(\frac{R_m^j}{a_t}\right)\right) \\ \leq& \sigma_t^2 \exp\left(-\frac{R_0}{a_t}\right)\left(\exp\left(\frac{R_m^i}{a_t}\right) - \exp\left(\frac{R_m^j}{a_t}\right)\right)\left(\exp\left(-\frac{\triangle R}{a_t}\right) - 1\right) \end{aligned} \tag{5.45}$$

Since $\sigma_t, a_t, \triangle R > 0$, $\exp(-\triangle R/a_t) - 1 < 0$. For those motion quality layers $j < i$, the corresponding motion bit rates R_m^j are smaller than R_m^i. Therefore, we have $(\exp(R_m^i/a_t) - \exp(R_m^j/a_t)) > 0$. Eq. (5.45) becomes,

$$D^i < D^j, \forall j < i. \tag{5.46}$$

Here we have proven that when bit rate increases, the best motion quality layer never decreases, i.e. the monotonically non-decreasing property. By applying this property, many testing scenarios can be omitted without sacrificing the performance of the final extractor output. In Table 5.2, for example, the motion quality layer $a = 0$ need not be tested for decoding bit rates which are greater than 384 *kbps*.

Moreover, the monotonically non-decreasing property also provides an even simpler way to describe the extractor information than the one shown in Table 5.3. A series of critical rates, $\{R^{a,*} | D^a(R^{a,*}) = D^{a+1}(R^{a,*})\}$, can be found and recorded. An example is shown in Table 5.4. Note that the monotonically non-decreasing property limits the number of critical rates to $A - 1$.

Table 5.4: Extractor information for the model-assisted method (BUS @ CIF 30 *fps*)

Motion quality layer	Critical rate (*kbps*)
0 - 1	272
1 - 2	816

Recall from Section 5.2 that the prior knowledge on $\{\lambda_a\}$ indicates the set of best operating points on which the motion quality layers are obtained. Therefore,

the corresponding set of decoding bit rates, $\{R^a\}$, can be derived from $\{\lambda_a\}$. Given R^a and R^{a+1}, our goal is to determine the critical rate, $R^{a,*}$, where $R^a < R^{a,*} < R^{a+1}$. Based on the monotonically non-decreasing property, $R^{a,*}$ can be approached using the bisection method. For each iteration, only two motion quality layers, i.e. a and $a+1$, are tested. In practice, the iterative algorithm can be terminated whenever $|D^a(\hat{R}^{a,*}) - D^{a+1}(\hat{R}^{a,*})| < \epsilon$, where ϵ is a stopping threshold and $\hat{R}^{a,*}$ is an approximation to $R^{a,*}$. Finally, $\{\hat{R}^{a,*}|0 \leq a < A-1\}$ is stored and transmitted as the optimal extractor output.

Unimodal Property

The unimodal property, on the other hand, states that given a fixed decoding rate, the decoding quality as a function of the motion quality layer is unimodal, i.e. the decoding quality is monotonically decreasing on both sides of the optimal motion quality layer. This property is especially useful when finding the maximal decoding quality (or minimal decoding distortion). Once a decrease in decoding quality is identified, there is no need to test the following motion quality layers.

The unimodal property can be proved as follows. We focus only on the monotonically increasing distortion along the side of increasing motion quality layers. The other side can be proved in exactly the same manner. Again, suppose at a certain decoding rate R_0, the minimal distortion is achieved with motion quality layer i. For layer $j > i$, we have the relationship as shown in (5.44). According to the mean value theorem, there exist $R_m^{ij}, R_m^i < R_m^{ij} < R_m^j$ and $R_t^{ji}, R_t^j < R_t^{ji} < R_t^i$ such that

$$D_m(R_m^i) - D_m(R_m^j) = (R_m^i - R_m^j)D'_m(R_m^{ij}) = -\triangle R^{ij} D'_m(R_m^{ij}) \tag{5.47}$$

$$D_t(R_t^j) - D_t(R_t^i) = (R_t^j - R_t^i)D'_t(R_t^{ji}) = -\triangle R^{ij} D'_t(R_t^{ji}) \tag{5.48}$$

where $\triangle R^{ij} \triangleq R_m^j - R_m^i = R_t^i - R_t^j > 0$. Since layer i produces the minimal distortion, we have

$$D_m(R_m^i) + D_t(R_t^i) < D_m(R_m^j) + D_t(R_t^j) \tag{5.49}$$

Equivalently,

$$D'_m(R_m^{ij}) > D'_t(R_t^{ji}) \tag{5.50}$$

Similarly, for another layer $k > j$,

$$\begin{aligned} D_m(R_m^j) - D_m(R_m^k) &= -\triangle R^{jk} D'_m(R_m^{jk}) \\ &< -\triangle R^{jk} D'_m(R_m^{ij}) \\ &< -\triangle R^{jk} D'_t(R_t^{ji}) \\ &< -\triangle R^{jk} D'_t(R_t^{kj}) \\ &= D_t(R_t^k) - D_t(R_t^j) \end{aligned} \tag{5.51}$$

Note that the first and last inequalities in (5.51) result from the monotonically increasing slopes as shown in (5.41) and (5.42), together with the fact that $R_m^{ij} < R_m^{jk}$ and $R_t^{kj} < R_t^{ji}$. The following relationship can now be concluded.

$$D^j < D^k, \forall\{j,k|i<j<k\}. \tag{5.52}$$

In other words, the decoding distortion is monotonically increasing (decreasing) on the right (left) hand side of the optimal motion quality layer. This proves the unimodal property.

5.4.3 Model-Based Method

Theoretically, the critical rate, $R^{a,*} \in (R^a, R^{a+1})$, is the decoding bit rate at which both motion quality layers, a and $a+1$, produce the same distortion.

$$D^a(R^{a,*}) = D^{a+1}(R^{a,*}) \tag{5.53}$$

Linear Model

For simplicity, we begin with an approximate model of the total distortion function where the contributions from motion (both motion distortion and motion rate) in (5.39) are ignored.

$$D(R) \cong D_t(R) = \sigma_t^2 \exp\left(-\frac{R}{a_t}\right) \tag{5.54}$$

Since the distortion-rate plot is usually depicted in logarithmic scale using Peak Signal-to-Noise Ratio (PSNR) notation, we then have

$$PSNR(R) = 10\log_{10}\left(\frac{255^2}{D(R)}\right) \cong \left(\frac{10}{a_t \ln 10}\right) R + 10\log_{10}\left(\frac{255^2}{\sigma_t^2}\right) \triangleq \alpha R + \beta \tag{5.55}$$

(α, β) can be estimated to reflect the individual characteristic of the video sequence from at least two actual operating points on the PSNR-rate curve. In order to approach the critical rate $R^{a,*}$, the best operating points for estimating (α, β) should be $(R^a, PSNR(R^a))$ and $(R^{a+1}, PSNR(R^{a+1}))$. The estimate for $R^{a,*}$ is then the rate at the intersection of the two lines, i.e. $PSNR^a(R)$ and $PSNR^{a+1}(R)$, where the former is the PSNR-rate curve using motion quality layer a and the latter is using layer $a+1$.

