# R-D Optimal Scalable Video Coding Using Soft Decision Quantization

by

Krzysztof Hebel

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2010

## AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

# Abstract

In this thesis, we study the concept of scalable video coding as implemented in the extension to the H.264 video coding standard. Specifically, for the spatial and quality scalability scenarios, we propose an optimization algorithm based on the Soft Decision Quantization (SDQ) concept, which aims at jointly optimizing all layers being encoded. The performance of the algorithm was assessed in an SVC implementation. Experimental results show, that the proposed method significantly improves the coding efficiency when compared to an unmodified SVC encoder.

## Acknowledgements

First and foremost, I would like to express my gratitude to my Supervisor, Prof. En-Hui Yang for his guidance, valuable advice and his support during my entire graduate studies at the University of Waterloo.

I would like to thank Prof. George H. Freeman and Prof. Zhou Wang for being the readers for this thesis and for their helpful comments and suggestions.

I would also like to thank all my friends at the Multimedia Communications Laboratory at the University of Waterloo: Yuhan Zhou, Chang Sun, Jin Meng, Lin Zheng and Jie Zhang. Special words of thanks go to Dr. Xiang Yu for many valuable discussions related to this work and to my roommate, Jakub Gawryjołek, for his assistance in the implementation process.

Last, but no way least, I would like to thank my girlfriend, Jodian Fairclough, for her unfailing love, support, understanding and encouragement without which completion of this thesis would not be possible.

# Table Of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The research in the area of video coding has gained a considerable amount of momentum recently, especially with the onset of on-line Internet services such as YouTube [5] or Hulu [1] and the rapidly expanding market of mobile devices. Customers nowadays expect seamless access to streamed video regardless of location or device they are using, whether it is a cellphone or a high definition television set. This poses a significant challenge to the service providers for whom an efficient and effective way of broadcasting video content becomes a necessity.

## 1.1 Video Coding and Scalability

Even with new technologies embedded in the state-of-the-art H.264 standard [9], and increasingly more powerful processors, the real-time transmission of multimedia, particularly voice and video, over packet networks continually poses major challenges for the users. The current network multimedia applications put a number of requirements on transmission capabilities (e.g. bandwidth, packet loss, end-to-end delay), but, at the same time, are required to support a broad range of endpoint devices, varying in processing power, display resolution and connection bandwidth. Therefore, it becomes desirable to be able to provide a single bit stream to serve all potential recipients, and shift the decision regarding the decoded video quality and resolution to the endpoint terminals.

The concept of scalability (containing several layers of varying quality/resolution/frame rate within a single bit stream) has been investigated several times in the past. The conventional approach of multilayer coding followed by H.264/SVC is already a part of such standards as H.262—MPEG-2 Video [6], H.263 [7], and MPEG-4 Visual [8]. Those attempts are worthy of recognition, yet they failed to provide a feasible alternative to broadcasting multiple bit streams.

It was not until the introduction of SVC however, that scalable video coding became a viable option for video distribution. It offered relatively similar coding efficiency compared to single-layer coding for each subset of the scalable bit stream. SVC managed to bring the bit rate overhead associated with scalability down to 10%, which can be considered tolerable, depending on the application. After a thorough investigation of the standard and the embedded techniques, we decided to investigate and analyze possible improvements to the encoding process that would allow closing the bit rate gap between scalable and single-layer solutions and potentially achieving parity between these two approaches.

## 1.2 Research Problems and Motivations

In this thesis, our purpose is to investigate possible improvements to the inter-layer prediction mechanism of the SVC standard. Additionally, we intend to verify, via experimental means, the performance of a subset of such improvements.

The research problems to be investigated in this proposal are:

1. What is the impact of the inter-layer residual prediction on the rate-distortion performance of spatially-scalable video coding?

2. What is the impact of the inter-layer residual prediction on the rate-distortion performance of quality-scalable video coding?

3. Our intuition tells us that it is theoretically achievable for scalable video coding to at least match single-layer coding in terms of rate-distortion performance. Therefore, our goal is to empirically verify whether it is possible to close the performance gap between spatially-scalable SVC and single-layer H.264.

The motivation driving our research lies in the suboptimal performance of current inter-layer prediction mechanisms, as implemented in the SVC standard. Based on the state-of-the art technique of Soft Decision Quantization we are aiming to improve the accuracy of inter-layer prediction and consequently improve the overall rate-distortion performance of the SVC codec.

## 1.3 Thesis Organization and Contribution

The thesis is organized as follows.

In Chapter 2 we provide the background for our research. A brief description of modern video standards is given using H.264 standard as the prominent scheme, followed by a detailed discussion of Scalable Video Coding and the inter-layer prediction techniques in particular. Subsequently, the concept of Soft Decision Quantization is explained and brief review of previous work on this topic is included.

In Chapter 3 we introduce our RD-optimal approach to scalable video coding. We present the algorithm for obtaining optimal quantized residual for both the base and enhancement layer. We then proceed to discussing its application within the frames of spatial scalability and quality scalability scenarios.

In Chapter 4 we describe the implementation of the proposed algorithm for both discussed scenarios (spatial and quality scalability). Subsequently, we present the obtained experimental results and provide our interpretation and discussion.

Finally, we summarize the entire thesis, provide our conclusions and discuss potential future work directions in Chapter 5.

## 1.4 Notation

Throughout this thesis a following notation is used. We generally use capital letters to denote quantities associated with the enhancement layer and lowercase letters for those associated with the base layer. Particularly, we follow the commonly used naming convention for signals in a video encoder/decoder:

- $\boldsymbol{X}, \boldsymbol{x}$ - pixel domain frame, enhancement and base layer, respectively (as explained above)

- $\boldsymbol{Z}, \boldsymbol{z}$ - pixel domain residuals, obtained after subtracting the motion prediction from the original frame
- $\boldsymbol{P_m}, \boldsymbol{p_m}$ - pixel domain motion prediction signal
- $\boldsymbol{P_r}$ - residual prediction signal (present only in the enhancement layer, hence capital letter), obtained from the reconstructed base layer residual
- $f_{up}(\cdot)$ - up-sampling method, additional superscript will indicate pixel ($px$) or transform ($dct$) domain
- $\mathrm{T}[\cdot]$ - DCT transform as implemented in the H.264/SVC standard (reference!)
- $Q, q$ - quantization step size
- $d(\cdot)$ - distortion measure

Furthermore, we denote the objective cost function by $J$ and we use $D_B, R_B, D_E, R_E$ to represent base layer distortion and rate, respectively, and enhancement layer distortion and rate, respectively. Additionally, $\lambda$ represents the Lagrangian multiplier.

Finally, throughout this thesis we assume a two layer scenario, with one base layer and one enhancement layer (hence the notation above), however our approach can be easily extended to multi-layer scenarios. In such case both spatial and quality scalability can be used concurrently with a number of layers for each scalability type.

# Chapter 2

# Background

This chapter provides the theoretical background for our research on Scalable Video Coding. The overview of the state-of-the-art H.264/AVC coding standard is included, followed by a description of its scalable extension, SVC. Finally, the concept of Soft Decision Quantization is reviewed.

## 2.1 H.264/AVC Video Coding Standard

### 2.1.1 Introduction

The development of modern digital video coding standards truly began in the early 1990 when the ITU-T's[1] H.261 video codec was introduced. It started a line of recommendations, which shaped the field of video compression. Throughout the years following the announcement of H.261, both ITU-T and ISO/IEC[2] have released recommendations for a number of standards, namely MPEG-1, MPEG-2, H.263, H.263+ and MPEG-4 Part 2 (also known as MPEG Visual), all of which are basically enhancing the feature set of their predecessor. Each subsequent recommendation would introduce improvements such as half-pixel motion compensation (first proposed in MPEG-2) or multiple reference frames (starting from H.263+). The newest and most feature-rich standard is the state-of-the-art

[1] ITU-T - International Telecommunication Union - Telecommunication Standardization Sector

[2] ISO - International Organization for Standardization, IEC - International Electrotechnical Commission

ITU-T's H.264 (also referred to as MPEG-4 Part 10) developed by the Joint Video Team (JVT) in 2003. From its inception, it targeted a wide range of multimedia applications ranging from low resolution mobile video to high definition television.

### 2.1.2 General Structure

The H.264/AVC video coding standard was designed for a wide range of technical solutions and applications. Those include, but are not limited to, broadcasting, storage, as well as conversational, video-on-demand and multimedia streaming services. In order to make the standard more flexible and easily adaptable, the H.264 design distinguishes a Video Coding Layer (VCL) which covers all techniques related to the video coding process and the Network Abstraction Layer (NAL), which formats the coded video stream and appends header information for easier transport and storage [20]

#### Network Abstraction Layer

As mentioned above, NAL was introduced in order to facilitate media transport over various networks as well as its storage on a physical medium. All coded data is contained within a NAL unit which contains a header and the video payload. It serves as a generic format for a wide range of systems, both packet-based (one example might include real-time video transmission using RTP[3] protocol) and bitstream. One of the features of NAL is the possibility to include an importance parameter which would indicate the significance level of the data contained within a particular unit.

#### Video Coding Layer

The Video Coding Layer is the more familiar part of the H.264 encoder. It encompasses all the components responsible for processing and encoding video content: temporal and spatial prediction combined with transform coding and quantization. Figure 2.1 represents a high-level diagram of the VCL.

Encoding of the input video sequence begins typically with an intra frame (referred to as I frame) which means that only information contained within that frame is used.

[3] RTP - Real-time Transport Protocol

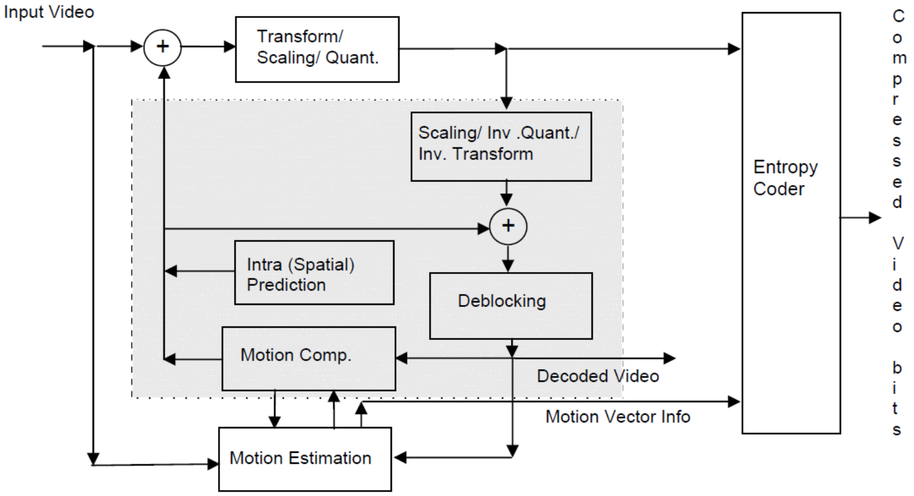

Figure 2.1: High-level encoder architecture [23]

All subsequent frames can be encoded using either intra prediction or inter prediction (i.e. using information from previously coded frames). The residual signal (also known as the prediction error) is then subjected to DCT transform and quantization which are followed by entropy coding. This data, together with additional information regarding the prediction used, comprises the output bitstream.

