IMPROVED REFINEMENT COEFFICIENTS' CODING IN SCALABLE H.264

by

RAHUL PANCHAL

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2007

ACKNOWLEDGEMENTS

I am grateful to my advisor Dr. K. R. Rao for his continued encouragement and guidance throughout this thesis. Sincere thanks to UTA, my Alma Mater, for recognizing my aptitude and giving me an opportunity to pursue my master's studies under the guidance of noble faculty and for providing me with all the required resources and environment to excel.

I would also like to mention my gratitude to Dr. Marta Karczewicz, Senior Manager of Video Team, Corp R&D dept (CRD), Qualcomm, for the support and trust she has shown in me to carry out this work and for all the resources provided at Qualcomm facility. I am also grateful to Dr. Scott Ludwin and the entire video codec team at CRD, Qualcomm. Without their cooperation this thesis would not be complete. Thanks to all my colleagues at Qualcomm for all their help and cooperation in completion of this thesis.

I would also like to sincerely thank my thesis committee members Dr. S. Oraintara and Dr. Z. Wang for being a part of the culmination of my thesis.

Finally, I would like to take this opportunity to thank my family and above all GOD who has always supported me in my endeavors, without whom it would not have been possible to come this far.

April 20, 2007

ABSTRACT

IMPROVED REFINEMENT COEFFICIENTS' CODING IN SCALABLE H.264

Publication No. _____

Rahul Panchal, M.S.

The University of Texas at Arlington, 2007

Supervising Professor: Dr. K. R. Rao

This thesis proposes to replace the adaptation used to select VLC table for coding of the refinement coefficients by signaling to the decoder which table should be used for which macroblock type (Inter or Intra) in scalable H.264. This part of the proposal aims at reducing decoder complexity and ensuring proper table selection when both macroblock types are present within one slice.

The contribution further proposes to extend the method used in SVC to increase coding efficiency of CABAC refinement coefficient coding to “VLC” refinement coefficients coding. All the coding tools are integrated into JSVM_7_10 and the performance is tested. The proposed changes mainly affect Intra coded slices and for these slices the improvements are in the range of 3-7% for 3 FGS layers.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF ACRONYMS

AR : Adaptive Reference

AVC: Advanced Video Coding

CABAC: Context-based Adaptive Binary Arithmetic Coding

CAVLC: Context-based Adaptive Variable length Coding

CGS: Coarse Granularity Scalability

DB: Decibel

DC: Direct Current

DCT: Discrete Cosine Transform

DPB: Decoded Picture Buffer

DSP: Digital Signal Processor

DVD: Digital Video Disc

DWT: Discrete Wavelet Transform

FGS: Fine Granularity Scalability

FREXT: Fidelity Range Extensions

GOP: Group of Pictures

HDTV: High Definition Television

HSDPA: High Speed Downlink Packet Access

HVS: Human Visual System

IDR: Instantaneous Decoding Refresh

IEC: International Engineering Consortium

ILP: Inter-Layer Prediction

ISO: International Standards Organization

ITU: International Telecommunication Union

JD: Joint Draft

JM: Joint Model

JVT: Joint Video Team

KLT: Karhunen-Loève Transform

MANE: Media-Aware Network Elements

MB: Macroblock

MC: Motion Compensation

MCPE: Motion Compensated Prediction Error

MCTF: Motion-Compensated Temporal Filtering

ME: Motion Estimation

MMCO: Memory Management Control Operation

MPEG: Motion Picture Experts Group

MSE: Mean Square Error

MVC: Multi-View Coding

NALU: Network Abstraction Layer Unit

NTSC: National Television System(s) Committee

PAL: Phase Alternating Line

POCS: Projection onto Convex Sets

PR: Progressive Refinement

PSNR: Peak Signal to Noise Ratio

QP: Quantization Parameter

RGB: Red Green Blue

ROI: Region-Of-Interest

RPLR: Reference Picture List Re-ordering

RV: Real Video

SI: Switching Intra

SIMD: Single Instruction Multiple Data

SMPTE: Society of Motion Picture and Television Engineers

SNR: Signal-to-Noise Ratio

SP: Switching Prediction

SQ: Scalar Quantization

SVC: Scalable Video Coding

SVD: Singular Value Decomposition

UMTS: Universal Mobile Telecommunications Service

VCEG: Video Coding Experts Group

VCL: Video Coding Layer

VLC: Variable Length Coding

VLIW: Very Long Instruction Word

VOD: Video On Demand

VQ: Vector Quantization

CHAPTER 1

INTRODUCTION

1.1 Background

With a growing number of devices capable of receiving or displaying digital video signals, there is a need for compressed video sequences to be "scalable" so that data can be selectively removed from the bit stream to yield a sequence with degraded characteristics, such as lower frame rate, spatial resolution, or visual quality. This need for scalability has been acknowledged by the standardization bodies of ISO/IEC and ITU-T, which have tasked the JVT (Joint Video Team) with creating a scalable extension to the recent H.264/AVC [30] video coding standard. A stated requirement for the scalable extension is that it must exhibit Fine Granularity Scalability (FGS), defined as the ability to scale the bit rate in rate increments of 10% or less. Such scalability may be useful when video is transmitted over a shared medium without pre-allocation of bandwidth, so that the bit rate of the video can be reduced (by a router, for example) when necessary.

1.2 Outline of the work

In this thesis, we reviewed some challenges associated with FGS, and introduce an improved refinement coefficients coding to achieve coding gain with minimal loss in quality of the video. Correlation between the base layer and

enhancement layer has been exploited to achieve coding gain. And also the variable length coder (VLC) table used in current Joint Draft 6 (JD1) of the H.264/AVC scalable extension [85] draft to code refinement coefficients has been modified to best suit the refinement coefficients statistics at higher FGS layers so as to code them with minimum bits and thus achieving gain in coding efficiency.

The thesis is organized as follows: Chapter 2 provides basic theory behind video coding; Chapter 3 give the introduction to H.264/AVC video coding standard; Chapter 4 extends the idea from H.264 to scalable extension to H.264; Chapter 5 describes the proposed method on improved refinement coefficients coding and Chapter 6 concludes with the comparison of the obtained results and future research directions.

CHAPTER 2

BASIC VIDEO CODING THEORY

2.1 Video Compression

The digital video compression technology has been gaining popularity for many years [9]. Today, when people enjoy HDTV (high definition television), movie broadcasting through Internet or the digital music such as MP3, the convenience that the digital video industry brings to us cannot be forgotten. All of these should attribute to the advances in compression technology, enhancement on mass storage media or streaming video/audio services. As the main contributor to all of these, video compression technology is the focus of this chapter. Some basic video compression concepts will be introduced as the basis of chapter 3.

2.1.1 RGB and YUV color spaces

RGB (red-green-blue) color space is well suited to capture and display of color images. The image consists of three grayscale components (sometimes referred to as channels) [3]. The combination of red, green and blue with different weights can produce any visible color. A numerical value is used to indicate the proportion of each color. The drawback of RGB representation of color image is that all 3 colors are equally important and should be stored with the same amount of data bits. It is found

that HVS (human visual system) is less sensitive to color than to brightness. In order to take advantage of this finding, a new color space called YUV (luminance-chrominance (blue)-chrominance (red)) is proposed. Instead of using the color of the light, YUV chooses the luminance (Y) and chrominance (UV) of the light to represent a color image. YUV uses RGB information, but it creates a gray scale image (luma) from the full color image and then subtracts the three primary colors resulting in two additional signals (chroma /C_b, C_r) to describe color. Combining the three signals back together results in a full color image [3]. The luminance information Y can be calculated from R, G and B according to the following equations:

$$Y = k_r R + k_g G + k_b B \tag{2.1}$$

where k are the weighting factors, $k_r + k_g + k_b = 1$

The color difference information (Chroma) can be derived as:

$$C_b = \frac{0.5}{1-k_b}(B-Y) \tag{2.2}$$

$$C_r = \frac{0.5}{1-k_r}(R-Y) \tag{2.3}$$

In reality, only three components (Y, C_b and C_r) need to be transmitted for video coding because C_g (green chroma) can be derived from Y, C_b and C_r. As recommended by ITU-R [30], k_b=0.114, k_r = 0.299. The Equations (2.1) through (2.3) can be rewritten as:

$$Y = 0.299R + 0.587G + 0.114B \tag{2.4}$$

$$C_b = 0.564(B - Y) \tag{2.5}$$

$$C_r = 0.713(R - Y) \tag{2.6}$$

$$R = Y + 1.402C_r \tag{2.7}$$

$$G = Y - 0.344C_b - 0.714C_r \tag{2.8}$$

$$B = Y + 1.772C_b \tag{2.9}$$

In reality, images are looked as 2D arrays. Figure 2.1a shows the red, green and blue components of a color image in comparison to chroma components C_b, C_r and C_g of Fig. 2.1b.

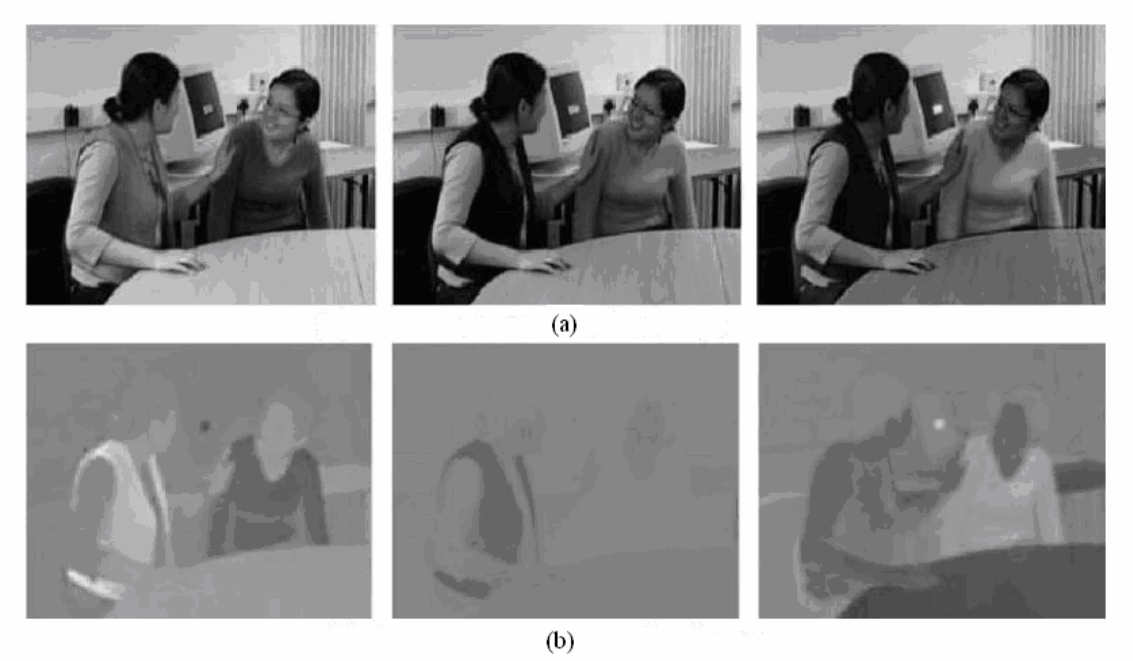


Figure 2.1 Components of an image: (a) R, G, B components, (b) C_b, C_r, C_g components [3]

2.1.2 Video Sampling

The video source is normally a bit stream consisting of a series of frames or fields in decoding order [30]. There are three YC_bC_r sampling modes supported by MPEG-4 and H.264 (Figure 2.2).

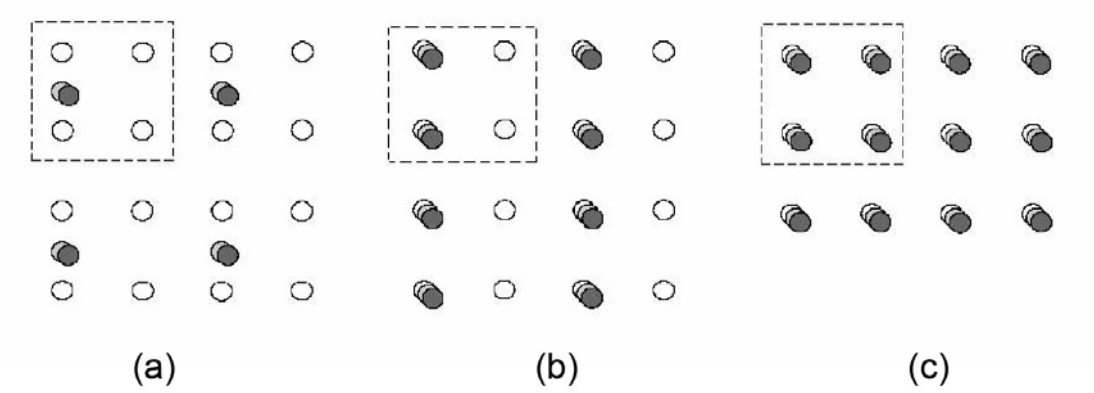


Figure 2.2 Color formats, (a) 4:2:0, (b) 4:2:2, (c) 4:4:4 [3]

4:2:0 is the most common used sampling pattern. The sampling interval of luminance sample Y is the same as the video source. The C_b and C_r have twice the sampling intervals as luminance on both vertical and horizontal directions (Figure 2.2a). In this case, every 4-luma samples have one C_b and one C_r sample. As HVS is less sensitive to color than to brightness, it is possible to reduce the resolution of chrominance part without degrading the image quality apparently. This makes 4:2:0 very popular in current video compression standards [3]. This mode is widely used for consumer applications such as video conferencing, digital television and DVD (digital versatile disc) storage. For 4:2:2 mode, C_b and C_r have the same vertical resolution as luma but half the horizontal resolution as luma (Figure 2.2b). This mode is used for high quality

color representation. 4:4:4 mode has the same resolution for Y, C_b and C_r on both directions (Fig. 2.2c).

2.1.3 Redundancy Reduction

The basic idea of video compression is to compress an original video sequence (raw video) into a smaller one with fewer numbers of bits. The compression is achieved by removing redundant information from the raw video sequence. There are totally three types of redundancies present: temporal, spatial and frequency domain redundancies.

Spatial and temporal redundancies: Pixel values are not independent, but are correlated with their neighbors both within the same frame and across frames [3]. Spatial redundancy is the little variation in the content of the image within a frame (Figure 2.3). Utilizing spatial redundancy, the value of a pixel is predictable from the known values of neighboring pixels. In time domain, there is little variation in frame content between consecutive frames except for the case when the object or content of the video is changing quickly. This is often known as temporal redundancy (Figure 2.3).

Figure 2.3 Spatial and temporal redundancies [3]

Frequency domain redundancy: The HVS is more sensitive to lower frequencies [6] than to higher frequencies.

2.1.4 Video Codec

The redundancies mentioned above can be removed by different methods. The temporal and spatial redundancies are often reduced by motion estimation/compensation and predictive schemes .The frequency redundancy is commonly reduced by DCT and quantization aided by HVS weighting. After these operations, entropy coding is employed to the data to achieve further compression.

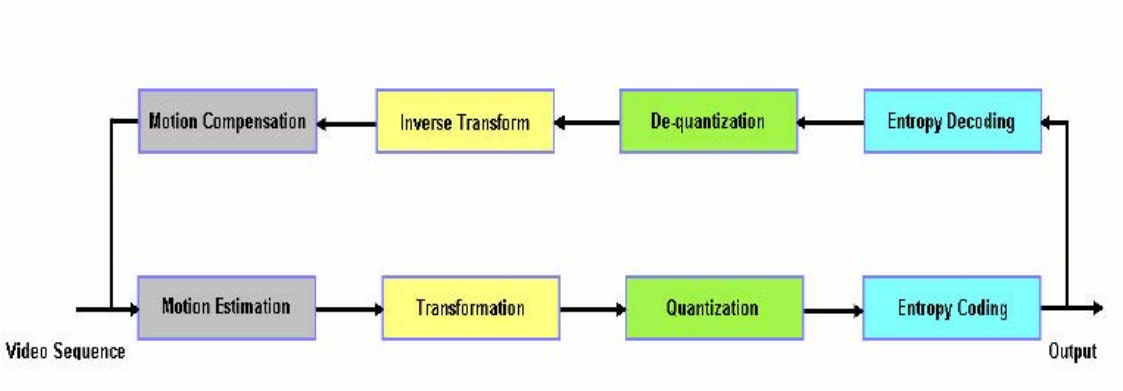


Figure 2.4 Common video coding flow [3]

Each function block of the common video coding flow (Figure 2.4) will be addressed in the order that it exists in the video coding process.

2.1.5 Motion Estimation

The input to the coding system is an uncompressed video sequence. In motion estimation, we find the best match for the current block from previous or future frames. We find the best match for current block by selecting an area in the reference

frame (past or future reference frames) within a search window that minimizes the residual energy (Figure 2.5). In motion compensation process, the chosen candidate region is subtracted from the current block to form a residual block.

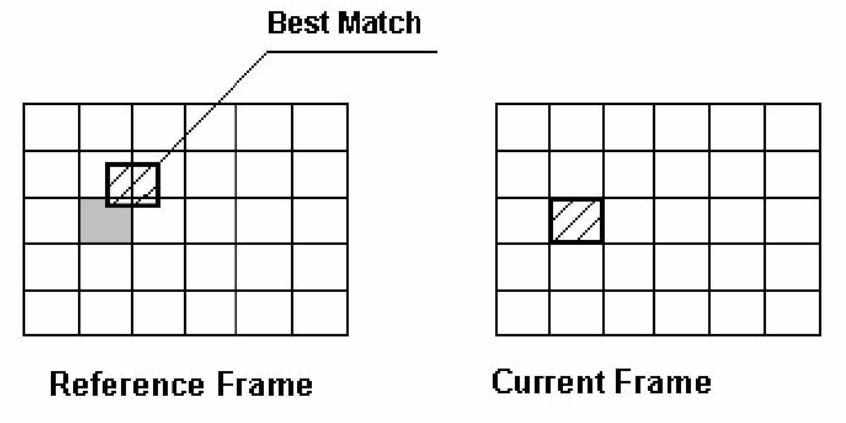


Figure 2.5 Motion estimation procedure

In practice, motion estimation and compensation are often based on rectangular blocks (MxN or NxN). The most common size of the block is 16x16 for luminance component and 8x8 for chrominance components (4:2:0 format). A 16x16 pixel region called macroblock is the basic data unit for motion compensation in current video coding standards (MPEG series and ITU-T series). It consists of one 16x16 luminance sample block, one 8x8 C_b sample block and one 8x8 Cr sample block (Figure 2.6). In MPEG-2, there is also 16x8 block ME for field based coding [7].

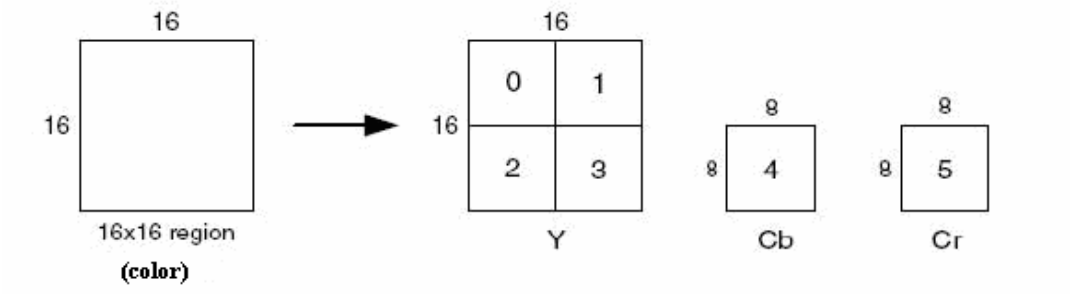


Figure 2.6 Macro block representation in 4:2:0 format

Theoretically, the smaller the block size is (in 4:2:0 format), the better the motion estimation performance.

2.1.6 Motion Vectors

Motion vector is a two-value pair (Δx, Δy), which indicates the relative position offsets of the current macroblock compared to its best matching region in both vertical and horizontal directions (Figure 2.7). Motion vector is encoded and transmitted together with the residual.

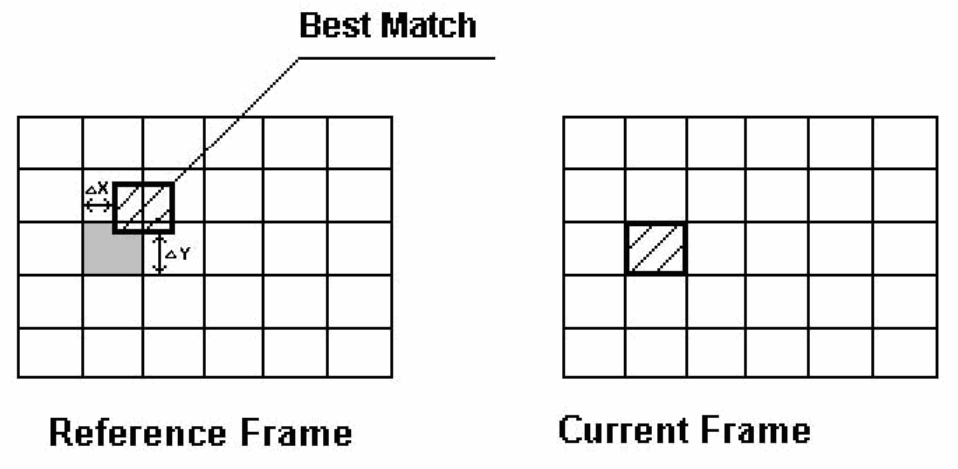


Figure 2.7 Motion vector representation

During the decoding process, the residual should be added to the matching region to recover the current frame. With the help of motion vectors, the matching region can be found from the reference frame.

2.1.7 Block Size Effect

H.264/AVC supports motion compensation block sizes ranging from 16x16 to 4x4 luminance samples with many options between the two. The luminance component of each macroblock (16x16 samples) may be split up in 4 ways as shown in Figure 2.8: 16x16, 16x8, 8x16 or 8x8. Each of the sub-divided regions is a macroblock partition. If the 8x8 mode is chosen, each of the four 8x8 macroblock partitions within the macroblock may be split in a further 4 ways as shown in Figure 2.8: 8x8, 8x4, 4x8 or 4x4 (known as macroblock sub-partitions). These partitions and sub-partitions give rise to a large number of possible combinations within each macroblock. This method of partitioning macroblocks into motion compensated sub-blocks of varying size is known as *tree structured motion compensation*.

Figure 2.9 shows the residual of 2 successive frames based on different block sizes. Figure 2.9a and Figure 2.9b are the original frames. Figure 2.9c is the residual without motion estimation. Figure 2.9d, Figure 2.9e and Figure 2.9f are the MCPE (motion compensated prediction errors) based on 16x16, 8x8 and 4x4 block (Figure 2.8) motion estimations respectively. Residual is the difference of frame 1 and frame 2. The mid-gray in the residual indicates that the subtract result is zero. The light or dark in the residual indicates the result is positive or negative. The more mid-gray area is, the more redundant information is reduced. In order to achieve higher compression efficiency, H.264 has chosen smaller block size for motion estimation. However, as the redundant information within residual is reduced, there is increase in

motion vectors that need to be encoded and transmitted. Therefore H.264 supports changing the block size dynamically according to the content of the frame.

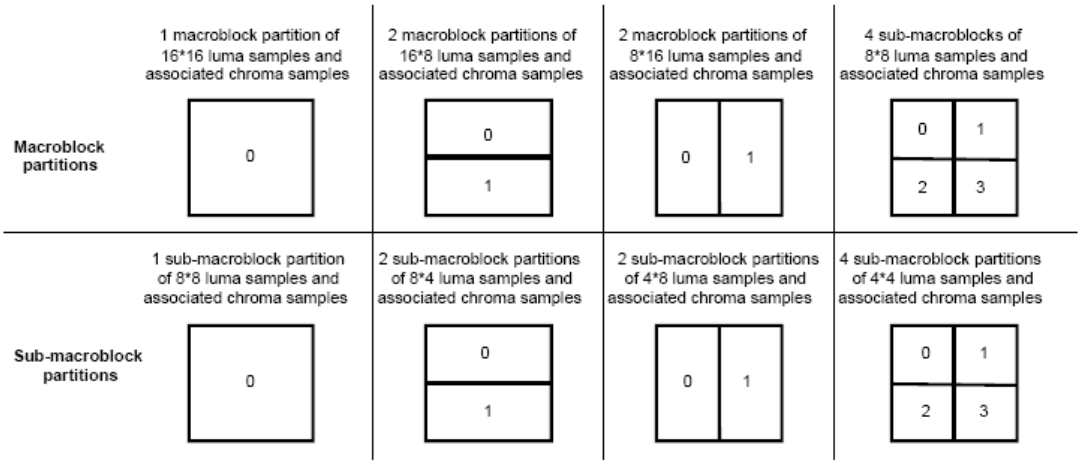


Figure 2.8 Macroblock partitions for motion estimation and compensation [18]

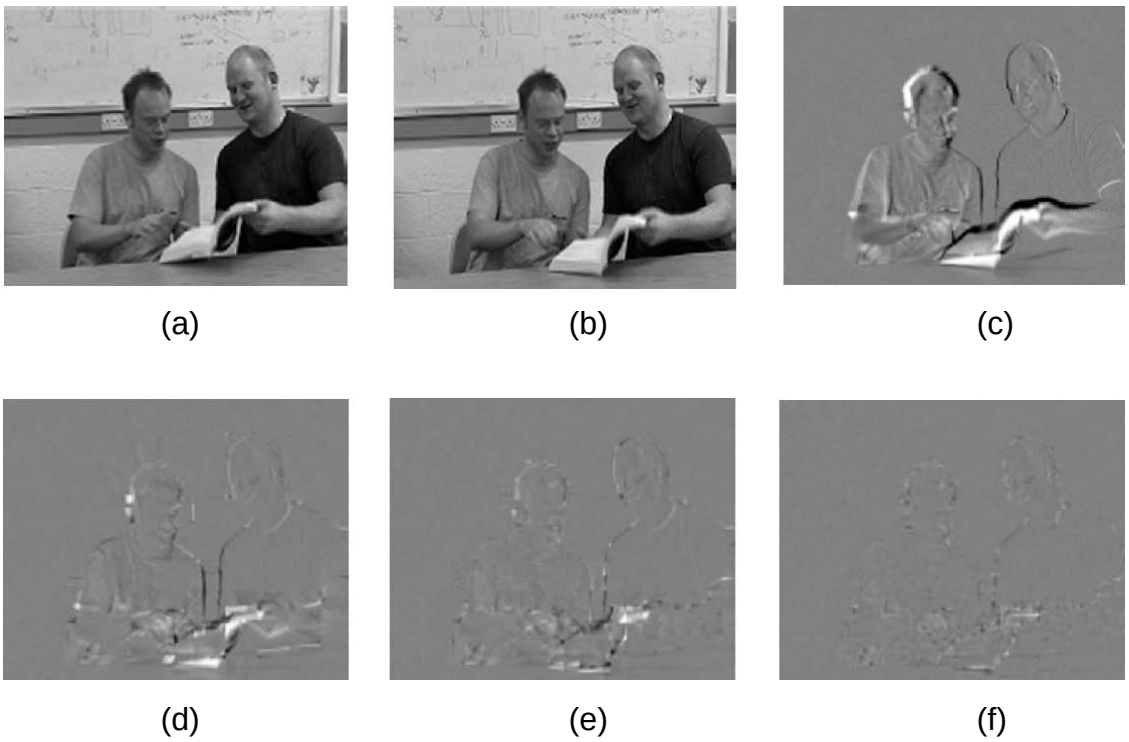


Figure 2.9 Block size effects on motion estimation, (a) Frame 1, (b) Frame 2, (c) No motion estimation (Inter-frame difference), (d) 16x16 block (MCPE), (e) 8x8 block (MCPE), (f) 4x4 block (MCPE) [3].

2.1.8 Sub-pixel Interpolation

The accuracy of motion compensation is in units of distance between pixels. In case the motion vector points to an integer-sample position, the prediction signal consists of the corresponding samples of the reference picture; otherwise the corresponding sample is obtained using interpolation to generate non-integer positions [3]. Non-integer position interpolation (Figure 2.10) gives the encoder more choices when searching for the best matching region compared to integer motion estimation; and thus the redundancy in the residual can be reduced further.

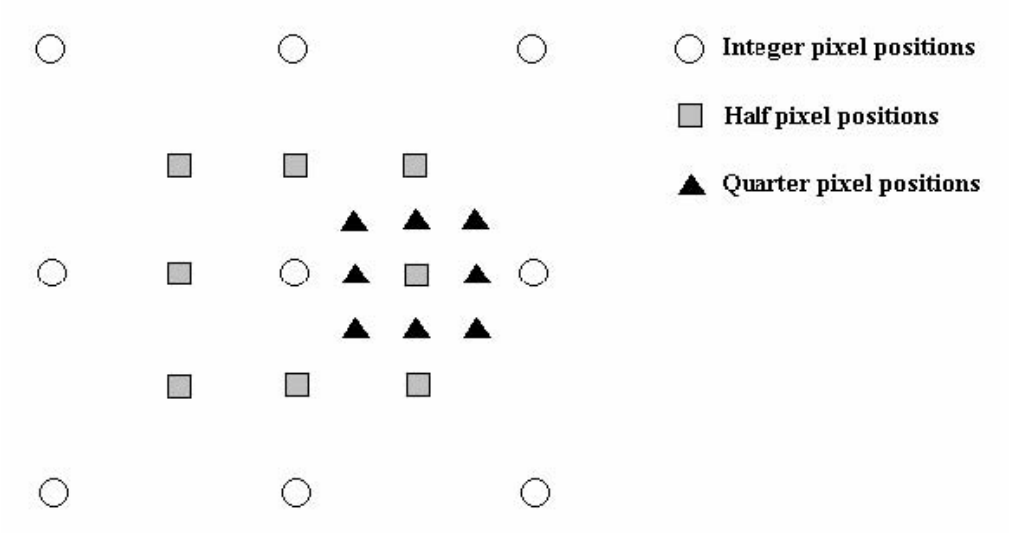


Figure 2.10 Sub-pixel interpolation [3]

2.1.9 Discrete Cosine Transform

After the motion estimation, the residual data can be converted into another domain (transform domain) to minimize the frequency redundancy. The choice of a transform depends on number of criteria: a) Data in the transform domain should be decorrelated and compact. b) The transform should be reversible. c) The transform

should be computationally tractable. The most popular transforms fall into two categories: block based and image based. Examples of block-based transforms include KLT (Karhunen-Loève Transform), SVD (Singular Value Decomposition) and DCT (Discrete Cosine Transform). Examples of image-based transforms include DWT (Discrete Wavelet Transform). DCT is the most popular transform of all these and is being currently employed in most video coding standards [93].

H.264/AVC employs smaller size of transform (4x4 Integer DCT and in FREXT 8x8 Integer DCT) compared to earlier standards. There is a tradeoff associated with the size of transform used. The large transforms can provide a better energy compaction and better preservation of detailed features in a quantized signal than a small transform does. But at the same time large transform introduces more ringing artifacts caused by quantization than small transform does.

2.1.10 Quantization

After DCT, quantization is employed to truncate the magnitude of DCT coefficients in order to reduce the number of bits that represent the coefficients. Quantization can be performed on each individual coefficient, which is known as scalar quantization (SQ). This can also be performed on a group of coefficients together, and this is known as vector quantization (VQ) [8].

2.1.11 Zigzag Scan

After quantization, most of the non-zero DCT coefficients are located close to the upper left corner in the matrix. Through zigzag scan (Figure 2.11), the order of the coefficients is rearranged in the order that most of the zeros are grouped together in the output data stream. In the following stage using run length coding, this string of zeros can be encoded with very few numbers of bits.