Because the actual PSNR-rate curve is approximated using a line with slope α and offset β, this approach is called the linear model method. In the linear model method, each iteration for determining an estimate of $R^{a,*}$ requires four operating points, i.e. $(R^a, PSNR^a(R^a))$, $(R^{a+1}, PSNR^a(R^{a+1}))$, $(R^a, PSNR^{a+1}(R^a))$, and $(R^{a+1}, PSNR^{a+1}(R^{a+1}))$. Two of these operating points should be updated with $(\hat{R}^{a,*}, PSNR^a(\hat{R}^{a,*}))$ and $(\hat{R}^{a,*}, PSNR^{a+1}(\hat{R}^{a,*}))$ from iteration to iteration, where $\hat{R}^{a,*}$ is the estimated $R^{a,*}$.

Additive Model

The additive model method, as a refined version to the linear model method, is realized by directly applying (5.39) without ignoring the effect of motion. However, motion distortion, which is an exponential function of motion rate in (5.38), is generalized to a function of total rate to compensate for the inter frame motion dependency within a GOP structure.

$$D(R) = D_m(R) + D_t(R - R_m) \tag{5.56}$$

where R_m is the motion rate. Note that $D_m(R)$ can be estimated using the PSNR approach, which is similar to (5.55). The required operating points include $(R^a, PSNR_m^a(R^a))$, $(R^{a+1}, PSNR_m^a(R^{a+1}))$, $(R^a, PSNR_m^{a+1}(R^a))$, and

$(R^{a+1}, PSNR_m^{a+1}(R^{a+1}))$, where $PSNR_m^a$ denotes the motion compensated PSNR using motion quality layer a.

$$PSNR_m^a(R) = \left(\frac{PSNR_m^a(R^{a+1}) - PSNR_m^a(R^a)}{R^{a+1} - R^a}\right)(R - R^a) + PSNR_m^a(R^a) \tag{5.57}$$

$$D_m^a(R) = 255^2 10^{-PSNR_m^a(R)/10} - D_m^* \tag{5.58}$$

where D_m^* is the minimal distortion, which is achieved using the highest motion quality layer. Note that in the additive distortion model, the motion distortion that is derived from the motion compensated PSNR should always be offset by D_m^*. D_m^* can also be viewed as the energy of the un-quantized texture signal.

In order to determine the parameters σ_t and a_t, in addition to the same four operating points from the linear model method, the motion rates for encoding the different motion quality layers, $\{R_m^a\}$, are also required.

Since D_m^* and $\{R_m^a\}$ for $0 \leq a < A$ are known after encoding, $R^{a,*}$ can be derived by solving (5.53). By plugging in (5.56), we have

$$\sigma_t^2 \left(\exp\left(-\frac{R^{a,*} - R_m^{a+1}}{a_t}\right) - \exp\left(-\frac{R^{a,*} - R_m^a}{a_t}\right)\right) = D_m^a(R^{a,*}) - D_m^{a+1}(R^{a,*}) \tag{5.59}$$

$$\exp\left(-\frac{R^{a,*}}{a_t}\right) = \frac{D_m^a(R^{a,*}) - D_m^{a+1}(R^{a,*})}{\sigma_t^2\left(\exp(R_m^{a+1}/a_t) - \exp(R_m^a/a_t)\right)} \tag{5.60}$$

In order to have an accurate $R^{a,*}$, a_t and σ_t must be correctly estimated. A widely accepted concept for estimating model parameters for video coding is adaptive model fitting. The parameters are updated from frame to frame in order to track the non-stationary characteristics of the video contents.

5.5 Experimental Results

The evaluation of the proposed SMM will be performed on the low complexity WSVC framework proposed in Section 2.3. Fig. 5.12 shows the system diagram with the insertion of the scalable motion model.

The format of the input testing sequences is CIF at 30 *fps* and the SVC will generate scalable bitstreams with maximum bit rates for various decoding scenarios

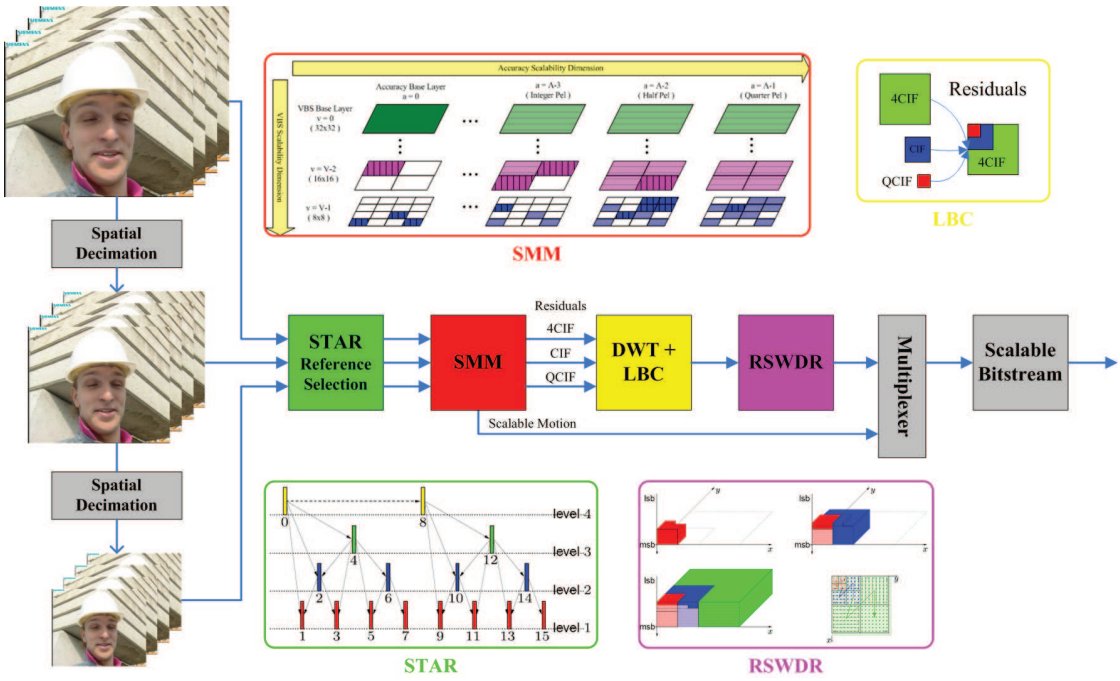


Figure 5.12: SVC system diagram with proposed SMM embedded.

as listed in Table 5.5. Our SMM will be compared side-by-side with a non-scalable motion model for both CIF and QCIF decoding formats with different frame rates.