Every H.264 encoder also contains a decoder within itself. It performs dequantization and inverse transform on the quantized coefficients provided by the encoder. These are then added to the prediction signal and together form a reconstructed frame, which can be then used as reference for encoding subsequent frames. Additionally, in order to improve the prediction quality and remove blocking artifacts, an in-loop deblocking filter is applied to the reconstruction.

In the following sections we will provide more detail on the H.264/AVC video coding standard and its features.

### 2.1.3 Codec Features

In this section we will briefly describe the most important features of the H.264/AVC video coding standard. Some were present in the previous standards and were just improved or extended, while a number of entirely new coding features has been added in order to further improve the overall performance of the new standard.

### Intra Prediction

Depending on a particular frame/slice type, each macroblock can be coded using a number of modes. In case of I frames and I slices, the intra prediction is employed with two possible types: Intra_4x4 and Intra_16x16 [24]. The first type allows for prediction of 4x4 luma blocks with 9 possible prediction modes. Due to the block size, it is well suited for coding frame parts with fine detail present. The Intra_16x16 type allows for prediction of 16x16 luma blocks with 4 possible prediction modes and is better suited for coding smooth areas of a particular picture.

Additionally, the H.264 standard specifies an I_PCM coding type which allows for a complete bypass of prediction and transform coding. In this case the values of encoded samples are sent directly. Such coding mode can be beneficial when a precise representation of the sample values is required.

The intra prediction in H.264 is performed entirely in spatial domain by referring to the neighboring samples of previously coded blocks. This constitutes a significant difference with respect to H.263 and MPEG-4 Visual where it would be performed in the transform domain.

### Inter Prediction

The inter prediction (inter meaning that it reaches outside of the current frame) refers to the motion compensated prediction for the currently processed macroblock. Apart from previously described intra prediction coding types, the H.264 standard defines a number of such predictive types (referred to as P macroblock types), that allow for obtaining a prediction based on one or more of the previously coded frames.

In order to improve the accuracy of motion prediction with respect to previous standards, partitions with luma block sizes of 16x16, 16x8, 8x16 and 8x8 are supported in H.264. In case when 8x8 partition is used, an additional syntax element is transmitted to indicate subpartitioning - the block can be further divided into 8x4, 4x8 or 4x4 size blocks. Figure 2.2 depicts supported partition and subpartition sizes.

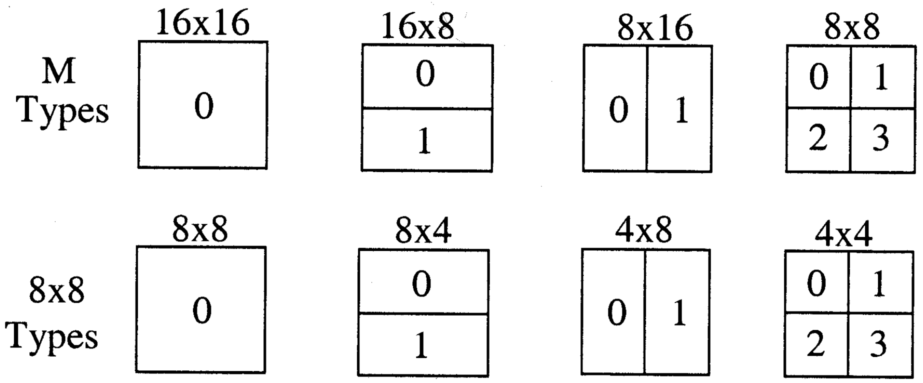

Figure 2.2: Macroblock partitioning for motion compensation. Top row: macroblock partitions, bottom row: subpartitions [24]

During the motion prediction process, the prediction signal is obtained by translating the corresponding block from the reference frame by a displacement vector (referred to as a motion vector). A maximum of 16 motion vectors can be transmitted for each macroblock (in such scenario there are four 8x8 partitions and each is further divided into 4 4x4 subpartitions). The prediction signal is then subtracted from the current frame in the process referred to as motion compensation. The difference, also known as the prediction error, constitutes the residual signal which is further subjected to DCT transform and quantization.

The H.264 encoder maintains a relatively low prediction error, thanks to the quarter-pel motion vector accuracy. At the cost of increased computational complexity a far more accurate motion prediction can be obtained. When the motion vector points to integer positions, the corresponding samples in the reference picture are used. In case a fractional position occurs in the process, interpolation is used to generate non-integer samples (one dimensional 6-tap FIR[4] filter for half-pel and additional averaging for quarter-pel) [24, 20].

An additional improvement over previously used video coding standards is the ability to utilize multiple reference frames for motion compensation. H.264 encoder may store up to 16 frames in the reference buffer and use a weighted average when calculating a particular motion vector. Similarly to quarter-pel accuracy, this feature greatly increases computational and storage complexity, allows however for better video quality.

[4] FIR - Finite Impulse Response

The accuracy of motion prediction depends greatly on the video content. In some cases, e.g. still frames or slow camera pan, it might be actually more beneficial to skip motion compensation altogether. H.264 standard defines an additional mode, designed with such scenarios in mind. In the P_SKIP mode, no motion vectors or prediction error signal is transmitted and, subsequently, no transform or quantization occurs.

The bitrate of the encoded video sequence can be further reduced by employing bi-predicted frames referred to simply as B frames. When performing motion prediction, encoder uses a weighted average of two distinct motion-compensated prediction values based on at one or more previously coded pictures stored in two reference buffers (List 0 and List 1). First buffer stores frames which are temporally before current frame, while the second buffer stores frames which are ahead of current frame. Each buffer can store up to 16 frames, effectively increasing the maximum number of reference frames to 32. In case of B frames, four different types of inter prediction are supported: list 0 (forward prediction), list 1 (backward prediction), bi-predictive and direct (similar to P_SKIP mode in P frames).

### Transform and Quantization

During the encoding process in H.264, each macroblock is transformed, quantized and finally undergoes entropy coding [9]. In most of the previous popular standards (namely MPEG-1, MPEG-2, MPEG-4 Part 2: Visual and H.263) an 8x8 Discrete Cosine Transform (DCT) is used as the basic transform. The "baseline" profile of H.264 specifies three types of transforms to be applied, depending on the residual data to be processed: a transform for the 4x4 array of DC luma coefficients in 16x16 intra predicted macroblocks, a transform for the 2x2 array of chroma coefficients and a transform for all remaining 4x4 blocks of residual data (including full inter predicted macroblocks). When the Fidelity Range Extensions (FRExt, [23]) are enabled, an 8x8 transform is also available for processing luma coefficients.

Figure 2.3 on page 11 represents the transmission order of the macroblock data. If the macroblock is intra coded, the block labeled -1 (containing the DC coefficients of each luma 4x4 block) is transmitted first, followed by the luma residual blocks labeled 0-15 (the DC coefficients in these blocks are all set to 0 in case of intra prediction). Next, the blocks containing 2x2 arrays of DC chroma coefficients are sent, followed by the remaining

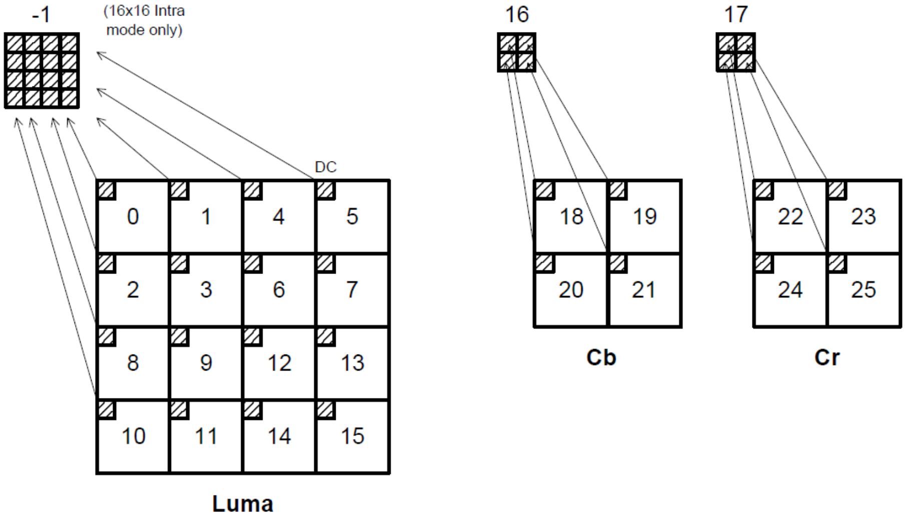

Figure 2.3: Scanning order of residual blocks within a macroblock [18]

residual chroma coefficients.

The transform in H.264 is applied to the residual data after motion-compensated prediction (inter) or intra prediction. While it is commonly referred to as a DCT transform, in reality it is a close approximation to one with a number of important differences [reference to H.264 transform white paper]:

1. It is a fully integer transform and all operations can be carried out with integer arithmetic (without loss of accuracy) which simplifies the implementation and reduces computational complexity.
2. There is a matching inverse transform specified for the decoder side which should prevent any mismatch from occurring during the decoding process.
3. The core part of the transform is devoid of multiplications (contains only additions and bit-shifts), which also contributes toward reducing computational complexity.
4. A scaling factor is integrated into the transform as a part of the quantizer and reduces the total number of multiplications.

The entire transform process can be therefore carried out using 16-bit integer arithmetic and includes only one multiplication per coefficient.

The basic quantizer in video coding is defined as follows [18]:

$$Z_{ij} = \texttt{round}\left(Y_{ij}/Q_{step}\right) \tag{2.1}$$

where $Y_{ij}$ denotes a transform coefficient resulting from the forward transform described above, $Q_{step}$ is the quantization step-size and $Z_{ij}$ is the resulting quantized coefficient.

There is a total number of 52 quantization step values and they are selected by providing the quantization parameter (QP, ranging from 0 to 51) to the encoder, i.e:

$$\begin{gathered}\texttt{QP} = 0 \quad \rightarrow \quad Q_{step} = 0.625\\ \texttt{QP} = 1 \quad \rightarrow \quad Q_{step} = 0.6875\\ \vdots\\ \texttt{QP} = 51 \quad \rightarrow \quad Q_{step} = 224\end{gathered}$$

and the $Q_{step}$ doubles in size for every increment of 6 in QP (`QP` is the H.264 quantization parameter which determines the quantization step-size).