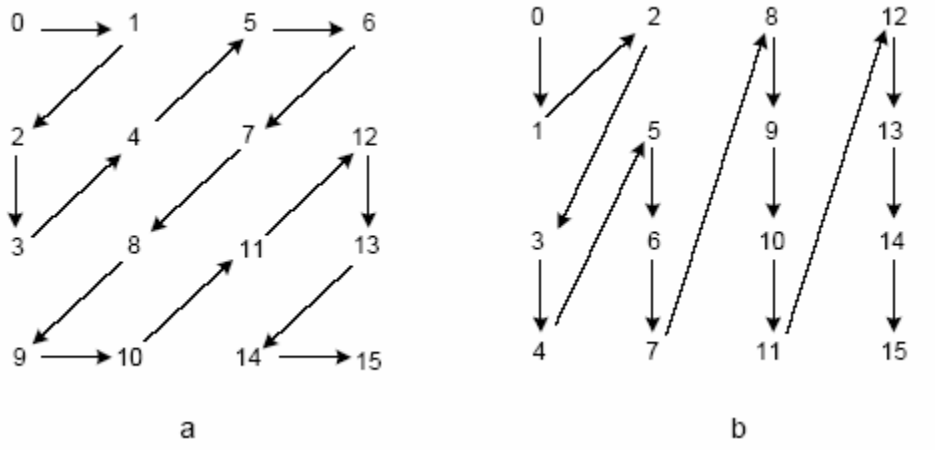


Figure 2.11 4x4 block scans. (a) Zig-zag scan. (b) Field scan (informative)[30]

2.1.12 Run-length Encoding

Run-length coding chooses to use a series of (run, level) pairs to represent a string of data. For example: For an input data array: {2, 0, 0, 0, 5, 0, 3, 7, 0, 0, 0, 1….} the output (run, level) pairs are: (0, 2), (3, 5), (1, 3), (0, 7), (3, 1)… Run here means the number of zeros in a data before the next non-zero data. Level is the value of the non-zero data. For details refer to [10].

2.1.13 Entropy Coding

The last stage in Figure 2.4 is entropy coding. Entropy encoder compresses the quantized data into smaller number of bits for future transmission. This is achieved by giving each value a unique code word based on its probability in the data stream. The more the probability of a value, the fewer bits are assigned to its code word. The most commonly used entropy coders are the Huffman encoder and the arithmetic encoder, though for applications requiring fast execution, simple run-length encoding (RLE) has proven very effective [10].

Two advanced entropy-coding methods know as CAVLC (Context Adaptive Variable Length Coding) [10] and CABAC (Context Adaptive Binary Arithmetic Coding) [10] are adopted by H.264/AVC. These two methods have improved coding efficiency compared to the methods applied in previous standards.

2.2 MPEG and H.26x

2.2.1 ISO/IEC, ITU-T and JVT

ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) and ITU-T (International Telecommunication Union) are two main international standards organizations for recommending coding standards of video, audio and their combination. H.26x family of standards is designed by ITU-T. As the ITU Telecommunication Standardization Sector, ITU-T is a permanent organ of ITU responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a

world-wide basis. H.261 is the first version of H.26x series started since 1984. During the following years, H.262, H.263, H.263+, H.263++ and H.264 are released by ITU-T subsequently [30].

The MPEG (moving picture experts group) family of standards includes MPEG-1, MPEG-2, MPEG-4, MPEG-7 and MPEG-21 [11] formally known as ISO/IEC-11172, ISO/IEC-13818 and ISO/IEC-14496. MPEG is originally the name given to the group of experts that developed these standards. The MPEG working group (formally known as ISO/IEC JTC1/SC29/WG11) is part of JTC1, the Joint ISO/IEC Technical Committee on Information Technology. The Joint Video Team (JVT) consists of members from ISO/IEC JTC1/SC29/WG11 (MPEG) and ITU-T SG16 Q.6 (VCEG). They published H.264 Recommendation/MPEG-4 part 10 standard [30].

2.2.2 H.261

H.261 is first developed by ITU-T in 1990. It is a video compression standard, which targets on low bit- rate real time applications (down to 64 kbps), such as visual telephone service. The basic idea of video coding is based on DCT, VLC entropy coding and simple motion estimation technique for reducing the redundancy of the video information. [25]

2.2.3 MPEG – 1

The MPEG-1 standard [26], published in 1992, was designed to produce reasonable quality images and audio at low bit rates. MPEG-1 provides the resolution of

352x240 (SIF- Source Input Format) for NTSC (National Television System(s) Committee) or 352x288 for PAL (Phase Alternating Line) at 1.5 Mbps. The target applications are focused on the CD-ROM, video-CD, and stream media applications like video over digital telephone networks, video on demand (VOD) etc. The picture quality level almost equals to VHS tape. MPEG-1 can also be encoded at bit rates as high as 4-5 Mbps. MPEG-1 specified the compression of audio signals, simply called layer-1,-2,-3. Layer-3 is now very popular in the digital music distribution over Internet known as MP3.

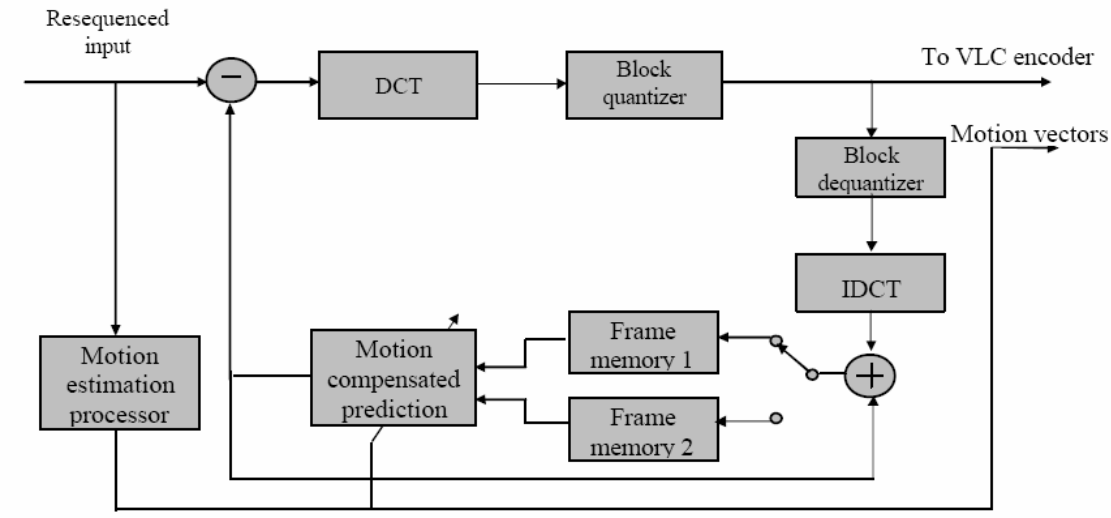


Figure 2.12 Typical MPEG-1 Encoder Structure

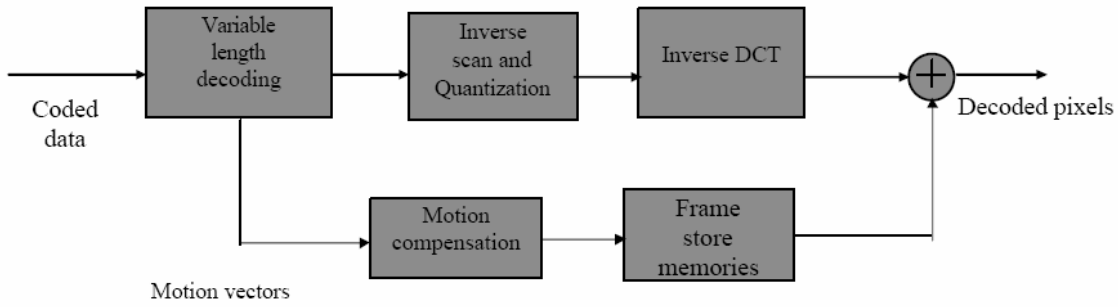


Figure 2.13 Simplified MPEG-1 Video Decoder

2.2.4 H.262 and MPEG – 2

MPEG–2 standard [27] was established by ISO/IEC in 1994. The purpose of this standard is to produce enhanced data rate and better video quality compared to MPEG–1. The coding technique of MPEG-2 is the same as MPEG-1 but with a higher picture resolution of 720x486.The unique feature of MPEG-2 is the layered structure, which supports a scalable video system. In this system, a streaming video can be decoded to videos with different qualities according to the situation of the network and the customer requirements. Field and frame picture structure makes the standard compatible with interlaced video. For the consistency of the standards, MPEG-2 is also backward compatible with MPEG-1, which means a MPEG-2 player can play back MPEG-1 video without any modification. This standard is also adopted by ITU-T referred to as H.262 [27].

2.2.5 H.263/H.263+/H.263++

H.263 (1995) [28] is the improvement of H.261. Compared to the former standards, H.263 provides (achieves) better picture quality and higher compression rate by using half pixel interpolation and more efficient VLC coding. H.263 version 2 (H.263+) and H.263 version 3 (H.263++) give more options to the coding standard on the basis of H.263 which achieves higher coding efficiency, more flexibility, scalability support and error resilience support.

2.2.6 MPEG-4

MPEG-4 (ISO/IEC 14496) became the international standard in 1999 [11]. The basic coding theory of MPEG-4 still remains the same as previous MPEG standards but more networks oriented. It is more suitable for broadcast, interactive and conversational environments. MPEG-4 introduced 'objects' concept: A *video object* in a scene is an entity that a user is allowed to access (seek, browse) and manipulate (cut and paste). It serves from 2 kbps for speech, 5 kbps for video to 5 Mbps for transparent quality video, 64 kbps per channel for CD quality audio [11]. The MPEG-4 Visual standard [29] allows the hybrid coding of natural (pixel based) images and video together with synthetic (computer generated) scenes.

2.2.7 MPEG-4 part-10/ H.264

ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC MPEG jointly developed the newest standard, H.264/AVC (also known as MPEG-4 part 10)

[30]. The motivation of this standard comes from the growing multimedia services and the popularity of HDTV, which need more efficient coding method. At the same time, various transmission media especially for the low speed media (Cable Modem, xDSL or UMTS) also called for the significant enhancement of coding efficiency. By introducing some unique techniques, H.264/AVC aims to increase compression rate significantly (save up to 50% bit rate as compared to MPEG-2 at the same picture quality) while transmitting high quality video at both high and low bit rates. The standard can increase resilience to errors by supporting flexibility in coding as well as organization of coded data. Network adaptation layer allows H.264 bit stream to be transported over different networks. The increase in coding efficiency and coding flexibility comes at the expense of increase in complexity as compared to the other standards. These features are discussed in chapter 3.

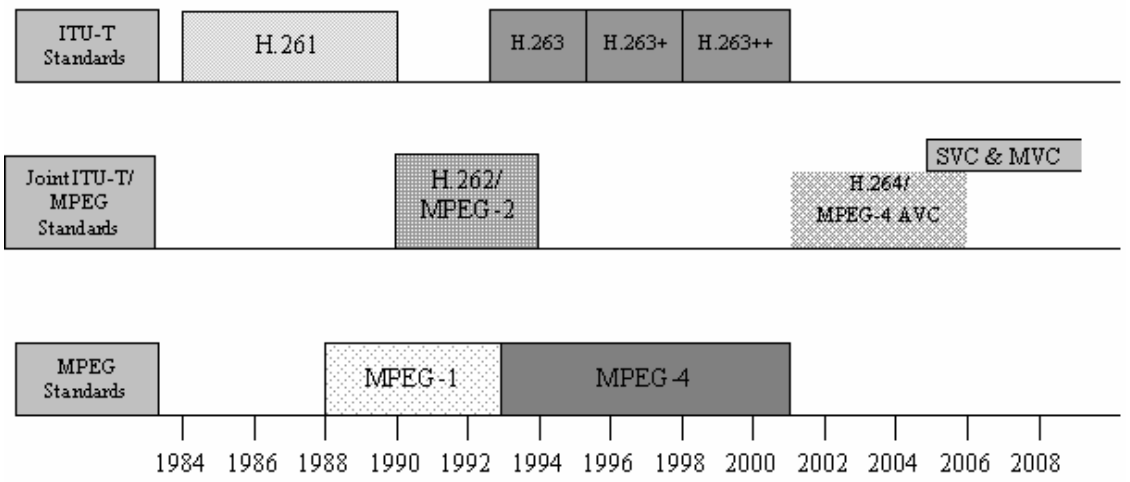


Figure 2.14 Progression of the ITU-T recommendations and MPEG standards

2.2.8 VC-1/ WMV-9

VC-1 [12] is a video codec specification that has been standardized by the Society of Motion Picture and Television Engineers (SMPTE) and implemented by Microsoft as Microsoft® Windows Media® Video (WMV) 9. Formal standardization of VC-1 represents the culmination of years of technical scrutiny by over 75 companies. The VC-1 codec is designed to achieve state-of-the-art compressed video quality at bit rates that may range from very low to very high. The codec can easily handle 1920 pixel × 1080 pixel presentation at 6 to 30 megabits per second (Mbps) for high-definition video. VC-1 is capable of higher resolutions such as 2048 pixels × 1536 pixels for digital cinema, and of a maximum bit rate of 135 Mbps. An example of very low bit rate video would be 160 pixel × 120 pixel presentation at 10 kilobits per second (Kbps) for modem applications. The basic functionality of VC-1 involves a block-based motion compensation and spatial transform scheme similar to that used in other video compression standards since MPEG-1 and H.261. However, VC-1 includes a number of innovations and optimizations that make it distinct from the basic compression scheme, resulting in excellent quality and efficiency. Unlike earlier versions of the Windows Media Video implementation, VC-1 is transport and container independent. This provides even greater flexibility for device manufacturers and content services. For further reading on VC-1, reader is referred to [12]

2.2.9 RV-10

RealVideo 10 [13] is a motion compensated hybrid coder that employs RealNetworks patented, and patent pending, technology including:

- Highly accurate motion modeling
- Proprietary spatial pixel prediction methods
- Multi-resolution residual analysis/synthesis stage
- Context adaptive entropy coding
- Psycho-visually tuned segmentation and filtering schemes
- Rate-Distortion optimized encoding algorithms
- Two-Pass encoding

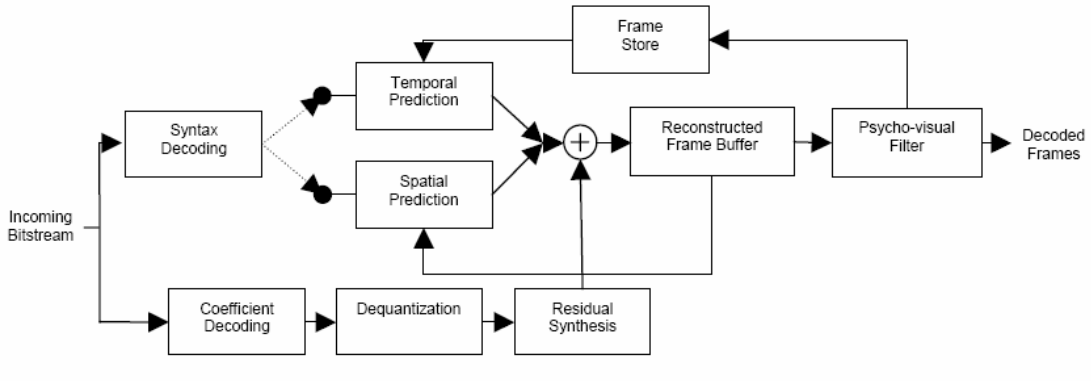


Figure 2.15 The RealVideo 10 Decoder

To learn more about RealVideo, RealAudio, and the RealSystem®, please refer [13].

CHAPTER 3

OVERVIEW OF H.264/AVC STANDARD

As broadband wire and wireless communication is booming in the world, streaming video has become one of the most important applications both in internet and telecom industries. 3G wireless service has been launched throughout the world and enhanced data service such as HSDPA (high-speed downlink packet access) is introduced with bandwidth more than 384kbps. Thus multimedia streaming including video and audio are expected to be delivered to end users. However the total bandwidth is still limited and the costs for end user are proportional to the reserved bit rate or the number of bits transmitted on the data link. At the same time, since harsh transmission environment in wireless communications such as distance attenuation, shadow fading, and multi-path fading can introduce unpredictable packet-loss and errors during transmission, compression efficiency and error resilience are the main requirements for video coding standard to succeed in the future.

Currently there are several image and video-coding standards that are widely used such as JPEG, JPEG2000, MPEG-2, and MPEG-4 [8]. In 2003, H.264/AVC was introduced with significant enhancement both in compression efficiency and error resilience. Compared with former video coding standards such as MPEG-2 and MPEG-4 part 2, it saves approximately 50% in bit rate [55] and provides important characteristics such as error resilience, stream switching, fast

forward/backward etc. It is believed to be the most competitive video coding standard in this new era. However, the improvement in performance comes at the expense of increase in computational complexity, which requires higher speed both in hardware and software. H.264/AVC targets the applications like video conferencing (full duplex), video storage or broadcasting (half duplex) with enhanced compression efficiency as well as network friendliness. The scope of H.264/AVC covers two layers: network abstraction layer (NAL) and video coding layer (VCL) (Figure 3.1). While NAL gives a better support for the video transmission through a wide range of network environments, VCL mainly focuses on how to enhance the coding efficiency.

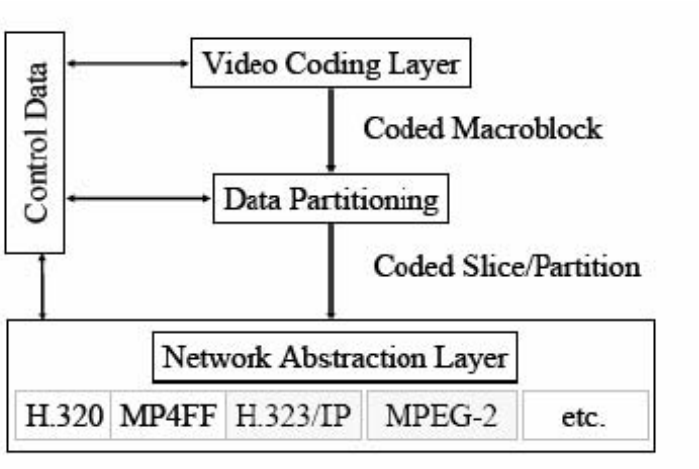


Figure 3.1 H.264/AVC layer structure [14]

In this chapter, the features that make H.264/AVC achieve the performance improvement compared to former existing standards will be investigated. The improvement to the video coding layer will be addressed in more detail. Before discussing the H.264/AVC technical features, some important terminologies should be introduced first.

Video coding standards commonly use hierarchical syntax. A video sequence is divided into group of pictures. A picture is divided into slices. A slice is divided into macroblocks. A macroblock is divided into blocks (Figure 3.2). In H.264/AVC additionally a block can be further divided into sub blocks.

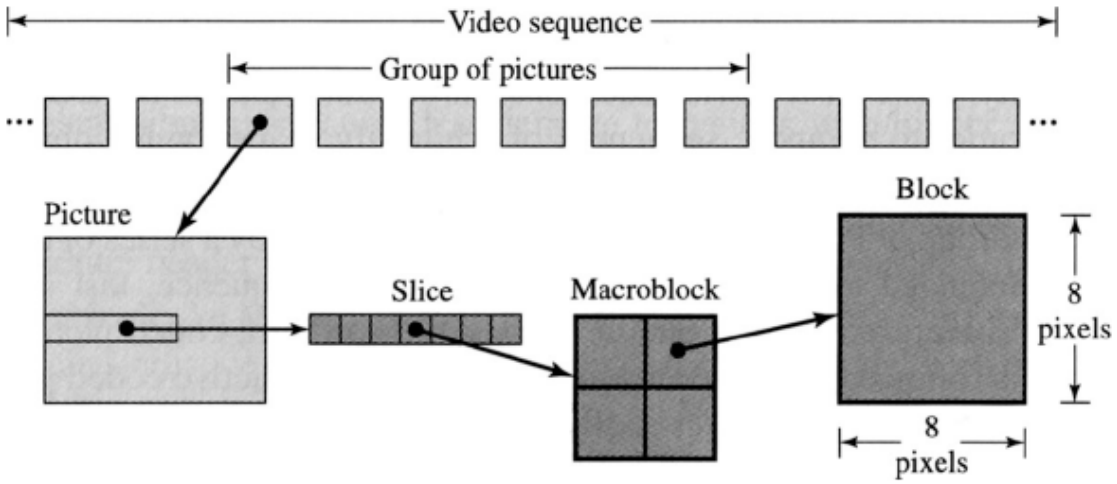


Figure 3.2 Hierarchical syntax [3]

Coded picture: A coded picture in this standard refers to a field (interlaced video) or a frame (progressive or interlaced video) (Figure 3.3). Each coded frame has a unique frame number, which is signaled in the bit stream. The frame number is not necessarily the same as the decoding order of the frame. For those interlaced frames or progressive frames, each field has an associate *picture order count,* used to indicate the decoding order between the two fields.

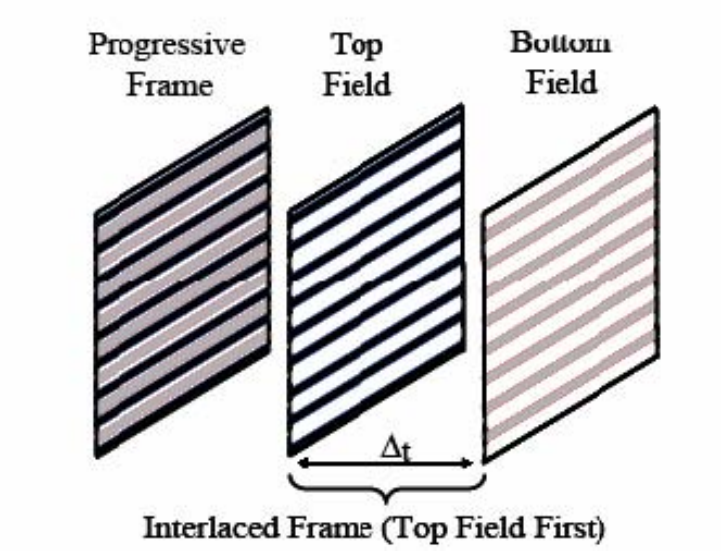


Figure 3.3 Progressive and interlaced frames [3]

Each previously coded picture can be used as the reference picture for future decoded pictures. One notable feature here is the reference pictures are managed by one or two lists (list0 and list1). Macroblock is the basic data unit for video coding operations. A set of macroblocks is further grouped into a slice in raster scan order. A frame may be split into one or more slices (Figure 3.4).

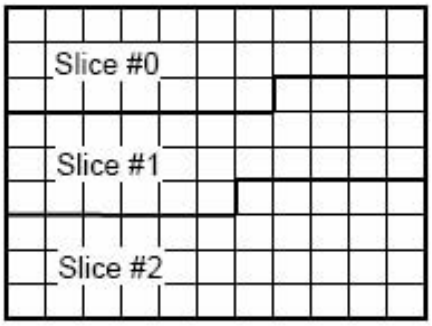


Figure 3.4 Subdivision of a picture into slices [55]

For each slice, the macro blocks within it are coded independently from those within other slices. There are totally 5 types of slices defined in H.264 /AVC:

I slice: All the macro blocks in this slice are I macro blocks. They are coded without referring to previously coded pictures, but may use the decoded samples within the same slice (current picture) as reference (intra prediction).

P slice: Macro blocks in this type of slice can be P macro blocks or I macro blocks. P macro block is predicted by referring to one previously decoded picture in list 0.

B slice: In addition to the coding types available in a P slice, macro blocks of a B slice can also be coded using inter prediction with two reference pictures per predicted block (one from list 0 and/or one from list 1) that are combined using a weighted average.

SP slice: A so-called switching P slice that is coded such that efficient switching between different video streams becomes possible without the large number of bits needed for an I slice [15].

SI slice: A so-called switching I slice that allows an exact match of a macro block in an SP slice for random access or error recovery purposes.

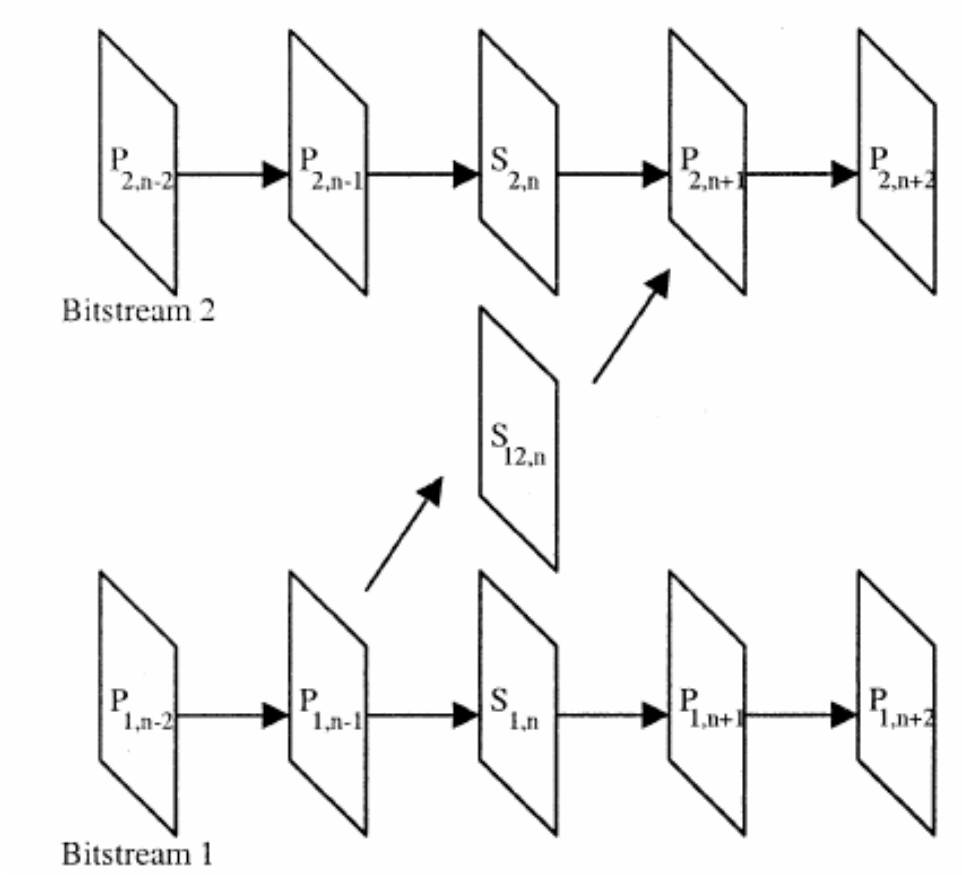


Figure 3.5 Switching between bitstreams using SP-frames. [15]

The last two types of slices are newly added in H.264/AVC and the first three slices are similar to those used in earlier standards.

Profile: Profile defines a set of coding algorithms or functions that a coding standard may use. In H.264 the following profiles are defined known as baseline profile (lower capability plus error resilience), main profile (high compression quality), extended profile (added features for efficient streaming) and high profile (Figure 3.6).

Level: The performance limits for codecs are defined as a collection of levels, each places a restriction on the configurations of the coding process, such as decoding speed, sample rate, number of blocks per second etc.

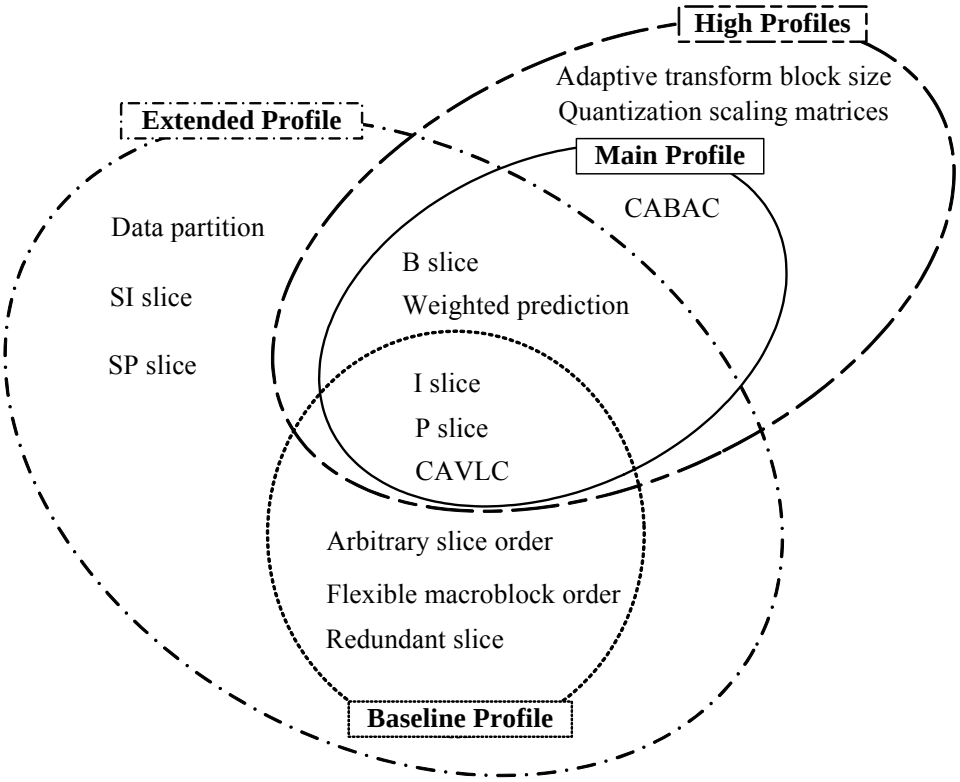


Figure 3.6 The specific coding parts of profile in H.264 [16]

3.1 Network Abstraction Layer

The NAL (network abstraction layer) is designed to provide friendly transmission for video data through different network environments. The coded video data is packetized into NAL units in order to support most of the existing packet switched based network environments. Each NAL unit is a packet that contains an integer number of bytes. The first byte of each NAL unit is the header that contains an indication of the data type in this NAL unit, and the following bytes contain the data

indicated by the header [14]. For more detailed information about NAL, please refer to [14].

3.2 Video Coding Layer

Modifications have been made in the video coding layer of H.264 in order to achieve significant compression efficiency as compared to previous standards. The basic encoding structure of H.264/AVC is shown in Figure 3.7. In H.264, first the video source is divided into blocks of luma and chroma. Then the motion estimation and prediction are employed to exploit the temporal and spatial redundancies. Then transform coding, quantization and entropy coding are applied in serial, which finally generate the output bit stream. This bit stream can be used to transmit through networks or stored with optical or magnetic storage devices.