Table 5.5: Maximum bit rate allocation for the SVC encoder (*kbps*)

	30 *fps*	15 *fps*	7.5 *fps*
CIF	1536	768	-
QCIF	512	256	128

The rate distortion curve for the FOOTBALL sequence in CIF size at full frame rate is shown in Fig. 5.13(a). Here we try two different settings of w_r, where $w = (1, 4)$ puts equal weightings on both CIF and QCIF sizes while $w = (1, 12)$ puts more weight on the QCIF size sequence. It is clear that both settings outperform the non-scalable motion model. Moreover, lower decodable bit rates can be achieved with the proposed SMM. Among the two settings using SMM, $w = (1, 4)$ yields slightly better coding efficiency on the CIF sequence as expected. This result verifies that our SMM has the ability to fine tune the coding performance

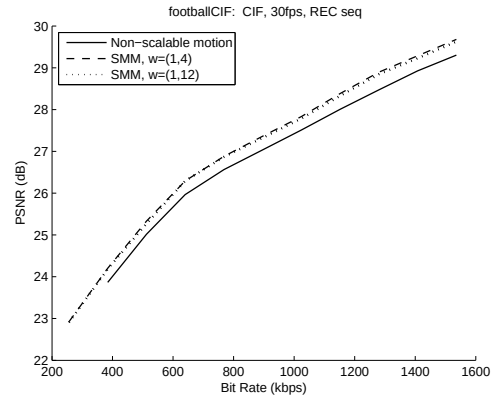
toward a preferred resolution using the distortion multiplier w_r. The details of the PSNR difference to non-scalable motion are listed in Table 5.6, along with the results from BUS and FOREMAN.

Fig. 5.13(b) and Fig. 5.13(c) demonstrate two additional decoding scenarios, i.e. reduced frame rate and reduced spatial resolution, respectively. While Fig. 5.13(b) shows a similar trend to that in Fig. 5.6, the performance of the proposed SMM is virtually the same or slightly worse than the non-scalable motion in QCIF. This result differs from that of a t+2D SVC, where better performance is usually observed for reduced resolution using scalable motion. However, it also reflects the fact that in a 2D+t+2D SVC, reduced resolution has its own motion information, which eliminates the incoherence from motion compensation. Since dedicated motion information is estimated for reduced resolutions, it is reasonable that scalable motion does not provide better performance than non-scalable ones. Again, slightly better coding efficiency on the QCIF sequence is observed with $w = (1, 12)$. The details for BUS and FOREMAN sequences can be found in Table 5.7 and Table 5.8.

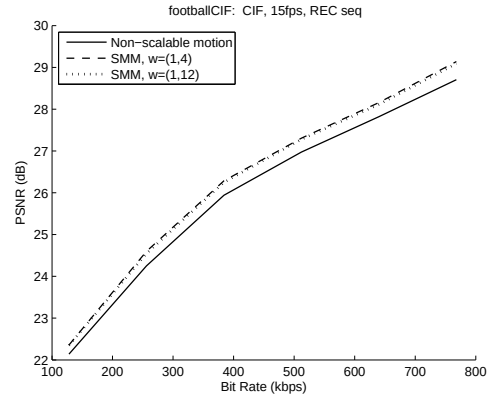
Table 5.6: RD comparison: CIF 30 *fps*

	Bit rate (*kbps*)	128	256	384	512	1024	1536
FOOTBALL	Non-scalable	-	-	23.87	25.02	27.51	29.30
	SMM, w=(1,4)	-	-	0.34	0.32	0.32	0.38
	SMM, w=(1,12)	-	-	0.32	0.25	0.27	0.32
BUS	Non-scalable	-	23.58	24.80	25.70	28.36	30.21
	SMM, w=(1,4)	-	0.08	0.07	0.06	0.07	0.16
	SMM, w=(1,12)	-	0.08	0.05	0.03	0.03	0.13
FOREMAN	Non-scalable	26.58	28.99	30.16	30.89	35.86	37.76
	SMM, w=(1,4)	0.22	0.13	0.05	0.07	0.01	0.05
	SMM, w=(1,12)	0.22	0.12	0.04	0.04	-0.03	0.00

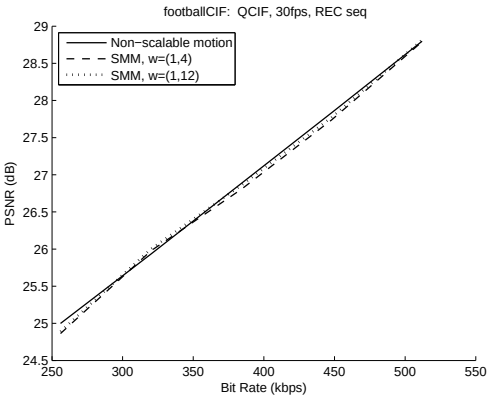
On the other hand, in order to evaluate the coding techniques of our proposed SMM structure, SMVF using SPIHT is compared with SMVF without SPIHT in Table 5.9. Note that E_a denotes the accuracy level from which SPIHT



(a)



(b)



(c)

Figure 5.13: Comparison of RD curves between non-scalable motion model (solid line) and the proposed SMM (dashed line with $w = (1, 4)$ and dotted line with $w = (1, 12)$) using FOOTBALL as input sequence. (a) CIF 30 *fps*. (b) CIF 15 *fps*. (c) QCIF 30 *fps*.

Table 5.7: RD comparison: CIF 15 *fps*

	Bit rate (*kbps*)	128	256	384	512	640	768
FOOTBALL	Non-scalable	22.14	24.25	25.94	26.97	27.82	28.71
	SMM, w=(1,4)	0.22	0.35	0.35	0.33	0.34	0.43
	SMM, w=(1,12)	0.21	0.30	0.31	0.31	0.29	0.38
BUS	Non-scalable	23.13	25.14	26.75	27.76	28.67	29.63
	SMM, w=(1,4)	0.04	0.07	0.13	0.05	0.11	0.22
	SMM, w=(1,12)	0.06	0.04	0.14	0.04	0.08	0.20
FOREMAN	Non-scalable	28.31	30.41	33.85	35.10	36.07	37.15
	SMM, w=(1,4)	0.25	0.07	0.17	-0.04	0.04	0.13
	SMM, w=(1,12)	0.23	0.05	0.23	-0.07	0.01	0.08

starts the sorting pass. No level smaller than E_a is decodable. However, the coding efficiency is increased for those levels equal to or larger than E_a. In other words, some scalabilities can be traded for higher coding efficiency if they are irrelevant. According to Table 5.9, 63% savings can be achieved in the low bit rate regime, i.e. $a = 0$. In general, the savings margin decreases for larger a.

One cause of the reduction of the bit saving percentage in high rate regimes is the inefficiency of significance bits coding. By introducing the additional run length coding on the LIV significance bits, we are able to effectively relieve the problem. From the RD curves shown in Fig. 5.14, the effectiveness of the proposed RLC coding can be clearly verified.

Fig. 5.15 shows three RD curves using different SMVD refinement codebook design methods. We observe that the optimal design outperforms the original one for all bit rates, with or without RDO. The very low bit rate design, on the other hand, has the worst motion compensated PSNR. However, it also requires the least bits, which satisfies the requirement of its possible applications.