The complete forward transform, scaling and quantize process can now be described as follows [18]:

1. The input to the process is a 4x4 matrix of pixel values **Z**
2. The core transform is applied first resulting in a 4x4 matrix of transform coefficients **W**:

   $$\mathbf{W} = \mathbf{C}_f\mathbf{Z}\mathbf{C}_f^{\mathbf{T}} \tag{2.2}$$

   where

   $$\mathbf{C}_f = \begin{bmatrix} 1 & 1 & 1 & 1\\ 2 & 1 & -1 & -2\\ 1 & -1 & -1 & 1\\ 1 & -2 & 2 & -1 \end{bmatrix}. \tag{2.3}$$

   is the forward transform matrix.

3. The transform coefficients are subsequently scaled and quantized using integer arithmetic (i.e. bit shifts):

$$\mathbf{U} = \mathbf{W} \cdot \frac{\texttt{PF}}{Q_{step} \cdot 2^{qbits}} \tag{2.4}$$

   where $\texttt{PF}$ is the scaling factor and $qbits = 15 + \text{floor}(QP/6)$.

The complete inverse transform, re-scaling and dequantize process can be described as follows [reference to the transform white paper]:

1. First, the quantized coefficients need to be dequantized (this step will also include transform pre-scaling):

$$\mathbf{W}' = \mathbf{U} \cdot Q_{step} \cdot \texttt{PF} \cdot 64 \tag{2.5}$$

2. Subsequently, the inverse core transform is applied:

$$\mathbf{Z}' = \mathbf{C}_i^T \mathbf{W}' \mathbf{C}_i \tag{2.6}$$

   where

$$\mathbf{C}_i = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}. \tag{2.7}$$

   is the inverse transform matrix.

3. Finally, post-scaling is applied:

$$\mathbf{Z}'' = \texttt{round}\left(\mathbf{Z}'/64\right) \tag{2.8}$$

4. The output of the process is a 4x4 block of reconstructed residual samples: $\mathbf{Z}''$

In general, the H.264 standard uses a scalar quantizer, however its definition and implementation are complicated by the requirement to perform only integer calculations as well as the need to incorporate the previously described scaling matrices.

**Other Features**

There is a number of other features that distinguish the H.264 video coding standard from its predecessors that also contribute to its greatly improved performance:

- In-loop deblocking filter: A common challenge in designing block-based video coding methods is the appearance of a blocking effect in the reconstructed frames. A well designed filter, when placed inside the reconstruction loop, can significantly minimize such artifacts and therefore improve the resulting video quality. Since some frames will be later used for reference, deblocking filter also increases the accuracy of inter prediction.

- Quarter-pel motion compensation: H.264 was the first standard to increase the motion vector precision from previously used half-pel to quarter-pel. This allows for improved motion compensation.

- Multiple reference frames: In the H.264 standard up to 16 frames can be used for reference during motion prediction which can improve the performance at the cost of increased complexity (both computational and memory)

- Context-based adaptive entropy coding: H.264 features two entropy coding methods, CAVLC (Context-Adaptive Variable Length Coding) and CABAC (Context-Adaptive Binary Arithmetic Coding) with the latter offering better coding performance at the cost of increased complexity. Both methods utilize context-based adaptivity which allows for increased coding efficiency.

## 2.2 Scalable Video Coding

Earlier, in the Introduction, we outlined reasons that were behind the concept of scalable video coding. The idea was so appealing that MPEG group has issued a call for proposals on SVC technology in October 2003. In the subsequent competition process, a proposal made by the Image Communication Group at the Heinrich Hertz Institute was chosen as a starting point for the SVC standardization effort. Since the new solution was largely based on the state-of-the-art H.264/AVC video coding standard, the MPEG[5] group along with the VCEG[6] group have agreed to include the SVC project as an amendment of the H.264 standard. Finally, SVC has officially become a standard in July 2007.

Arguably, one of the most important features of the SVC design is its broad compliance with the H.264/AVC standard. Motion compensation, intra prediction, transform and entropy coding, de-blocking as well as NAL-packetization are all used as specified in the original standard. The base layer generated by the SVC encoder is by default compliant with H.264/AVC and can be decoded by any standard conforming H.264/AVC decoder. The new tools provide additional functionality in terms of spatial, temporal or quality scalability and can be utilized by the SVC decoder, while still providing basic backward compatibility [22, 26].

Generally speaking, a video bit stream is deemed scalable when a part of the stream can be removed while leaving a substream which is decodeable by some target decoder. At the same time, the removed part forms an equally valid substream representing the source content, albeit with lower quality and significantly lower bit rate. Conversely, if a bit stream does not have this property, it is considered a single-layer bit stream [22].

The typical modes of scalability include spatial, temporal and quality scalability. Spatial scalability represents a scenario in which subsets of the bit stream (referred to as layers) contain varying in size representations of the source content. Temporal scalability on the other hand, refers to a scenario in which different layers contain representations of varying frame rate. Finally, quality scalability provides means for generating layers of same spatio-temporal resolution, but different fidelity, where by fidelity we usually understand signal-to-noise ratio (SNR).

### 2.2.1 Spatial Scalability

As mentioned earlier in this proposal, SVC follows the conventional approach to multilayer coding. In the case of spatial scalability it is reflected by each layer corresponding to a particular spatial resolution. Each such layer is then being referred to as a spatial layer with the base layer being the lowest supported resolution.

[5] MPEG - Motion Picture Experts Group

[6] VCEG - Video Coding Experts Group

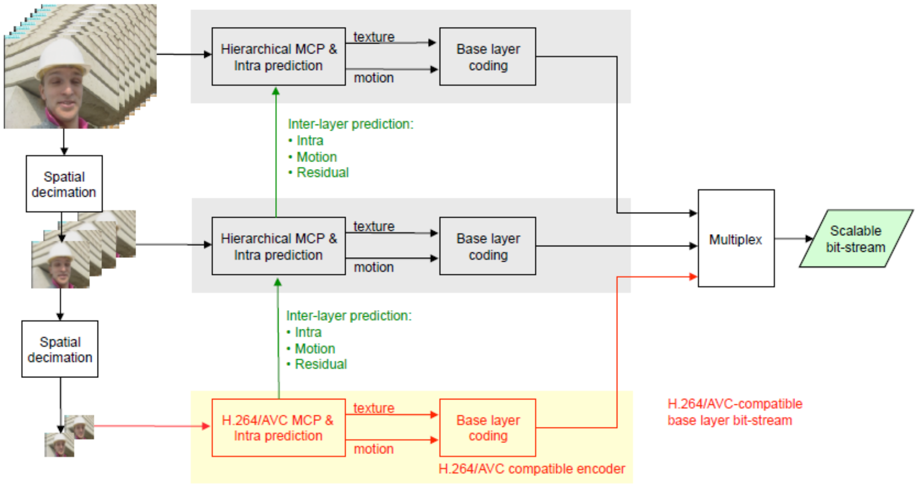

Figure 2.4: High level SVC encoder architecture

During encoding, each spatial layer undergoes same processes as during single-layer coding, i.e. motion compensation, intra prediction, transform, quantization and entropy coding. However, in order to increase coding efficiency, additional mechanisms are employed that aim at utilizing the spatial correlation between layers. The main goal lying at the base of those mechanisms is to use as much of the information from the lower layer as it is possible.

There were previous attempts at providing spatial scalability within the encoded bitstream (H.262, MPEG-2, H.263, MPEG-4 Visual), however those solutions only used upsampled version of the reconstructed base layer signal. There are numerous circumstances when it is not the best choice and might even degrade the resulting video quality (e.g. little motion between frames, very fine texture, high amount of detail present).

In the following subsections we will described the three SVC inter-layer prediction techniques that are included in the SVC standard. It has been demonstrated [22], that with all those techniques enabled, the efficiency of SVC encoder can be brought within 10% away from that of a single-layer H.264 encoder.

### Inter-layer motion prediction

In order to facilitate the use of lower layer motion information, SVC standard defines additional macroblock modes. When selected, the partitioning of the co-located 8x8 block

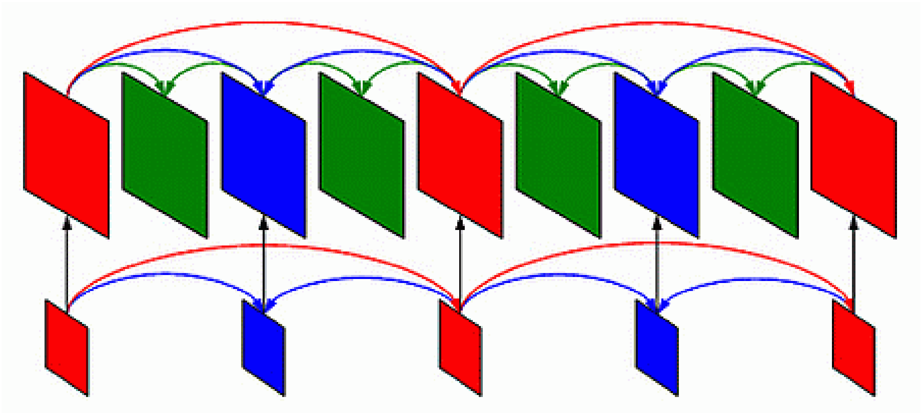

Figure 2.5: Multi-layer structure with additional inter-layer prediction (black-arrows) [4]

in the lower resolution layer is upsampled and applied to the current layer. The reference picture indices are copied in the same process as well. Additionally, the associated motion vectors are scaled by a factor of 2 (assuming doubling the resolution from layer to layer, generally speaking motion vectors are scaled by the same factor as the resolution difference). In order to increase motion prediction accuracy, an additional quarter-pel motion vector refinement is possible, otherwise only the scaled lower layer vector is used. Even if non-scalable macroblock mode is selected, the scaled motion vector can be still utilized, serving as a predictor for conventional macroblock modes [26, 22, 4]

### Inter-layer residual prediction

Apart from motion information, the encoder can also choose to utilize the lower layer residual information. As part of the syntax, a flag signaling such option is transmitted. In such case, base layer signal of the co-located block is block-wise up-sampled and used as a prediction for the residual signal of the current macroblock. In effect, only the resulting difference is coded (preferably of lower energy than initial residual signal). Such mechanism can be useful in coding of frames with large flat areas containing little or no texture.

### Inter-layer intra prediction

There is also an additional intra prediction macroblock mode introduced in the SVC standard. When selected, it causes the encoder to use the up-sampled reconstruction signal from the co-located lower layer block.

### 2.2.2 Quality (SNR) Scalability

Another feature of the SVC standard is support for quality scalability, also referred to as SNR scalability. It can be described as a special case of the spatial scalability with the same picture sizes for base and enhancement layers. In this scenario the encoder generates multiple layers with the same resolution, but different quality. There are two types of SNR scalability included in the specification: coarse-grain scalability (CGS) and fine-grain scalability (FGS).

Similarly to the spatial scalability scenario, it has been demonstrated that the performance gap separating SVC from a single-layer encoder can be reduced to as low as 10% bitrate overhead.