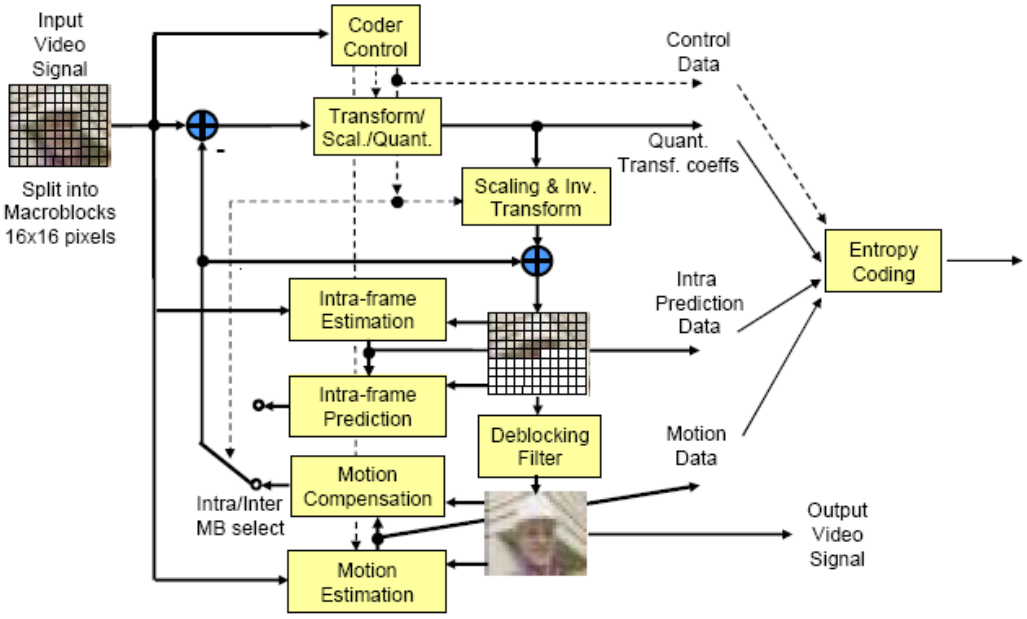


Figure 3.7 H.264 encoder [17]

The decoding flow consists of a series of reversed operations in terms of the encoding process (Figure 3.8). The only operation added to the decoding flow is the loop-filter (deblocking filter). The purpose of this filter is to minimize the block distortions introduced by block based transformations and motion estimations. The video decoding procedure is defined in existing standards (also for H.264/AVC), which means by imposing the decoding process with a collection of restrictions (such as the restrictions on bit stream and syntax), any encoding process that produces a decodable bit stream (decodable by the standard decoding process) is an applicable encoder. By this way, the developers have strong flexibility in developing the encoders in order to incorporate different applications with various requirements (such as compression quality, implementation cost, time to market, etc) [14].

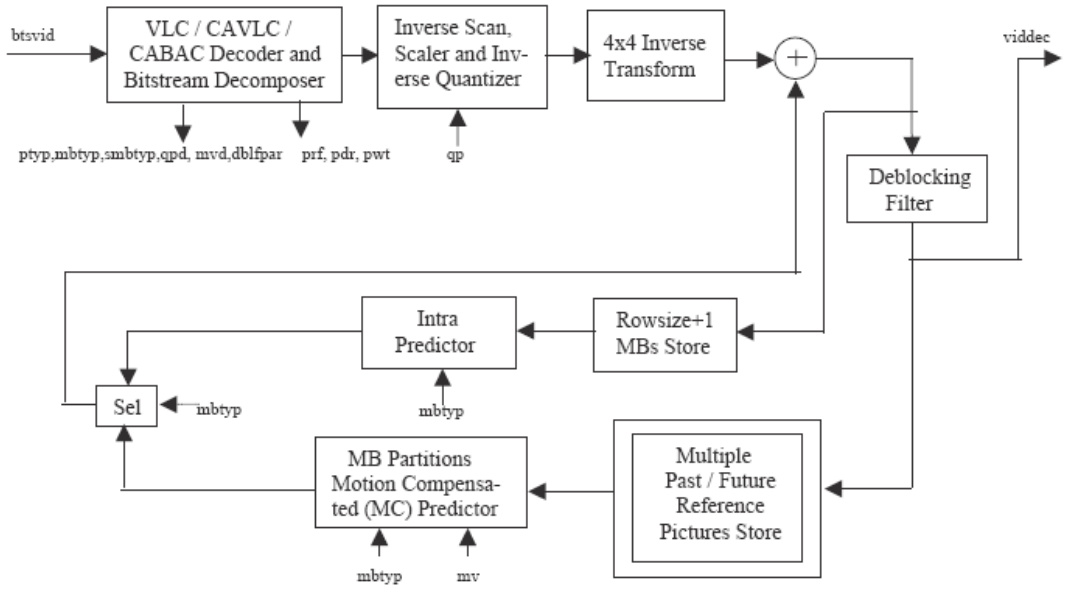


Figure 3.8 H.264 decoder [18]

3.2.1 Motion Estimation and Compensation for Inter frames

The key features added to motion estimation and compensation part of H.264/AVC include (1) variable block-size motion compensation with small block sizes, (2) quarter-pixel motion estimation accuracy, and (3) multiple reference pictures selection.

1. Variable block-size motion compensation with small block sizes:

In previous standards, motion estimation is based on 16x16 macro block for luma component and as 8x8 block for chroma component for 4:2:0 format. But in H.264/AVC, different block sizes are supported for motion compensation. The luminance component (Y) of each macro block can be partitioned in 4 ways: one 16x16 macro block, two 16x8 rectangular blocks, two 8x16 rectangular blocks or four 8x8 blocks (Figure 3.9). If the 8x8 mode is chosen, each 8x8 block may be further divided into four ways. One 8x8 block, two 8x4 sub blocks, two 4x8 sub blocks or four 4x4 sub blocks (Figure 3.9).

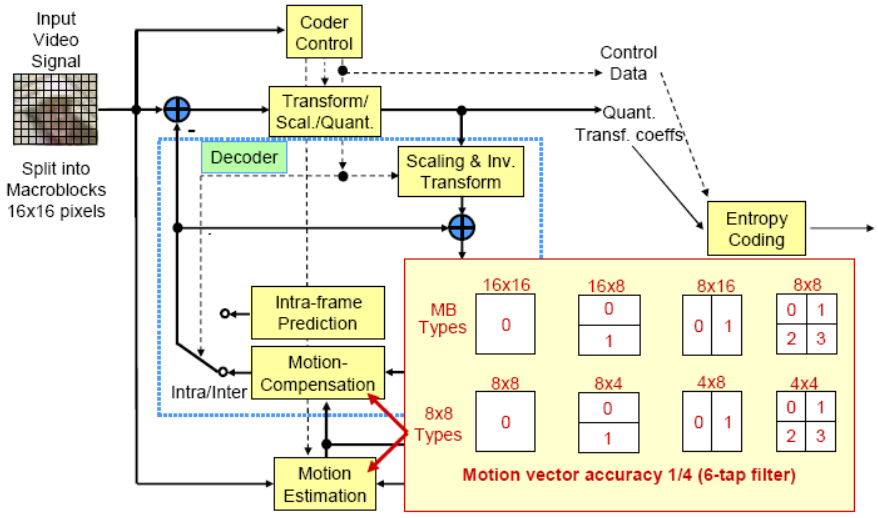


Figure 3.9 Motion compensation accuracy [17]

Variable block size is chosen by transmitting one additional syntax element for each 8x8 partition. This syntax element specifies whether the corresponding 8x8 partition should be divided further. The partition strategy can be looked as a tree structure to some extent. The partitions for chroma components (Cb, Cr) in 4:2:0 format are done in the same manner except the size of each partition is only half of the luma partition in both horizontal and vertical coordinates (16x8 in luma corresponding to 8x4 in chroma, 8x4 in luma corresponding to 4x2 in chroma) [3]. The smaller the block is split, the less energy is left within the residual. By using the combination of seven different block sizes, the bit rate savings of up to 12% can be achieved as compared to using only a 16x16 block size [19].

2. Quarter-pixel motion estimation accuracy:

Most of the existing standards support the motion estimation accuracy up to half sample pixel. In H.264/AVC, the maximum accuracy is enhanced to quarter pixel. Each partition or sub-macro block partition in an inter-coded macro block is predicted from an area of the same size in a reference picture. The offset between the two areas (the motion vector) has quarter-sample resolution for the luma component and one-eighth sample resolution for the chroma components. The luma and chroma samples at sub-sample positions do not exist in the reference picture and so it is necessary to create them by using interpolation from nearby coded samples.

Figure 3.10 shows the quarter pixel interpolation of a 4x4 luma block. The gray dots noted with upper case indicate the integer-position samples .The white dots noted with lower case indicate the half and quarter-pixel samples. First the half

sample positions are obtained by applying a 6-tap filter with tap values: (1, -5, 20, 20, -5, 1)/32. Quarter sample positions are obtained by averaging samples at integer and half sample positions. In practice, the motion vectors (MV) of the block use one or two bits to indicate if the motion estimation is integer, half-pixel or quarter-pixel. The quarter-pixel accuracy gives 20% bit rate savings [20] as well as more accurate motion representation compared to integer-pixel spatial accuracy.

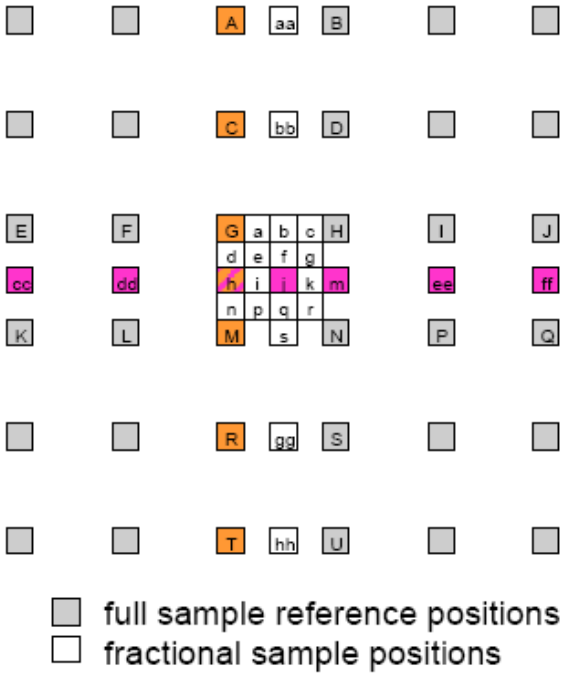


Figure 3.10 Quarter sample luma interpolation [3]

3.2.2 Multiple Reference Pictures Selection

The H.264/AVC standard gives flexibility for the encoder to select large number of decoded reference pictures. This flexibility increases the requirement of memory size for both encoder and decoder but enhances the compression efficiency at

the same time. For P macro block, the reference picture can be chosen from multiple former decoded pictures (Figure 3.11). Therefore not only the motion vectors but also a reference index parameter Δ (which indicates which picture should be referenced) is transmitted. The reference index parameter is transmitted for each motion-compensated 16x16, 16x8, 8x16, or 8x8 luma block. Motion compensation for regions smaller than 8x8 uses the same reference index for prediction of all blocks within the 8x8 region.

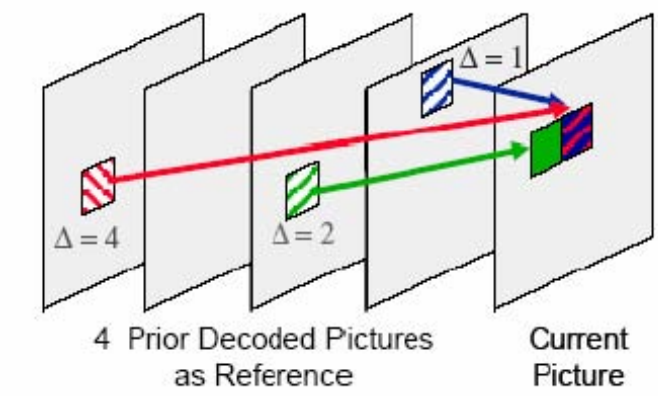


Figure 3.11 Multiple reference frames and generalized bi-predictive frames [55]

Because of the increased complexity in motion prediction, the H.264/AVC standard employs two distinct lists of reference pictures (list 0 and list 1). For P slice, only the list 0 is used to store the reference pictures whereas B slice needs both list 0 and list 1. For detailed reference picture management, please refer to [3] and [30].

The motion compensated-prediction for B slice is in the same manner except that it is a bi-directional prediction. In B slices, four different types of inter-picture prediction are supported: list 0, list 1, bi-predictive, and direct prediction. For the bi-predictive mode, the prediction signal is formed by a weighted average of motion-compensated list 0 and list 1. The direct prediction mode is inferred from

previously transmitted syntax elements and can be either list 0 or list 1 prediction or bi-predictive.

Multiple reference pictures in the new standard yield about 5-20% coding efficiency [20] compared to former standards that use only one reference frame.

3.2.3 Intra Prediction

Intra prediction allows the current macro block to be predicted by the previously decoded samples within the same slice at the decoder. The encoder can switch between intra and inter prediction dynamically according to the content of the frame. The directional spatial prediction for intra coding improves the quality of the prediction signal (Figure 3.12).

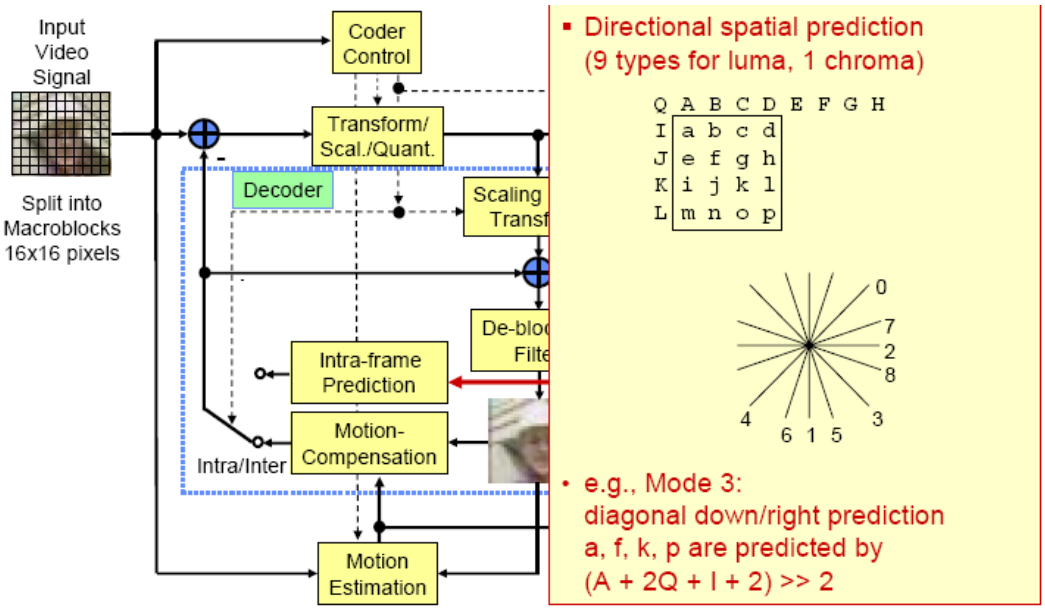


Figure 3.12 Intra prediction in H.264 [17]

Luma intra prediction either has a single prediction for entire 16x16 macro block or 16 individual predictions of 4x4 blocks. In high profiles there is also 8x8 intra prediction. [21] There are 9 intra 4x4 (DC, 8 directional) and 4 intra 16x16 (vertical, horizontal, DC, planar) prediction modes for luma components. For chroma components, 4 8x8 based intra prediction modes (vertical, horizontal, DC, planar) are supported and both the chroma components of the same macro block (C_b and C_r) use the same prediction mode. In addition to the intra prediction modes above, another intra coding mode (I_PCM) is also used for some special cases. I_PCM just sends the image samples without prediction or transformation. The I_PCM mode guarantees a limit on the expansion of white noise during compression.

The 16x16 and 4x4 intra prediction direction modes are shown in the Figures. 3.13 and 3.14 respectively.

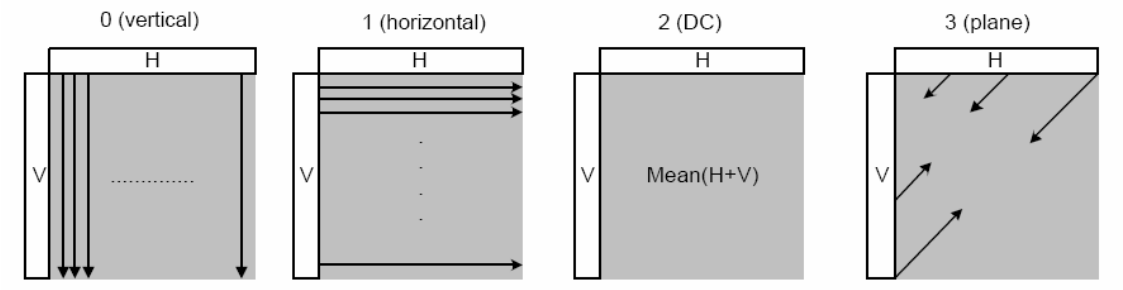


Figure 3.13 16x16 intra prediction directions [3]

For regions with less spatial detail (flat regions), H.264 supports 16x16 intra predictive coding (Figure 3.14). The prediction mode for each block is efficiently

coded by assigning shorter symbols to more likely modes, where the probability of each mode is determined based on the modes used for coding the surrounding block.

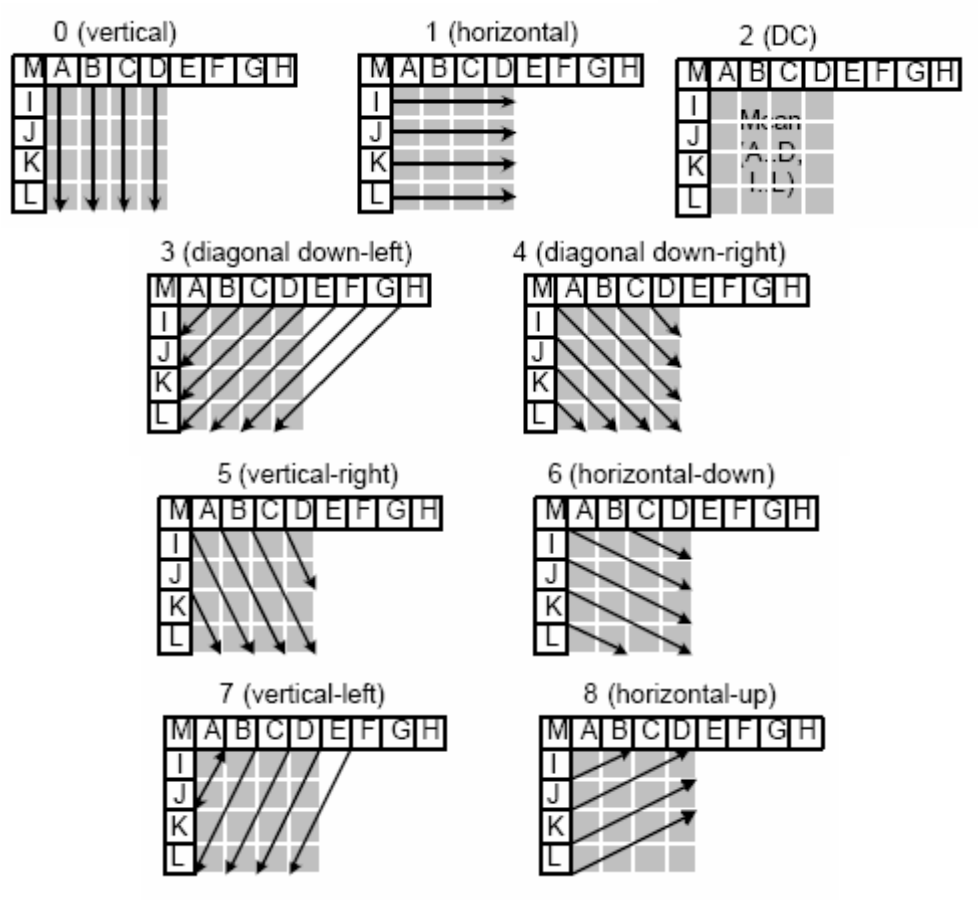


Figure 3.14 4x4 intra prediction directions [3]

3.2.4 Transform and Quantization

H.264/AVC employs integer spatial transform which is primarily 4x4 in shape (Figure 3.15), as opposed to the usual floating point 8x8 DCT [22] specified with rounding error tolerances as used in earlier standards. H.264 can use 3 transforms depending on the type of the residual data that is to be coded: a transform for the 4x4 array of luma DC coefficients in intra macro blocks (predicted in 16x16 mode),

a transform for the 2x2 array of chroma DC coefficients (in intra macro block) and integer DCT for all other 4x4 blocks in the residual data.

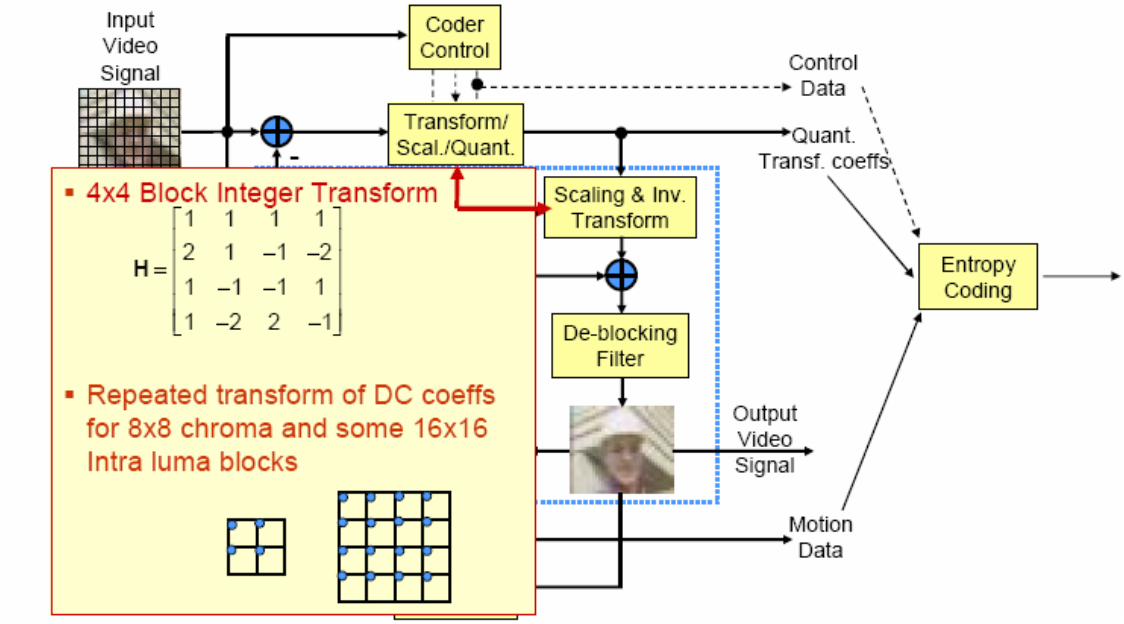


Figure 3.15 Transform coding [17]

The characteristics of transform used in H.264/AVC are as follows:

1. Transform is applied in 2 stages for 16x16 intra prediction. In the first stage 4x4 integer DCT is applied. Hadamard transform is applied in the second stage to the DC coefficients of the first stage transform.
2. Transform of block size 4x4 is separable.
3. Integer transform: Accuracy mismatch at the encoder/decoder can be eliminated.
4. As it consists of only adds and shifts, it is easy to implement in the transform.
5. Different norms for even and odd rows of the matrix.
6. Due to the use of small size transform, it reduces ringing artifacts in a frame [22].

The characteristics of quantization used in H.264/AVC are as follows:

1. Scaling part of the transform in H.264/AVC is integrated into the quantizer.
2. Logarithmic step size control.
3. Extended range of step sizes. QP is in the range of 0-51.
4. Smaller step size for chroma compared to luma.
5. Quantization reconstruction is just one multiply, one add and one shift.

3.2.5 Deblocking Filter

Deblocking filter (Figure 3.7) is introduced in H.264/AVC standard to minimize the block distortion caused by the present compression technique. By controlling the strength of the filtering with some parameters, the block edges can be smoothed and the appearance of the decoded frames is also enhanced. The deblocking filter is an in-loop filter, which means it is not a kind of post processing. For post filter, the input is a completely reconstructed frame, but for in-loop filter, the input is the current MB and the boundaries of each decoded macroblock are filtered immediately. Deblocking filter is added to the encoder after inverse transformation and before the frame store. For decoder, the filter (Figure 3.8) is located after the reconstruction of the frame for display. For more details about deblocking filter refer [3].

3.2.6 Entropy Coding

Exp-Golomb code is used universally for all symbols except for transform coefficients. The following two entropy-coding algorithms are used in H.264 standard:

1. CAVLC (Context Adaptive Variable Length Coding)
    a. No end-of block, but number of coefficients is decoded.
    b. Coefficients are scanned backwards and contexts are built depending on transform coefficients.
    c. Transform coefficients are coded with the following elements: number of non-zero coefficients, levels and signs for all non-zero coefficients, total number of zeros before last non-zero coefficient, and run before each non-zero coefficient.
    d. The VLC table to use is adaptively chosen based on the number of coefficients in the neighboring blocks.
2. CABAC (Context Adaptive Binary Arithmetic Coding)
    a. Overview of CABAC is shown in Figure 3.16.
    b. Usage of adaptive probability models for most symbols.
    c. Exploiting symbol correlations by using contexts.
    d. Discriminate between binary decisions by their positions in the binary sequence.
    e. Probability estimation is realized via look up table.

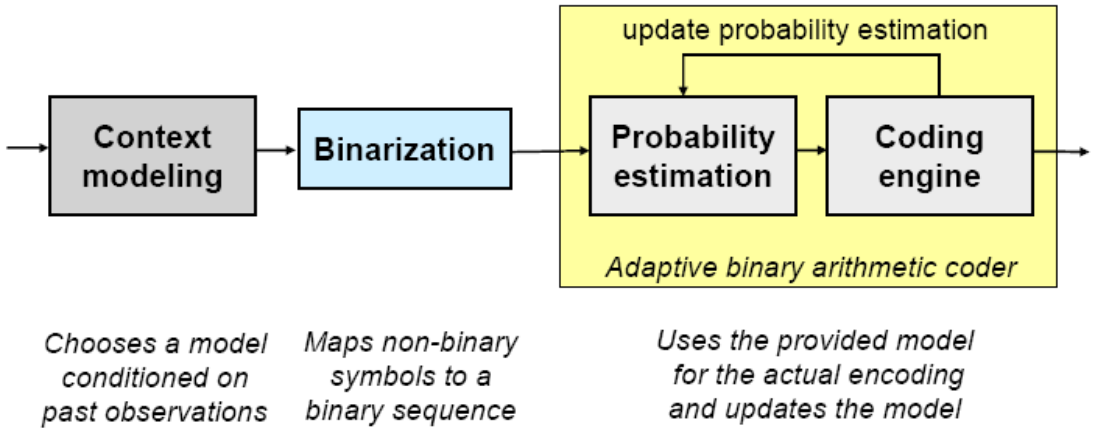


Figure 3.16 CABAC overview [17]

3.3 Conclusions

H.264/AVC gives significant enhancement both in compression efficiency and error resilience. Compared with earlier video coding standards such as MPEG-2 and MPEG-4 part 2, it saves more than 40% in bit rate [20] and provides important characteristics such as error resilience, stream switching, fast forward/backward etc. It is believed to be the most competitive video coding standard in this new era. However, the improvement in performance also introduces significant increase in computational complexity, which requires higher speed both in hardware and software.

CHAPTER 4

SCALABLE EXTENSION OF THE H.264/AVC STANDARD

4.1 Introduction

The advances in video coding technology and standardization [25]-[30] along with the improvements of network infrastructures and the increasing growth of computing power are bringing digital video into our daily lives. Application areas today range from multimedia messaging, video telephony, video conferencing over mobile TV, wireless and Internet video streaming to standard and high-definition TV broadcasting. In particular, the Internet as well as wireless networks gain more and more in importance for video applications. But in comparison to traditional video transmission systems like terrestrial, cable, or satellite TV, they are characterized by varying connection quality. Moreover, the source content is usually accessed by devices with unknown capabilities ranging from cell phones with small screens and restricted processing power to up-to-date PCs with high-definition displays. For today's and future multimedia streaming services it is generally required that multiple bit-streams of the same source content that differ in coded picture size, frame rate, and bit-rate are provided by the same service. Internet and wireless video applications should cope with the unpredictable bandwidth variations and the relatively high packet loss rate. A reduction in connection quality should not result in an interruption of the service or a

highly corrupted output. Hence, error resilience schemes with graceful degradation as well as efficient mechanisms for stream switching are highly desired.

An attractive solution to several challenges of modern video transmission systems is scalable video coding (SVC) [113]. A video bit-stream is called scalable when parts of it can be removed in a way that the resulting sub-stream forms another valid bit-stream for a given decoder, which represents the source content with a reduced reconstruction quality compared to the original bit-stream. Bit-streams that do not provide this property are referred to as single-layer streams. The usual modes of scalability are temporal, spatial, and quality scalabilities. Spatial and temporal scalabilities describe cases, in which subsets of the bit-stream represent the source content with a reduced picture size (spatial resolution) or frame rate (temporal resolution), respectively. With quality scalability, the sub-stream provides the same spatio-temporal resolution as the global bit-stream, but a lower fidelity or signal-to-noise ratio (SNR). Quality scalability is also commonly referred to as SNR scalability. More rarely required scalability modes are region-of-interest (ROI) and object-based scalability, in which the sub-streams represent special parts of the original picture area. The different types of scalability can usually be combined, so that a multitude of representations with different spatio-temporal resolutions and bit-rates are provided inside a single bit-stream.

It is obvious that efficient scalable video coding is a highly desired feature for video transmission over heterogeneous networks [31]-[33]. The source content has to be encoded only once with the highest required resolution and bit-rate,

but representations with lower resolution and/or quality can be obtained by partial decoding. A client with restricted resources (display resolution, processing power, or battery power) can decode only the required parts of the delivered bit-stream. In a multicast scenario, terminals with different capabilities can be served by a single scalable bit-stream. Since scalable bit-streams usually contain parts with different importance, it is possible to efficiently combine scalable video coding with unequal error protection schemes. By a stronger protection of the more important information, error resilience with graceful degradation can be achieved up to a certain degree of transmission errors. Media-aware network elements (MANE), which receive feedback messages about the terminal capabilities and/or channel conditions, can remove the non-required parts from a scalable bit-stream, before they forward it. Thus, the loss of important transmission units due to congestion can be avoided, and the error robustness of the video transmission service can be further improved. Scalable video coding is also highly desired for video surveillance applications, in which videos do not only need to be viewed on multiple devices ranging from high-definition monitors to video phones or PDAs, but also need to be stored and archived. With scalable video coding, high-resolution/high-quality parts of a bit-stream can be deleted after certain expiration time, so that only low-quality copies of the video are kept for the archive.

Scalable video coding has been an active research area for about 20 years. The video coding standards MPEG-2 Video / H.262 [27], H.263 [28], and MPEG-4 Visual [29] already include several tools, by which the most important scalability modes, temporal, spatial, and SNR scalabilities, are supported. MPEG-4

Visual additionally provides tools for supporting object-based scalability and fine-grain SNR scalability (FGS) [34][35], which allows a much more flexible bit-stream adaptation than the conventional SNR scalable coding modes in MPEG-2 Video / H.262, H.263, and MPEG-4 Visual. However, the scalable profiles of these standards have been rarely used, mainly because spatial and SNR scalabilities come along with a significant loss in coding efficiency as well as an increase in decoder complexity. Even though scalable video coding schemes offer many nice features, coding efficiency remains the most important requirement. It should be noted that two or more single-layer streams can always be transmitted by the method of simulcast. Such a parallel transmission provides similar properties as scalable bit-streams. Furthermore, features like switching pictures and redundant pictures ([30]) have been developed to improve rate adaptation and error resilience for single-layer codecs. Due to the rare use of the scalable profiles in prior standards, the newest video coding standard H.264/AVC does not contain any tools for providing spatial or SNR scalability. Temporal scalability is supported, but only because it comes along with a coding efficiency improvement (sec. 4.3.5). Scalable extension to H.264/AVC supports all three kinds of scalability, i.e. spatial, temporal and SNR scalability.