Finally, we test our SMM using all proposed coding and estimation algorithms side-by-side with a non-scalable motion model. BUS is chosen as the input sequence, which has moderate motion between FOREMAN and FOOTBALL. The RD curve is shown in Fig. 5.16, with the solid line (circle marker) depicting non-

Table 5.8: RD comparison: QCIF 30 *fps*

	Bit rate (*kbps*)	256	320	384	448	512
FOOTBALL	Non-scalable	25.00	25.93	26.88	27.84	28.80
	SMM, w=(1,4)	-0.14	0.04	-0.08	-0.10	-0.01
	SMM, w=(1,12)	-0.11	0.07	-0.02	-0.06	0.02
BUS	Non-scalable	26.04	26.98	28.01	28.87	29.70
	SMM, w=(1,4)	-0.18	-0.10	-0.11	-0.05	0.04
	SMM, w=(1,12)	-0.15	-0.06	-0.08	-0.05	0.05
FOREMAN	Non-scalable	32.74	34.04	35.25	36.50	37.66
	SMM, w=(1,4)	-0.17	0.01	-0.08	-0.09	0.063
	SMM, w=(1,12)	-0.13	0.04	-0.03	-0.03	0.11

Table 5.9: SMM structure coding comparison

	Original	SPIHT					
		$E_a = 0$	Saving	$E_a = 1$	Saving	$E_a = 2$	Saving
$a = 0$	3271	1209	63%	56	-	56	-
$a = 1$	3584	1790	63%	1522	57%	56	-
$a = 2$	4372	3022	31%	2754	37%	2486	43%

scalable motion and the dashed line (square marker) depicting the proposed SMM. Note that the dashed line is the convex hull of three different dotted lines, each of them depicting the RD curve using different motion qualities. As observed, our SMM is comparable to non-scalable motion in the high-rate regime, despite the tradeoff of efficiency for scalability. Performance crossing starts in the middle-rate range, as our SMM starts taking more advantage of the flexible tradeoff between motion and residuals. In the low-rate regime, the SMM not only dominates the performance comparison but also extends the decodable range to a region of much lower rate.

To evaluate the different bitstream extractor realizations, we test 5 decoding scenarios, i.e. CIF 30 *fps*, CIF 15 *fps*, QCIF 30 *fps*, QCIF 15 *fps*, and QCIF 7.5 *fps*, for 4 MPEG reference sequences, i.e. BUS, FOREMAN, FOOTBALL, and MOBILE. For each decoding scenario, two experiments will be performed to gen-

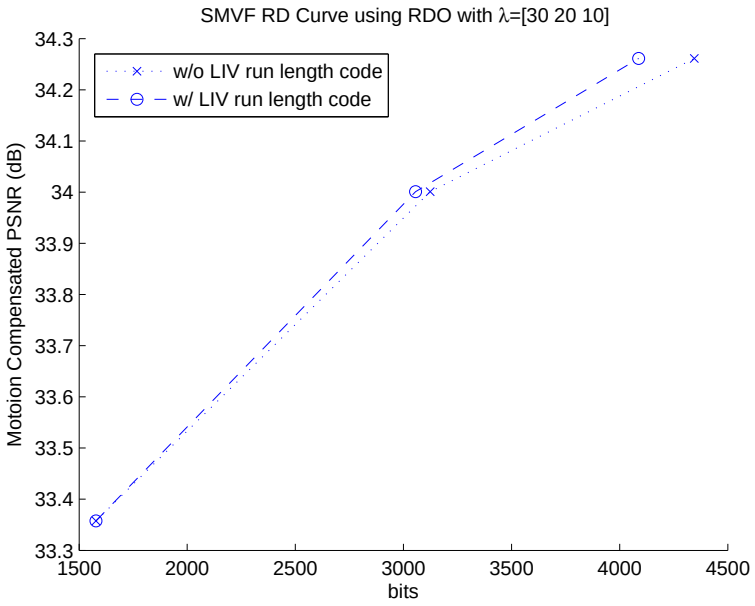


Figure 5.14: RD curve comparison using LIV RLC.

erate the extractor output. In the first experiment, only a discrete set of decoding rates will be tested. The possible range of decoding rate is uniformly divided into 2^N segments, which are indexed from 1 to 2^N. The goal is to find the segment in which the optimal motion quality layer changes and record the corresponding index. Brute force (BF) and model-assisted (MA) (including progressive (PR) and bisection (BI) search) methods will be compared side-by-side in this experiment. The results are shown in Table 5.10 for $N = 3$.

In Table 5.10, the columns labeled with "$a1$" denote the index (from 1 to 8) of the rate segment in which the optimal motion quality layer transitions from $a = 0$ to $a = 1$. The same rules apply to the columns labeled with "$a2$". Those entries marked with "-" indicate that the transition never happens. The columns labeled with "#" denote the number of decoding times required to complete the extractor output. This is a measurement of the extractor complexity where lower is better.

Observed from the table, the model-assisted method provides exactly the same output as the brute force method, while saving a tremendous amount of com-

Table 5.10: Extractor comparison with discrete decoding bit rates

		BUS			FOOTBALL			FOREMAN			MOBILE		
		$a1$	$a2$	#	$a1$	$a2$	#	$a1$	$a2$	#	$a1$	$a2$	#
CIF	BF	3	7	24	4	8	24	2	6	24	2	8	24
30 *fps*	MA(PR)	3	7	14	4	8	15	2	6	12	2	8	16
	MA(BI)	3	7	11	4	8	10	2	6	10	2	8	10
CIF	BF	3	8	24	5	-	24	3	7	24	2	-	24
15 *fps*	MA(PR)	3	8	16	5	-	14	3	7	14	2	-	16
	MA(BI)	3	8	11	5	-	9	3	7	11	2	-	10
QCIF	BF	5	-	16	8	-	16	4	-	16	4	-	16
30 *fps*	MA(PR)	5	-	7	8	-	11	4	-	6	4	-	6
	MA(BI)	5	-	6	8	-	5	4	-	5	4	-	5
QCIF	BF	6	-	16	-	-	16	6	-	16	5	-	16
15 *fps*	MA(PR)	6	-	9	-	-	10	6	-	10	5	-	8
	MA(BI)	6	-	6	-	-	4	6	-	6	5	-	6
QCIF	BF	-	-	16	-	-	16	-	-	16	7	-	16
7.5 *fps*	MA(PR)	-	-	11	-	-	8	-	-	13	7	-	11
	MA(BI)	-	-	5	-	-	3	-	-	6	7	-	6

putational power and time. This verifies the effectiveness of the additive distortion assumption along with the exponential function model, from which the monotonically non-increasing property and the unimodal property are derived. Moreover, the advantage of bisection search over progressive search on reducing the complexity is also verified throughout various testing sequences and decoding scenarios.