#### Coarse-grain SNR scalability

Coarse-grain SNR scalability, as implemented withing the SVC is very similar to the spatial scalability mechanism. The only difference in this case is the lack of up-sampling during inter-layer prediction. In fact, the process involves requantizing the residual refinement in the enhancement layer with a lower quantization step-size with respect to the lower layer. This concept, while it facilitates quality scalability, is limited in the number of rate points it can effectively support.

Our optimization method proposed in this thesis assumes use of CGS for quality scalability.

#### Fine-grain SNR scalability

The fine-grain SNR scalability addresses the shortcomings of the coarse-grain scalability feature. It employs the concept of Progressive Refinement (PR) slices - each of them represents a refinement of the lower layer residual signal. Additionally, the ordering of the transform coefficients in PR slices allows the corresponding NAL units to be truncated at arbitrary positions and therefore provide a fine-grain quality tuning.

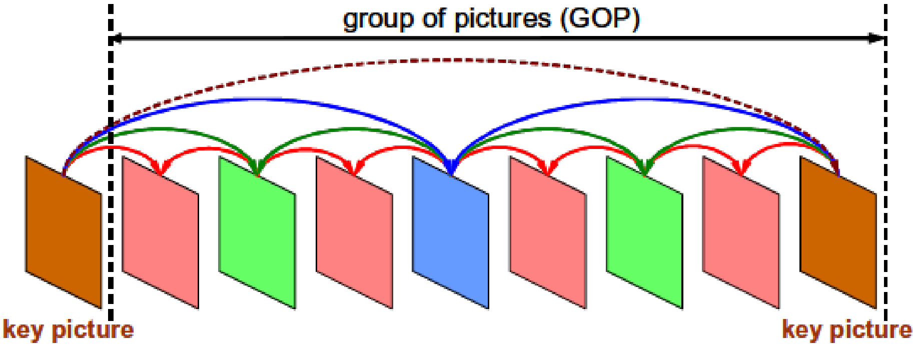

Figure 2.6: Hierarchical prediction structure with 4 dyadic levels [22]

### 2.2.3 Temporal Scalability

It is said that a bitstream offers temporal scalability if one can easily decode its subset with a framerate lower than that of the highest available enhancement layer. It is achieved by the means of hierarchical prediction structures (e.g. hierarchical B-pictures). The encoder regularly inserts key pictures into the bitstream, that are predicted using only previously encoded key pictures. The frames that appear between two consecutive key pictures are then hierarchically predicted (bi-predicted), in order to provide finer temporal resolutions. The sequence of only key pictures represents then the lowest temporal resolution and the decoder can refine it by processing pictures of higher temporal prediction levels (see Figure 2.6). It is obvious then that the entire mechanism can be implemented using tools that are already present in the H.264/AVC standard (i.e. B-pictures) and thanks to the fact that H.264 treats coding and display orders independently of each other.

## 2.3 Soft Decision Quantization

Traditional approach to scalar quantization, as implemented in all modern video coding standards, involves calculating the quantized residual, based on the transform coefficient and the quantization step-size, as a simple division. It can be referred to as Hard Decision Quantization (HDQ) since once the step-size is fixed, the output of the quantizer is determined. Therefore, optimization within the encoder usually targeted the quantization step-size as its parameter. It is however possible to look at this problem from a different angle. In the so-called Soft Decision Quantization (SDQ), the quantized residual itself becomes the free parameter that can be optimized during the coding process in order to improve the overall performance [29]. In the following subsections we will cover the Soft Decision Quantization framework for H.264 as well as the accompanying algorithm design [13, 30, 29].

### 2.3.1 Soft Decision Quantization Framework

The SDQ algorithm for given motion compensation (modes and motion vectors) and quantization step-size is formulated as,

$$\min_{\boldsymbol{U}} d\left(\boldsymbol{X} - \boldsymbol{P}, \hat{\boldsymbol{Z}}\left(\boldsymbol{q}, \boldsymbol{U}\right)\right) + \lambda \cdot r\left(\boldsymbol{U}\right) \tag{2.9}$$

where $\boldsymbol{X}$ is the given frame, $\boldsymbol{P}$ denotes the prediction signal (obtained during motion compensation), $\hat{\boldsymbol{Z}} = \left\{\hat{\boldsymbol{z}} : \hat{\boldsymbol{z}} = \mathrm{T}^{-1}\left[q \cdot \boldsymbol{u}\right]\right\}$ $\mathrm{T}^{-1}$ denotes the inverse DCT transform in H.264, $d(\cdot)$ is a distortion measure, $r(\cdot)$ is the rate function for CAVLC, $\lambda$ is the Lagrangian coefficient and, finally, $\boldsymbol{U}$ denotes the set of quantized coefficients undergoing optimization.

Based on the above formulation, a specific, CAVLC-tailored SDQ algorithm has been designed aiming at calculating optimal quantized coefficient values given a particular entropy coding method [29].

### 2.3.2 Soft Decision Quantization Algorithm Design

In this subsection we will briefly present the core SDQ algorithm for 2.9, as described in [29]. The optimization of the entire frame can be solved in a block-by-block manner, with each block being 4x4 in size (since H.264 uses a 4x4 implementation of a DCT transform, same block size is used in the entropy coding process). Hence, 2.9 can be reduced to:

$$\boldsymbol{u} = \arg\min_{u} d\left(\boldsymbol{x} - \boldsymbol{p}, \mathrm{T}^{-1}\left(\boldsymbol{u} \cdot q\right) + \lambda \cdot r(\boldsymbol{u})\right) \tag{2.10}$$

where $r(\mathbf{u})$ is the number of bits required to encode the quantized residual for a particular block using the CAVLC entropy coding method given that its two neighboring blocks have been optimized [29, 30].

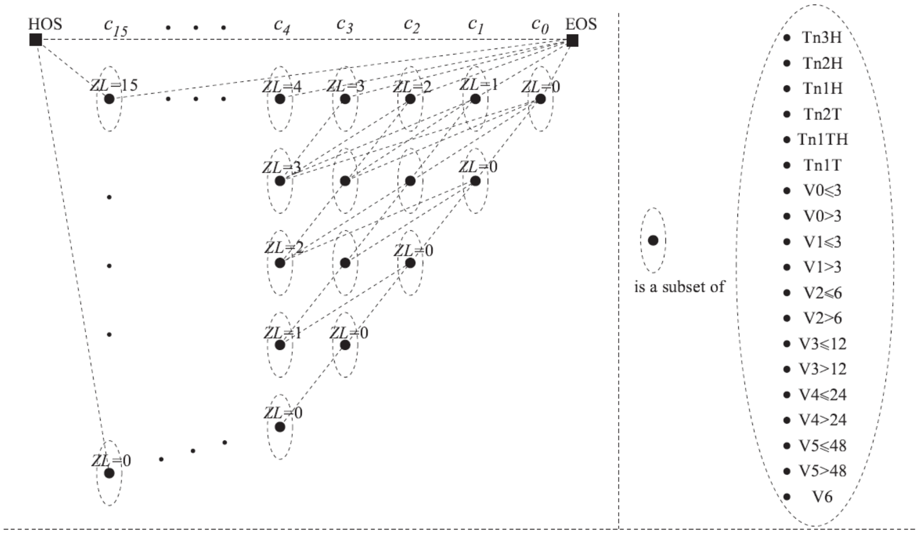

Figure 2.7: The graph structure for Soft Decision Quantization based on CAVLC [30]

By applying Euclidean distance as a distortion measure (since DCT maintains Euclidean distance), 2.10 becomes:

$$\boldsymbol{u} = \arg\min_{u} \|\boldsymbol{c} - \boldsymbol{u} \cdot q\|^2 + \lambda \cdot r(\boldsymbol{u}) \tag{2.11}$$

where $\boldsymbol{c} = \mathrm{T}\left[\boldsymbol{x} - \boldsymbol{p}\right]$ is computed beforehand, in order to reduce computational complexity. An added benefit to calculating the distortion in the DCT domain is the possibility to apply dynamic programming for solving the problem. Consequently, a graph is constructed to represent the vector space of the possible quantization outputs, with each transition corresponding to a particular (*run*, *level*) pair and each path in the graph results in a unique sequence of quantization output (see Figure 2.7 for the complete graph). The optimal solution to the previously stated optimization problem is the one associated with the path having lowest cost among all paths in the graph.

# Chapter 3

# RD Optimal Coding for Scalable Video Coding

In this chapter we are presenting our main contribution, which is the application of Soft Decision Quantization (SDQ) to the Scalable Video Coding by the means of an inter-layer optimization algorithm. In Chapter 2, Section 2.3 we have reviewed the concept of Soft Decision Quantization and briefly described the optimal algorithm design for the H.264 video coding standard. In this thesis, we are extending this idea to the scalable scenario and implementing it within the Scalable Video Coding. Following sections will describe this process in more detail.

## 3.1 Improving Inter-layer Prediction in SVC

In Chapter 2, Section 2.2 we have described the scalable extension to the H.264 standard, the Scalable Video Coding. We have particularly focused on the scalability features of the new codec. Let us now perform a deeper analysis of a subset of those features.

The design of Scalable Video Coding was largely influenced by the required backwards compatibility with H.264. One of the requirements was the ability to decode the base layer by a standard-compliant decoder. Consequently, all attempts to utilize the existing inter-layer correlation were limited to techniques which fit within the general structure of an H.264 encoder. The inter-layer motion prediction and inter-layer residual prediction are

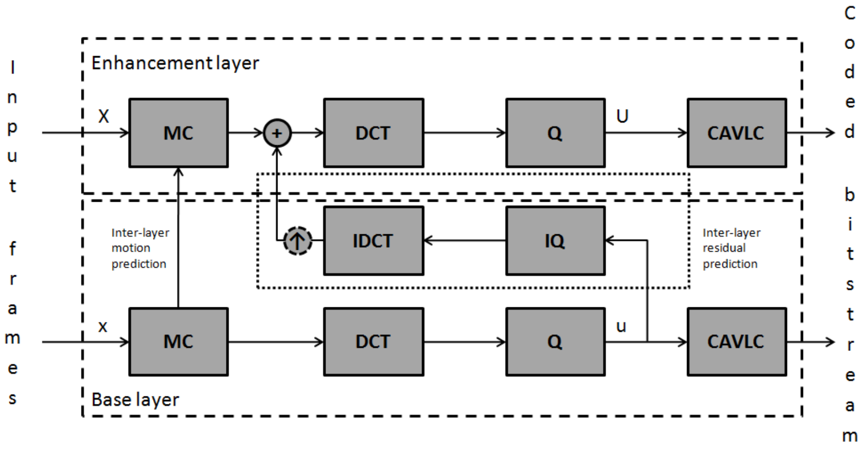

Figure 3.1: Encoding of a single P frame in SVC with the use of residual prediction

perfect examples of such techniques (both have been covered in more detail in Chapter 2, Section 2.2.1).