Given the requirements for today's and future video applications as well as the experiences with scalable profiles in the past, it can be summarized that for the success of scalable video coding, it has to be characterized by the following features:

- The support of temporal, spatial and SNR scalabilities.
- The support of simple bit-stream adaptations.

- A small increase in decoding complexity compared to state-of-the-art single-layer coding.
- A decoding complexity that scales with the decoded spatio-temporal resolution and bit-rate.
- A reasonable coding efficiency compared to single-layer coding and simulcast.

In any case, the coding efficiency should clearly be superior to that of simulcasting the supported spatio-temporal resolutions and bit-rates using a state-of-the-art single-layer video codec. But in comparison to single-layer coding, bit-rate increases of 10% to 50% or even more for the same fidelity may be tolerable depending on the needs of an application and the supported degree of scalability.

Even though the scalability tools in MPEG-2 Video / H.262, H.263, and MPEG-4 Visual haven not been accepted by the market, the investigation of scalable coding techniques continued. On the one hand, significant progress has been made in improving SNR scalability, especially fine-grain SNR scalable coding [36]-[43] for conventional hybrid video codecs. On the other hand, scalable video coding based on a 3-d wavelet transform has been investigated in detail. With these schemes a spatio-temporal transform is applied to a group of pictures. Codecs of this type [44] are known for more than 15 years, but they did not present an alternative for hybrid video coding schemes until motion compensation was incorporated into the 3-d wavelet filtering [45][46]. Later it was discovered that the motion-compensated temporal filtering (MCTF) can be elegantly realized by employing the lifting representation of wavelets [47]-[49]. This made it possible to use any motion model and interpolation method in

the context of MCTF while preserving the perfect reconstruction property of the 3-d transform.

The progress in 3-d wavelet coding caused the ISO/IEC Moving Picture Experts Group (MPEG) to start an activity on exploring inter-frame wavelet video coding techniques. As a result MPEG issued a call for proposals for efficient scalable video coding technology in October 2003 with the intention to develop a new scalable video coding standard. 12 of the 14 submitted proposals [50] represented scalable video codecs based on a 3-d wavelet transform, while the remaining two proposals were extensions of H.264/AVC [30]. After a 6 month evaluation phase, in which several subjective tests for a variety of conditions have been carried out and the proposals have been carefully analyzed regarding their potential for a successful future standard, the scalable extension of H.264/AVC as proposed in [51] has been chosen as the starting point [52] of MPEG's scalable video coding (SVC) project in October 2004. In January 2005, MPEG and the ITU-T Video Coding Experts Group (VCEG) agreed to jointly finalize the SVC project as an Amendment of their H.264/AVC standard. For more information about SVC, the reader is referred to the current draft standard [53].

4.2 Basic Concepts for extending H.264/AVC towards a Scalable Video Codec

The most recent video coding standard H.264/AVC provides a significantly improved coding efficiency in comparison to all prior standards [56]. H.264/AVC has attracted a lot of attention from industry and was adopted by various application standards. It is expected that in future H.264/AVC will be widely used in

most application areas of video coding. As explained in sec. 4.1, the most important points for developing a successful scalable video coding standard are coding efficiency and complexity as well as the support for easy bit-stream adaptation. And given the fact that SVC is developed as an extension of H.264/AVC, the investment that has already been taken place for preparing and developing H.264/AVC products should additionally be taken into account. Thus, one of the main design goals was that SVC should represent a straightforward extension of H.264/AVC. As much as possible, components of H.264/AVC should be re-used, and new tools should only be added for efficiently supporting the required types of scalability. As for any other video coding standard, coding efficiency has always to be seen in connection with complexity in the design process.

4.3 Temporal Scalability

A bit-stream provides temporal scalability when the set of its access units can be partitioned into a temporal base layer and one or more temporal enhancement layers with the following property. Let the temporal layers be identified by a temporal level, which starts from 0 for the base layer and is increased by 1 from one temporal layer to the next. Then for each natural number *k*, the bit-stream that is obtained by removing all access units of all layers with temporal level greater than *k* shall form another valid bit-stream for the given decoder.

With hybrid video codecs, temporal scalability can be generally enabled by restricting the motion-compensated prediction to reference pictures with a temporal level that is less than or equal to the temporal level of the picture to be predicted. The video coding standards MPEG-2 Video / H.262 [27], H.263 [28], and MPEG-4 Visual [29] support temporal scalability, but with due restrictions in the syntax and decoding process only with one temporal enhancement layer. H.264 / MPEG-4 AVC [30], however, provides a significantly increased flexibility on a picture and sequence level. It allows the coding of picture sequences with arbitrary temporal dependencies, which are only restricted by the maximum usable decoded picture buffer (DPB) size. Thus, for supporting temporal scalability with a reasonable number of temporal levels, no changes of H.264 / MPEG-4 AVC are required. The only related change in SVC refers to the signaling of temporal layers, which is described in sec. 4.4.

Temporal scalability with n dyadic temporal enhancement layers can be very efficiently provided with the concept of hierarchical B pictures [57], [58] as illustrated in Figure 4.1(a). A sub-sequence with $1/(2n)$-th of the full frame rate, which represents the temporal base layer T_0, is coded independently of all other pictures. It starts with an instantaneous decoding refresh (IDR) access unit, and each picture is either intra-coded, e.g. in order to enable random access, or temporally predicted by using the previous picture of this temporal base layer as reference. Coding and display order are identical for the base layer. The pictures of a temporal enhancement layer T_X are always located in the middle between two successive pictures with a temporal level less than X. The enhancement layer pictures are generally coded as B pictures, where

the reference picture lists 0 and 1 are restricted to the temporally preceding and succeeding pictures, respectively, with a temporal level less than X. Each set of temporal layers $\{T_0,\ldots,T_X\}$ can be decoded independently of all layers with a temporal level $Y > X$. In the following, the set of pictures between two successive pictures of the temporal base layer together with the succeeding base layer picture is also referred to as a group of pictures (GOP).

Although the described prediction structure with hierarchical B pictures provides temporal scalability and also shows excellent coding efficiency, as it will be demonstrated later, it is characterized by unnecessary restrictions. In general, hierarchical prediction structures for enabling temporal scalability can always be combined with the multiple reference picture concept of H.264 / MPEG-4 AVC. This means that the reference picture lists can be constructed by using more then one reference picture, and they can also include pictures with the same temporal level as the picture to be predicted. Furthermore, hierarchical prediction structures are not restricted to the dyadic case. As an example, Figure 4.1(b) illustrates a non-dyadic hierarchical prediction structure, which provides two independently decodable sub-sequences with 1/9th and 1/3rd of the full frame rate. It should further be noted that it is possible to arbitrarily modify the prediction structure of the temporal base layer, e.g. in order to increase the coding efficiency. It does not even need to be constant over time.

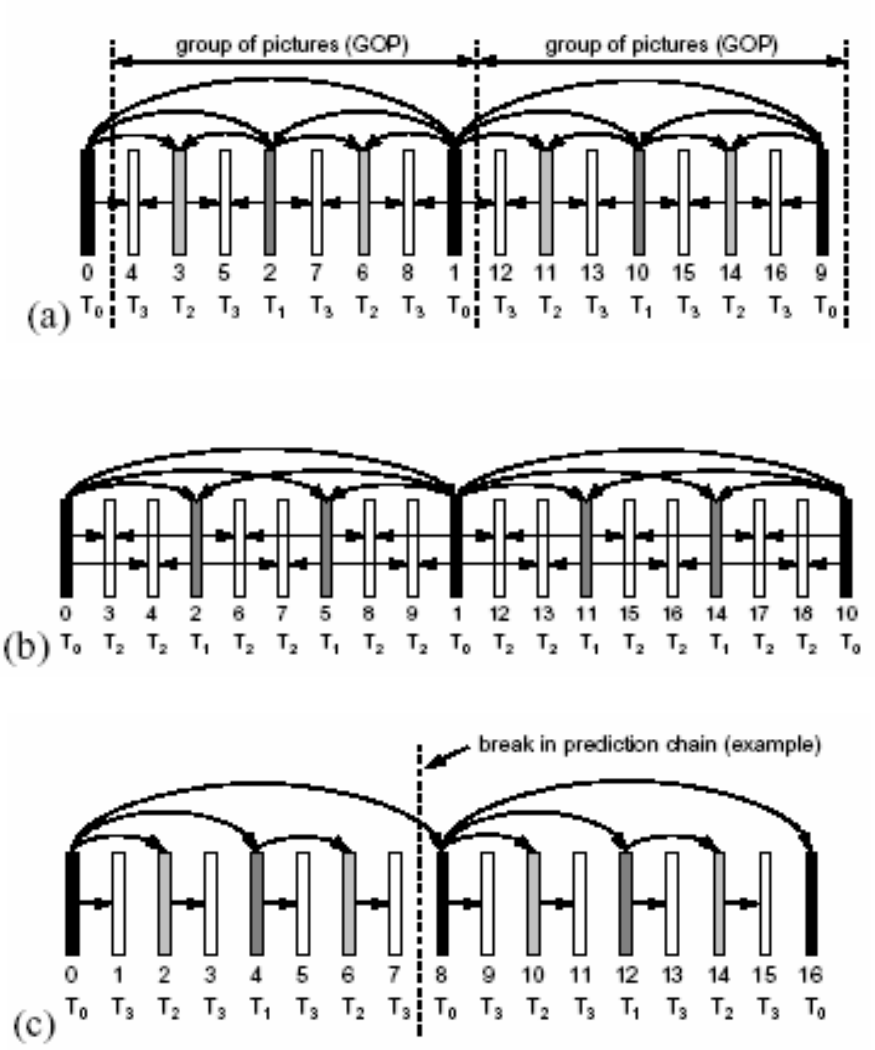


Figure 4.1 Hierarchical prediction structures for enabling temporal scalability: (a) coding with hierarchical B pictures, (b) non-dyadic hierarchical prediction structure, (c) hierarchical prediction structure with a structural encoder-decoder delay of 0. The numbers directly below the pictures specify the coding order; the symbols T_X specify the temporal layers with $_X$ representing the corresponding temporal level [24]

Furthermore, the enhancement layer pictures do not need to be coded using the B slice syntax. Actually, it is even possible to arbitrarily adjust the structural delay between encoding and decoding a picture (sec. 4.3.2) by restricting the motion-compensated prediction from pictures that follow the picture to be predicted in display

order. As an example, Figure 4.1(c) shows a hierarchical prediction structure, which does not employ motion-compensated prediction from pictures in the future. Although this structures provides the same degree of temporal scalability as the prediction structure of Figure 4.1(a), its structural delay is 0 compared to a structural delay of 8 pictures for the prediction structure in Figure 4.1(a). However, the usage of such low-delay structures generally decreases the coding efficiency. Especially, at low rates the reconstructed video is also characterized by subjectively disturbing temporal blocking artifacts, which result from breaks in the prediction chain. One of these breaks is illustrated in Figure 4.1(c). While picture 7 is predicted from picture 0 via the pictures 4 and 5, picture 8 is directly predicted from picture 0. As a result the reconstruction errors of these two pictures have different characteristics, and these differences are visible as fluttering signal components in the reconstructed video. A suitable usage of B pictures can significantly reduce such artifacts, since bi-predictive blocks, in which a forward and backward prediction signals are combined, allow a smooth transition between the different coding artifacts. That is exactly why the so-called open-GOP structure, in which B pictures are inserted before a picture, is recommended for MPEG-2 coded video.

4.3.1 Coding Order

The coding order for hierarchical prediction structures has to be chosen in a way that reference pictures are coded before they are employed for motion-compensated prediction. This can be ensured by different strategies, which mostly differ

in the associated decoding delay and memory requirement. In the following, we describe a coding order that guarantees minimal decoding delay. First, all pictures that are directly or indirectly used for motion-compensated prediction of the first picture of a coded video sequence in display order and the first picture itself are coded. Next, all pictures that are required for coding the second picture in display order and the second picture itself are coded, etc. At this, it has to be kept in mind that the required reference pictures have to be coded in the order, in which they are employed for motion-compensated prediction.

4.3.2 Delay

It is differentiated between decoding, encoding, and structural delay. The delay can be measured in units of pictures or equivalently in units of time. The decoding delay specifies the maximum number of decoded, but not outputted pictures that have to be stored in the decoder. Similarly, the encoder delay specifies the maximum number of pictures that need to be buffered between capturing and encoding. The structural delay represents the required delay for video transmission when assuming an infinite transmission rate and infinite processing power at both encoder and decoder. Hence, the delay that is introduced in real-world interactive applications is always somewhat larger than the structural encoder-decoder delay. In contrast to MCTF-based coding [59], the structural delay for hybrid video coding is always identical to the corresponding encoding delay.

Both encoding and decoding delays are determined by the usage of pictures for motion-compensated prediction that follow the picture to be predicted in display order. When assuming a minimum delay coding order as the one described above, the decoding delay in pictures is equal to the maximum number of prediction stages in the existing backward prediction paths. As illustrated in Figure 4.2(a), a backward prediction path is a path in the prediction structure that only consists of backward prediction stages. The encoding delay in pictures is equal to the maximum absolute difference between the display order number of the first and the last picture of a backward prediction path. For the example in Figure 4.2(a), decoding and encoding delay are equal to 4 and 15 pictures, respectively.

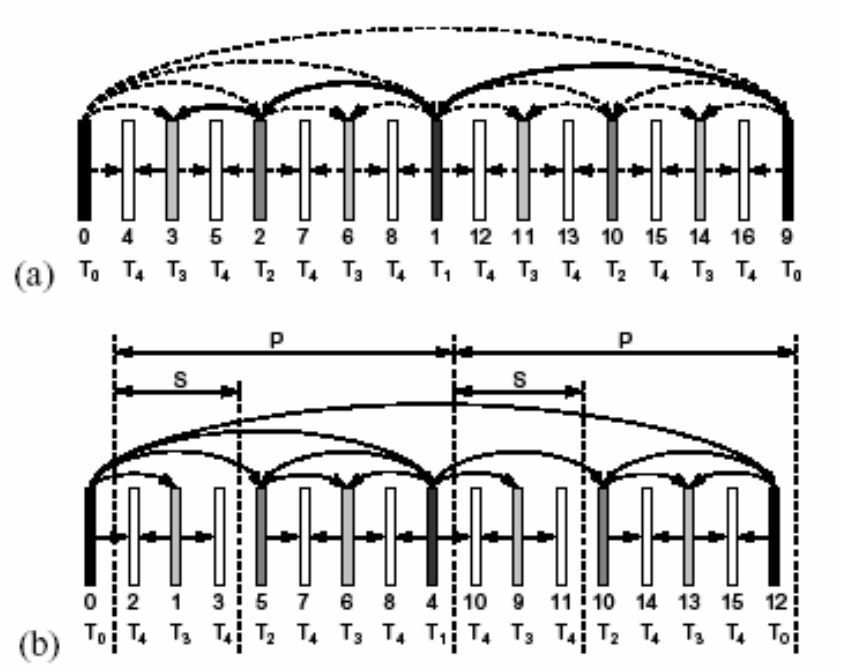


Figure 4.2 Adjusting the delay for hierarchical coding structures: (a) backward prediction path, (b) partitioning of an image sequence with a GOP size of 16 pictures for restricting the encoding delay to 4 pictures [24]

The encoding and decoding delay can be arbitrarily adjusted by suitably restricting the usage of backward prediction. A possible strategy for adjusting the delay ([59]), which was especially designed for avoiding as much as possible temporal blocking artifacts (see above), is described in the following. Let d_{enc} and d_{dec} specify the encoding and decoding delays in pictures, respectively. The sequence of pictures in partitioned into groups as illustrated in Figure 4.2(b), and backward prediction is only allowed inside these subgroups. The partitioning is specified by a partition size *P*s and a sub-partition size *S*, which are determined as follows. Let *d* be equal to

$$d = \max(d_{enc}, 2^{(1+d_{dec})} - 2)$$

then *P* and *S* are given by

$$P_s = 2^{\lceil \log_2 (1+d) \rceil}$$

$$S = \max(0, P_s - 1 - d)$$

And when further assuming that the reference picture list 1 is always constructed in a way that it only includes future pictures in display order, the number of active entries *Nact* for the reference picture list 1 can be determined via the relationship

$$N_{act} = \left\lfloor \frac{\{[(i \bmod P_s) > S \,?\, P_s : S] - (i \bmod P_s) + 1\}}{2} \right\rfloor$$

with *i* representing the display order number of the picture to be predicted.

4.3.3 Memory Management

For the following analysis of the memory requirements we assume that the reference picture lists for all pictures are constructed using only directly neighboring pictures of a coarser or the same temporal level (Figure 4.1(a)). Since the DPB will generally hold additional pictures that are marked as "used for reference", it is still possible to use multiple reference pictures for motion-compensated prediction. We further assume that the pictures are coded in the order described in sec. 4.3.1. It can easily be verified that it is not possible to find a coding order that allows a smaller DPB size. In addition, as pointed out above, the selected coding order guarantees minimal decoding delay. Pictures of the highest temporal level (e.g. the pictures T_3 in Figure 4.1(a) are always coded as non-reference pictures. These pictures do not need to be stored in the DPB and can be outputted just after decoding, since all of these non-reference pictures are coded in display order. Hence, the required DPB size (in units of pictures) is equal to the maximum number of reference pictures that need to be stored in the DPB, and consequently, it is also equal to the minimum required value of the syntax element *num_ref_frames*.

With the above described coding order it is always sufficient, when the 2 surrounding pictures of the temporal base layer and 1 picture for each temporal enhancement layer – with exception of the finest temporal level – are marked as "used for reference". Thus, when a base layer picture is decoded, it should replace the base layer picture before the previous base layer picture in the DPB. All temporal enhancement pictures that are coded as reference pictures should replace the previous

picture of the same temporal level. Although H.264 / MPEG-4 AVC provides several memory management control operation (MMCO) commands by which the behavior of the DPB can be arbitrarily controlled, it should be noted that MMCO commands are included in the slice syntax. Thus when decoding a temporal sub-stream the MMCO commands in the temporal enhancement pictures are missing, and the DPB might be operated in a wrong way. Since H.264 / MPEG-4 AVC does not allow a repetition of MMCO commands in different access units, the only possibility to enable temporal scalable coding with the minimum memory requirement is to encode all pictures as so-called long-term pictures. Two long-term indices are used for the pictures of the temporal base layer and one additional long-term frame index for each temporal enhancement layer, for which the pictures are coded as reference pictures. This method significantly reduces the memory requirement in comparison to the default sliding window marking process. The minimum required DPB size in pictures is equal to the number of temporal layers. As an example, a DPB of only 6 frames is sufficient for coding groups of 32 pictures with a dyadic hierarchical structure. With the sliding window marking process it is not even possible to encode groups of 32 pictures, since the maximum allowed number of frame storages (16) would be exceeded. It should be noted that when all pictures are coded as long-term pictures it is generally required to transmit reference picture list re-ordering (RPLR) commands for all slices in order to specify a suitable reference list construction process. When using the default sliding window algorithm, however, RPLR commands are only required for P slices.

4.3.4 Encoder Control

Independent of the prediction structure, an encoder should always operated using a rate-distortion optimized encoder control as described in [61] and specified in the Joint Model [62]. For the coding with hierarchical prediction structures, however, the coding efficiency can be further improved when the following points are taken into account.

4.3.4.1 Cascading of quantization parameters

The coding efficiency for hierarchical prediction structures is highly dependent on how the quantization parameters are chosen for pictures of different temporal levels. Intuitively, the base pictures should be coded with highest fidelity, since they are directly or indirectly used as references for motion-compensated prediction of all other pictures. For the next temporal level a larger quantization parameter should be chosen, since the quality of these pictures influences less pictures. Following this rule, the quantization parameter should be increased for each subsequent hierarchy level. Additionally, the optimal quantization parameter also depends on the local signal characteristic. An optimal selection of the quantization parameters can be achieved by a computationally expensive rate-distortion analysis similar to the strategy presented in [63]. In order to avoid such a complex operation, we have chosen the following strategy, which proved to be sufficiently robust for a wide range of tested sequences [24]. Based on a given quantization parameter QP_0 for pictures of the temporal base layer, the quantization parameters for enhancement layer pictures of a

given temporal level $k > 0$ are determined by $QP_k = QP_0 + 3 + k$. Although this strategy for cascading the quantization parameters over hierarchy levels results in relatively large PSNR fluctuations inside a group of pictures, subjectively, the reconstructed video appears to be temporally smooth without any annoying temporal artifacts.

4.3.4.2 Temporal direct mode

The decoding process of H.264 / MPEG-4 specifies that the motion vectors for the temporal direct mode are derived by scaling the co-located list 0 motion vector (when available) of the first picture of reference pictures list 1. When using hierarchical B pictures with more than 2 temporal levels, for about half of the B pictures unsuitable motion vectors are used for deriving the direct mode motion vectors. A similar problem also remains when specifying RPLR commands in order to modify the first entry of reference picture list 1. That is why it is generally recommended to use the spatial direct mode in connection with hierarchical B pictures. It is of course also possible to choose the direct mode for each picture separately.

4.3.4.3 Motion search

The Joint Model [62] specifies that motion vectors for bipredicted blocks are determined by independent motion searches for both reference lists. It is, however, well-known that the coding efficiency for B slices can be improved when the combined prediction signal (weighted sum of list 0 and list 1 predictions) is considered during the motion search, e.g. by employing the iterative algorithm presented in [64].

4.4 Spatial Scalability

For supporting spatial scalable coding, SVC follows the conventional approach of multiple-layer coding, which is also used in MPEG-2 Video / H.262, H.263, and MPEG-4 Visual. Each layer corresponds to a supported spatial resolution and is identified by a layer or dependency identifier *D*. The layer identifier *D* for the spatial base layer is equal to 0, and it is increased by 1 from one spatial layer to the next. In each layer, motion-compensated prediction and intra coding are employed as for single-layer coding. But in order to improve the coding efficiency in comparison to simulcasting different spatial resolutions, additional inter-layer prediction mechanisms are incorporated as illustrated in Figure 6. Although the basic concept for supporting spatial scalable coding is similar to that in prior video standards, SVC contains new tools that simultaneously improve the coding efficiency and reduced the decoder complexity overhead in relation to single-layer coding. In order to limit the memory requirements and decoder complexity, SVC requires that the coding order in base and enhancement layer is identical. All representations with different spatial resolutions for a time instant form an access unit and have to be transmitted successively in increasing order of their layer identifiers *D*. But as illustrated in Figure 4.3, lower layer pictures do not need to be present in all access units, which make it possible to combine temporal and spatial scalability.

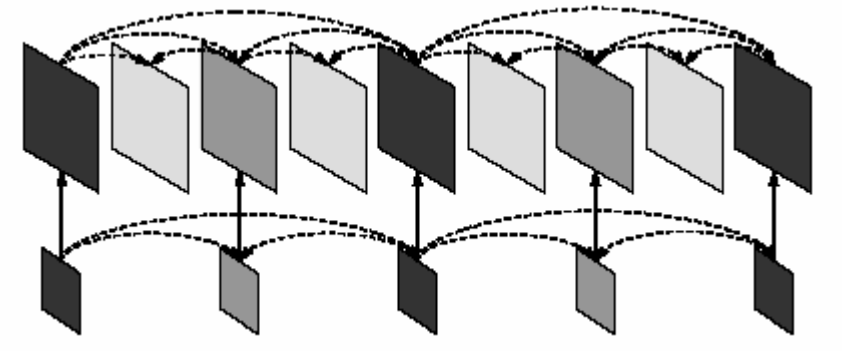


Figure 4.3 Multi-layer structure with additional inter-layer prediction for enabling spatial scalable coding [24]

4.4.1 Inter-layer Prediction

With multi-layer concepts, the prediction between spatial layers is the most important aspect for improving the coding efficiency of the enhancement layers in relation to simulcast. The main goal consists of designing inter-layer prediction tools that enable the usage of as much as possible base layer information for improving the rate-distortion efficiency of the enhancement layers. The only supported inter-layer prediction concept in MPEG-2 Video / H.262, H.263, and MPEG-4 Visual employs the reconstructed samples of the base layer signal. The prediction signal is either formed by upsampling the reconstructed base layer signal or by averaging this upsampled signal with a temporal prediction signal. Although the reconstructed base layer samples represent the complete base layer information, they are not necessarily the most suitable data that can be used for inter-layer prediction. Usually the inter-layer predictor has to compete with the temporal predictor, and especially for sequences with slow motion and high spatial detail, the temporal prediction signal mostly presents a better approximation of the original signal than the upsampled base layer reconstruction. In

order to improve the coding efficiency for spatial scalable coding, two additional inter-layer prediction concepts have been added in SVC [67]: The prediction of motion parameters and the prediction of the residual signal. When neglecting the minor syntax overhead for spatial enhancement layers, the coding efficiency of spatial scalable coding should never become worse than that of simulcast, since in the extreme case none of the inter-layer prediction tools doesn't need to be used. That's why all inter-layer prediction mechanism have been made switchable in SVC, so that an encoder can freely chose between intra- and inter-layer prediction based on the local signal characteristic. Inter-layer prediction can only take place inside an access unit, and from a layer with a lower layer identifier *D* than that of the layer to be predicted. The layer which is employed for inter-layer prediction is signaled in the slice header of the enhancement layer slices. Since the incorporated inter-layer prediction concepts include techniques for motion and residual prediction, the prediction structures of all spatial layers should be temporally aligned for an efficient use of inter-layer prediction. Although the SVC design supports spatial scalability with arbitrary resolution ratios, in the following, only the original inter-layer prediction concepts based on simple dyadic spatial scalability are described. Extensions of these concepts are briefly summarized in sec. 4.4.2.

4.4.1.1 Inter-layer motion prediction

For spatial enhancement layers, the SVC design includes a new macroblock mode, which is referred to as BlSkip. In this mode only a residual signal, but no additional side information as intra prediction modes or motion parameters is

transmitted. With conventional dyadic spatial scalability a macroblock in an enhancement layer corresponds to an 8x8 sub-macroblock in its base layer. When a macroblock is coded using the BlSkip mode and the corresponding 8x8 base layer block lies inside an intra-coded macroblock, the macroblock is predicted by inter-layer intra prediction as it will be explained in sec. 4.4.1.3. When, however, the base layer macroblock is inter-coded, the enhancement layer macroblock is inter-coded, too, and its macroblock partitioning together with the associates reference indices and motion vectors are derived from the co-located 8x8 block in the base layer. The macroblock segmentation is obtained by upsampling the partitioning of the co-located 8x8 block in the lower resolution layer. When the base layer 8x8 block is not divided into smaller blocks, the enhancement layer macroblock is not partitioned. Otherwise, each axb sub-macroblock partition in the base layer block corresponds to a (2a)x(2b) macroblock partition in the enhancement layer macroblock (Figure 4.4). For macroblocks or sub-macroblocks that are coded in direct mode, the partitioning usually depends on the derived motion vectors. But, identical decoding results are obtained when it is assumed that these blocks are always divided into 4x4 sub-macroblock partitions. For the obtained macroblock partitions, the same reference indices as for the corresponding sub-macroblock partitions of the 8x8 base layer block are used; and both components of the associated motion vectors are scaled by a factor of 2. In addition to this new macroblock type, the SVC concept includes the possibility to use a scaled motion vector of the lower resolution as motion vector predictor for the conventional motion-compensated macroblock modes. A flag that is transmitted with each motion vector

difference indicates whether the motion vector predictor is build by conventional spatial prediction or by the corresponding scaled base layer motion vector.

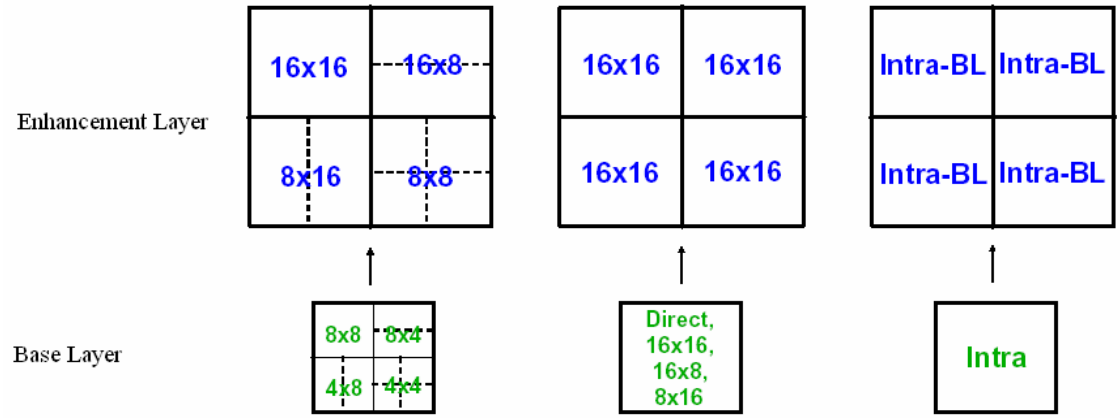


Figure 4.4 Spatial prediction of data

4.4.1.2 Inter-layer residual prediction

When employing the inter-layer motion prediction by using the BlSkip mode, the motion rate that has been transmitted for the co-located sub-macroblock in the base layer is virtually at least partly re-used in the enhancement layer. However, the bits that have been transmitted for coding the base layer prediction error represent useless side information for the enhancement layer. In order to provide the possibility to benefit from this information for the enhancement layer coding, interlayer residual prediction was added to the SVC design. The inter-layer residual prediction can be employed for all inter-coded macroblocks regardless whether they are coded in the new BlSkip mode or with any of the conventional macroblock types. A flag is added to the macroblock syntax for spatial enhancement layers, which signals the usage of inter-layer residual prediction. When this flag is true, the residual signal of the corresponding 8×8 base layer sub-macroblock is blockwise upsampled using a bi-linear filter and used

as prediction for the residual signal of the enhancement layer macroblock, so that only the corresponding difference signal is coded in the enhancement layer. The upsampling of the base layer residual is done on a transform block basis in order to ensure that no filtering is applied across transform block boundaries, by which disturbing signal components could be generated.

4.4.1.3 Inter-layer intra prediction

When an enhancement layer macroblock is coded in BlSkip mode and the co-located 8×8 sub-macroblock in its base layer is intra-coded, the prediction signal of the enhancement layer macroblock is obtained by inter-layer intra prediction, for which the corresponding reconstructed intra signal of the base layer is upsampled. For upsampling the luma component, one-dimensional 6-tap FIR filters ([1, -5, 20, 20, -5, 1]/32) are applied horizontally and vertically. The chroma components are upsampled by using a simple bi-linear filter. The filtering is always performed across sub-macroblock boundaries using the samples of neighboring intra blocks. When the neighboring blocks are not intra-coded, the required samples are generated by specific border extension algorithms. It is always avoided to reconstruct inter-coded macroblocks in the base layer in order to allow single-loop decoding, which will be explained in sec. 4.4.3. To prevent the generating of disturbing signal components, the H.264 / MPEG-4 AVC deblocking filter is applied to the reconstructed intra signal of the base layer before it is used for upsampling (Figure 4.11).

4.4.2 Generalized Spatial Scalability

Similar to MPEG-2 Video / H.262 and MPEG-4 Visual, SVC supports spatial scalable coding with arbitrary resolution ratios. The only restriction is that both the horizontal and vertical resolution must not decrease from a base to an enhancement layer. With the SVC design it is further possible that an enhancement layer picture represents only a selected rectangular area of the corresponding base layer picture, which is coded with a higher or identical spatial resolution. Or in the enhancement layer picture, additional parts beyond the borders of the base layer picture are added. The base and enhancement layer cropping, which may also be combined, can even be modified on a picture basis. Furthermore, the SVC design also includes tools for spatial scalable coding of interlaced sources. For both extensions, the generalized spatial scalable coding with arbitrary resolution ratios and cropping as well as for the spatial scalable coding of interlaced sources, the three basic inter-layer prediction concepts are maintained. But especially the derivation process for motion parameters as well as the design of appropriate upsampling filters for residual and intra blocks needed to be generalized. For a detailed description of these extensions the reader is referred to [68] and [69].