In the second experiment, a search is conducted for the exact critical rates at which the optimal motion quality layer switches, i.e. $\{R^{a,*}|a = 0, \cdots, A-2\}$. For practical reason, the searching process for $R^{a,*}$ stops whenever $|PSNR^a(\hat{R}^{a,*}) - PSNR^{a+1}(\hat{R}^{a,*})| \leq 0.01\ dB$. The approximate critical rates $\{\hat{R}^{a,*}\}$ are recorded as the extractor output. The model-assisted method with bisection search (MA(BI)) will be compared with the model-based method using the linear model (MB(LM)). The results are shown in Table 5.11.

Table 5.11: Extractor comparison with continuous critical rates

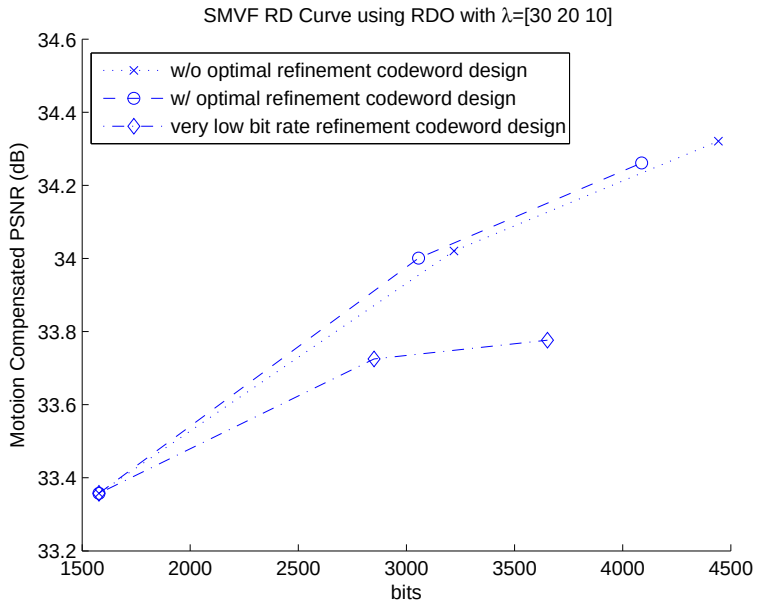
		BUS			FOOTBALL			FOREMAN			MOBILE		
		$a1$	$a2$	#	$a1$	$a2$	#	$a1$	$a2$	#	$a1$	$a2$	#
CIF	MA(BI)	272	816	23	420	976	26	203	704	26	181	896	24
30 *fps*	MB(LM)	272	816	19	401	974	16	203	698	22	148	899	15
CIF	MA(BI)	176	480	17	280	-	27	136	448	17	120	-	28
15 *fps*	MB(LM)	178	481	21	277	-	23	134	448	13	118	-	23
QCIF	MA(BI)	285	-	18	448	-	5	208	-	9	208	-	9
30 *fps*	MB(LM)	259	-	8	448	-	5	210	-	11	209	-	7
QCIF	MA(BI)	190	-	14	-	-	14	160	-	6	140	-	12
15 *fps*	MB(LM)	191	-	8	-	-	14	174	-	12	141	-	6
QCIF	MA(BI)	-	-	13	-	-	7	-	-	14	96	-	4
7.5 *fps*	MB(LM)	-	-	13	-	-	7	-	-	14	96	-	4

Some observations can be drawn from Table 5.11. In general, the model-based method using the linear model demonstrates better or equal performances than the model-assisted method using bisection search in about 85% of the cases. The linear model assumption is more accurate in the QCIF than in the CIF sequences. The reconstructed PSNR-rate plots for the BUS sequence are shown in Fig. 5.17, which clearly illustrates the benefit of using the linear model in QCIF size sequences.

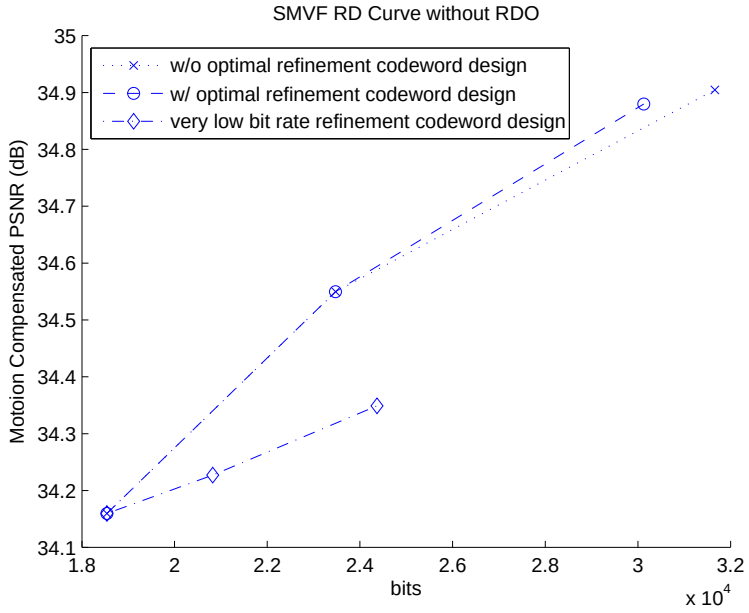
5.6 Acknowledgement

Portions of this chapter appear in "A Fully Scalable Motion Model for Scalable Video Coding", Meng-Ping Kao and Truong Nguyen, in the June 2008 issue of the *IEEE Transactions on Image Processing*; "A Fully Scalable Motion Model for Scalable Video Coding", Meng-Ping Kao and Truong Nguyen, in Proceedings of the *IEEE International Conference on Image Processing*, Sep. 2007; and also in "Coding and Optimization of a Fully Scalable Motion Model", Meng-Ping Kao and Truong Nguyen, in Proceedings of the *SPIE Applications of Digital Image*

Processing, Oct. 2007. The dissertation author was the primary author of these publications, and the listed co-author directed and supervised the research that forms the basis for this chapter.



(a)



(b)

Figure 5.15: RD curve comparison using different SMVD refinement codebook designs. (a) With RDO. (b) Without RDO.