The question that becomes obvious once we take a closer look at the scalability implementation in SVC, is whether it is the best possible approach. After all, the standard was initially designed to handle single-layer scenarios and was only extended to multi-layer cases via several relatively straightforward additions. The key fact here, is that at no point during the encoding process is the impact of lower layer on higher layer taken into account. The inter-layer prediction is formed *post factum*, once the lower layer has been processed and the encoder has moved on to the next layer. At this point, we are simply hoping that due to the existing correlation between different layer content, some portion of that previously encoded lower layer information can be reused.

Let us analyze the scenario illustrated in Figure 3.1. It represents dual-layer coding of a single P frame and includes all stages of the process: motion compensation (MC), DCT transform, quantization and entropy coding (CAVLC). Additionally, inter-layer motion prediction and inter-layer residual prediction paths have been included. Based on this illustration, the reconstructed frame distortion in the base layer can be written as:

$$D_B = \|(\boldsymbol{x} - \boldsymbol{p}_m) - \hat{\boldsymbol{x}}\|^2 = \left\|(\boldsymbol{x} - \boldsymbol{p}_m) - \mathrm{T}^{-1}\left[\boldsymbol{u} \cdot q\right]\right\|^2 \tag{3.1}$$

where $\hat{\boldsymbol{x}}$ denotes the reconstructed frame.

Similarly, the reconstructed frame distortion in the enhancement layer can be written as:

$$D_E = \left\|(\boldsymbol{X} - \boldsymbol{P}_m - \boldsymbol{P}_r) - \hat{\boldsymbol{X}}\right\|^2 = \left\|(\boldsymbol{X} - \boldsymbol{P}_m - \boldsymbol{P}_r) - \mathrm{T}^{-1}\left[\boldsymbol{U} \cdot Q\right]\right\|^2 \tag{3.2}$$

the notable difference here being the inclusion of the inter-layer residual prediction term $\boldsymbol{P}_r$ (the inter-layer motion prediction has been accounted for by the $\boldsymbol{P}_m$ term). Analogically, $\hat{\boldsymbol{X}}$ denotes the reconstructed frame.

We have mentioned before, that processing of each particular layer bears great resemblance to the H.264 encoder (with the base layer processing being nearly identical and fully compatible). Therefore, implementing SDQ within SVC becomes a straightforward task - we simply follow the algorithm described in Chapter 2, Section 2.3 and, more exhaustively, in [13, 30, 29]. The optimization procedure becomes, for base and enhancement layer, respectively:

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} d\left(\boldsymbol{x} - \boldsymbol{p}_m, \mathrm{T}^{-1}\left[\boldsymbol{u} \cdot q\right]\right) + \lambda \cdot r\left(\boldsymbol{u}\right) \tag{3.3}$$

$$\boldsymbol{U} = \arg\min_{\boldsymbol{U}} d\left(\boldsymbol{X} - \boldsymbol{P}_m - \boldsymbol{P}_r, \mathrm{T}^{-1}\left[\boldsymbol{U} \cdot Q\right]\right) + \lambda \cdot r\left(\boldsymbol{U}\right) \tag{3.4}$$

where $d(\cdot)$ is the distortion measure, the distortion calculation is performed as shown in Equations 3.1 and 3.2 and $r(\cdot)$ is the rate function for CAVLC which calculation is performed according to the graph structure described in [30].

It is quite obvious at this point that the above approach is hardly a novel one. We have simply applied the existing SDQ method to each layer independently, effectively treating each one as a standalone H.264 encoder. Let us, however, take a closer look at the inter-layer residual prediction $P_r$. From Figure 3.1 it is clear that $P_r$ can be represented as follows:

$$\boldsymbol{P}_r = f_u\left(\mathrm{T}^{-1}\left[\boldsymbol{u} \cdot q\right]\right) \tag{3.5}$$

where $f_u$ is the up-sampling filter (assuming spatial scalability scenario, in case of quality scalability this operation is simply omitted). Let us plug this expression into Equation 3.2:

$$D_E = \left\| (\boldsymbol{X} - \boldsymbol{P}_m - P_r) - \hat{\boldsymbol{X}} \right\|^2 = \left\| \left(\boldsymbol{X} - \boldsymbol{P}_m - f_u\left(\mathrm{T}^{-1}\left[\boldsymbol{u} \cdot q\right]\right)\right) - \mathrm{T}^{-1}\left[\boldsymbol{U} \cdot Q\right] \right\|^2 \quad (3.6)$$

At this point it becomes clear that the reconstruction quality and the resulting enhancement layer distortion is dependent on the quantization output of the previously encoded base layer frame. Therefore, it can be potentially beneficial, to take that impact into account when performing the base layer optimization. In the following section, we explore a combined approach to SDQ within SVC, namely, we propose an iterative optimization procedure which aims at lowering the total cost of encoding one frame in both supported layers.

## 3.2 Soft Decision Quantization in Scalable Video Coding

Let us assume a dual-layer scenario, with one base layer and one enhancement layer. Our objective is to obtain a set of optimal quantized transform coefficients for base and enhancement layer that would minimize the total cost for encoding one frame in both layers. In that process, we aim at utilizing the existing inter-layer dependency according to the discussion in previous section (Section 3.1). We can therefore formulate a following optimization problem:

$$\min_{u,U} D_B + \alpha \cdot D_E + \lambda \left(R_B + R_E\right) \quad (3.7)$$

where $D_B, D_E$ denote distortion for base and enhancement layer, respectively, and $R_B, R_E$ denote rate for encoding a given frame in base and enhancement layer, respectively.

Let us rewrite Equation 3.7 in order to emphasize the distortion calculation and the inter-layer dependency:

$$\min_{u,U} d(\mathrm{T}^{-1}_{4x4}(\boldsymbol{u} \cdot q), \boldsymbol{z}) + \alpha \cdot d(\mathrm{T}^{-1}_{8x8}(\boldsymbol{U} \cdot Q), \boldsymbol{Z} - f_{up}(\mathrm{T}^{-1}_{4x4}(\boldsymbol{u} \cdot q))) + \lambda \left(r(\boldsymbol{u}) + r(\boldsymbol{U})\right) \quad (3.8)$$

and the notation follows the convention that we outlined in Chapter 1, Section 1.4, $\alpha$ is the weighting parameter for the enhancement layer distortion and $r(\cdot)$ is the rate function for CAVLC.

According to the SVC standard ([26]), the up-sampling of the residual prediction signal is performed via an up-sampling filter in the pixel domain. However, in our formulation, we need to ensure additivity of our cost function. Consequently, we will now introduce a simplified up-sampling method, performed in the DCT domain, that will allow us to perform block-wise optimization:

$$f_{\mathrm{up}}^{\mathrm{DCT}}\left(\mathrm{T}_{4x4}\left[\boldsymbol{x}\right]\right) = \begin{pmatrix} \mathrm{T}_{4x4}\left[\boldsymbol{x}\right] & 0 \\ 0 & 0 \end{pmatrix} \tag{3.9}$$

where $\boldsymbol{x}$ is the input 4x4 block in the pixel domain and $f_{up}(\cdot)$ outputs an 8x8 block of DCT coefficients. This remark, naturally, does not affect the optimization of quality scalability scenario where the up-sampling of residual prediction signal does not take place.

The distortion for a single 8x8 block in the enhancement layer can now be written as follows:

$$\begin{aligned} D_{E,8x8} &= \left|\left|\left[(\boldsymbol{X}-\boldsymbol{P}) - \mathrm{T}_{8x8}^{-1}(f_{\mathrm{up}}^{\mathrm{DCT}}(\boldsymbol{u}\cdot q))\right] - \mathrm{T}_{8x8}^{-1}(\boldsymbol{U}\cdot Q)\right|\right|^2 & (3.10) \\ &= \left|\left|\left[\boldsymbol{Z} - \mathrm{T}_{8x8}^{-1}\begin{pmatrix} \boldsymbol{u}\cdot q & 0 \\ 0 & 0 \end{pmatrix}\right] - \mathrm{T}_{8x8}^{-1}(\boldsymbol{U}\cdot Q)\right|\right|^2 & (3.11) \end{aligned}$$

In order to simplify the above expression and make it more readable, let us denote then $\boldsymbol{Z} - \mathrm{T}_{8x8}^{-1}(\boldsymbol{U}\cdot Q)$ by $\boldsymbol{Z'}$.

$$D_{E,8x8} = \left|\left|\boldsymbol{Z'} - \mathrm{T}_{8x8}^{-1}\begin{pmatrix} \boldsymbol{u}\cdot q & 0 \\ 0 & 0 \end{pmatrix}\right|\right|^2$$

DCT transform maintains Euclidean distance, hence we can apply it to the above expression and obtain the following result:

$$\begin{aligned} D_{E,8x8} &= \left|\left|\mathrm{T}_{8x8}\left(\boldsymbol{Z'}\right) - \mathrm{T}_{8x8}\left(\mathrm{T}_{8x8}^{-1}\left(\begin{bmatrix} \boldsymbol{u}\cdot q & 0 \\ 0 & 0 \end{bmatrix}\right)\right)\right|\right|^2 & (3.12) \\ &= \left|\left|\mathrm{T}_{8x8}\left(\boldsymbol{Z'}\right) - \begin{bmatrix} \boldsymbol{u}\cdot q & 0 \\ 0 & 0 \end{bmatrix}\right|\right|^2 & (3.13) \\ &= \sum_{i\in\mathrm{UL}} \left|\left|\mathrm{T}_{8x8}\left(\boldsymbol{Z'}\right)_i - \left(\boldsymbol{u}\cdot q\right)_i\right|\right|^2 + \sum_{i\in\mathrm{UR,LL,LR}} \left|\left|\mathrm{T}_{8x8}\left(\boldsymbol{Z'}\right)_i\right|\right|^2 & (3.14) \end{aligned}$$

where UL, UR, LL, LR denote sets of pixels belonging to upper-left, upper-right, lower-left, lower-right 4x4 block of the given 8x8 block, respectively:

$$X_{8,8} = \begin{bmatrix} [\mathrm{UL}]_{4,4} & [\mathrm{UR}]_{4,4} \\ [\mathrm{LL}]_{4,4} & [\mathrm{LR}]_{4,4} \end{bmatrix} \tag{3.15}$$

Equation 3.14 shows clearly the advantage of selecting the up-sampling method as 3.9. At this point there is a one-to-one correspondence between the coefficients of the upper-left quadrant of the 8x8 enhancement layer block and the 4x4 base layer block.