It should be noted that in an extreme case of spatial scalable coding, both the base and enhancement layer have an identical spatial resolution and no cropping is applied. This case actually represents SNR scalable coding, which is also referred to as coarse-grain SNR scalable (CGS) coding (sec. 4.5). As a specific feature of this configuration, the deblocking of the base layer intra signal for inter-layer intra

prediction is omitted, since the transform block boundaries in base and enhancement layer are aligned. It is however still possible to use a 4×4 transform in the base layer and an 8×8 transform in the enhancement layer, or vice versa.

4.4.3 Complexity Considerations

The design of the SVC inter-layer prediction concepts was not only conducted from a coding efficiency point of view, but also by complexity considerations. The possibility of employing inter-layer intra prediction is restricted to selected enhancement layer macroblocks. The coding efficiency can generally be improved (see sec. 4.4.4) by allowing this prediction mode for all enhancement layer macroblocks as it was done in the initial design [67]. In [51] and [70] it was however shown that the decoder complexity can be significantly reduced by constraining the usage of inter-layer intra prediction. The general idea is to avoid the computationally complex operations of motion-compensated prediction and deblocking for all inter-coded base layer macroblocks. This can be realized when the usage of inter-layer intra prediction is only allowed for enhancement layer macroblock, for which the co-located base layer signal is completely intra-coded. It is further required that the base layer and all intermediate layers are coded using constrained intra prediction, so that the intra macroblocks can be constructed without reconstructing any interceded macroblock. With these restrictions, which are mandatory in SVC, each supported layer can be decoded with a single motion-compensation loop. Note that the complexity reduction is even more important for CGS coding than for e.g. dyadic spatial scalable coding. The

decoder complexity overhead in comparison to single-layer coding for SVC is smaller than that for previous video coding standards, which all require multiple motion-compensation loops at the decoder side.

The feature of single-loop decoding also reduces the memory requirement, since decoded samples of lower layers do not need to be stored in the decoded picture buffer for inter-layer prediction. This is also a reason why inter-layer prediction is only allowed inside an access unit. Additionally, it should be mentioned that a CGS or spatial enhancement layer NAL unit can be parsed independently of the corresponding base layer NAL units, which provides the possibility to further reduce the complexity of decoder implementations [71].

4.4.4 Coding Efficiency

The effectiveness of the inter-layer prediction concepts for spatial scalable coding is evaluated in comparison to single-layer coding as well as simulcast. The base layer was coded at a fixed bit-rate, for encoding the spatial enhancement layers, the bit-rate as well as the amount of enabled inter-layer prediction mechanisms was varied. Additional simulations were run with multiple-loop decoding. For these runs, the restriction for the inter-layer intra prediction in the current SVC design was removed. Only the first access unit was intra-coded and CABAC was used as entropy coding method. Simulations have been carried out for a GOP size of 16 pictures as well as for IPPPP coding. All encoders have been rate-distortion optimized following [61]. For each access unit, first the base layer picture is encoded, and given the corresponding

coding parameters, the enhancement layer picture is coded [65]. The inter-layer prediction tools are considered as additional coding options for the enhancement layer pictures in the operational encoder control. The lower resolution sequences have been generated following the method in [65]. The simulation results for the sequence "City are depicted in Figure 4.5. All inter-layer prediction (ILP) tools, intra (I), motion (M), and residual (R) prediction, improve the coding efficiency in comparison to simulcast. However, the effectiveness of a tool strongly depends on the sequence characteristic. Multiple-loop decoding can further improve the coding efficiency. But the gain is often minor and comes along with a significant increase in decoder complexity. Particularly, the coding efficiency for multi-loop decoding with only inter-layer intra prediction enabled, which is comparable to the concepts of MPEG-2 Video / H.264, H.263, and MPEG-4 Visual, is usually worse than the coding efficiency for the SVC design. Furthermore, it can be noted that the usage of hierarchical prediction structures does not only improve the overall coding efficiency, but also the effectiveness of the inter-layer prediction mechanisms.

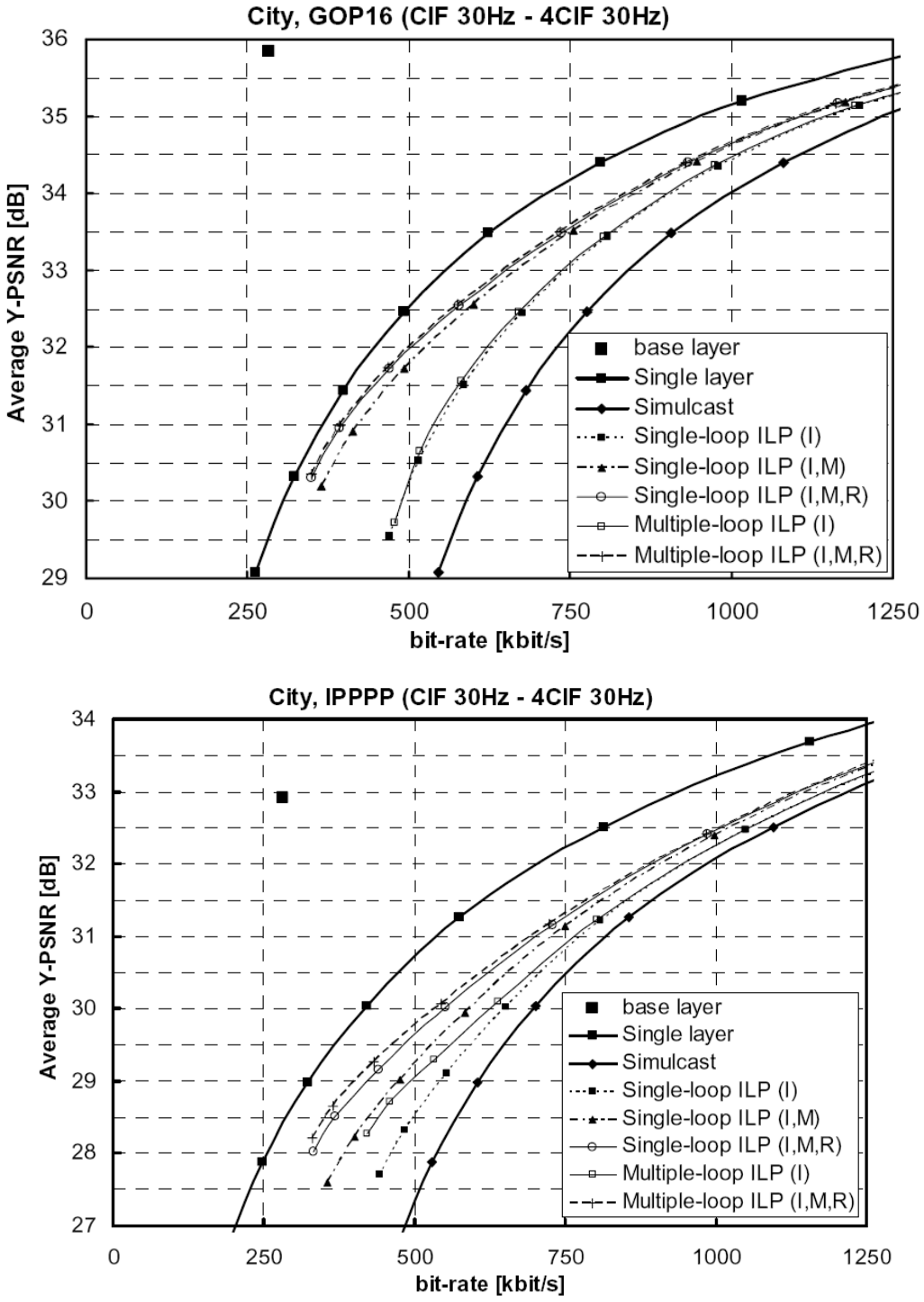
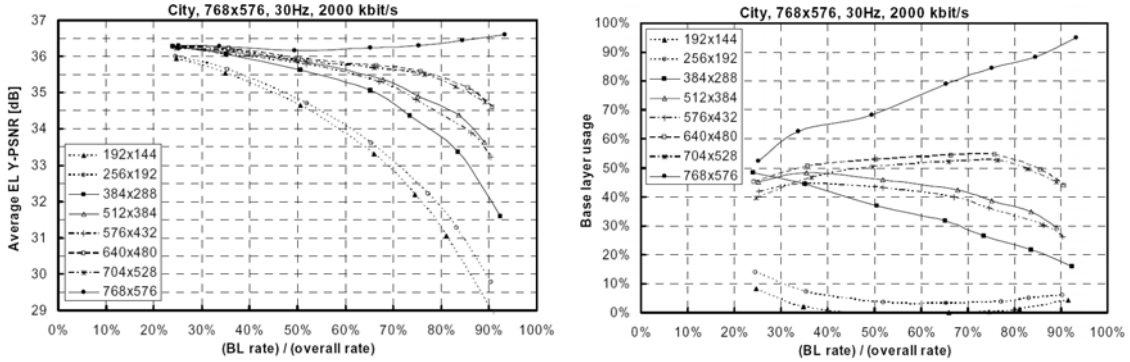


Figure 4.5 Analysis of the efficiency of the inter-layer prediction concepts in SVC for different prediction structures. The rate-distortion point for base layer is plotted inside the diagrams, but it should be noted that it corresponds to a different spatial resolution [24]

In a second experiment, the efficiency of spatial scalable coding was investigated in dependence on the resolution and rate ratio between base and enhancement layer. The enhancement layer resolution was set to 768×576 luma samples, and the overall bit-rate was fixed at 2000 kbit/s. Both the spatial resolution of the base layer as well as the ratio between base layer and overall bit-rate was varied. The simulation results for the sequence "City" are summarized in Figure 4.5. In Figure 4.6(a), the coding efficiency is measured as the average luma PSNR of the enhancement layer. However, the enhancement layer PSNR is only partly suitable for evaluating the efficiency of spatial scalability, because it does not consider the amount of the base layer rate that is used for enhancement layer coding.



(a) (b)

Figure 4.6 Efficiency of the inter-layer prediction in dependence of resolution and bit-rate ratios of (a) enhancement layer and (b) base layer [24]

In Figure 4.6(b), the coding efficiency is expressed by an alternative measure, the base layer usage Bu, which is calculated by Bu = 1 – (RE – RS) / RB. The overall and the base layer bit-rates are presented by RE and RB, respectively. RS is the bit-rate that would be required for obtaining the same PSNR as for the spatial enhancement layer, but by using single-layer coding. This rate is calculated by interpolating a single-layer rate-distortion curve for the enhancement layer resolution using cubic spline interpolation. It should be noted that a base layer usage Bu of 0% corresponds to the coding efficiency of simulcast, while Bu equal to 100% represents the efficiency of single-layer coding. The experiment shows that the effectiveness of spatial scalable coding depends on the resolution ratio and the ratio of base layer and overall bit-rate as well as on the sequence characteristic.

4.4.5 Encoder Control

The encoder control as usually employed for multi-layer coding [65] represents a bottom-up process. For each access unit, first the coding parameters of the base layer are determined, and given these data, the enhancement layer is coded. This however might limit the achievable enhancement layer coding efficiency, since the chosen base layer coding parameters are only optimized for the base layer, but they are not necessarily suitable for an efficient enhancement layer coding. A similar effect is obtained by using downsampled sequences as an input for the base layer coding. Given these sequences the encoder control minimizes the reconstruction error in relation to these downsampled originals. By using a different downsampling process it might

however find base layer coding parameters that are more suitable for the enhancement layer coding, while both reconstructed base layer sequences show a comparable subjective quality. First experimental results for an improved multi-layer encoder control are presented in [72]. The algorithm determines the coding parameters based on a weighted sum of the Lagrangian costs for the base and enhancement layers. Via the corresponding weighting factor it is possible to trade off base and enhancement layers coding efficiencies. In Figure 4.7, this is demonstrated for CGS coding using an IPPPP coding structure and the sequence "Soccer".

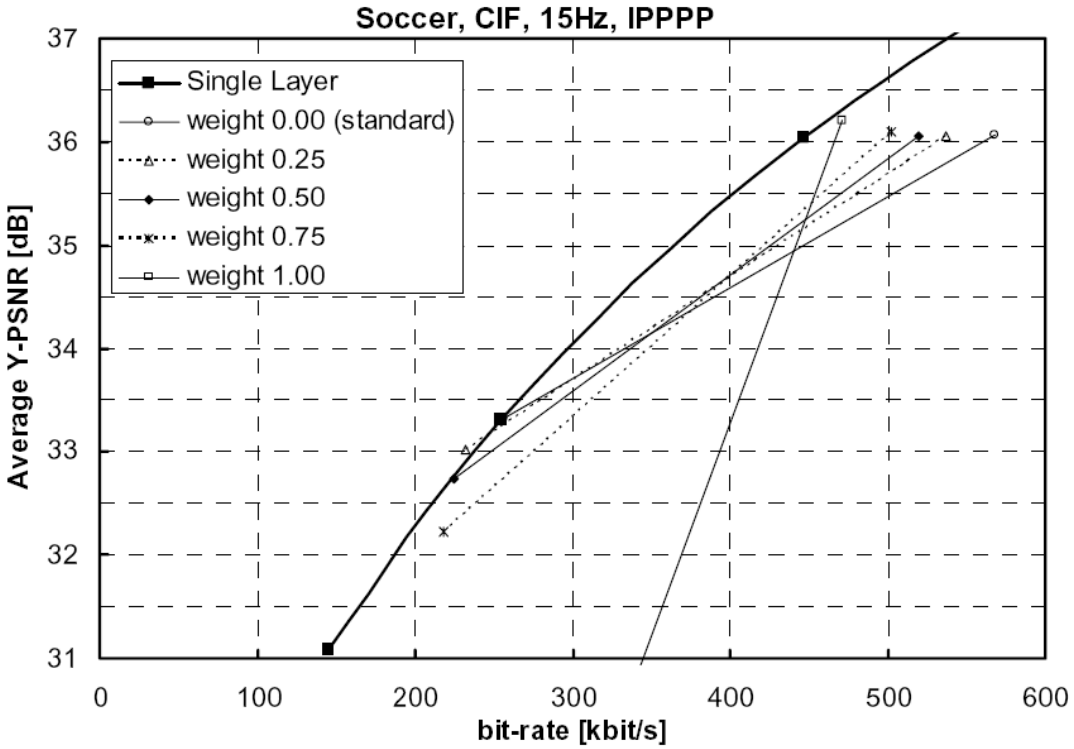


Figure 4.7 Joint encoder control for multi-layer coding [24]

By increasing the weight for the enhancement layer, its coding efficiency is improved, but at the same time the base layer coding efficiency gets worse. In the

extreme case that only the enhancement layer is considered for determining all coding parameters, the enhancement layer coding efficiency is nearly identical to that of single-layer coding. However, the base layer becomes useless, since its reconstruction quality is not taken into account by the encoder control. The base layer is purely used as a data partition for the enhancement layer coding. However, such an encoder control allows adjusting the importance of base and enhancement layer coding efficiency according to the needs of an application.

4.5 Quality / SNR Scalability

SNR scalability can be considered as a special case of spatial scalability for which the picture sizes of base and enhancement layer are identical. As already mentioned in sec. 4.4, this case is supported by the general concept for spatial scalable coding and it is also referred to as coarse-grain SNR scalable (CGS) coding. The same inter-layer prediction mechanisms as for spatial scalable coding are employed, but without the corresponding upsampling operations. However, with this multi-layer concept for SNR scalable coding only a few selected bit-rates can be supported in a scalable bit-stream. In general, the number of rate points is identical to the number of layers. A switching between different layers can only be done at defined points in the bit-streams (see sec. 4.6). Furthermore, as it will be demonstrated in sec. 4.4.4 the multi-layer concept for SNR scalable coding becomes inefficient, when the relative rate difference between successive layers is small. Although CGS coding is simple and

characterized by a low decoder complexity overhead in relation to single-layer coding, it does not provide enough flexibility for all applications.

Especially for increasing the flexibility for bit-stream adaptations and the error robustness, but also for improving the coding efficiency for bit-streams that have to provide a variety of bit-rates, an additional approach for fine-granular SNR scalable (FGS) coding is included in the SVC design. FGS coding is based on so-called progressive refinement (PR) slices. The SVC syntax allows that up to 3 layers of PR slices are coded on top of a base layer picture. These SNR refinement layers are identified by a quality layer identifier Q, which is equal to 0 for the base layer pictures and increases by 1 for each SNR refinement layer. The NAL units containing PR slices have the unique property that they can be truncated at any byte-aligned position or arbitrarily discarded from an SVC bit-stream without influencing its decodability. When PR slices are employed in connection with a suitable encoder configuration (see sec. 4.5.1), any truncation or discarding of the corresponding NAL units reduces the reconstruction quality of the bit-stream in a fine-granular way. FGS coding is thus especially suitable for streaming applications in which the video bit-rate has to be frequently adapted to the channel conditions, or in combination with unequal error protection for environments that are characterized by significant packet loss rate.

4.5.1 Controlling the Drift in SNR scalable coding

Regardless of the technique that is applied for encoding the actual SNR refinement signal, the process of motion-compensated prediction for SNR scalable

coding has to be carefully designed, since it determines the trade-off between enhancement layer coding efficiency and drift ([73]). The drift describes the effect that the motion compensation loop in the decoder and the motion compensation loop in the encoder, which is employed for controlling the encoder, are not synchronized, e.g. because SNR refinement packets are truncated or discarded from a bit-stream. Depending on the actual prediction structure the impact can range from a hardly noticeable, graceful degradation of the reconstructed video to an extremely bad reconstruction quality, which is useless for any type of application.

For the FGS coding in MPEG-4 Visual, the prediction structure was chosen in a way that drift is completely omitted. As illustrated in Figure 4.8(a), motion-compensated prediction is only performed using the base layer reconstruction as reference, and thus any loss or modification of an SNR refinement packet does have any impact on the motion compensation loop. The drawback of this approach is that it significantly decreases the enhancement layer coding efficiency in comparison to single-layer coding. Since only that base layer signal is used for motion-compensated prediction, the bit-rate that is spend for encoding the enhancement layer of a picture cannot be employed for the coding of following pictures. For the SNR scalable coding in MPEG-2 Video / H.262, the other extreme case of possible prediction structures that is depicted in Figure 4.8(b) was specified. Here, the reference with the highest available quality is always employed for motion-compensated prediction.

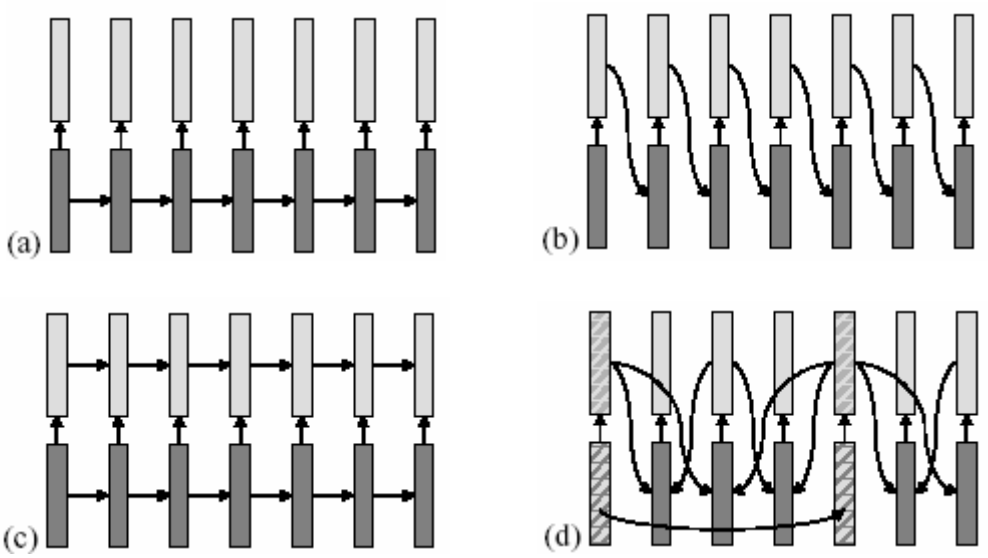


Figure 4.8 Different concepts for trading off enhancement layer coding efficiency and drift: (a) base layer only control, (b) enhancement layer only control, (c) two loop control, (d) key picture concept of SVC for hierarchical prediction structures where the so-called key pictures are marked by the hatched boxes. [24]

This ensures a high coding efficiency for the enhancement layer as well as low complexity, since only a single reference picture need to be stored for each time instant, however, any loss or modification of an SNR refinement packet results in a drift that can only be controlled by intra updates. This concept is not suitable for enabling flexible bit-rate adaptations, it can only be used as a method for increasing the error robustness of a video transmission. As an alternative, an approach with two motion compensation loops as illustrated in Figure 4.8(c) could be employed. This concept is similar to spatial scalable coding as specified in MPEG-2 Video / H.262, H.263, and MPEG-4 Visual. Although the base layer is not influenced by any modification of the enhancement layer, any loss of information in the enhancement layer results in a drift for the enhancement layer reconstruction.

The SVC design includes two different concepts for allowing a reasonable adjustment of the trade-off between enhancement layer coding efficiency and drift. The first concept was especially designed for enabling a simple but effective drift control for hierarchical prediction structures. For each picture a flag is transmitted, which signals whether the base layer representation (when available) or the enhancement layer representation of the reference pictures are employed for motion-compensated prediction. Pictures that use the base layer representation for motion-compensated prediction are referred to as key pictures. In order to limit the memory requirements, a second syntax elements signals whether the base representation of a pictures is additionally stored in the decoded picture buffer. The enhancement layer reconstruction is always inserted in the decoded picture buffer, except for non-reference pictures. Figure 4.8(d) illustrates how the key picture concept can be efficiently combined with hierarchical prediction structures. The pictures of the coarsest temporal level are transmitted as key picture, and only for these picture the base representation is inserted in the decoded picture buffer. Thus, no drift is introduced in the motion compensation loop of temporal level 0. In contrast, all temporal refinement pictures employ the reference with the highest available quality for motion-compensated prediction, which results in a high coding efficiency for these pictures. Since the key pictures serve as re-synchronization points between encoder and decoder reconstruction, the drift propagation is efficiently limited to the inside of a group of pictures. The trade-off between enhancement layer coding efficiency and drift is adjusted by the GOP size or the number of hierarchy stages. It should be noted that both the SNR scalability

structure in MPEG-2 Video / H.262 (no picture is coded as key picture) and the FGS coding approach in MPEG-4 Visual (all pictures are coded as key pictures) represent special cases of the key pictures concept in SVC. And furthermore, similar to spatial scalable coding in SVC, only a single motion compensation process is required for each picture.

With the key picture concept alone, efficient SNR coding can only be realized in connection with sufficiently large GOP sizes. However, for low-delay applications often require SNR scalable coding based on a conventional IPPPP structure. In order to also improve the coding efficiency for such scenarios, leaky prediction concepts following the basic ideas in [39]-[43] have been additionally included in the SVC design. With leaky prediction enabled, the motion-compensated prediction signal for key pictures is given as the weighted average of the base and enhancement layer reconstruction of the corresponding reference picture. Thus the drift that is introduced due to encoder-decoder mismatches of the enhancement layer references decays over time. For optimizing the coding efficiency, the weighting factors are adaptively adjusted based on local signal statistics. Furthermore, the strength of the enhancement layer weighting can be signaled in the bit-stream. For more detailed information on the leaky prediction concept in SVC, the reader is referred to [74].

4.5.2 Progressive Refinement Slices

Progressive refinement slices have been designed for efficiently representing SNR refinements and allowing the truncation of the corresponding NAL

units. Each PR slice basically corresponds to a bisection of the quantization step size or an increase by 6 of the quantization parameter QP. The quantization parameter QP for a macroblock can only be freely chosen, when no non-zero transform coefficient was transmitted in the co-located macroblock of any subordinate layer. Otherwise, the quantization parameter is derived from the QP of the corresponding lower layer macroblock. At the decoder side, the transform coefficient levels of the base and all enhancement layers are scaled by the scaling factor which is determined by the quantization parameters, these scaled values are added up, and a single inverse transform is applied to obtain the reconstructed residual signal.

Although the quantization step size is usually halved from one layer to the next, the SVC concept for SNR scalable coding substantially differs from bit-plane coding as it is applied in MPEG-4 Visual or most of the 3-d wavelet codecs. The SVC design generally allows a greater freedom in encoder decision for determining transform coefficient levels. It was mainly influenced by the observation that the possibility to arbitrarily adjust the transform coefficient levels of each SNR refinement layer has a much greater impact on the coding efficiency than an efficient bit-plane coding for given transform coefficients. As an example, with bit-plane coding, a transform coefficient level unequal to 0 has to be transmitted for any transform coefficient that lies outside the quantization interval around zero; otherwise, a very large reconstruction error in following refinement layers would be obtained. With the SVC concept of re-quantization, however, a transform coefficient level equal to 0 could be transmitted in the base layer, because this would minimized the rate-distortion cost in

the base layer and thus maximizing its coding efficiency. Then, in the next enhancement layer, a transform coefficient level greater than 1 could be transmitted, since this would maximize the coding efficiency of this enhancement layer. The only restriction in the current SVC draft is that for transform coefficients, for which a non-zero level was transmitted in one of the subordinate layers, only the level values -1, 0, and 1 are allowed.

With PR slices, the SNR refinement signal is represented in a coarse-to-fine representation. The transform coefficient levels are usually not transmitted macroblock by macroblock. Instead they are processed in several scans, and in each scan only a few transform coefficient levels for each transform block are coded. The coefficient scanning can be influenced by syntax elements of the slice header, and thus it is possible to adjust the trade-off between decoder complexity, which increases with the number of scans, and the quality of the coarse to fine representation, which determines the coding efficiency for truncated FGS layers. For more details on this so-called cyclic block scanning in SVC the reader is referred to [74].

In SVC, it is also possible to include a refinement of motion parameters in PR slices. This is especially useful, when SNR scalability has to be provided for a large bit-rate interval. With a single motion vector field the trade-off between motion and texture rate can only be optimized for a single bit-rate or a small interval. Similarly to CGS coding, the possibility to refine the motion vector field of the base layer for the enhancement layer coding allows to increase the coding efficiency for larger rate intervals. In order to limit the decoder complexity, the motion refinement in PR slices is

not allowed for key pictures, here only the concept of leaky prediction can be employed. More details about the refinement of motion information in PR slices are given in [75].

It should be further noted that it is not only possible to truncate NAL units that contain PR slices, but also to distribute the data of a PR slice to several NAL units. Thus, for example only the first part of a PR slice is employed as a reference for inter-layer prediction of the next spatial layer (sec. 4.6). But the entire PR slice can be used for decoding the lower layer resolution. The PR NAL units that are not employed for interlayer prediction are also called discardable sub-streams.

4.5.3 Encoder Control

As described in sec. 4.5.3, except for key pictures, the motion-compensated prediction for SNR scalable coding is always performed by employing the highest available quality of the corresponding reference picture. However, during the encoding process it is not known what representation will be available in the decoder. The encoder has to decide what reference it will use for motion estimation, mode decision, and the determination of the residual signal to be coded (motion compensation). This decision influences the coding efficiency for the supported rate points. Several investigations [76][77] turned out that a suitable coding efficiency is obtained when the prediction loop in the encoder is closed at the highest rate point. I.e. for the processes of motion estimation, mode decision, and motion compensation the reference with the highest reconstruction quality is employed. Note that this is different from so-called open-loop coding, in which the original of the reference pictures is used.

In [76][77] it is additionally pointed out that the base layer coding efficiency can be improved by a two-loop encoder control, in which the base layer residual to be coded is determined by a second motion compensation process for which the base layer references are used. The impact on enhancement layer coding efficiency is usually minor.

4.5.4 Bit-stream Extraction

For extracting a sub-stream with a specific average bit-rate from a given SNR scalable bit-stream usually a huge number of possibilities exist. The same average bit-rate can be adjusted by truncating or discarding different SNR refinement packets. However, the coding efficiency that corresponds to the bit-rate is dependent on the extraction method. With a very simple method that is widely used in experiments, all refinement packets are truncated by the same percentage. In a more sophisticated method, a priority identifier is assigned to each packet by an encoder. During the bit-stream extraction, first all packets with the lowest priority value are truncated or discarded, and when the target bit-rate is not reached the packets of the next priority values or truncated or discarded, etc. The priority identifiers can either be fixed by the encoder based on the employed coder structure or determined by a rate-distortion analysis. The SVC syntax (sec. 4.6) provides different means to include such priority information in the bitstream. For more detailed information about the concept of optimized bit-stream extraction, which is also referred to as quality layers, the reader is referred to [78].

4.5.5 Coding Efficiency

In a first experiment the different concepts for controlling the drift are evaluated for both hierarchical B pictures with a GOP size of 16 pictures and IPPPP coding. With exception of the 2-loop control, all configurations could be realized with an SVC compliant coder. The results for the sequence "City" and "Crew" are summarized in Figure 4.9. When the motion compensation loop is only closed in the base layer (BL-only control) as in MPEG-4 FGS, no drift occurs, but the enhancement layer coding efficiency is very low, especially for sequences like "City" for which the motion-compensated prediction works very well. By closing the loop only at the enhancement layer (EL-only control) as it is done in the SNR scalable mode of MPEG-2 Video / H.262, a high enhancement layer coding efficiency can be achieved. But any modification to the enhancement layer sub-stream results in a serious drift, and the reconstructed video quickly becomes unusable, especially for IPPPP coding structures. A similar behaviour can also be observed for the 2-loop control, but here the reconstruction quality stabilizes for low rates at the base layer level. For the sequence "Crew" these impacts are less obvious, since a substantial part of the macroblock is intra-coded and the differences only apply for inter coding. With the SVC key picture concept (adapt. BL/EL control), in which the pictures of the coarsest temporal level are coded as key pictures, a reasonable coding efficiency for the entire supported rate interval can be achieved in connection with hierarchical prediction structures. For IPPPP coding, the best coding efficiency over the entire rate range is obtained with the

leaky prediction concept. For hierarchical B pictures, however, the additional gain relative to the simply key picture concept is minor.