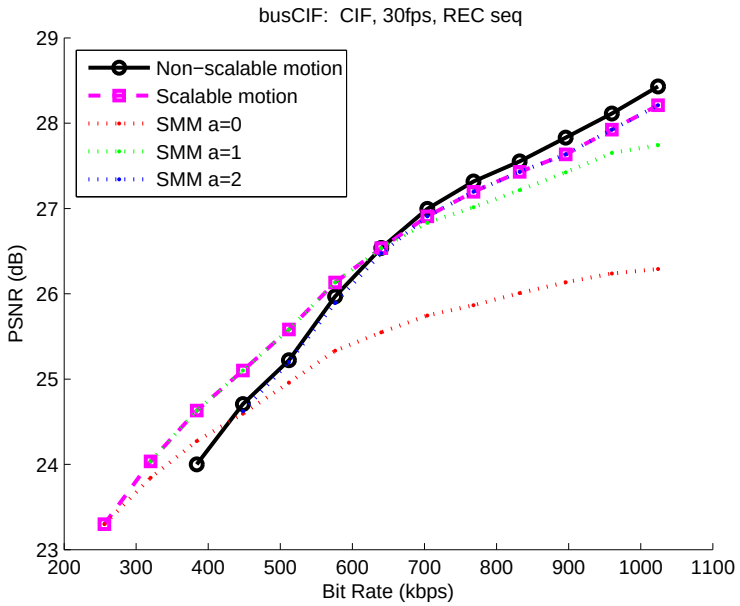
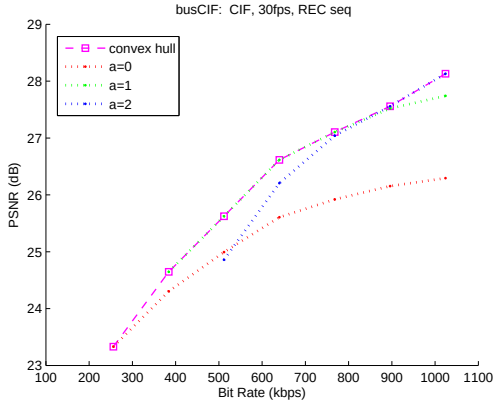
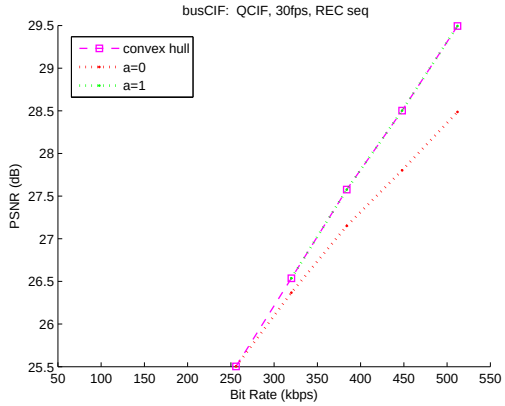


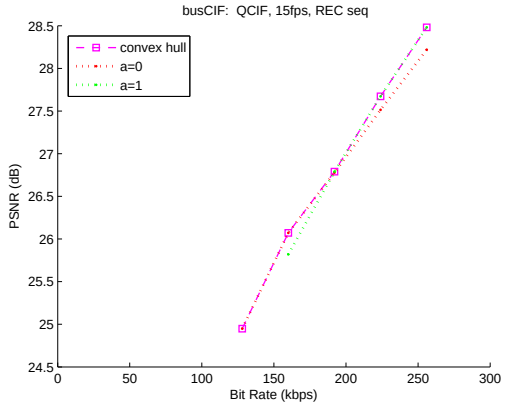
Figure 5.16: BUS RD curve comparison between non-scalable motion and proposed SMM.



(a)



(b)



(c)

Figure 5.17: BUS reconstructed PSNR plots. (a) CIF 30 *fps*. (b) QCIF 30 *fps*. (c) QCIF 15 *fps*.

6 Conclusions

This dissertation focuses mainly on scalable motion, a topic that has flourished with the rapid development of scalable video coding.

We approach this topic in various ways and from different point of views. First, the redundancy of motion information required for decoding lower resolution sequences is identified in Chapter 3. This is also a common entry point to the field of scalable motion, as illustrated in the history review in Chapter 4. The remainder of Chapter 4 involves a full understanding of required functionalities of scalable motion within an SVC framework. The importance of scalable motion on quality scalability is realized, and the mathematical model for scalable motion is built. Based on this model and the associated assumptions, the advantages of scalable motion can be easily understood and an optimal bitstream extractor can also be designed accordingly.

Integrating our knowledge from comprehensive studies on scalable motion, we propose a novel and fully scalable motion model for SVC based on block motion. The proposed model helps to achieve coding optimality over a wide range of bit rates, resolutions and frame rates. The tailored rate distortion optimization algorithm provides the tool, via the newly introduced rate and distortion multipliers, to further optimize the coding efficiency towards a preferred decoding scenario, with minimal degradation to other scenarios. In order to provide better tools for SMM coding, which in turn reduce the cost in the RDO process, several optimal and suboptimal algorithms are proposed, including SMM structure coding and optimal codebook design for SMVD refinements. Exhaustive simulations have shown positive results that verify the promised functionalities of our SMM.

Finally, hybrid video coding, which efficiently explores the great temporal redundancy of natural video sequences by motion prediction, has dominated the digital video coding technologies for more than 30 years. Even with the emergence of various scalability demands, e.g. SVC, its status as the key core in virtually every codec design remains unchanged. However, the initial consensus of lossless coding on non-scalable motion information no longer holds, especially for low resolution or low bit rate decoding scenarios in SVC. This phenomenon opens a new research field, i.e. scalable motion. In this dissertation, we extensively discuss this topic, with a possibly outdated premise that the coded texture is obtained as the difference between the current block and the best motion predicted block. However, with the introduction of scalable motion, other versions of coded texture are also available while adjusting the motion quality layers. This is another research topic that requires more investigation. In addition, an accurate model describing the relationship between $\{\lambda_a\}$ and $\{R^a\}$ is also important for practical applications.

Bibliography

[1] Codecs for video conferencing using primary digital group transmission. Technical report, ITU-T Rec. H.120, ITU-T, 1984.

[2] Generic coding of moving pictures and associated audio information part 2: video. Technical report, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2 Video), ITU-T and ISO/IEC JTC 1, Nov. 1994.

[3] Information technology - JPEG 2000 image coding system - Part 1: Core coding system. Technical report, ISO/IEC 15444-1:2000, 2000.

[4] Coding of audio-visual objects part 2: visual. Technical report, ISO/IEC 14492-2 (MPEG-4 Visual), ISO/IEC JTC 1, Version 1: Apr. 1999, Version 2:Feb. 2000, Version 3: May 2004.

[5] Advanced video coding for generic audiovisual services. Technical report, ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1, Version 1: May 2003, Version 2: May 2004, Version 3: Mar. 2005, Version 4: Sept. 2005, Version 5 and Version 6: June 2006, Version 7: Apr. 2007, Version 8 (including SVC extension): Nov. 2007.

[6] Video codec for audiovisual services at p x 64 kbit/s. Technical report, ITU-T Rec. H.261, ITU-T, Version 1: Nov. 1990, Version 2: Mar. 1993.

[7] Video coding for low bit rate communication. Technical report, ITU-T Rec. H.263, ITU-T, Version 1: Nov. 1995, Version 2: Jan. 1998, Version 3: Nov. 2000.

[8] N. Adami, M. Brescianini, R. Leonardi, and A. Signoroni. SVC CE1: STool - A native spatially scalable approach to SVC ISO/IEC JTC1/SC29/WG11. 70th MPEG Meeting, Palma de Mallorca, Spain, M11368, Oct. 2004.

[9] N. Adami, A. Signoroni, and R. Leonardi. State-of-the-art and trends in scalable video compression with wavelet-based approaches. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1238–1255, Sept. 2007.

[10] L. Alparone, M. Barni, F. Bartolini, and V. Cappellini. Adaptively weighted vector-median filters for motion-fields smoothing. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2267–2270, 1996.