Based on our discussion above, the total cost for encoding one frame in both the base and enhancement layer can be expressed as follows:

$$J = D_B + \alpha \cdot D_E + \lambda \left( R_B + R_E \right) \tag{3.16}$$

which can be now expanded as follows:

$$\begin{aligned} J \;=\; & \left|\left|\boldsymbol{z} - \mathrm{T}^{-1}(\boldsymbol{u} \cdot q)\right|\right|^2 + \alpha \left[ \sum_{i \in \mathrm{UL}} \left|\left|\mathrm{T}_{8x8} \left(\boldsymbol{Z'}\right)_i - \left(\boldsymbol{u} \cdot q\right)_i\right|\right|^2 \right. & (3.17) \\ & \left. + \sum_{i \in \mathrm{UR,LL,LR}} \left|\left|\mathrm{T}_{8x8} \left(\boldsymbol{Z'}\right)_i\right|\right|^2 \right] + \lambda \left( R_B + R_E \right) & (3.18) \end{aligned}$$

where:

$$\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{p}_m$$
$$\boldsymbol{Z'} = (\boldsymbol{X} - \boldsymbol{P}) - \mathrm{T}_{8x8}^{-1}(\boldsymbol{U} \cdot Q)$$
$$R_B = R_B(\boldsymbol{u}, \boldsymbol{m}, \boldsymbol{f}, \boldsymbol{v})$$
$$R_E = R_E(\boldsymbol{U}, \boldsymbol{M}, \boldsymbol{F}, \boldsymbol{V})$$

The focus of our discussion however, is the quantization operation only and in our algorithm design we assume that the prediction modes, reference frame indices, motion vectors and quantization step-size are given and are not the object of optimization.

It becomes obvious, when looking at the cost function in Equation 3.16 (and its subsequent expansion), that the solution to the problem is an iterative optimization algorithm. The general procedure is shown in Algorithm 1.

In the above algorithm, the first step carries the bulk of our contribution, as finding the optimal quantized transform coefficients for the base layer will include taking into account the impact on the subsequent coding of the enhancement layer. Once the encoder

**Algorithm 1 : General optimization algorithm**

```
1. Fix $\boldsymbol{U}$, solve the following optimization problem:
       $\min_{\boldsymbol{u}} J$
2. Fix $\boldsymbol{u}$, solve the following optimization problem:
       $\min_{\boldsymbol{U}} J$
3. Repeat until the decrement of the cost is less than a prescribed threshold.
```

proceeds to encode the enhancement layer frame, the procedure becomes analogical to the optimization described in Chapter 2, Section 2.3 and, more exhaustively, in [13, 30, 29]. Subsequently, the encoder will need to return to the base layer while still processing the same frame, in order to update the enhancement layer residuals for distortion calculation (see Equation 3.8). Those steps will be performed iteratively until the cost decrement reaches a predefined threshold.

At this point it is not clear whether the described optimization algorithm will converge to the global optimum solution for both layers at the same time. However, the iterative optimization algorithm does converge in the sense that the calculated rate distortion cost is decreasing at each iteration step. This observation is analogical to that of the original SDQ proposed for the H.264 standard [29].

Let us now proceed to presenting specific optimization algorithms, tailored to particular scenarios (spatial and quality scalability) which use the above approach to solve the problem of optimal SVC quantization for each case.

## 3.3 Iterative Inter-layer Optimization Algorithm

In this section we will present the optimization algorithm for two scenarios: spatial scalability and SNR (quality) scalability. In general, our algorithm follows the path outlined by the original SDQ algorithm implemented in the H.264 standard (see Chapter 2, Section 2.3 for details). As discussed in the previous section though, it does include several new elements associated with the scalability feature and utilizing the existing inter-layer dependency resulting from inter-layer residual prediction.

The SVC encoder, when processing an input video sequence, does so in a frame-by-frame manner and each frame is being encoded in each supported layer. Due to this structure, our algorithm needs to involve iterative optimization of one frame in each layer in an alternating fashion. Additionally, an initialization phase is required in order to learn all residual values prior to the inter-layer optimization. Consequently, the process will include the initialization phase, in which the optimization is constrained to one layer only and allows for obtaining all required transform coefficients, and the actual inter-layer optimization phase where all necessary information is available when processing the base layer. During the latter phase, the process takes into account the impact of the base layer on the enhancement layer, making it the core component of our contribution.

The following subsections describe specific algorithms tailored to the particular scalability scenario (spatial and quality).

### 3.3.1 Spatial Scalability Scenario

Let us assume encoding of one P frame for two input sequences with resolution ratio 2:1 (dyadic case). Block size is assumed to be 4x4 for base layer and 8x8 for enhancement layer. Due to the difference in resolution an up-sampling routine is present and taken into account during optimization.

The proposed alternating optimization algorithm is summarized as Algorithm 2:

**Remark:** Steps 1 and 2 constitute one iteration (the aforementioned initialization phase), so do steps 3 and 4 (optimization with inter-layer dependency).

In the following chapter we will describe the implementation of Algorithm 2 and present the obtained experimental results.

### 3.3.2 Quality Scalability Scenario

Let us assume encoding of one P frame for two input sequences with the same resolution. Block size is assumed to be 4x4 for both base layer and enhancement layer. Since resolution

**Algorithm 2 : Optimal Soft Decision Quantization for Double Layer SVC**

```
1. Compute optimal quantized residual for the base layer ($u^0$) by:
       $\min_{\boldsymbol{u}^0} d(\mathrm{T}_{4\times4}^{-1}(\boldsymbol{u}^0 \cdot q), \boldsymbol{z}) + r(\boldsymbol{u}^0)$
2. Compute optimal quantized residual for the enhancement layer ($U^0$) by:
       $\min_{\boldsymbol{U}^0} d(\mathrm{T}_{8\times8}^{-1}(\boldsymbol{U}^0 \cdot Q), \boldsymbol{Z}^0) + r(\boldsymbol{U}^0)$
3. For a given $\boldsymbol{U}^{t-1}$ compute $\boldsymbol{u}^t$ by:
       $\min_{\boldsymbol{u}^t} d(\mathrm{T}_{4\times4}^{-1}(\boldsymbol{u}^t \cdot q), \boldsymbol{z}) + \alpha \cdot d(\mathrm{T}_{8\times8}^{-1}(\boldsymbol{U}^{t-1} \cdot Q), \boldsymbol{Z} - \mathrm{T}_{8\times8}^{-1}(f_{up}(\boldsymbol{u}^t \cdot q))) + \lambda \cdot r(\boldsymbol{u}^t)$
4. Compute $\boldsymbol{U}^t$ by:
       $\min_{\boldsymbol{U}^t} d(\mathrm{T}_{8\times8}^{-1}(\boldsymbol{U}^t \cdot Q), \boldsymbol{Z}^t) + \lambda \cdot r(\boldsymbol{U}^t)$
5. Repeat steps 3 and 4 until the decrement of the RD cost is less than a prescribed threshold.
```

is the same for both layers, the up-sampling routine has been omitted from the algorithm. Other than that, the algorithm is analogical to that in Section 3.3.1

The proposed alternating optimization algorithm is summarized as Algorithm 3.

**Algorithm 3 : Optimal Soft Decision Quantization for Double Layer SVC**

1. Compute optimal quantized residual for the base layer ($\boldsymbol{u}^0$) by:

$$\min_{\boldsymbol{u}^0} d(\mathrm{T}_{4\times4}^{-1}(\boldsymbol{u}^0 \cdot q), \boldsymbol{z}) + \lambda \cdot r(\boldsymbol{u}^0)$$

2. Compute optimal quantized residual for the enhancement layer ($\boldsymbol{U}^0$) by:

$$\min_{\boldsymbol{U}^0} d(\mathrm{T}_{4\times4}^{-1}(\boldsymbol{U}^0 \cdot Q), \boldsymbol{Z}^0) + \lambda \cdot r(\boldsymbol{U}^0)$$

3. For a given $\boldsymbol{U}^{t-1}$ compute $\boldsymbol{u}^t$ by:

$$\min_{\boldsymbol{u}^t} d(\mathrm{T}_{4\times4}^{-1}(\boldsymbol{u}^t \cdot q), \boldsymbol{z}) + \alpha \cdot d(\mathrm{T}_{4\times4}^{-1}(\boldsymbol{U}^{t-1} \cdot Q), \boldsymbol{Z} - \mathrm{T}_{4\times4}^{-1}(\boldsymbol{u}^t \cdot q)) + \lambda \cdot r(\boldsymbol{u}^t)$$

4. Compute $\boldsymbol{U}^t$ by:

$$\min_{\boldsymbol{U}^t} \beta \cdot d(\mathrm{T}_{4\times4}^{-1}(\boldsymbol{U}^t \cdot Q), \boldsymbol{Z}^t) + \lambda \cdot r(\boldsymbol{U}^t)$$

5. Repeat steps 3 and 4 until the decrement of the RD cost is less than a prescribed threshold.

**Remark:** Steps 1 and 2 constitute one iteration (the aforementioned initialization phase), so do steps 3 and 4 (optimization with inter-layer dependency).

In the following chapter we will describe the implementation of Algorithm 3 and present the obtained experimental results.

# Chapter 4

# Experimental Results

In this chapter we are presenting experimental results obtained using a practical implementation of the algorithm from Chapter 3. In the following sections we will provide more details on the programming process as well as what assumptions were made during our experiments. The final sections contain numerical results in terms of rate and distortion as well as comparison with benchmark results. Each section is supplemented by discussion of obtained results.

## 4.1 Testing Methodology

In this section we will describe the methodology of our testing process, namely comment on the implementation and state our underlying assumptions.

### 4.1.1 Implementation

The proposed solution was implemented within the SVC Reference Software, commonly referred to as the Joint Scalable Video Model [2]. A substantial modification to the core encoding component was necessary in order to facilitate iterative inter-layer optimization. A combination of buffers and data structures was used for preserving the encoder state from iteration to iteration.

The whole implementation process was conducted in C++ programming language on a Linux operating system. KDevelop [3] Integrated Development Environment (IDE) was used in order to streamline the process. Subsequent testing has revealed, that implemented solution has significant computational complexity and code optimization will be desirable in the future.

### 4.1.2 Assumptions and Limitations

The JSVM software and SVC in general offer a large number of features and allow for providing numerous encoding parameters. When implementing our solution and performing the experiments, we were aiming at simplifying the testing procedure as much as possible without sacrificing the accuracy and validity of the results. Therefore, the following assumptions and limitations were introduced:

- As it was mentioned before, throughout this thesis we have focused on a dyadic case, i.e. one base layer and one enhancement layer. It has to be noted however, that the proposed solution can be easily extended to a multi-layer scenario with an arbitrary number of spatial and quality layers. It would however impose a significant computational burden on the machine performing the encoding.

- The motion prediction mode has been limited to 16x16 only in order to reduce the amount of computation necessary to evaluate all allowed prediction modes. While reducing the resulting video quality, this limitation does not affect the validity of the results, it serves merely the purpose of significantly reducing the computation time. From the programming point of view, such limitation also greatly simplifies the implementation of the iterative optimization scheme. In the future, the solution can be conceivably extended to support more prediction modes, although that would lead to a more complicated optimization process of collocated base and enhancement layer blocks.