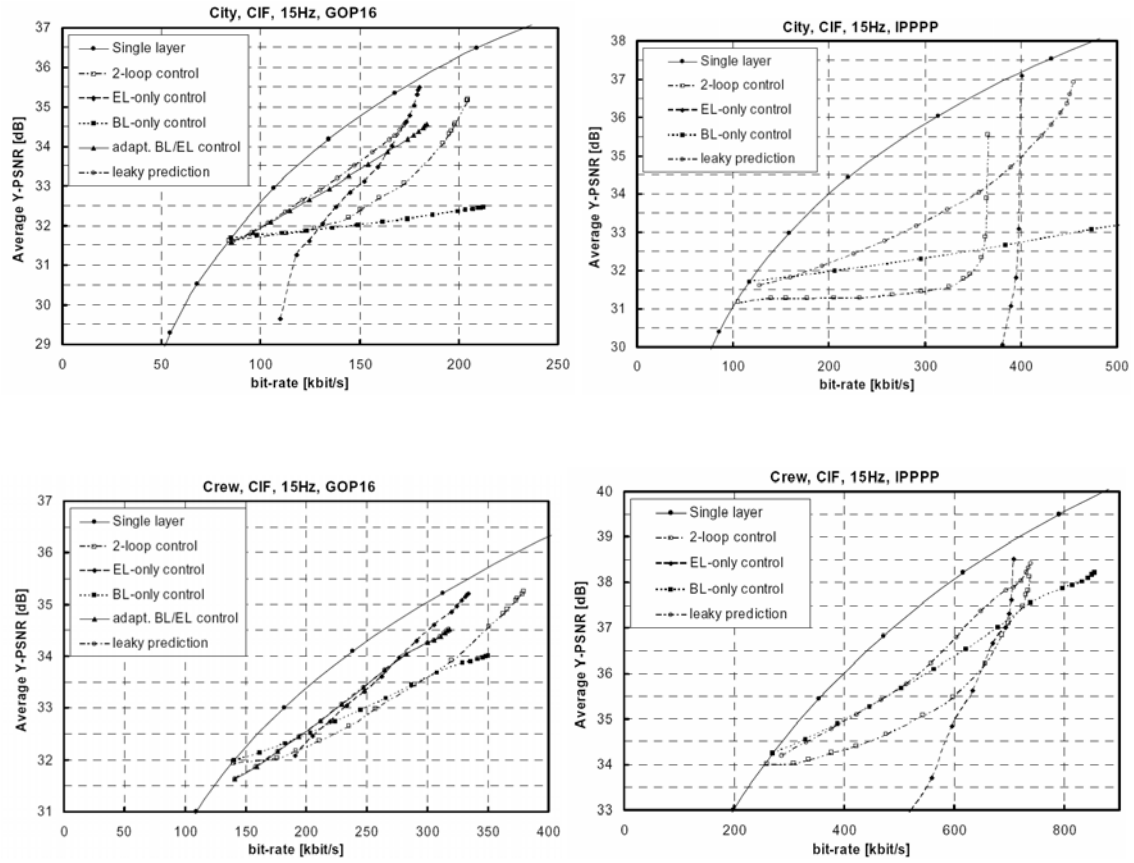


Figure 4.9 Comparison of concepts with different tradeoffs between enhancement layer coding efficiency and drift [24]

In a second experiment the different concepts for providing SNR scalability are evaluated. In Figure 4.10(a) and (b), the coding efficiency of CGS and FGS coding is compared to that of single layer coding for different coding structures. For the SNR scalable bit-streams, the bit-rate interval between the lowest and highest supported rate point correspond to a QP difference of 12. For hierarchical B pictures

with a GOP size of 16 pictures, the results of two FGS runs, one with and one without motion refinement, are plotted inside the diagrams. It can clearly be seen that the possibility to refine motion information in FGS slices can improve the coding efficiency when large rate intervals need to be supported. By comparing the different CGS runs, it can be seen that the coding efficiency generally decreases with an increasing number of supported rate points. In the CGS run that are labeled as "non-adaptive" all possible inter-layer prediction mechanisms are always employed. The results show that similarly to spatial scalable coding, it is important that the usage of inter-layer prediction is adaptively controlled based on the local signal statistics. Especially the trade-off between motion and residual rate needs to be optimized for each layer. The diagrams also contain rate-distortion curves for CGS with multiple-loop decoding, which is not supported by the SVC design. The results indicate that also for CGS coding, the usage of multiple-loop decoding only slightly increases the coding efficiency. But as discussed in sec. 4.4.3 it significantly increases the decoder complexity. Figures 4.10(c) & (d) compare different FGS runs that have been obtained by varying the Lagrangian parameter for the motion estimation and mode decision process. It demonstrated how the coding efficiency of the lower and higher rate points can be traded off. Additionally this figure also shows the effect of using optimized methods for bit-stream extraction (sec. 4.5.4). In Figure 4.10(d) the effect of using different strengths of leaky prediction is demonstrated. The presented simulation results also show that the coding efficiency loss of SNR scalable coding in relation to single layer coding is usually smaller when employing hierarchical prediction structures.

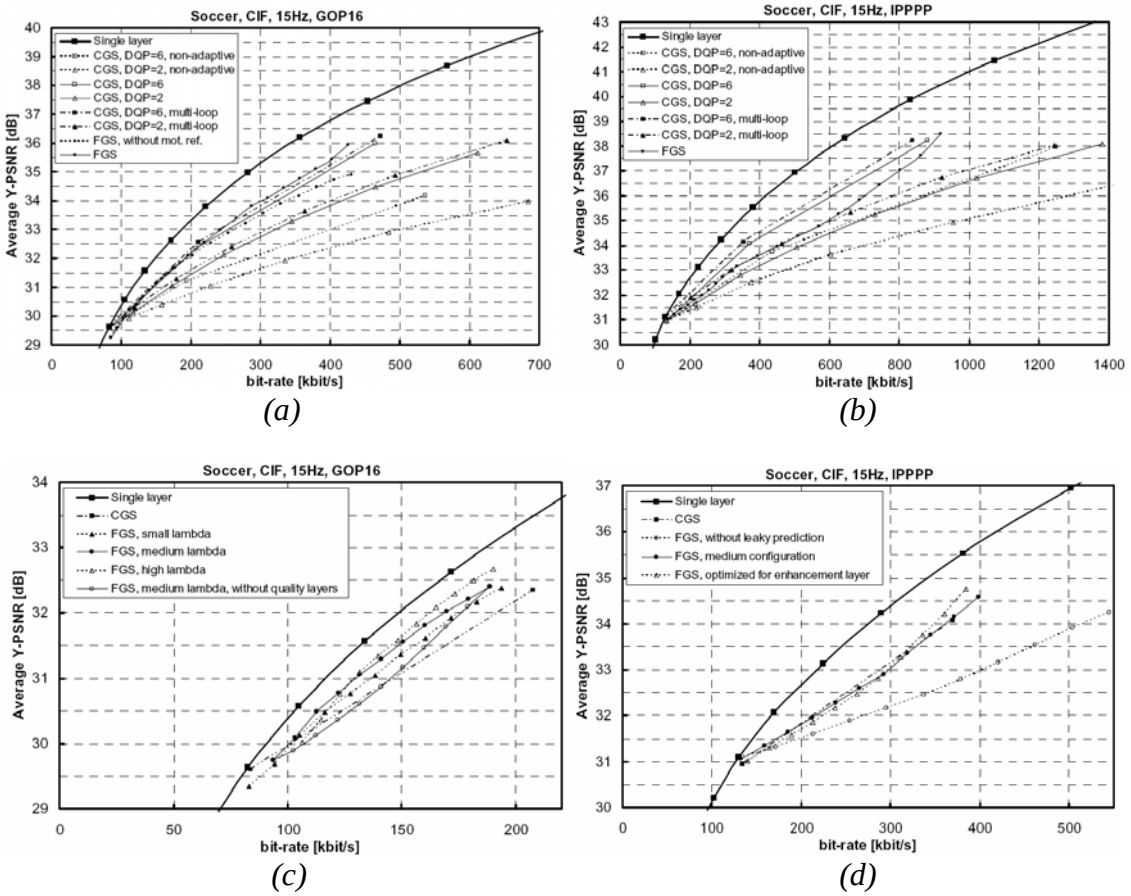


Figure 4.10 Comparison of coarse-grain and fine-grain SNR scalable coding with different configurations [24]

4.6 SVC Design

In the SVC design, the basic concepts for temporal, spatial, and SNR scalability as described in sec. 4.3, 4.4 and 4.5 are combined. In order to enable simple bit-stream adaptation SVC additionally provides means by which the sub-streams that are contained in a global scalable bit-stream can be easily identified. An SVC bit-stream doesn't need to provide all types of scalability. Since the support of SNR and spatial scalability usually comes along with a loss in coding efficiency relative to single-layer

coding, the trade-off between coding efficiency and the provided degree of scalability can be adjusted according to the needs of an application. For a comparison of spatial, SNR, and combined scalability with single layer coding the reader is referred to [79].

4.6.1 Combined Scalability

The general concept for combining spatial, SNR, and temporal scalability is illustrated in Figure 4.11, which shows an example encoder structure with two spatial layers. The SVC coding structure is organized in layers. A layer usually represents a specific spatial resolution. In an extreme case it is also possible that the spatial resolution for two layers is identical (CGS). Layers are identified by a layer identifier D. The spatial resolution must not decrease from one layer to the next. For each layer, the basic concepts of motion-compensated prediction and intra prediction are employed as in single-layer coding, the redundancy between layers is exploited by additional interlayer prediction concepts as explained in sec. 4.4.1.

The concepts for fine-grain SNR scalability as described in sec. 4.5 are inserted within a layer. That means that progressive refinement slices can be coded in each layer, usually in order to refine the reconstruction quality of the specific spatial resolution as illustrated in Figure 4.11. The SNR refinement levels inside each layer are identified by a quality level identifier Q. When however the spatial base layer contains different SNR representation, it needs to be signaled which of these is employed for inter-layer prediction.

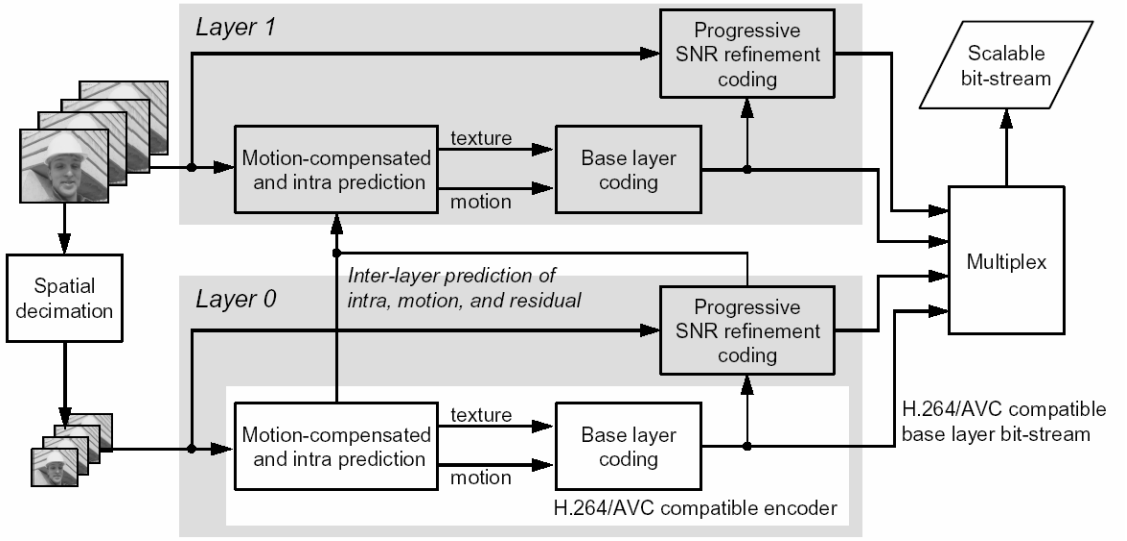


Figure 4.11 SVC Encoder structure example [24]

Therefore, SVC slices include a syntax element which not only signals whether interlayer prediction is employed, but also the layer identifier D and the quality level identifier Q of the corresponding base layer. Since PR slices can also be partitioned into several NAL units as described in sec. 4.5.2, this syntax additional additionally comprises an identifier for the corresponding fragment number F. In order to limit the memory requirement for storing intermediate representations, all slices of a layer at a specific time instant have to use the same base representation identified by D, Q, and F for inter-layer prediction.

One important difference between the concept of layers and SNR levels is that a switching between different layers is only envisaged at defined switching point.

Whereas switching between different SNR levels is virtually possible in any access unit. The SVC design also allows the use of the SNR refinement signaling with layer identifier Q in connection with CGS layers [80]. This does not change the basic decoding process for CGS. Only the high-level signaling and the error detection capability are modified. It should be noted that in this mode, the decoder cannot detect whether a CGS refinement is missing or intentional discarded. This configuration, which is also referred to as medium-grain scalability is mainly suitable in connection with hierarchical prediction structures, for which several CGS refinements can be discarded without influencing the reconstruction for other access units.

In SVC all slice data NAL units for a time instant together with one or more non-VLC NAL units form an access unit. Since inter-layer prediction can only take place from a lower to a higher layer inside an access unit, spatial and SNR scalability can be easily combined with temporal scalability. To all slices of an access unit the same temporal level T is assigned.

In addition to the main scalability types temporal, spatial, and SNR scalability, SVC additionally supports region-of-interest (ROI) scalability. ROI scalability can be realized via the concepts of slice groups (cp. II.B), but the shape of the ROI is restricted the pattern that can be represented as a collection of macroblocks.

4.6.2 System Interface

An important goal of a scalable video coding standard is to support easy bit-stream manipulation. In order to extract a substream with a reduced spatio-temporal

resolution and/or bitrate, all packets that are not required for decoding the target resolution and/or bit-rate should be removed from a bit-stream. In addition, NAL units of PR slices might be additionally truncated. However, therefore parameters like the layer identifier L, the quality level Q, the temporal level T, and the fragment number F need to be known for each slice data NAL unit. Furthermore, it need to be known whether a NAL unit contains a PR slice and can thus be truncated and whether a NAL unit is required for inter-layer prediction of higher layers.

In order to assist easy bit-stream manipulations, the 1-byte header of H.264 / MPEG-4 AVC is extended by additional 3 bytes. This extended header includes all the data mentioned above as well as some additional information. One of the additional syntax elements is a priority identifier Pi, which signals the importance of a NAL unit. It can be used either for simple bit-stream adaptations with a single comparison per NAL unit or for an rate-distortion optimized bit-stream extraction using quality layer information (sec. 4.5.4). SVC also specifies an additional SEI message (SEI – Supplemental Enhancement Information), which contains information like the spatial resolution or bit-rate of the layers that are included in a bit-stream and can further assist the bit-stream adaptation process.

Each SVC bit-stream includes a sub-stream, which is coded in compliance with H.264 / MPEG-4 AVC, so that any standard conforming H.264 / MPEG-4 AVC decoder is capable of decoding this base representation when it is provided with an SVC bit-stream. However, standard H.264 / MPEG-4 AVC NAL units do not contain the extended NAL unit header information, which are very useful for bit-

stream adaptation. In order to attach this additional SVC related information to H.264 / MPEG-4 AVC NAL units, so-called suffix NAL units are introduced. These NAL units directly follow H.264 / MPEG-4 AVC NAL units in an SVC bit-stream, and include the above describe identifiers as well as syntax elements that are actually required for the SVC decoding process but are not transmitted in H.264 / MPEG-4 AVC NAL units.

SVC also defines additional SEI messages, which can further assist the extraction or decoding process. More detailed information on the system interface of SVC is provided in [81]. Information on an RTP payload format for SVC and an SVC file format are given in [82] and [83], respectively.

4.6.3 Bit-stream Switching

As mentioned above, switching between different SNR levels inside a layer is possible in each access unit. However, SVC specifies the switching between different layers, which are identified by a different layer identifier D, only at defined switching points. One of these switching points are IDR access units by which it is signaled that non of the previously transmitted access units is required for decoding all following access units. But in order to enable a more flexible bit-stream switching, SVC also defines so-called EIDR pictures. An EIDR picture in a layer D signals that the reconstruction of layer D for the current and all following access units is independent of all previously transmitted access units. Thus, it is always possible to switch to the layer for which an EIDR picture is transmitted in the current access unit. The advantage of the EIDR pictures in comparison to IDR access units can be seen in the fact that EIDR

pictures only define random access points for a specific layer. For instance when an EIDR picture is coded in an enhancement layer and thus no motion-compensated prediction can be used, it is still possible to employ motion-compensated prediction in the base layer in order to improve its coding efficiency.

Although SVC specifies switching between different layers only for well-defined points, a decoder can be implemented in a way that at least down-switching is possible in virtually any access unit. One way is to do multiple-loop decoding. That is when decoding an enhancement layer; the pictures of the subordinate layer are reconstructed and stored in additional decoded picture buffers although they are not required for decoding the enhancement layer picture. But, when the transmission switches to any of the subordinate layers in an arbitrary access unit, the decoding of this layer can be continued since the an additional DPB has been operated as when the corresponding layer would have been decoded for all previous access unit. Such a decoder behavior requires additional processing power. But it is also possible to restrict the multi-loop decoding to selected pictures like the pictures of the coarsest temporal level. This decreases the required processing power, but still provides access points for down switching in regular intervals. For up switching, the decoder usually has to wait for the next IDR access unit or the next EIDR picture. However, similar to random access in single-layer coding it can also immediately start with decoding the arriving NAL units in connection with suitable error concealment techniques, but it output base layer reconstructions until the reconstruction quality for the enhancement layer has been stabilized (gradual decoder refresh).

4.7 Conclusions

In comparison to the scalable profiles of prior video coding standards, the SVC design provides various tools that improve the efficiency of scalable coding relative to single-layer coding. The most important differences are:

- possibility to employ hierarchical prediction structures for providing temporal scalability with several levels, for improving the coding efficiency, and for increasing the effectiveness of SNR and spatial scalable coding
- inter-layer prediction of motion and residual for improving the coding efficiency of spatial scalable and coarse-grain SNR scalable coding
- single-loop decoding for reducing the complexity and the memory requirement for spatial scalable and coarse-grain SNR scalable coding
- key picture concept for efficiently controlling the drift of SNR scalable coding with hierarchical prediction structures
- leaky prediction for efficiently controlling the drift of SNR scalable coding for low delay scenarios
- efficient coding of progressive refinement slices

A further important point is that the underlying single-layer video coding standard H.264 / MPEG-4 AVC also provides a significant improved coding efficiency in comparison to previous standards. As a result the coding efficiency of SVC bitstreams, which support a huge degree of scalability, is often superior to that of single-layer coding based on prior standards such as MPEG-4 Visual for all supported spatio-temporal resolutions and bit-rates [79]. It should further be noted that the H.264 /

MPEG-4 AVC already supports the usage of hierarchical prediction structures, which turned out to not only provide an improved coding efficiency compared to conventional prediction structures, but also improve the effectiveness of spatial and SNR scalable coding.

Although the SVC design improves the effectiveness of scalable video coding, it cannot circumvent the intrinsic problems. One of these is the trade-off between drift and enhancement layer coding efficiency in SNR scalable video coding. In this context various researchers claimed that MCTF-based coding is drift-free, but still capable of achieving a high enhancement layer coding efficiency. When analyzing MCTF-based coding it becomes obvious that the only difference to the coding with hierarchical prediction structures are the additional motion-compensated update steps. Similar to hybrid video coding any loss of an enhancement layer packet results in a propagating reconstruction error. However, since the low pass picture, which represent the pictures of the coarsest temporal resolution, are usually intra-coded in MCTF-based codecs, the error propagation is limited to the inside of a group of pictures, exactly as for hierarchical prediction structures when using the SVC key picture concept. A fair comparison of MCTF-based coding and the coding with hierarchical B pictures [58][76], in which virtually identical codecs are used, turned out that MCTF does usually not improve the coding efficiency compared to the coding with hierarchical B pictures, neither for single-layer nor scalable coding. The main drawback of MCTF-based coding is the increase in decoder complexity. Furthermore, it requires an open-loop control which cannot compensated for quantization errors of the reference pictures.

Additionally, when decoding lower temporal layers with an MCTF coder, the pictures can contain ghosting artifacts, which are caused by the update steps and can only be suppressed by complicated adaptive weighting algorithms.

A general problem of multi-layer coding as spatial and CGS coding is that only a medium part of the base layer rate can usually be re-used for the enhancement layer coding. One of the reasons is that the quantization noise of base and enhancement layer is uncorrelated, and thus the non-coded error prediction signals are only weakly correlated. The effect could be seen in the experiment in sec. 4.4.5, in which a joint encoder control for base and enhancement layer was applied. Any modification of the coded base layer residual that results in an improved enhancement layer coding efficiency simultaneously leads to a loss in base layer coding efficiency. Nonetheless, we believe that further investigations of encoder control concepts could further improve the efficiency of scalable coding. In the literature it is sometimes argued that pyramid concepts as employed in the SVC design are unsuitable for spatial scalable coding, since the number of coded samples is increased. Following this argumentation it is claimed that 3-d wavelet codecs are more suitable, since they intrinsically provide a spatial scalable representation without increasing the number of samples to be coded. But with this argumentation other, more important points are ignored. When for instance analyzing the popular “t+2d” concept, in which a MCTF is first applied and then followed by a 2-d wavelet transform of the temporal subband pictures, the following can be observed: Since a single set of motion parameters is used for all spatial resolutions, which is usually optimized for the enhancement layer, the

trade-off between motion and texture rate for the base layer significantly differs from its optimal adjustment, and thus the coding efficiency of the base layer is significantly affected. Although it has been tried to circumvent this issue by a scalable coding of motion parameters [84], a second intrinsic problem of this approach exists. When decoding a lower spatial resolution, the motion vectors that have been obtained by MCTF with the full resolution are applied to the lower resolution signal, and thus the prediction errors do not fit to the motion parameters. The effect is that the "t+2d" coding schemes are usually characterized by a good coding efficiency at the enhancement layer, but for the base layer the coding efficiency is significantly worse than with single-layer coding.

CHAPTER 5

IMPROVED REFINEMENT COEFFICIENTS' CODING

5.1 A brief history of FGS

The concept of multi-rate coding has been well studied for some time, and was initially applied to still image coding ([86], [87]) then to video coding in [88]. Early papers recognized the relationship between scalability and time-varying channels, but focused on achieving bit rate scalability through a reduction in spatial resolution. In a precursor to modern FGS video, Taubman and Zakhor [89] specifically identified "rate-scalability" as being of interest, and proposed a system for achieving the same.

Rate scalability gained in popularity during development of MPEG-4's Streaming Video Profile [90]. Li provides a thorough overview of MPEG-4 fine-grained scalability (FGS) in [34]. The term 'granularity' indicates the precision with which rate can be controlled; thus a "fine-grained scalability" scheme permits rate to be added in small increments. While the objective was laudable, MPEG-4 FGS introduced a tremendous coding efficiency penalty. This penalty came about because the motion compensation associated with the low-rate base layer becomes less optimal as rate increases, and because correlation between the base and FGS layers (or even between bit planes within the FGS layer) was under-exploited. Although an attempt was made to address the former with the addition of Progressive FGS (P-FGS) [36], coding efficiency remained poor. To date, the demand for scalability as a whole has been small,

in part due to limited interaction between different "application spheres" of devices using digital video. This, coupled with the existence of other mechanisms that achieve a similar effect to scalability (e.g. bit stream switching) and poor compression efficiency, mean that MPEG-4 FGS has failed to achieve widespread deployment. However, as digital video becomes more pervasive and content is more readily shared by devices possessing markedly different characteristics, scalability is likely to have a "second chance" in the marketplace, provided the key defect of poor coding efficiency can be overcome.

5.2 More recent approaches to FGS

Working Draft 1 (WD1) of the H.264/AVC scalable extension [91] contains a much-updated version of FGS. There remains a coding efficiency penalty associated with FGS when compared to discrete quality layers (i.e. those containing motion information). However, the loss is smaller than that for MPEG-4 FGS. The improvement is firstly because the enhancement layer coding occurs in an "open loop" structure, where the encoder uses the original frame rather than a reconstruction in its motion model. Consequently the motion model at higher bit rates is not sub-optimal to the extent of MPEG-4 FGS. Second, there is the provision for discrete SNR layers containing additional motion information, so that a single FGS layer does not need to span a vast range of bit rates. Third, the arithmetic coder of H.264/AVC (CABAC) is used [92]. Context selection promotes coding efficiency by exploiting correlation with base layer information, or with that previously encoded in the current FGS layer. A

more detailed description of the FGS encoding process in [91] is warranted, as it forms the basis for this thesis.

Within the encoder, a base layer is encoded using a non-embedded scheme. A reconstruction of the encoded version is subtracted from the original, yielding a differential; a 4x4 or 8x8 transform is applied to each color component, and the transform coefficients are separated into subbands by frequency. Following separation into subbands, one or more bit planes are encoded for each FGS layer, where each bit plane involves categorizing coefficients and encoding each in one of three passes: (a) The "significance pass" identifies those coefficients that had reconstructed values of zero in the previous bit plane, and which had one or more neighboring coefficients with a non-zero reconstructed value in the previous bit plane. An encoded binary digit serves as a "significant coefficient flag" (SCF) indicating whether the coefficient transitions from zero to non-zero in the current bit plane. (b) The "refinement pass" identifies those coefficients that had reconstructed non-zero values in the previous bit plane. An encoded binary digit refines the precision of these coefficients in the current bit plane. (c) The "remainder pass" encodes those coefficients not already identified in the first or second passes.

5.3 Current Refinement Coefficient Coding in SVC

In the CAVLC mode in Joint Draft 6 (JD6) [85], refinement coding is done block-by-block. After all the significant coefficients within a block have been coded, the refinement coding pass of the current block starts. All of the refinement

coefficients in the current block are sent before the FGS coder moves on to the next block.

When VLC coding mode is used, coding of the refinement coefficient c_n at layer *n* is performed by encoding two flags: *coeff_ref_flag* and *coeff_ref_dir_flag*. The first flag indicates "Is the coefficient equal to 0 or not?", the second flag informs is the sign of the refinement coefficient same (*coeff_ref_dir_flag*=0) or different (*coeff_ref_dir_flag*=1) than the sign of the collocated coefficient reconstructed based on the information received in the previous FGS layers. The two refinement flags are combined into an alphabet of three refinement symbols (Table 5.1).

Table 5.1 Refinement Symbols [100]

coeff_ref_flag	*Coeff_ref_dir_flag*	*ref_symbol*
0	-	0
1	0	1
1	1	2

Three consecutive refinement symbols (*ref_symbols*) are grouped together and sent using one of predefined VLC tables (Table 5.2). First table assumes higher probability of refinement symbols equal to 1 than when refinement symbols equal to 2. The second table assumes equal probability of these two symbols. Which VLC table should be used for the currently coded macroblock, is adaptively determined based on the statistics of the previously coded macroblocks. It is done by comparing how many refinement symbols 0, 1 and 2 were received for the previous macroblocks.

Table 5.2 VLC table to code refinement symbols in SVC [95][100]

	Table 1		Table 2	
Group of ref symbol	Code length	Code word	Code length	Code word
{0,0,0}	1	1	1	1
{0,0,1}	4	0011	5	00111
{0,0,2}	5	00101	5	00110
{0,1,0}	3	011	4	0111
{0,1,1}	6	000101	7	0001001
{0,1,2}	8	00000101	7	0001000
{0,2,0}	5	00100	4	0110
{0,2,1}	7	0000101	7	0000111
{0,2,2}	9	000000101	6	001001
{1,0,0}	3	010	4	0101
{1,0,1}	6	000100	7	0000110
{1,0,2}	8	00000100	7	0000101
{1,1,0}	6	000011	6	001000
{1,1,1}	9	000000100	8	00000111
{1,1,2}	10	0000000011	8	00000110
{1,2,0}	7	0000100	6	000111
{1,2,1}	10	0000000010	8	00000101
{1,2,2}	12	000000000011	8	00000100
{2,0,0}	5	00011	4	0100
{2,0,1}	7	0000011	7	0000100
{2,0,2}	9	000000011	6	000110
{2,1,0}	8	00000011	6	000101
{2,1,1}	10	0000000001	8	00000011
{2,1,2}	12	000000000010	8	00000010
{2,2,0}	9	000000010	5	00101

Table 5.2 - continued

{2,2,1}	12	000000000001	8	00000001
{2,2,2}	12	000000000000	8	00000000

5.4 Improved Refinement Coefficient Coding in SVC

5.4.1 Proposal 1: Removing Table Adaptation

The quantizer dead-zone parameter which decides the probability of refinement symbols 1 and 2 is usually set in the encoder depending on the macroblock and frame type (I, P, or B frame), and is different for Inter and Intra macroblocks. The probability of the refinement symbols will differ between the neighboring macroblocks if they are of different types, i.e. an Intra macroblock having neighboring macroblock as Inter or vice versa. Hence currently used adaptation not only adds to the complexity of the decoder but may also lead to decreased coding efficiency when frame contains macroblocks of both types (e.g., intra macroblocks when used to increase error resilience).

We propose to use information about the macroblock type in deciding which table should be used [1], [2]. The information about which table is used for which type of macroblock is signaled to the decoder for each FGS layer. Only 2 bits for each FGS layer have to be sent (1 bit to indicate which VLC table is used to code intra macroblocks and 1 bit to indicate which VLC table is used to code inter macroblocks) (Table 5.4). Encoder can determine the appropriate table using, for example, information about its quantization or statistic gathered from previously encoded frames. Furthermore, we noticed that it is beneficial to replace the tables with the new ones –

the second table assumes higher probability of symbol 2 instead of equal probability of symbols 1 and 2. The new tables used are shown in Table 5.3

Table 5.3 Proposed VLC table to code refinement symbols in SVC

	Table 1		Table 2	
Group of ref symbol	Code length	Code word	Code length	Code word
{0,0,0}	1	0x01	1	0x01
{0,0,1}	3	0x00	5	0x09
{0,0,2}	8	0x3a	4	0x05
{0,1,0}	3	0x02	5	0x0d
{0,1,1}	5	0x05	8	0x1f
{0,1,2}	10	0xee	7	0x0e
{0,2,0}	7	0x1c	4	0x07
{0,2,1}	10	0xed	7	0x0b
{0,2,2}	11	0x124	6	0x11
{1,0,0}	3	0x03	4	0x00
{1,0,1}	6	0x0f	8	0x1e
{1,0,2}	10	0xef	7	0x0c
{1,1,0}	5	0x06	7	0x08
{1,1,1}	7	0x13	10	0x4c
{1,1,2}	11	0x126	9	0x25
{1,2,0}	9	0x4a	7	0x31
{1,2,1}	11	0x127	10	0x4d
{1,2,2}	14	0xec6	9	0x35
{2,0,0}	6	0x08	3	0x01
{2,0,1}	9	0x48	7	0x0a
{2,0,2}	12	0x3b0	6	0x10
{2,1,0}	9	0x4b	7	0x30

Table 5.3 – continued

{2,1,1}	11	0x125	9	0x24
{2,1,2}	14	0xec5	9	0x34
{2,2,0}	11	0x1d9	6	0x19
{2,2,1}	14	0xec7	9	0x27
{2,2,2}	14	0xec4	8	0x1b

Table 5.4 Signaling Table showing which VLC table to use according to MB type, where x=don't care

Frame Type	Quality Level	Temporal Level	VLC Table to be used	
			Intra	Inter
I	1	x	0	0
I	2	x	1	0
I	3	x	1	0
B	1	x	0	0
B	2	x	1	0
B	3	x	1	0
P	1	x	0	0
P	2	x	1	0
P	3	x	1	0
P	1	0	1	1
P	2	0	1	1
P	3	0	1	1

5.4.2 Proposal 2: All Zero Refinement Coefficients

In the second part of the proposal we extend the ideas, which are used to increase coding efficiency of "CABAC" refinement coefficients coding in SVC coder, to "VLC" coding [97], [1], [2].