[11] I. Amonou, N. Cammas, S. Kervadec, and S. Pateux. Optimized rate-distortion extraction with quality layers in the scalable extension of H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1186–1193, 2007.

[12] Y. Andreopoulos, M. van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens, and J. Cornelis. Fully-scalable wavelet video coding using in-band motion compensated temporal filtering. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 417–420, Apr. 2003.

[13] J. Barbarien, Y. Andreopoulos, A. Munteanu, P. Schelkens, and J. Cornelis. Motion vector coding for in-band motion compensated temporal filtering. In *Proceedings IEEE International Conference on Image Processing*, volume 2, pages 783–786, Sept. 2003.

[14] J. Barbarien, A. Munteanu, F. Verdicchio, Y. Andreopoulos, J. Cornelis, and P. Schelkens. Scalable motion vector coding. In *Proc. IEEE Int. Conf. Image Process.*, volume 2, pages 1321–1324, 2004.

[15] J. Barbarien, A. Munteanu, F. Verdicchio, Y. Andreopoulos, J. Cornelis, and P. Schelkens. Motion and texture rate-allocation for prediction-based scalable motion-vector coding. *EURASIP Signal Processing: Image Communication*, 20:315–342, Apr. 2005.

[16] H. Bi and W.Y. Chan. Rate-distortion optimization of hierarchical displacement fields. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(1):18–24, 1998.

[17] V. Bottreau, M. Benetiere, B. Felts, and B. Pesquet-Popescu. A fully scalable 3D subband video codec. In *Proc. IEEE Int. Conf. Image Process.*, volume 2, pages 1017–1020, Oct. 2001.

[18] P. Burt and E. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.

[19] M.H. Chan, Y.B. Yu, and A.G. Constantinides. Variable size block matching motion compensation with applications to video coding. *IEE Proceedings I Communications, Speech and Vision*, 137(4):205–212, 1990.

[20] L. Chau, Y. Liang, and Y. Tan. Motion vector re-estimation for fractional-scale video transcoding. In *Proceedings IEEE International Conference on Information Technology: Coding and Computing*, pages 212–215, Apr. 2001.

[21] J.J. Chen and H.M. Hang. Source model for transform video coder and its application. II. Variable frame rate coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(2):299–311, 1997.

[22] P. Chen and J.W. Woods. Bidirectional MC-EZBC with lifting implementation. *IEEE Trans. Circuits and Syst. Video Technol.*, 14(10):1183–1194, Oct. 2004.

[23] G. Dane and T.Q. Nguyen. Smooth motion vector resampling for standard compatible video post-processing. In *Proceedings IEEE Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1731–1735, 2004.

[24] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):247–269, 1998.

[25] K. Fung and W. Siu. New DCT-domain transcoding using split and merge technique. In *Proceedings IEEE International Conference on Image Processing*, volume 1, pages 197–200, 2003.

[26] K. Fung and W. Siu. Diversity and importance measures for video downscaling. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1061–1064, 2005.

[27] Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[28] H. Gish and J. Pierce. Asymptotically efficient quantizing. *IEEE Transactions on Information Theory*, 14(5):676–683, 1968.

[29] W.J. Han. Responses of call for proposal for scalable video coding. ISO/IEC JTC1/SC29/WG11, 68th MPEG Meeting, Munich, Germany, M10569/S17, Mar. 2004.

[30] H.M. Hang and J.J. Chen. Source model for transform video coder and its application. I. Fundamental theory. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(2):287–298, 1997.

[31] H.M. Hang, S.S. Tsai, and T. Chiang. Motion information scalability for MC-EZBC: response to call for evidence on scalable video coding. ISO/IEC JTC1/SC29/WG11, 65th MPEG Meeting, Trondheim, Norway, M9756, Jul. 2003.

[32] S. Hsiang and J.W. Woods. Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling. In *Proceedings IEEE International Symposium on Circuits and Systems*, volume 3, pages 662–665, 2000.

[33] S.T. Hsiang and J.W. Woods. Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank. *Signal Process.: Image Commun.*, 16:705–724, 2001.

[34] Z. Hu, van der Schaar M., and B. Pesquet-Popescu. Scalable motion vector coding for MC-EZBC. In *Proceedings European Signal Processing Conference*, pages 657–660, Sept. 2004.

[35] ISO/IEC JTC1/SC29/WG11. Call for proposals on scalable video coding technology. 67th MPEG Meeting, Waikoloa, HI, N6193, Dec. 2003.

[36] ISO/IEC JTC1/SC29/WG11. Wavelet codec reference document and software manual. 73th MPEG Meeting, Poznan, Poland, N7334, Jul. 2004.

[37] ISO/IEC JTC1/SC29/WG11. Technical description of the Thomson proposal for SVC CE6, nonscalable motion vector coding. 72th MPEG Meeting, Busan, M12063, Apr. 2005.

[38] ISO/IEC JTC1/SC29/WG11. Text description of joint model reference encoding methods and decoding concealment methods. 75th MPEG Meeting, Hong Kong, N046, Jan. 2005.

[39] ISO/IEC JTC1/SC29/WG11. Text description of joint model reference encoding methods and decoding concealment methods. 71st MPEG Meeting, Hong Kong, N046, Jan. 2005.

[40] ISO/IEC JTC1/SC29/WG11. Joint scalable video model JSVM-9. 79th MPEG Meeting, Marrakech, Morocco, JVT-U202, Jan. 2007.

[41] M.P. Kao and T.Q. Nguyen. Motion vector field manipulation for complexity reduction in scalable video coding. In *Proc. IEEE Asilomar Conf. Signals Syst. Comput.*, pages 1095–1098, 2006.

[42] J.W. Kim and S.U. Lee. On the hierarchical variable block size motion estimation technique for motion sequence coding. *SPIE Visual Commun. Image Process.*, 1993.

[43] T. Kim and M.H. Ammar. Optimal quality adaptation for scalable encoded video. *IEEE Journal on Selected Areas in Communications*, 23(2):344–356, 2005.

[44] Y.S. Kim, Y.J. Jung, T.C. Thang, and Y.M. Ro. Bit-stream extraction to maximize perceptual quality using quality information table in SVC. volume 6077, page 607723. SPIE, 2006.

[45] J. Lee. Joint optimization of block size and quantization for quadtree-based motion estimation. *IEEE Transactions on Image Processing*, 7(6):909–912, 1998.

[46] D. Maestroni, A. Sarti, M. Tagliasacchi, and S. Tubaro. Fast in-band motion estimation with variable size block matching. In *Proceedings IEEE International Conference on Image Processing*, volume 4, pages 2287–2290, Oct. 2004.

[47] D. Maestroni, A. Sarti, M. Tagliasacchi, and S. Tubaro. Scalable coding of variable size blocks motion vectors. In *Proc. IEEE Int. Conf. Image Process.*, volume 2, pages 1333–1336, 2004.

[48] C. Mayer. Motion compensated in-band prediction for wavelet-based spatially scalable video coding. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 73–76, 2003.