- During the experiments conducted, only QCIF-CIF (spatial scenario) and CIF-CIF (quality scenario) sequence pairs were used, again, in order to reduce the computation time. In the future, further experiments with higher resolution sequences (including high definition, 1080p content) can be performed for deeper performance analysis.

- I-P-P-P... frame sequence was used for encoding with only P frames being optimized using SDQ. During every experiment I frames are encoded using standard quantization process as described by the standard specification.

- In all performed experiments, the adaptive inter-layer prediction is used, i.e. the up-sampled base layer motion prediction and residuals are applied selectively in the enhancement layer based on a rate-distortion criterion. In other words, the inter-layer prediction is utilized only if it is beneficial from the performance point of view.

- In most of the experiments (unless otherwise noted) 50 frames were processed and PSNR/rate results were averaged. The values for I frames have been excluded from calculations in order to prevent result skewing (I frames are always hard-decision quantized).

### 4.1.3 Test Cases and Result Presentation

Before we present our experimental results, we need to stress, that SVC poses a significant challenge when it comes to comparing the efficiency and R-D performance of its improvements with the unmodified version. This statement is especially true when considering the impact on potential commercial applications. As an example, let us consider here a video broadcasting scenario. Such case is one of the potentially successful ways of implementing SVC in a real world application. When analyzing the situation from the service provider point of view, the parameter of focus is the overall codec efficiency and minimizing the total bitrate becomes the goal (by total bitrate we understand the bandwidth consumed by the entire scalable bitstream which contains all the layers within itself). On the other hand, when looking from the end user point of view, the parameter of focus now becomes the bandwidth necessary to receive the video sequence with desired quality (i.e. the sum of bitrates of all layers up to the one that matches the user's demand). It is of no significance to that particular user that the bitstream they receive will contain all lower layers as well. From that perspective a single-layer coding scheme which will provide better efficiency for the requested quality will simply be more attractive regardless of the fact that it will no longer provide the scalability feature. We have described this example in order to visualize the challenges that are posed by SVC in terms of comparing it to other potentially viable solutions. It also serves to show that the content and service providers face significant trade-offs when choosing the best method of content processing and delivery.

## 4.2 Spatial Scalability Scenario

In this section we will present experimental results obtained using the encoder running in the spatial scalability mode. For simplicity, we assume a two-layer scenario with one base layer having QCIF resolution (176x144) and one enhancement layer having CIF resolution (352x288).

### 4.2.1 Testing Conditions and Assumptions

In order to obtain a series of results, the encoder was run for a range of QP parameter values. In each case, the parameter values for base layer and enhancement layer were identical. As mentioned in Chapter 3, a simplified up-sampling method was implemented within the encoder. However, tests have shown that the performance difference between the original bi-cubic interpolation filter and the simple DCT-based up-sampling is almost negligible.

### 4.2.2 Results

Tables 4.1, 4.2 and 4.3 show the efficiency comparison between unmodified SVC encoder, SVC encoder with standard SDQ (without layer interaction) and our iterative SDQ implementation for three different quality levels. Included are: the total numbers of bits required to code a particular frame in both layers as well as the bit savings achieved by using SDQ with and without interaction. For brevity, only the results for first five P frames are presented, however the average bit savings is calculated over the entire 50 P frames encoded. All tables show, that our iterative SDQ optimization achieves significant bit savings over the original SVC implementation when spatial scalability is used. On average, the number of bits required has been reduced by up to 8% with single frames achieving over 9% bit savings. Additionally, a clear advantage over the "no interaction" SDQ implementation can be observed, as our approach yields additional bit savings of 2-2.5%.

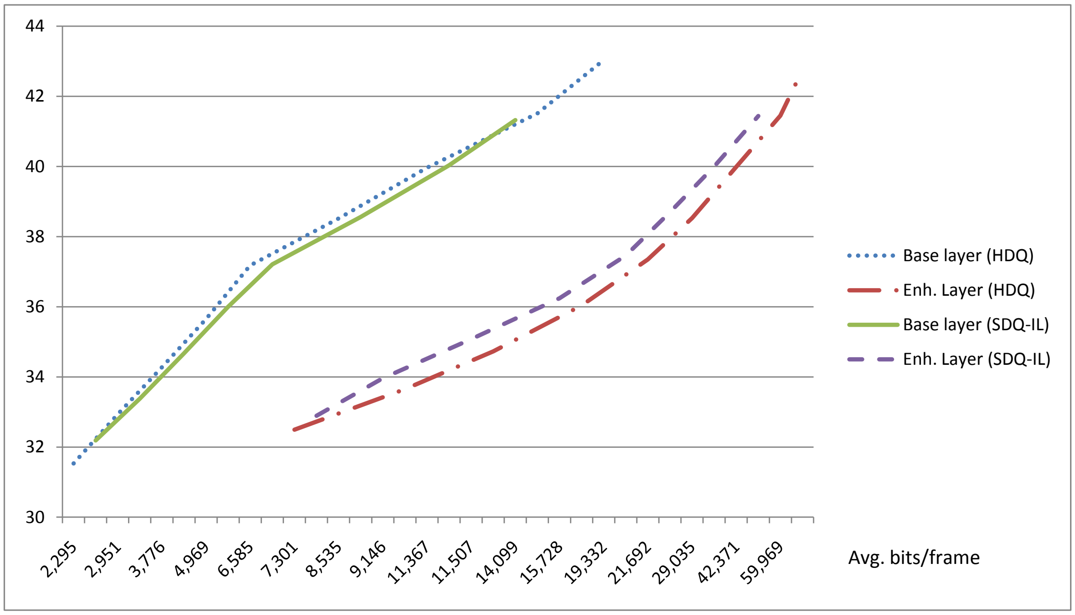

Figure 4.1: RD performance comparison of “Foreman” in the spatial scalability scenario

Figure 4.1 shows the RD curves for both implementations for base and enhancement layer. It can be clearly seen that the iterative SDQ optimization allows to trade a very small number of bits in the base layer for achieving significant savings in the enhancement layer. That leads to the already mentioned overall bit savings when considering the entire scalable bitstream.

Figures 4.2, 4.3, 4.4 compare the total number of bits spent on each frame for each quality level for the three aforementioned cases (HDQ: standard SVC, SDQ: SVC with SDQ, and SDQ-IL: SVC with SDQ and inter-layer optimization).

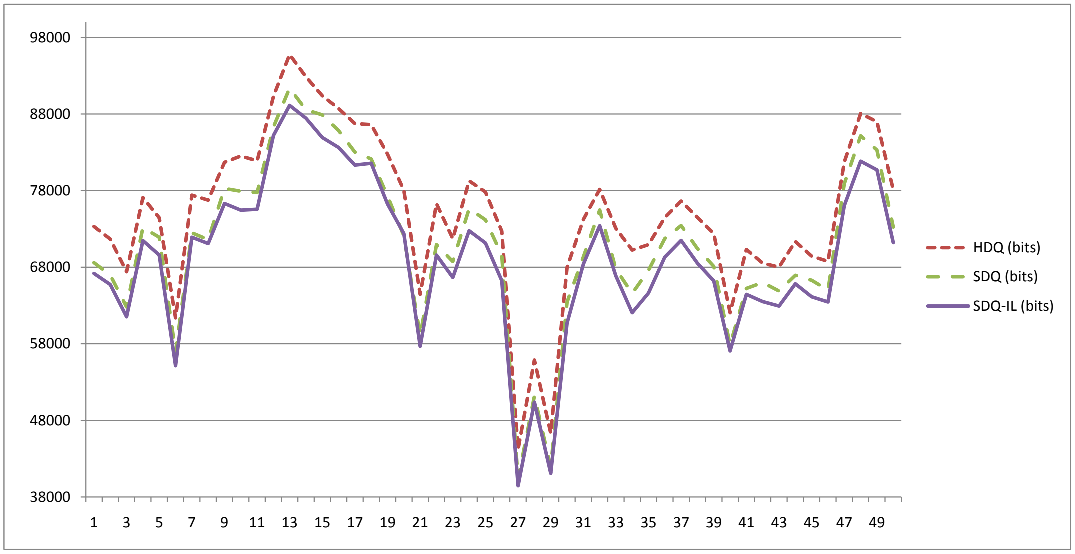

Figure 4.2: Comparison of total number of bits for each frame in the spatial scalability scenario. Quality was fixed at 41.3 and 41.4 dB PSNR for base and enhancement layer, respectively.

Table 4.1: Total bitrate comparison for first five frames of “Foreman” sequence in the spatial scalability scenario. Quality was fixed at 41.3 and 41.4 dB PSNR for base and enhancement layer, respectively

| **P Frame #** | **HDQ [b]** | **SDQ [b]** | **Savings [%]** | **SDQ-IL [b]** | **Savings [%]** |
|---|---|---|---|---|---|
| 1 | 73328 | 68608 | 6.44 | 67202 | 8.35 |
| 2 | 71680 | 66969 | 6.57 | 65747 | 8.28 |
| 3 | 67408 | 62820 | 6.81 | 61523 | 8.73 |
| 4 | 77096 | 73166 | 5.10 | 71487 | 7.28 |
| 5 | 74464 | 71907 | 3.43 | 69583 | 6.55 |
| **Average (50 fr.)** | | | 5.58 | | 7.93 |

Table 4.2: Total bitrate comparison for first five frames of “Foreman” sequence in the spatial scalability scenario. Quality was fixed at 37.2 and 37.5 dB PSNR for base and enhancement layer, respectively

| **P Frame #** | **HDQ [b]** | **SDQ [b]** | **Savings [%]** | **SDQ-IL [b]** | **Savings [%]** |
|---|---|---|---|---|---|
| 1 | 27088 | 25298 | 6.61 | 24800 | 8.45 |
| 2 | 29392 | 27797 | 5.43 | 27373 | 6.87 |
| 3 | 26800 | 25439 | 5.08 | 24929 | 6.98 |
| 4 | 29896 | 28066 | 6.12 | 27557 | 7.82 |
| 5 | 30408 | 28678 | 5.69 | 28157 | 7.40 |
| **Average (50 fr.)** | | | 5.24 | | 7.65 |

Table 4.3: Total bitrate comparison for first five frames of “Foreman” sequence. Quality was fixed at 32.2 and 32.9 dB PSNR for base and enhancement layer, respectively

| **P Frame #** | **HDQ [b]** | **SDQ [b]** | **Savings [%]** | **SDQ [b]** | **Savings [%]** |
|---|---|---|---|---|---|
| 1 | 8696 | 8074 | 7.15 | 7907 | 9.07 |
| 2 | 10368 | 9780 | 5.67 | 9603 | 7.38 |
| 3 | 10984 | 10332 | 5.94 | 10123 | 7.84 |
| 4 | 11440 | 10930 | 4.46 | 10715 | 6.34 |
| 5 | 10688 | 10118 | 5.33 | 9947 | 6.93 |
| **Average (50 fr.)** | | | 5.64 | | 7.31 |