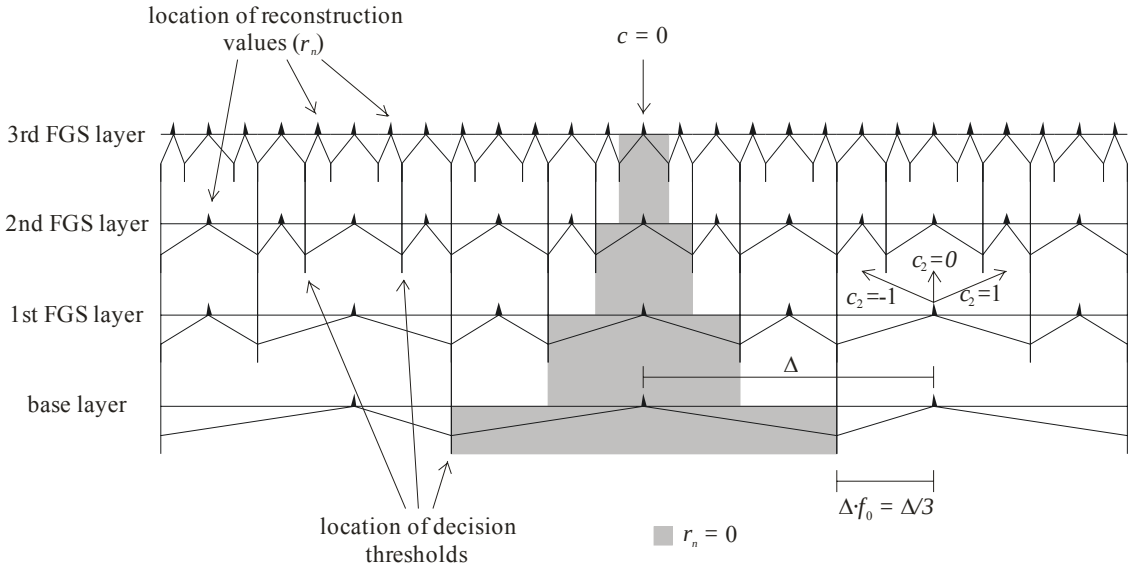


Figure 5.1 Reconstruction values and decision thresholds for fn = 1/3.

In Figure 5.1, the location of the reconstruction values for the different layers are shown as small solid triangles on the horizontal lines, where each such line indicates a different layer starting from the base layer at the bottom up to the third FGS layer on top of the graph. Also shown in Figure 5.1 are the decision thresholds corresponding to the quantization rule with a fixed choice of f_n = 1/3.

A closer inspection of the relative location of decision thresholds involved in the quantization of two consecutive FGS layers reveals that there are two types of decision intervals. As can be seen from Figure 5.1, there is one type of decision interval whose interval width keeps constant and another type of decision interval that gets subdivided into three intervals. However, from a coding point-of-view, there is currently no distinction between signaling the refinement symbols of both types. This leads to a substantial loss in coding performance in the second and third FGS layer.

In fact, for the special case of a fixed choice of f_n = 1/3, it would be possible to locate those intervals that keep constant at the decoder side and avoid signaling any refinement information for the corresponding reconstruction levels. Figure 5.1 shows that the two types of intervals are alternating and therefore, it is fairly easy to derive a rule specifying for which reconstruction values of the previous FGS layer, refinement information is necessary.

To each refinement coefficient in n-th FGS layer index h is assigned based on the refinement values of the collocated coefficients c_i in layers $i<n$, where n is the total number of FGS layer. Index h is calculated as follows:

```
h=0;
for (i=0; i<n; i++) {
    sig=(ci != 0)? 1 : 0;
    h=h+sig*(1<<i);
}
```

For some choices of a quantizer's dead-zone parameter, refinement coefficients having certain values of index *h* are most likely to be zero. E.g., for dead-zone parameter f_n = 1/3 (Figure 5.1), usually used for Intra macroblocks, refinement coefficients at FGS layer *n=2* for which value of index *h* is equal to 2 or 3 will be equal to zero with probability higher than 99%. We will call such refinement coefficients – "type-0" refinement coefficients.

For each FGS layer the values of index *h* which identify type-0 refinement coefficients are signaled to the decoder together with information macroblocks of which type (Inter or Intra) contain "type-0" refinement coefficients. Figure 5.2 shows the way to calculate the index *h*.

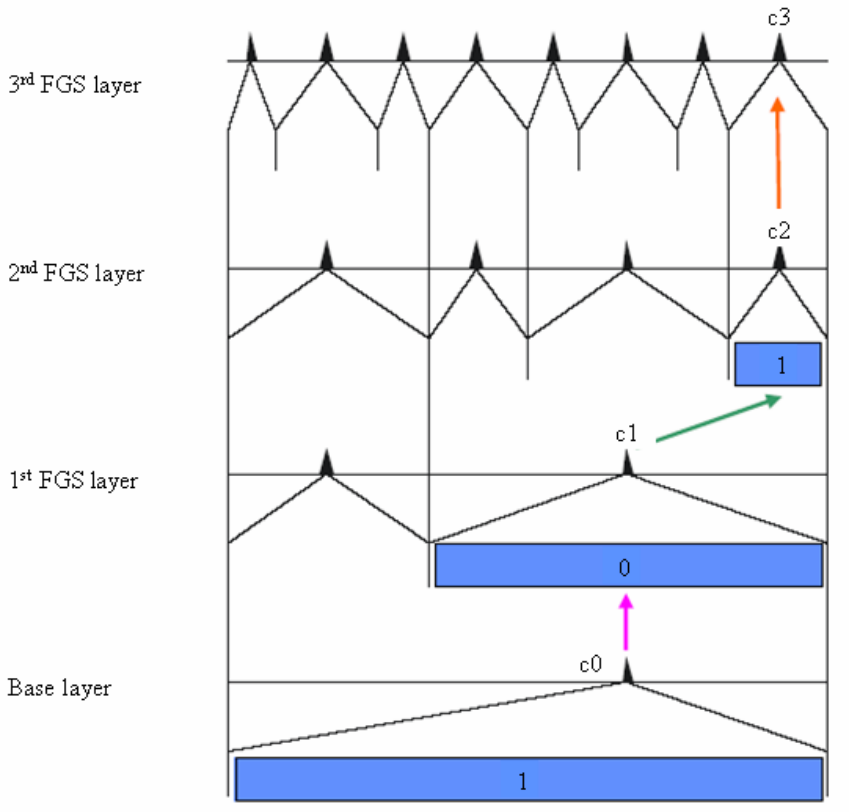


Figure 5.2 Figure showing how to generate History Map for "Type-0 coeff"

The refinement coefficient at quality level 3 uses the information from collocated coefficients at quality level 2, quality level 1 and base layer. So index *h* for refinement coefficients at quality level 3 can have values from 0 to 7. Similarly, the refinement coefficient at quality level 2 uses the information from collocated coefficients at quality level 1 and base layer. So index *h* for refinement coefficients at quality level 2 can have values of 0, 1, 2 or 3. As seen in Figure 5.2, refinement coefficient c3 is "type-0" refinement coefficients. Refinement coefficient at FGS layer 3, c3, is derived from its collocated refinement coefficient at FGS layer 2, c2, and the value of c2 is 1, and c2 is derived from c1, and the value of c1 is 0 and c1 is derived from c0 and the value of c0 is 1. Thus the index *h* for refinement coefficient c3 will be decimal of (c2, c1, c0). Index *h* = (101)b = 5. So in the history map of quality layer 3, we set bit5 to 1.

- For Quality Layer = 2, History Map = [0,0,1,1]
- For Quality Layer = 3, History Map = [0,0,0,0,1,1,0,0]

Furthermore, for each macroblock containing type-0 refinement coefficients, flag *coefRefMBType* is sent to the decoder (flag *coefRefMBType* is sent separately for luminance and chrominance):

- If the *coefRefMBType*=1 all type-0 refinement coefficients in this macroblock are set to zero and no further information needs to be transmitted for them. Remaining refinement coefficients for this macroblock are coded by sending 1

bit for *coeff_ref_flag* and, if *coeff_ref_flag* indicated that coefficient is non-zero, 1 bit for *coeff_ref_dir_flag*.

- Refinement coefficients for macroblocks for which *coefRefMBType*=0 are coded as currently.

Table 5.5 Signaling Table for proposal 2, where x=don't care and N/A=not applicable

Frame Type	Quality Level	Temporal Level	CodeType used		History Map
			Intra	Inter	
I	1	x	0	0	N/A
I	2	x	1	0	011
I	3	x	1	0	0001100
B	1	x	0	0	N/A
B	2	x	1	0	011
B	3	x	1	0	0001100
P	1	x	0	0	N/A
P	2	x	1	0	011
P	3	x	1	0	0001100
P	1	0	0	0	N/A
P	2	0	1	1	011
P	3	0	1	1	0001100

CHAPTER 6

RESULTS AND CONCLUSIONS

MPEG-4 FGS introduced a tremendous coding efficiency penalty and thus did not create a market for itself. H.264 FGS is way better than MPEG-4 FGS in every aspect, one of them being coding efficiency. The main aim of this thesis is to further improve the coding efficiency of H.264 SVC FGS and we are able to do that successfully putting forward the results in sec 6.2.

6.1 Test Conditions

The test sequences [94] include bus, city, crew, football, foreman, harbour, mobile and soccer with CIF@30Hz. Each sequence was tested at 4 different base layer Quantization Parameters (QP), i.e. QP = 29, 30, 35 and 36. Each of these CIF sequences (Figure 6.1 & Table 6.1) has 300 frames and was encoded-decoded fully for 3 FGS layers. Test sequences for 4CIF@60Hz include city, crew, harbour and soccer. Each of these 4CIF (Figure 6.1 & Table 6.1) sequences has 600 frames and was encoded-decoded fully for 3 FGS layers. Each sequence was partially decoded at 10 different quality levels as in 0.0, 0.3, 0.6, 1.0, 1.3, 1.6, 2.0, 2.3, 2.6 and 3.0, where for e.g. 1.3 means decoded video contains 100% quality level 1 and 30% of quality level 2. The test was performed on Intra only, Inter and AR-FGS schemes.

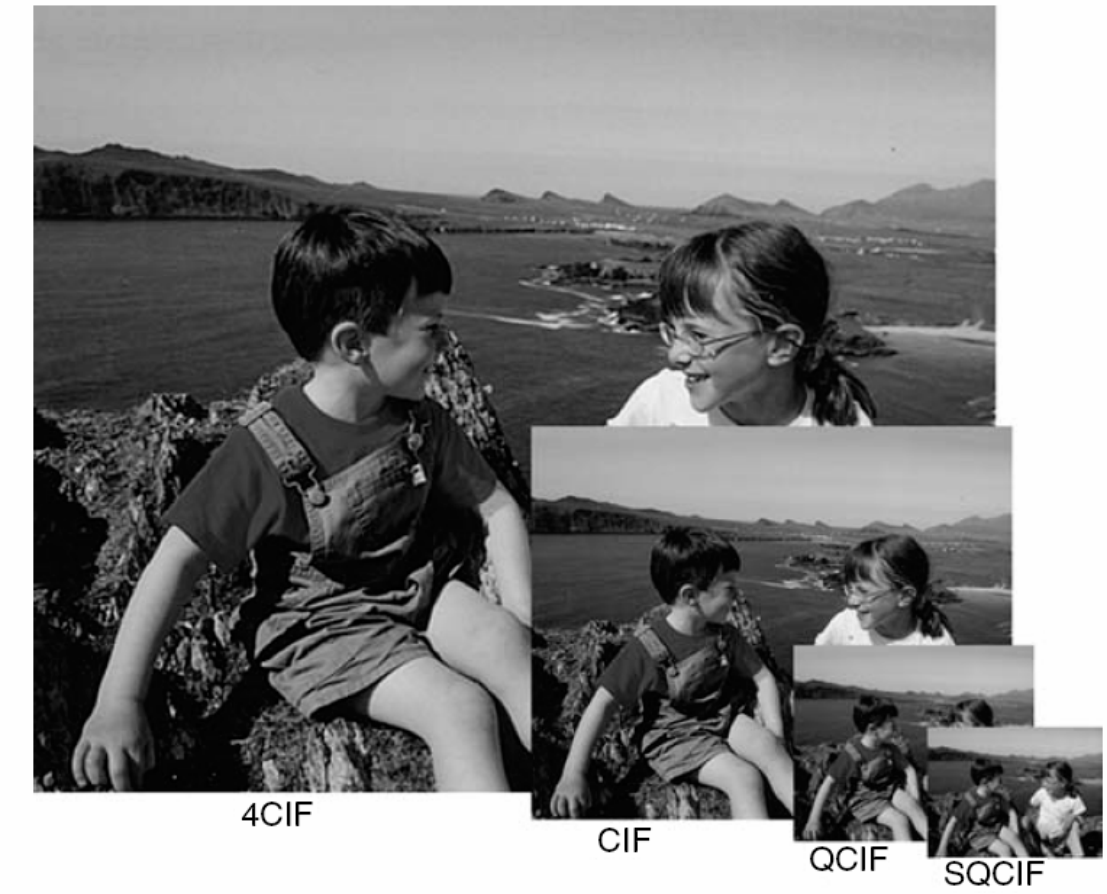


Figure 6.1 Video frame sampled at range of resolutions

Table 6.1 Video Frame Formats for 4:2:0 case

Format	Luminance Resolution (Horiz. x Verti.)	Chrominance Resolution (Horiz. x Verti.)
SQCIF	128 x 96	64 x 48
QCIF	176 x 144	128 x 96
CIF	352 x 288	176 x 144
4CIF	704 x 576	352 x 288

6.2 Results

To evaluate the performance of our proposed method, the proposed scheme was integrated into H.264 SVC reference software JSVM_7.10 [95]. When the proposed method is used, in the encoder, for macroblocks containing type-0 refinement coefficients, addition optimization is performed. If there are less than 4 refinement coefficients of type-0 in the macroblock all of them are forced to 0. Hence there is PSNR difference between results obtained using the proposed method and JSVM_7.10. However it is usually with in the range of 0.01-0.02 dB and for simplicity we neglect it when calculating bit-rate reduction.

The approximate percentage of the coding performance gain is derived from overall bit rates:

$$CodingGain = \frac{(BR_{ref} - BR_{proposed})}{BR_{ref}} * 100$$, where

BRproposed = overall bit rate (base + 3 FGS layers): proposed method

BRref = overall bit rate (base + 3 FGS layers): JSVM_7.10

The results for CIF Intra only coding are show in Tables 6.2 and 6.3. Table 6.4 shows CIF Inter coding results and Table 6.5 shows results for CIF AR-FGS coding scheme. Table 6.6, 6.7, 6.8 and 6.9 shows results for 4CIF Intra only with QP=29 and 30, 4CIF Intra only with QP=35 and 36, 4CIF Inter and 4CIF AR-FGS respectively. Figure 6.2 to 6.9 expresses the same results shown in Tables 6.2 as graphs. Detailed version of all these results can be found in [1] and [2].

Table 6.2 Results for CIF Intra Only for QP=29 and 30

Intra Only QP=29,30	Original (JSVM_7.10)		Proposed		Bitrate reduction (%)	ΔPSNR Y (Proposed-Original)
	Bitrate (kbps)	PSNR Y (dB)	Bitrate (kbps)	PSNR Y (dB)		
bus	15594.48	50.2044	14480.88	50.1708	**7.14%**	-0.0336
city	13892.16	49.3126	12926.4	49.279	**6.95%**	-0.0336
crew	9493.44	50.4573	8938.08	50.4218	**5.85%**	-0.0355
football	16210.56	50.1486	15045.12	50.1066	**7.19%**	-0.042
foreman	11265.36	50.3174	10561.68	50.2753	**6.25%**	-0.0421
harbour	16082.64	50.1452	14921.28	50.0936	**7.22%**	-0.0516
mobile	19937.04	50.2651	18399.36	50.2118	**7.71%**	-0.0533
soccer	13620.48	49.3005	12659.52	49.2726	**7.06%**	-0.0279

Table 6.3 Results for CIF Intra Only for QP=35 and 36

Intra Only QP=35,36	Original (JSVM_7.10)		Proposed		Bitrate reduction (%)	ΔPSNR Y (Proposed-Original)
	Bitrate (kbps)	PSNR Y (dB)	Bitrate (kbps)	PSNR Y (dB)		
bus	11159.52	44.7116	10405.92	44.7038	**6.75%**	-0.0078
city	9727.68	44.0118	9107.76	44.0031	**6.37%**	-0.0087
crew	5808.24	45.9139	5515.68	45.8979	**5.04%**	-0.016
football	11542.8	44.8843	10771.44	44.8544	**6.68%**	-0.0299
foreman	7233.6	45.2958	6844.32	45.2742	**5.38%**	-0.0216
harbour	11565.6	44.7529	10764.72	44.7258	**6.92%**	-0.0271
mobile	15188.16	45.0282	14067.36	44.9926	**7.38%**	-0.0356
soccer	9408.72	44.0007	8826.72	43.9938	**6.19%**	-0.0069

Table 6.4 Results for CIF Inter for QP=36

CIF Inter QP=36	Original (JSVM_7.10)		Proposed		Bitrate reduction (%)	ΔPSNR Y (Proposed-Original)
	Bitrate (kbps)	PSNR Y (dB)	Bitrate (kbps)	PSNR Y (dB)		
bus	2571.14	42.5928	2522.19	42.5874	**1.90%**	-0.0054
city	1492.27	42.8530	1463.76	42.8449	**1.91%**	-0.0081
crew	2311.09	43.6315	2250.81	43.6162	**2.61%**	-0.0153
football	2680.08	43.1606	2608.15	43.1535	**2.68%**	-0.0071
foreman	1599.44	43.0771	1570.94	43.0684	**1.78%**	-0.0087
harbour	2908.35	42.4041	2865.00	42.4067	**1.49%**	0.0026
mobile	2930.54	42.5117	2878.83	42.5017	**1.76%**	-0.01
soccer	1783.95	43.1676	1742.64	43.1566	**2.32%**	-0.011

Table 6.5 Results for CIF AR-FGS for QP=36

AR-FGS QP=36	Original (JSVM_7.10)		Proposed		Bitrate reduction (%)	ΔPSNR Y (Proposed-Original)
	Bitrate (kbps)	PSNR Y (dB)	Bitrate (kbps)	PSNR Y (dB)		
bus	4462.36	43.654	4204.40	43.5451	**5.78%**	-0.1089
city	3989.81	43.5091	3776.05	43.3744	**5.36%**	-0.1347
crew	3089.61	44.6658	2949.56	44.581	**4.53%**	-0.0848
football	3634.15	44.4228	3445.21	44.3534	**5.20%**	-0.0694
foreman	3341.51	44.0064	3192.78	43.9215	**4.45%**	-0.0849
harbour	4767.00	43.5404	4490.34	43.4071	**5.80%**	-0.1333
mobile	6239.90	43.4652	5878.95	43.3471	**5.78%**	-0.1181
soccer	3288.93	44.1392	3128.70	44.056	**4.87%**	-0.0832

Table 6.6 Results for 4CIF Intra Only for QP=29 and 30

Intra Only QP=29,30	Original (JSVM_7.10)		Proposed		Bitrate reduction (%)	ΔPSNR Y (Proposed-Original)
	Bitrate (kbps)	PSNR Y (dB)	Bitrate (kbps)	PSNR Y (dB)		
city	98297.28	49.3625	91766.4	49.3430	**6.64%**	-0.0195
crew	73662.24	50.5198	69577.92	50.4748	**5.54%**	-0.045
harbour	102059.52	50.1891	95355.84	50.1511	**6.57%**	-0.038
soccer	96046.08	49.5007	89544	49.4894	**6.77%**	-0.0113

Table 6.7 Results for 4CIF Intra Only for QP=35 and 36

Intra Only QP=35,36	Original (JSVM_7.10)		Proposed		Bitrate reduction (%)	ΔPSNR Y (Proposed-Original)
	Bitrate (kbps)	PSNR Y (dB)	Bitrate (kbps)	PSNR Y (dB)		
city	66726.72	44.3328	62744.16	44.3241	**5.97%**	-0.0087
crew	42138.72	45.7412	40370.4	45.7249	**4.20%**	-0.0163
harbour	67664.64	45.1237	63629.28	45.1029	**5.96%**	-0.0208
soccer	65788.8	44.4922	61750.56	44.483	**6.14%**	-0.0092

Table 6.8 Results for 4CIF Inter for QP=36

4CIF Inter QP=36	Original (JSVM_7.10)		Proposed		Bitrate reduction (%)	ΔPSNR Y (Proposed-Original)
	Bitrate (kbps)	PSNR Y (dB)	Bitrate (kbps)	PSNR Y (dB)		
city	28286.91	42.7473	27948.42	42.7406	**1.20%**	-0.0067
crew	30313.94	43.5211	29688.02	43.5007	**2.06%**	-0.0204
harbour	32545.71	42.7543	32084.71	42.7499	**1.42%**	-0.0044
soccer	23498.06	43.4313	23096.12	43.4200	**1.71%**	-0.0113

Table 6.9 Results for 4CIF AR-FGS for QP=36

AR-FGS QP=36	Original (JSVM_7.10)		Proposed		Bitrate reduction (%)	ΔPSNR Y (Proposed-Original)
	Bitrate (kbps)	PSNR Y (dB)	Bitrate (kbps)	PSNR Y (dB)		
bus	53069.35	43.8756	50470.38	43.713	**4.90%**	-0.1626
city	39022.25	44.7164	37678.03	44.6412	**3.44%**	-0.0752
crew	53930.87	43.8246	51244.15	43.678	**4.98%**	-0.1466
football	42379.12	44.6311	40481.26	44.5148	**4.48%**	-0.1163

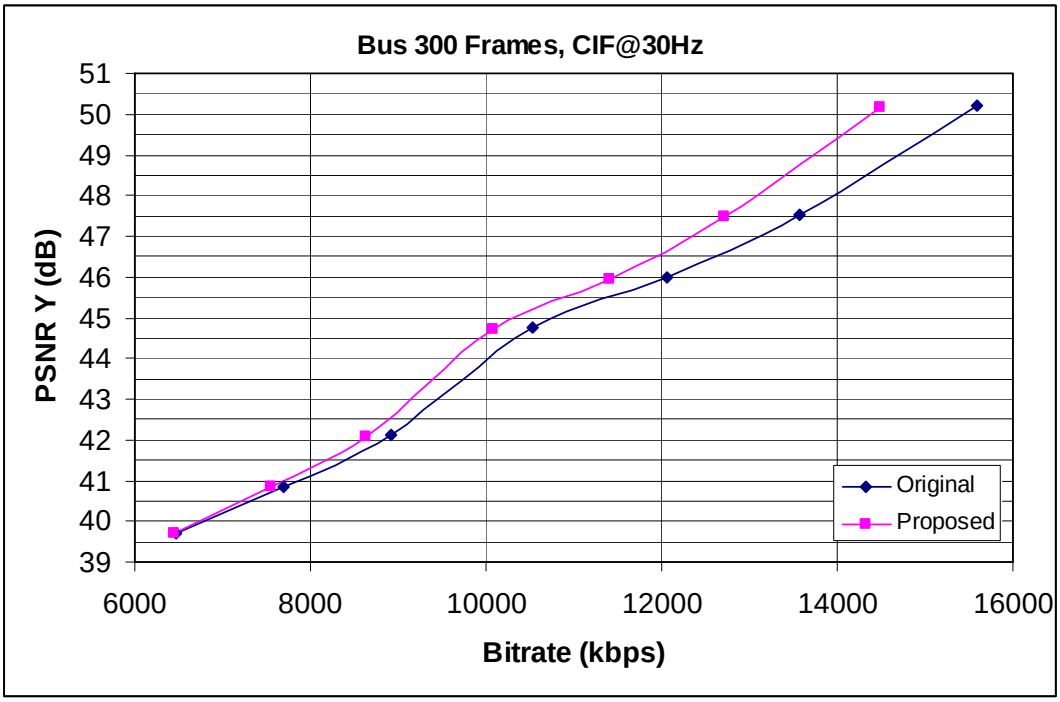


Figure 6.2 Sequence Bus, base layer QP 29, Intra only, 3 FGS-Layers (with 2nd and 3rd FGS layer shown)

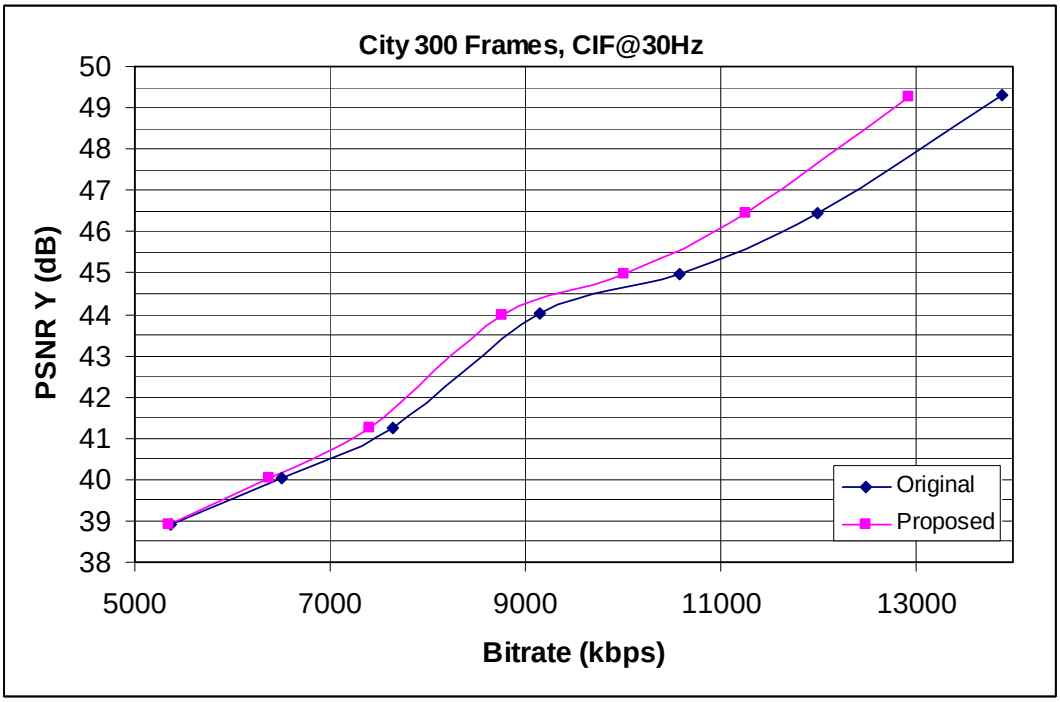


Figure 6.3 Sequence City, base layer QP 30, Intra only, 3 FGS-Layers (with 2nd and 3rd FGS layer shown)

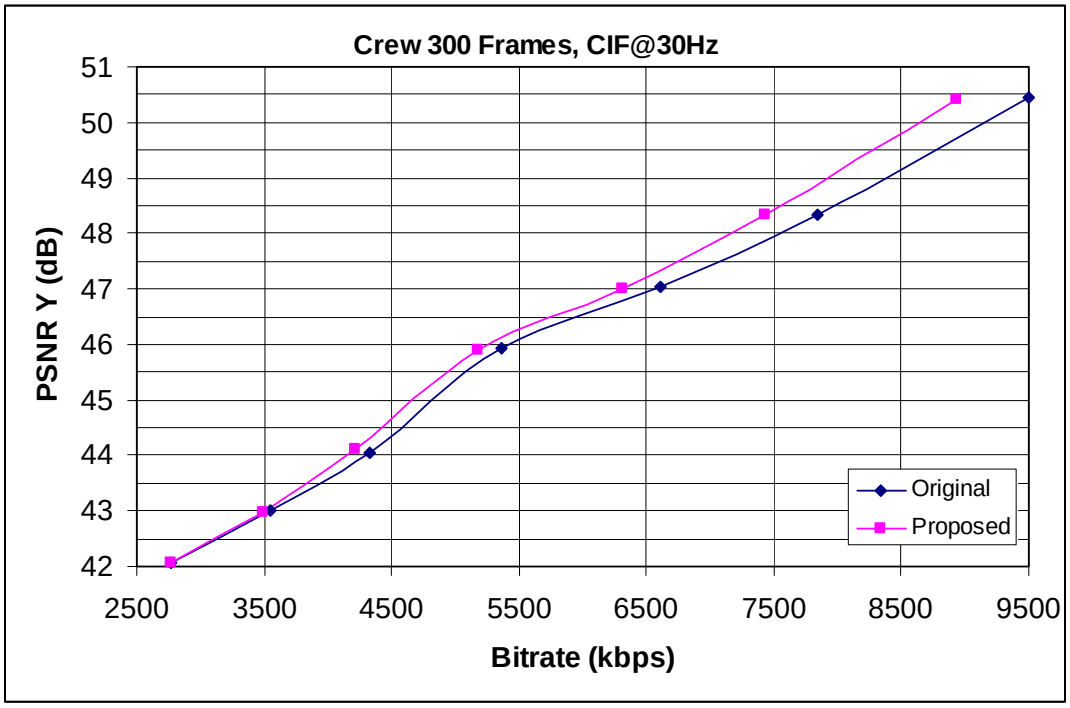


Figure 6.4 Sequence Crew, base layer QP 29, Intra only, 3 FGS-Layers (with 2nd and 3rd FGS layer shown)

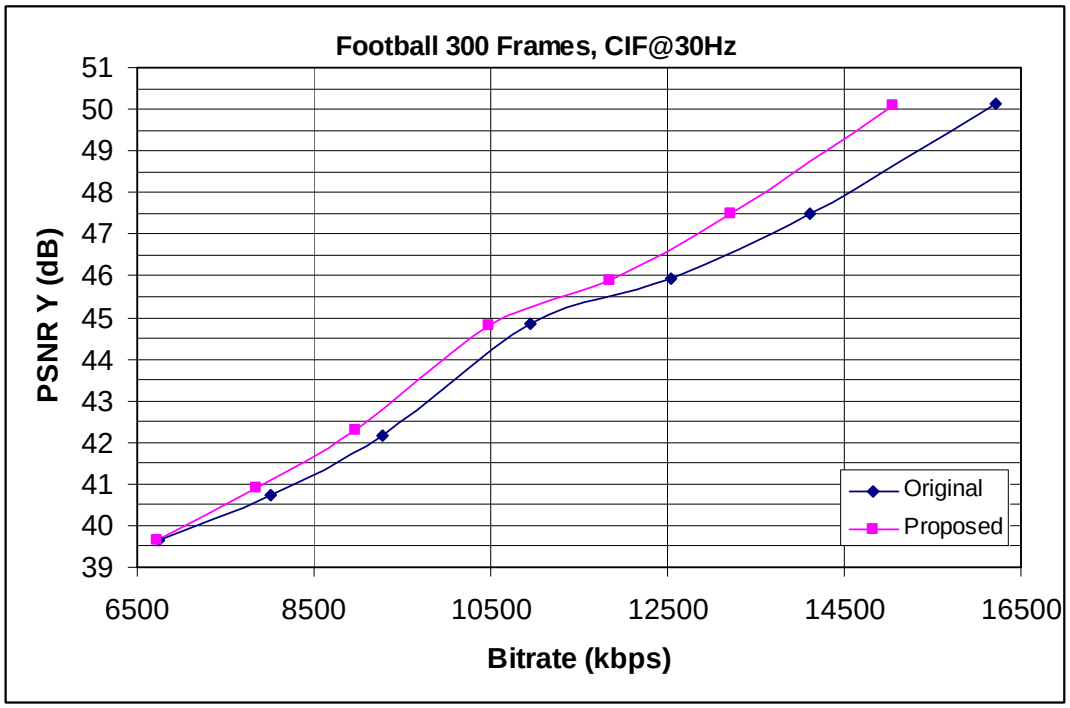


Figure 6.5 Sequence Football, base layer QP 29, Intra only, 3 FGS-Layers (with 2nd and 3rd FGS layer shown)