[49] M. Mrak, G.C.K. Abhayaratne, and E. Izquierdo. On the influence of motion vector precision limiting in scalable video coding. In *Proc. IEEE Int. Conf. Image Process.*, volume 2, pages 1143–1146, 2004.

[50] M. Mrak, N. Sprljan, and E. Izquierdo. A resolution adaptive interpolation technique for enhanced decoding of scalable coded video. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 353–356, Mar. 2005.

[51] M. Mrak, N. Sprljan, and E. Izquierdo. Evaluation of techniques for modeling of layered motion structure. In *Proc. IEEE Int. Conf. Image Process.*, pages 1905–1908, 2006.

[52] J.R. Ohm. Three-dimensional subband coding with motion compensation. *IEEE Trans. Image Process.*, 3(5):559–571, Sept. 1994.

[53] J.R. Ohm. Advances in scalable video coding. *IEEE Proceedings*, 93(1):42–56, Jan. 2005.

[54] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing (2nd Edition)*. Prentice Hall, 1999.

[55] A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, 15(6):23–50, 1998.

[56] Y. Peng, J. Boyce, and P. Pandit. Technical description of the Thomson proposal for SVC CE4. ISO/IEC JTC1/SC29/WG11, 71th MPEG Meeting, Hong Kong, M11682, Jan. 2005.

[57] B. Pesquet-Popescu and V. Bottreau. Three-dimensional lifting schemes for motion compensated video compression. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1793–1796, 2001.

[58] K. R. Rao and P. C. Yip. *The Transform and Data Compression Handbook*. CRC, 2001.

[59] A. Said and W.A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits Syst. Video Technol.*, 6(3):243–250, Jun. 1996.

[60] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, 2007.

[61] A. Secker and D. Taubman. Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression. *IEEE Transactions on Image Processing*, 12(12):1530–1542, 2003.

[62] A. Secker and D. Taubman. Highly scalable video compression with scalable motion coding. *IEEE Trans. Image Process.*, 13(8):1029–1041, Aug. 2004.

[63] C.A. Segall and G.J. Sullivan. Spatial scalability within the H.264/AVC scalable video coding extension. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1121–1135, 2007.

[64] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, 1993.

[65] D. Shckler, Y. Ozturk, and H. Abut. Variable size block motion estimation. In *Proceedings IEEE Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 868–872, Nov. 1998.

[66] B. Shen, I.K. Sethi, and B. Vasudev. Adaptive motion-vector resampling for compressed video downscaling. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(6):929–936, 1999.

[67] Gilbert Strang and Truong Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1997.

[68] G. Sullivan and T. Wiegand. Rate distortion optimization for video compression. *IEEE Signal Process. Magazine*, 15:74–90, Nov. 1998.

[69] G.J. Sullivan and R.L. Baker. Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks. In *Proceedings IEEE Global Telecommunications Conference*, pages 85–90, Dec. 1991.

[70] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.

[71] D. Taubman. High performance scalable image compression with EBCOT. In *Proceedings IEEE International Conference on Image Processing*, volume 3, pages 344–348, Oct. 1999.

[72] D. Taubman and A. Secker. Highly scalable video compression with scalable motion coding. In *Proceedings IEEE International Conference on Image Processing*, volume 3, pages 273–276, Sept. 2003.

[73] A.M. Tourapis and J. Boyce. Reduced resolution update mode extension to the H.264 standard. In *Proceedings IEEE International Conference on Image Processing*, volume 2, pages II–910–913, 2005.

[74] S.S. Tsai and H.M. Hang. Motion information scalability for MC-EZBC. *Signal Process.: Image Commun.*, 19:675–684, Aug. 2004.

[75] D.S. Turaga, M. van der Schaar, and B. Pesquet-Popescu. Temporal prediction and differential coding of motion vectors in the MCTF framework. In *Proceedings IEEE International Conference on Image Processing*, volume 2, pages 57–60, Sept. 2003.

[76] V. Valentin, M. Cagnazzo, M. Antonini, and M. Barlaud. Scalable context-based motion vector coding for video compression. In *Proceedings EURASIP Picture Coding Symposium*, 2003.

[77] M. van der Schaar and D.S. Turaga. Unconstrained motion compensated temporal filtering (UMCTF) framework for wavelet video coding. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 81–84, 2003.

[78] J.S. Walker and T.Q. Nguyen. Adaptive scanning methods for wavelet difference reduction in lossy image compression. In *Proceedings IEEE International Conference on Image Processing*, volume 3, pages 182–185, 2000.

[79] Y. Wang, S.F. Chang, and A.C. Loui. Subjective preference of spatio-temporal rate in video adaptation using multi-dimensional scalable coding. In *Proceedings IEEE International Conference on Multimedia and Expo*, volume 3, pages 1719–1722, 2004.

[80] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and J. Sullivan. Rate-constrained coder control and comparison of video coding standards. *IEEE Trans. Circuits Syst. Video Technol.*, 13:688–703, Jul. 2003.

[81] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):560–576, Jul. 2003.

[82] Y. Wu, A. Golwelkar, and J. Woods. MC-EZBC video proposal from Rensselaer Polytechnic Institute. ISO/IEC JTC1/SC29/WG11, 68st MPEG Meeting, Munich Germany, M10569/S15, Mar. 2004.

[83] Y. Wu and J. W. Woods. Scalable motion vector coding based on CABAC for MC-EZBC. *IEEE Trans. Circuits Syst. Video Technol.*, 17(6):790–795, Jun. 2007.

[84] R. Xiong, J. Xu, F. Wu, S. Li, and Y.Q. Zhang. Layered motion estimation and coding for fully scalable 3D wavelet video coding. In *Proc. IEEE Int. Conf. Image Process.*, volume 4, pages 2271–2274, 2004.

[85] J. Xu, R. Xiong, B. Feng, G. Sullican, M. C. Lee, F. Wu, and S. Li. 3D subband video coding using barbell lifting. ISO/IEC JTC1/SC29/WG11, 68st MPEG Meeting, Munich Germany, M10569/S05, Mar. 2004.

[86] J. Xu, Z. Xiong, S. Li, and Y.Q. Zhang. Three-dimensional embedded subband coding with optimized truncation (3-D ESCOT). *Applied and Computational Harmonic Analysis*, 10(3):290–315, May 2001.

[87] P. Yang, Y.W. He, and S.Q. Yang. An unsymmetrical-cross multi-resolution motion search algorithm for MPEG4-AVC/H.264 coding. In *Proceedings IEEE International Conference on Multimedia and Expo*, volume 1, pages 531–534, 2004.

[88] Y.H. Yu and C.J. Tsai. A model-based rate allocation mechanism for wavelet-based embedded image and video coding. In *Proceedings IEEE International Symposium on Circuits and Systems*, volume 6, pages 6066–6069, 2005.

[89] T. Zgaljic, N. Sprljan, and E. Izquierdo. Bitstream syntax description based adaptation of scalable video. In *Proceedings European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*, pages 173–178, Nov. 2005.