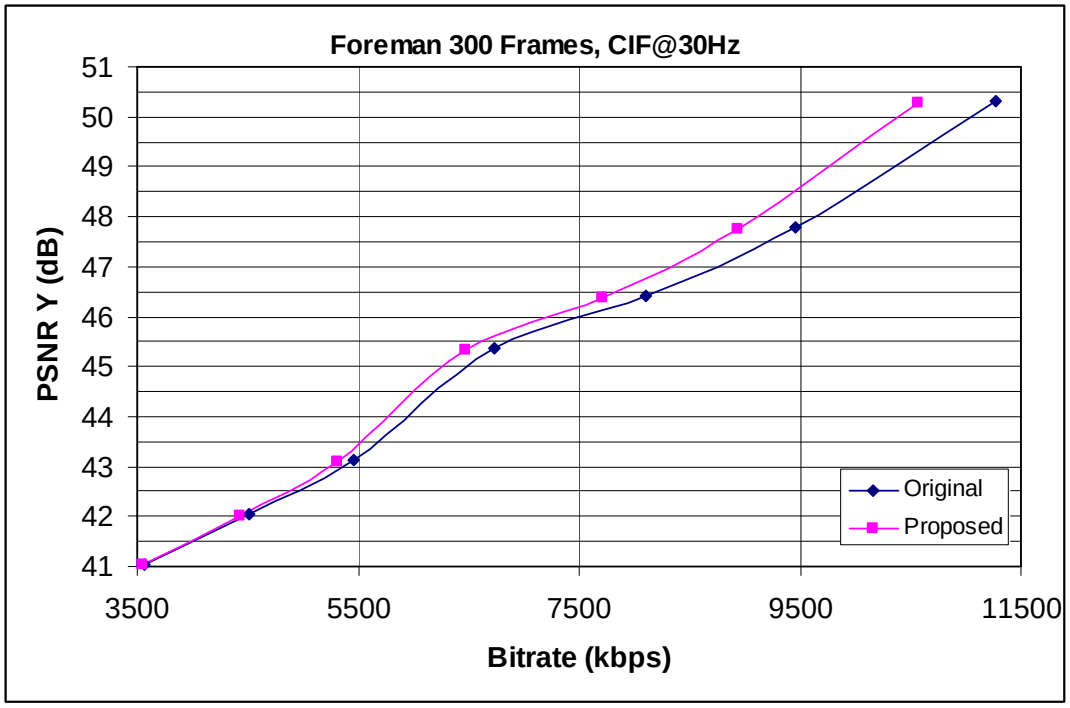


Figure 6.6 Sequence Foreman, base layer QP 29, Intra only, 3 FGS-Layers (with 2nd and 3rd FGS layer shown)

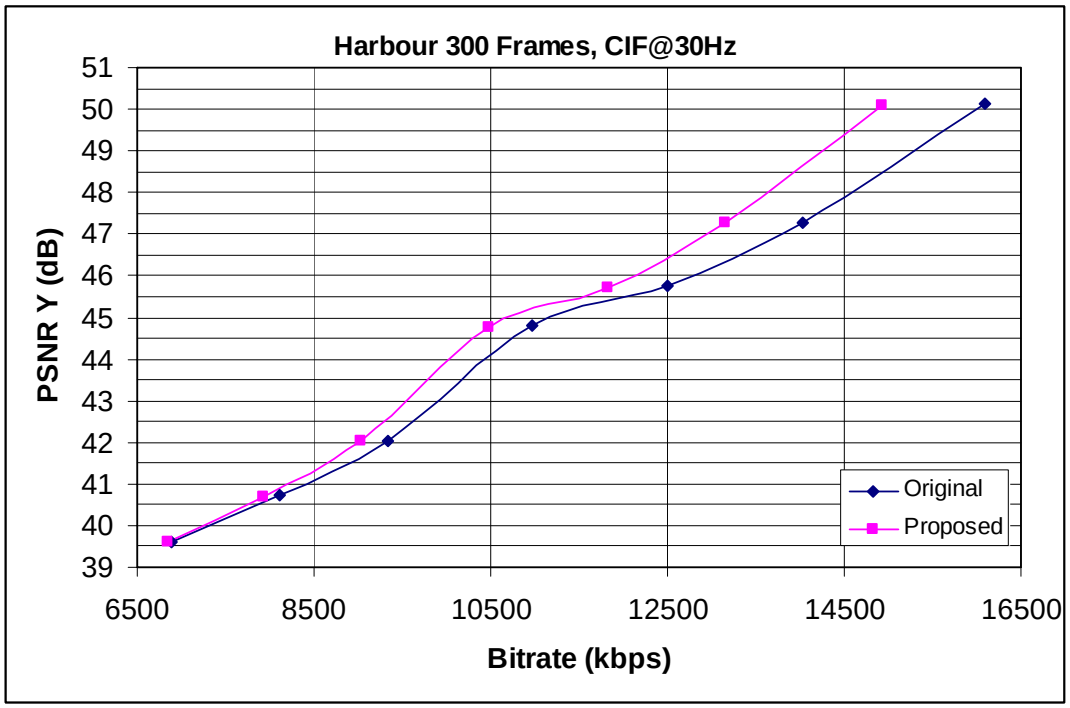


Figure 6.7 Sequence Harbour, base layer QP 29, Intra only, 3 FGS-Layers (with 2nd and 3rd FGS layer shown)

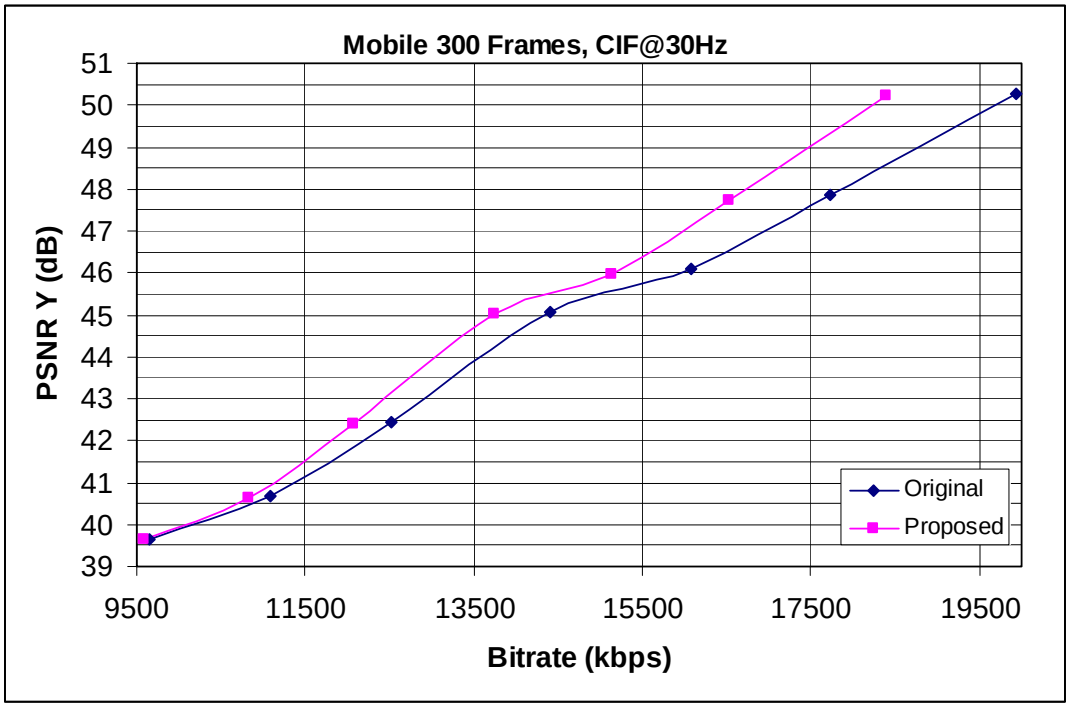


Figure 6.8 Sequence Mobile, base layer QP 29, Intra only, 3 FGS-Layers (with 2nd and 3rd FGS layer shown)

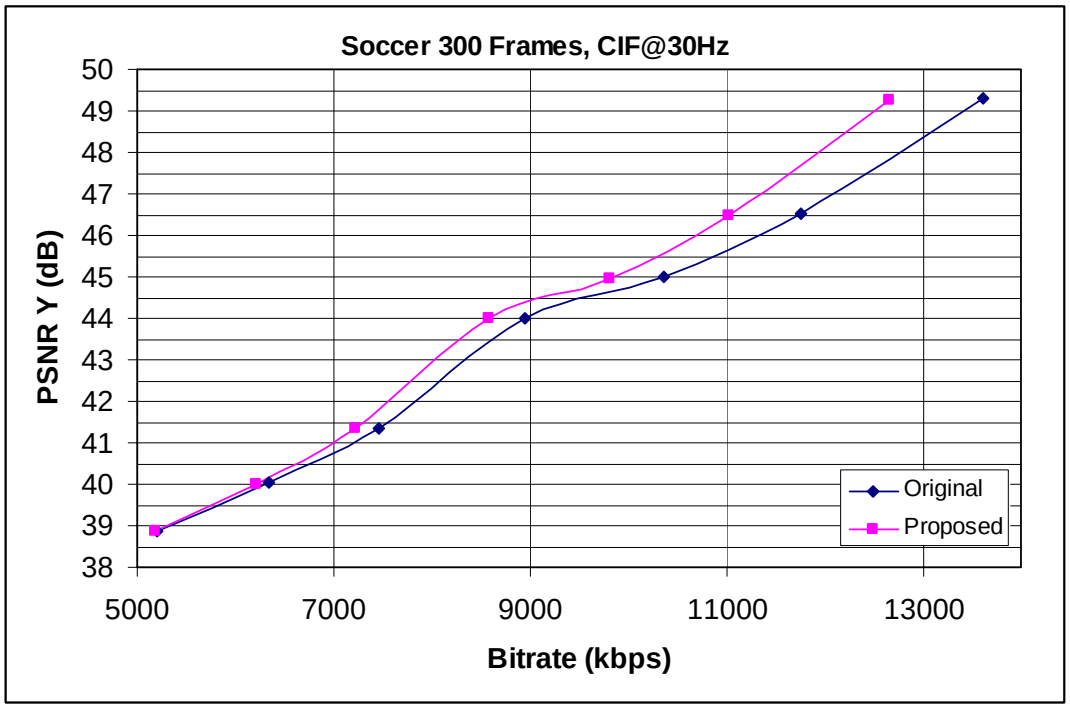


Figure 6.9 Sequence Soccer, base layer QP 30, Intra only, 3 FGS-Layers (with 2nd and 3rd FGS layer shown)

6.3 Conclusions

Context modeling for refinement coding in the current draft does not properly reflect the typical statistics for different possible choices of dead-zone parameters beyond the first FGS layer. With the improved CAVLC context modeling proposed scheme, an overall improvement in coding efficiency for coding of refinement information can be obtained for the second and third FGS layer. A new CAVLC context modeling scheme for coding of refinement information in PR slices has been introduced, showing a significant improvement in coding performance. The overall improvements are from 3 to 7%. Majority of the improvement is coming from proposed type-0 refinement coefficients coding.

6.4 Future Work

Future work involves developing and implementing algorithms to improve Fine Granularity Scalability within the Scalable Video Coder test model developed by JVT standardization forum.

- Combine proposed method (macroblock type based adaptation and type-refinement coefficients coding, JVT-V095) with fragments based decoding (JVT-V116).
- Combine proposed method with the method proposed in JVT-V077, i.e. joint significant and refinement coefficients coding.
- Test different methods of sign signaling for joint significant and refinement coefficients coding when used in conjunction with fragments based decoding.

- Investigate necessary extensions to SVC standard required to code high fidelity material in scalable manner, e.g. encode 10 bit material as 8 bit baseline sequence and 2 bit enhancement one. Similarly different chroma sampling patterns should be supported (4:2:0 as a baseline for 4:2:2).

APPENDIX A

HOW TO DOWNLOAD JSVM SOFTWARE

(1) Download TortoiseCVS from http://www.tortoisecvs.org/

(2) Install TortoiseCVS

(3) Right click on desktop and select CVS Checkout

(4) Copy following line in CVSROOT

:pserver:jvtuser@garcon.ient.rwth-aachen.de:/cvs/jvt

(5) Type "jsvm" in Module

(6) Click Fetch list.

(7) When ask for password, type "jvt.Amd.2"

And you get the latest JSVM version on your desktop...ENJOY... ☺

REFERENCES

[1] M. Karczewicz and R. Panchal, “Improved Refinement Coefficient Coding”, Joint Video Team, Doc. JVT-U132, Hangzhou, China, Oct 2006.

[2] M. Karczewicz and R. Panchal, “Report of CE1 on improved refinement coefficients coding”, Joint Video Team, Doc. JVT-V095, Marrakech, Morocco, Jan 2007.

[3] I.E.G Richardson, “H.264 and MPEG-4 Video Coding for Next Generation Multimedia”, John Wiley & Sons, 2003

[4] I.E.G Richardson, “Video Codec Design: developing image and video compression systems”, Chichester: John Wiley & Sons, 2002.

[5] K.R.Rao and J.J.Hwang, “Techniques and standards for Image, Video and Audio Coding”, Prentice Hall, 1996

[6] N. Jayant, J. Johnson, and R. Safranek, “Signal compression based on models of human perception”, Proc. IEEE, Vol. 81, pp. 1385-1422, Oct. 1993.

[7] M. Ghanbari, “Standard Codecs: Image Compression to Advanced Video Coding”, London, U.K.: Institution of Electrical Engineers, 2003, pp 187-194

[8] M. Ghanbari, “Video Coding: An introduction to standard codecs”, London, U.K.: Institution of Electrical Engineers, 1999

[9] http://en.wikipedia.org/wiki/Video_compression

[10] www.vcodex.com

[11] ISO / IEC JTC1/SC29, Generic coding of moving pictures and associated audio, ISO/IEC 13818-2, Draft International Standard, Nov. 1994

[12] http://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx

[13] http://www.realnetworks.com/solutions/leadership/realvideo.html

[14] T. Weigand, et al, “Overview of the H.264/AVC video coding standard”, IEEE Trans. CSVT, Vol.13, pp. 560-576, July 2003.

[15] M. Karczewicz and R. Kurceren, “The SP- and SI-Frames design for H.264/AVC”, IEEE Trans. CSVT, Vol.13, pp. 637-644, July 2003

[16] Soon-kak Kwon, A. Tamhankar and K.R. Rao, “Overview of H.264 / MPEG-4 Part 10 (pp.186-216)”, Special issue on “ Emerging H.264/AVC video coding standard”, J. Visual Communication and Image Representation, Vol. 17, pp.183-552, April 2006

[17] http://www.stanford.edu/class/ee398b/handouts/05 standardsH264JVT.pdf

[18] A. Puri, H. Chen and A. Luthra, "Video Coding using the H.264/MPEG-4 AVC compression standard", Signal Processing: Image Communication Vol. 19, pp.793-849, Oct. 2004

[19] M. Wein, “Variable block-size transforms for H.264/AVC”, IEEE Trans. CSVT,

Vol. 13, pp. 604-613, July 2003
[20] J. Ostermann, et al, "Video coding with H.264/AVC: Tools, performance and complexity", IEEE CAS Magazine, Vol.4, pp.7-34, I quarter, 2004
[21] Joint Video Team of ITU-T and ISO/IEC: "Draft text of H.264/AVC Fidelity Range Extensions Amendment", Doc.JVT-L047, Sept. 2004
[22] K. R. Rao and P. Yip, Discrete Cosine Transform, Academic Press, 1990.
[23] H. S. Malvar, et al, "Low-complexity transform and quantization in H.264/AVC", IEEE Trans. CSVT, Vol. 13, pp. 598-603 July 2003
[24] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Extension of the H.264/MPEG-4 AVC Video Coding Standard" Joint Video Team, doc. JVT-U145-L, Hangzhou, China, Oct 2006.
[25] ITU-T, "Video codec for audiovisual Services at 64 kbit/s," ITU-T Recommendation H.261, Version 1: Nov. 1990, Version 2: Mar. 1993.
[26] ISO/IEC JTC 1, "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 2: Video," ISO/IEC 11 172-2 (MPEG-1 Video), Mar. 1993.
[27] ITU-T and ISO/IEC JTC 1, "Generic coding of moving pictures and associated audio information – Part 2: Video," ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2 Video), Nov. 1994.
[28] ITU-T, "Video coding for low bit rate communication," ITU-T Recommendation H.263, Version 1: Nov. 1995, Version 2: Jan. 1998, Version 3: Nov. 2000.
[29] ISO/IEC JTC 1, "Coding of audio-visual objects – Part 2: Visual," ISO/IEC 14492-2 (MPEG-4 Visual), Version 1: Apr. 1999, Version 2: Feb. 2000, Version 3: May. 2004.
[30] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG4-AVC), Version 1: May 2003, Version 2: Jan. 2004, Version 3: Sep. 2004, Version 4: July 2005.
[31] A. Eleftheriadis, O. Shapiro, and T. Wiegand, "Video conferencing using SVC," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, pp , Apr. 2007.
[32] T. Schierl and T. Wiegand, "Mobile video transmission using SVC," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, pp , Apr. 2007.
[33] R. Cazoulat, A. Graffunder, A. Hutter, and M. Wien, "Real-time system for adaptive video streaming based on SVC," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, pp , Apr. 2007.
[34] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, no. 3, pp. 301-317, Mar. 2001.
[35] M. van der Schaar, "A hybrid temporal-SNR fine-granular scalability for internet video," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, no. 3, pp. 318-331, Mar. 2001.

[36] F. Wu, S. Li, and Y.-Q. Zhang, "A framework for efficient progressive fine granular scalable video coding," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, no. 3, pp. 332-344, Mar. 2001.

[37] X. Sun, F. Wu, S. Li, W. Gao, and Y.-Q. Zhang, "Macroblock-based progressive fine granular scalable (PFGS) video coding with flexible temporal-SNR scalabilities," IEEE, Proceedings of ICIP'01, Vol. 2, pp. 1025-1028, Thessaloniki, Greece, Oct. 2001.

[38] M. van der Schaar and H. Radha, "Adaptive motion-compensation fine granular-scalability (AMC-FGS) for wireless video," IEEE Transactions on Circuit and Systems for Video Technology, Vol. 12, no. 6, pp. 360- 371, June 2002.

[39] M. Ghanbari and V. Seferidis, "Efficient H.261-based two layer video codecs for ATM networks," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 5, no. 2, pp. 171-175, Apr. 1995.

[40] S. Han and B. Girod, "Robust and efficient scalable video coding with leaky prediction," IEEE, Proceedings of ICIP'02, Rochester, NY, USA, Vol. 2, pp. 41-44, Sep. 2002.

[41] H.-C. Huang, "A robust fine granularity scalability using trellis-based predictive leak," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, no. 6, June 2002.

[42] Y. Liu, P. Salama, Z. Li, and E. J. Delp, "An enhancement of leaky prediction layered video coding", IEEE Transaction on Circuits and Systems for Video Technology, Vol. 15, no. 11, pp. 1317-1331, Nov.2005.

[43] Y. Gao and L.P. Chau, "Efficient fine granularity scalability using adaptive leaky factor," IEEE Transactions on Broadcasting, Vol. 51, no. 4, pp. 512-519, Dec. 2005.

[44] G. Karlsson and M. Vetterli, "Three dimensional subband coding of video," Proceedings of ICASSP'88, Vol. 2, pp. 1100-1103, New York City, USA, Apr. 1988.

[45] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," IEEE Transactions on Image Processing, Vol. 3, no. 5, pp. 559-571, Sep. 1994.

[46] S.-J. Choi and J. W. Woods, "Motion-compensated 3-d subband coding of video," IEEE Transactions on Image Processing, Vol. 8, no. 2, pp. 155-167, Feb. 1999.

[47] E. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion-compensated video compression," Proceedings of ICASSP'01, pp. 1793-1796, Salt Lake City, UT, USA, May 2001.

[48] L. Luo, J. Li, S. Li, Z. Zhuang, and Y.-Q. Zhang, "Motion compensated lifting wavelet and its application in video coding," IEEE, Proceedings of ICME'01, pp. 365-368, Tokyo, Japan, Aug. 2001.

[49] A. Secker and D. Taubman, "Motion-compensated highly scalable video compression using an adaptive 3d wavelet transform based on lifting," Proceedings of ICIP'01, Vol. 2, pp. 1029-1032, Thessaloniki, Greece,Oct. 2001.

[50] MPEG video sub-group chair, "Registered responses to the call for proposals on scalable video coding," ISO/IEC JTC 1/SC29/WG11, doc. M10569, Munich, Germany, Mar. 2004.
[51] H. Schwarz, T. Hinz, H. Kirchhoffer, D. Marpe, and T. Wiegand, "Technical description of the HHI proposal for SVC CE1," ISO/IEC JTC 1/SC29/WG11, doc. M11244, Palma de Mallorca, Spain, Oct.2004.
[52] J. Reichel, M. Wien, and H. Schwarz, eds., "Scalable Video Model 3.0," ISO/IEC JTC 1/SC29/WG11, doc. N6716, Palma de Mallorca, Spain, Oct. 2004.
[53] T. Wiegand, G. J. Sullivan, J. Reichel, H. Schwarz, and M. Wien, eds.,"Joint Draft 7," Joint Video Team, Doc. JVT-T201, Klagenfurt, Austria, July 2006.
[54] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 560-576, July 2003.
[55] G. J. Sullivan and T. Wiegand, "Video compression – from concepts to the H.264/AVC standard," Proceedings of IEEE, Vol. 93, no. 1, pp. 18-31, Jan. 2005.
[56] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 688-703, July 2003.
[57] H. Schwarz, D. Marpe, and T. Wiegand, "Hierarchical B pictures," Joint Video Team, doc. JVT-P014, Poznan, Poland, July 2005.
[58] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," Proceedings of ICME'06, Toronto, Canada, July 2006.
[59] J.-R. Ohm, "Complexity and delay analysis of MCTF interframe wavelet structures," ISO/IEC JTC 1/WG11, doc. M8520, Klagenfurt, Austria, July 2002.
[60] H. Schwarz, J. Shen, D. Marpe, and T. Wiegand, "Technical description of the HHI proposal for SVC CE3," ISO/IEC JTC 1/WG11, doc. M11246, Palma de Mallorca, Spain, Oct. 2004.
[61] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constraint coder control and comparison of video coding standards," IEEE Transaction on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 668-703, July 2003.
[62] K.-P. Lim, ed., "Text description of Joint Model reference encoding methods and decoding concealment methods," Joint Video Team, doc. JVT-L046, Redmond, WA, USA, July 2004.
[63] K. Ramchandran, A. Ortega and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," IEEE Transactions on Image Processing, Vol. 13, no. 5, Sep. 1994.
[64] M. Flierl, T. Wiegand and B. Girod, "A locally optimal design algorithm for block-based multi-hypothesis motion-compensated prediction," Proceedings of Data Compression Conference, Apr. 1998.
[65] J. Reichel, H. Schwarz, M. Wien, eds., "Joint scalable video model 7 (JSVM 7)," Joint Video Team, doc. JVT-T202, Klagenfurt, Austria, July 2006.

[66] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, no. 1, pp. 70-84, Feb. 1999.
[67] H. Schwarz, D. Marpe, and T. Wiegand, "SVC Core Experiment 2.1: Inter-layer prediction of motion and residual data," ISO/IEC JTC 1/WG11, doc. M11043, Redmond, WA, USA, July 2004.
[68] A. Segall, S. Sun, and G. J. Sullivan, "Spatial scalability," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, pp , Apr. 2007.
[69] E. François and J. Vieron, "Interlaced coding in SVC," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, pp , Apr. 2007.
[70] H. Schwarz, D. Marpe, and T. Wiegand, "Further results on constrained inter-layer prediction", Joint Video Team, doc. JVT-O074, Busan, Korea, April 2005.
[71] H. Schwarz, D. Marpe, and T. Wiegand, "Independent parsing of spatial and CGS layers," Joint Video Team, doc. JVT-S069, Geneva, Switzerland, March 2006.
[72] H. Schwarz and T. Wiegand, "Preliminary results for an r-d optimized multi-loop SVC encoder," Joint Video Team, doc. JVT-T080, Klagenfurt, Austria, July 2006.
[73] J.-R. Ohm, "Advances in scalable video coding," Proceedings of the IEEE, Vol. 93, no. 1, pp. 42-56, Jan. 2005.
[74] J. Ridge, X. Wang, Y. Bao, Y. Ye, "Low-delay, low-complexity scalable bit-rate video coding," IEEE Transactions on Circuits and Systems for Video Technology, Vol 17, pp , Apr. 2007.
[75] M. Winken, H. Schwarz, D. Marpe, and T. Wiegand, "Adaptive refinement of motion information for fine-granular SNR scalable video coding," Proceedings of EuMob'06, Alghero, Italy, Sep. 2006.
[76] H. Schwarz, D. Marpe, and T. Wiegand, "Comparison of MCTF and closed-loop hierarchical B pictures," Joint Video Team, doc. JVT-P059, Poznan, Poland, July 2005.
[77] X. Wang, Y. Bao, M. Karczewicz, and J. Ridge, "Implementation of closed-loop coding in JSVM," Joint Video Team, doc. JVT-P057, Poznan, Poland, July 2005.
[78] I. Amonou, N. Cammas, S. Kervadec, and S. Pateux, "Optimized ratedistortion extraction with quality layers," IEEE Transactions on Circuits and Systems for Video Technology, Vol 17, pp , Apr. 2007.
[79] M. Wien, H. Schwarz, and T. Oelbaum, "Performance analysis of SVC," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, pp , Apr. 2007.
[80] I. Amonou, N. Cammas, S. Kervadec, and S. Pateux, "Enhanced SNR scalability for layered CGS coding using quality layers," Joint Video Team, doc. JVT-S044, Geneva, Switzerland, Apr. 2006.
[81] S. Pateux, Y.-K. Wang, M. Hannuksela, and A. Eleftheriadis, "System and transport interface of the emerging SVC standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, pp , Apr. 2007.

[82] S. Wenger and T. Schierl, "RTP payload for SVC," IEEE Transactions on Circuits and Systems for Video Technology, Vol 17, pp , Apr. 2007.
[83] D. Singer, T. Rathgen, and P. Amon, "File format for SVC," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, pp , Apr. 2007.
[84] A. Secker and D. Taubman, "Highly scalable video compression with scalable motion coding," IEEE Transactions on Image Processing, Vol. 13, no. 8, pp. 1029-1041, Aug. 2004.
[85] J. Reichel, H. Schwarz, and M. Wien (eds.), "Scalable Video Coding – Joint Draft 6," Joint Video Team, Doc. JVT-S201,Geneva, Switzerland, Apr. 2006
[86] W. J. Butera, "Multiscale coding of images", Master's thesis, Mass. Inst. of Tech., Sept. 1988
[87] M.R. Civanlar and A. Puri, "Scalable video coding in frequency domain", SPIE Symp. VCIP (Boston), Vol. 1818, pp. 1124-1134, Nov. 1992
[88] A. Singh, J. Bove and V. Mkhael, "Multidimensional quantizers for scalable video compression", IEEE Journal on Sel. Areas in Comm., Vol. 11, pp 36-45, Jan. 1993
[89] D. Taubman and A. Zakhor, "Highly scalable, low-delay video compression", Proc. IEEE Int'l Conf. on Image Proc. (ICIP-94), Vol. 2, pp. 13-16, Nov. 1994
[90] "Coding of audio-visual objects, Part 2-Visual, Amendment 4: Streaming video profile", ISO/IEC 1449602:2000
[91] J. Reichel, H. Schwarz and M. Wien (ed), "Scalable Video Coding: Working Draft 1", JVT document JVT-N020, Jan. 2005
[92] D. Marpe, H. Schwarz and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard", IEEE Trans. on CSVT, Vol. 13,Issue 7, pp 620-636, July 2003
[93] K.R. Rao and P. Yip, "Discrete Cosine Transform: Algorithms, Advantages, Application", Academic Press, 1990
[94] test sequences ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/
[95] Refer Appendix for JSVM software.
[96] H. Schwarz and T. Wiegand, "Review paper on SVC" to be published in IEEE Trans CSVT, Vol. 17, pp ,April 2007
[97] D. Marpe, H. Kirchhoffer, M. Winken and T. Wiegand, "Improved CABAC for progressive refinement slices", Joint Video Team, Doc. JVT-T077, Klagenfurt, Austria, Jul 2006.
[98] D. Marpe, H. Kirchhoffer, M. Winken and T. Wiegand, "CE3- Improved CABAC for PR slices", Joint Video Team, Doc. JVT-U082, Hangzhou, China, Oct 2006.
[99] Y. Ye and Y. Bao, "Adaptive Variable Length Coding for FGS", Joint Video Team, Doc. JVT-T086, Klagenfurt, Austria, Jul 2006.
[100] Y. Ye and Y. Bao, "Improvements to FGS layer Variable Length Coder", Joint Video Team, Doc. JVT-S066, Geneva, Switzerland, Apr 2006.
[101] J. Ridge, X. Wang, A. Hallapuro and M. Karczewicz, "Simplification and unification of FGS", Joint Video Team, Doc. JVT-S077, Geneva, Switzerland, Apr 2006.

[102] M. Winken, H. Schwarz, D. Marpe, and T. Wiegand, "Adaptive motion refinement for FGS slices", Joint Video Team, Doc. JVT-Q031, Nice, France, Oct 2005.
[103] J. Reichel, H. Schwarz, and M. Wien (eds.), "Scalable Video Coding – Joint Draft 6", Joint Video Team, Doc. JVT-S201, Geneva, Switzerland, Apr 2006.
[104] J. Reichel, H. Schwarz, and M. Wien (eds.), "Joint Scalable Video Model JSVM-6", Joint Video Team, Doc. JVT-S202, Geneva, Switzerland, Apr 2006.
[105] J. Reichel, H. Schwarz and M. Wien (ed), "Scalable video coding: Working Draft 2", Joint Video Team, Doc. JVT-O201, Busan, Korea, Apr 2005.
[106] X. Ji, D. Zhao, W. Gao, J. Xu and F. Wu, "An Efficient SNR Scalability Coding Framework -Hybrid Open-Close Loop FGS Coding", ISCAS 2006
[107] Kemal Ugur, Panos Nasiopoulos, Rabab Ward, "An efficient H.264 based fine-granular-scalable video coding system", SIPS 2005
[108] Y. Bao, M. Karczewicz, J. Ridge and X. Wang, "FGS block enhancements for scalable video coding", MPEG doc. M11428, Oct. 2004.
[109] J. Ridge, Y. Bao, M. Karczewicz and X. Wang, "Cyclical block coding for FGS", MPEG doc. M11509, Jan. 2005.
[110] J. Ridge, Y. Bao, M. Karczewicz and X. Wang, "FINE-GRAINED SCALABILITY FOR H.264/AVC" Signal Processing and Its Applications, 2005. Proceedings of the 8th International Symposium on August 28-31, 2005 Vol. 1, pp: 247- 250
[111] JVT documents http://ftp3.itu.int/av-arch/jvt-site/
[112] H.264 software (JM12.1) & JM Manual http://iphome.hhi.de/suehring/tml/
[113] Al Bovik, "Handbook of Image and Video Processing",2nd edition, Elsevier Academic Press, 2005

BIOGRAPHICAL INFORMATION

Rahul Panchal received his Bachelor of Engineering degree in Electronics and Communications engineering from Nirma University, Gujarat, India in December 2002. After working in the industry for two and half years developing embedded hardware and software for Clinical instruments like Spectrophotometer, Titrator, pH meter, etc, he pursued his masters studies at University of Texas at Arlington. He was the member of Multimedia Processing Laboratory research group guided by Dr. K. R. Rao. He received his M.S. degree in Electrical engineering in May 2007 from Universtiy of Texas at Arlington. He worked as intern in Nvidia and Qualcomm and is currently employed with Corp R&D Video codec group, Qualcomm, San Diego. His research interests are digital signal processing, multimedia processing and cutting edge video codec developments.