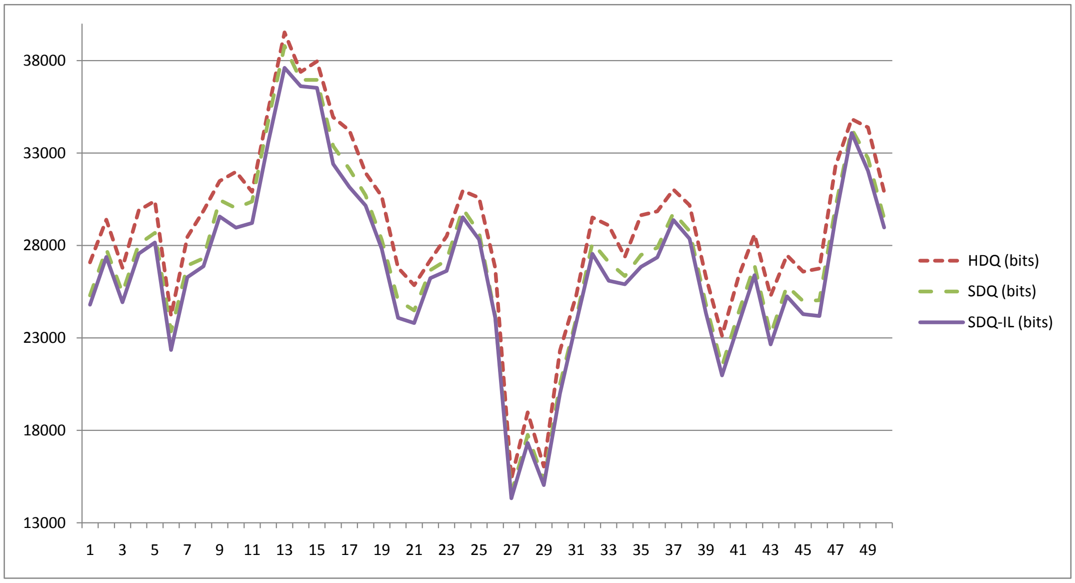

Figure 4.3: Comparison of total number of bits for each frame in the spatial scalability scenario. Quality was fixed at 37.2 and 37.5 dB PSNR for base and enhancement layer, respectively.

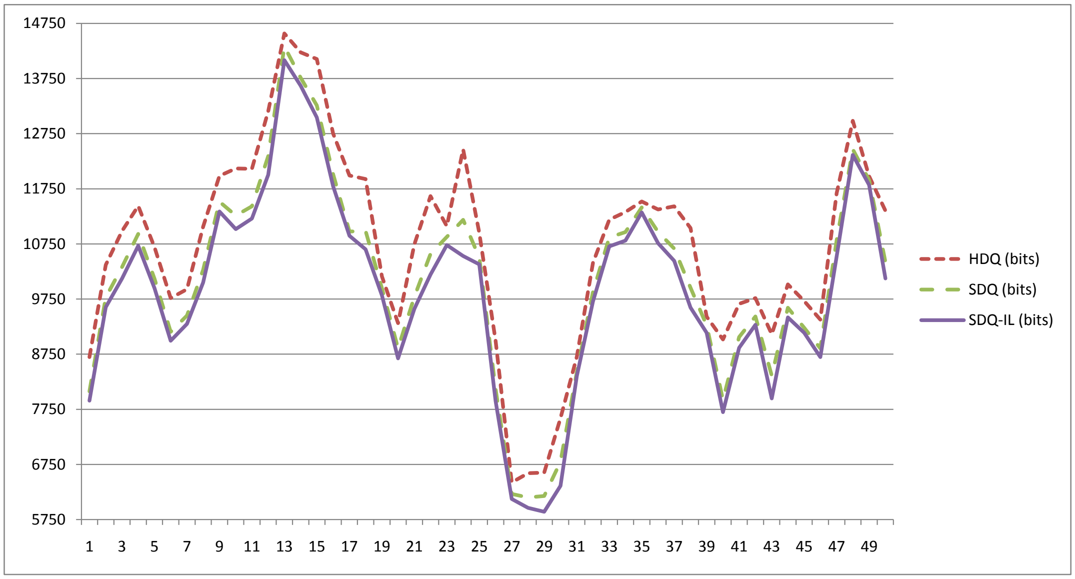

Figure 4.4: Comparison of total number of bits for each frame in the spatial scalability scenario. Quality was fixed at 32.2 and 32.9 dB PSNR for base and enhancement layer, respectively.

## 4.3 Quality Scalability Scenario

In this section we will present experimental results obtained using the encoder running in the quality scalability mode. For simplicity, we assume a two-layer scenario with both layers having CIF resolution (352x288).

### 4.3.1 Testing Conditions and Assumptions

In order to obtain a series of results, the encoder was run for a range of QP parameter values. In each case, if the parameter value for base layer was set as $QP$, the value for enhancement layer was set as $QP + 4$. As mentioned earlier in the thesis, the DCT transform was fixed at 4x4 which allowed to maintain the one-to-one relationship between blocks in different layers.

### 4.3.2 Results

Tables 4.4, 4.5 and 4.6 show the efficiency comparison between unmodified SVC encoder and our iterative SDQ implementation for three different quality levels. Included are: the total numbers of bits required to code a particular frame in both layers as well as the bit savings achieved by using our approach. For brevity, only the results for first five P frames are presented, however the average bit savings is calculated over the entire 50 P frames encoded. All tables show, that our iterative SDQ optimization achieves significant bit savings over the original SVC implementation when quality scalability is used. On average, the number of bits required has been reduced by up to 9% with single frames achieving over 14% bit savings.

Figure 4.1 shows the RD curves for both implementations for base and enhancement layer. It can be clearly seen that the iterative SDQ optimization allows significant savings in the enhancement layer (especially for higher bitrates) while still achieving some gain in the base layer. That leads to the already mentioned overall bit savings when considering the entire scalable bitstream.

Figure 4.6 compares the total number of bits spent on each P frame.

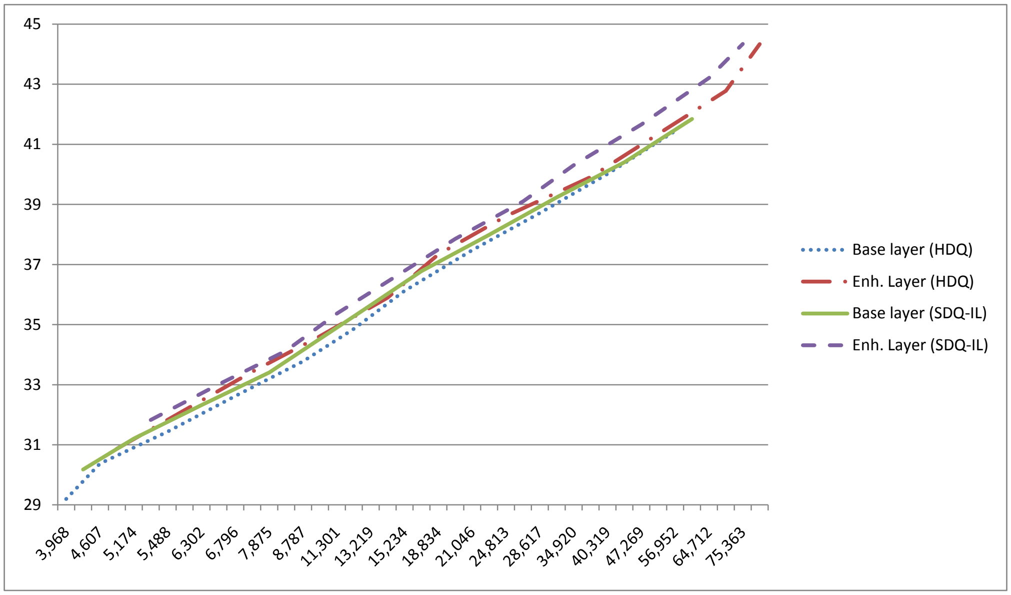

Figure 4.5: RD performance comparison of "Foreman" in the SNR scalability scenario.

Table 4.4: Total bitrate comparison for first five frames of "Foreman" sequence. Quality was fixed at 41.8 and 44.3 dB PSNR for base and enhancement layer, respectively.

| **P Frame No.** | **HDQ [bits]** | **SDQ [bits]** | **Bit savings [%]** |
|---|---|---|---|
| 1 | 140224 | 128248 | 8.54 |
| 2 | 142400 | 132768 | 6.76 |
| 3 | 137776 | 127240 | 7.65 |
| 4 | 154160 | 141608 | 8.14 |
| 5 | 152120 | 139816 | 8.09 |
| **Average (50 fr.)** | | | 8.50 |

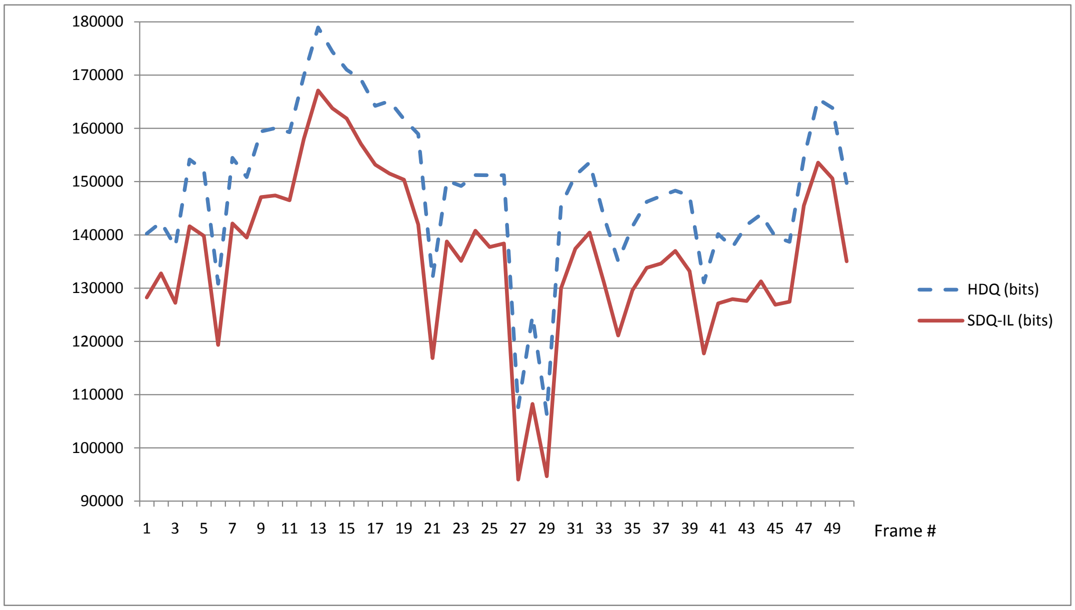

Figure 4.6: Comparison of total number of bits for each frame in the quality scalability scenario. Quality was fixed at 41.8 and 44.3 dB PSNR for base and enhancement layer, respectively.

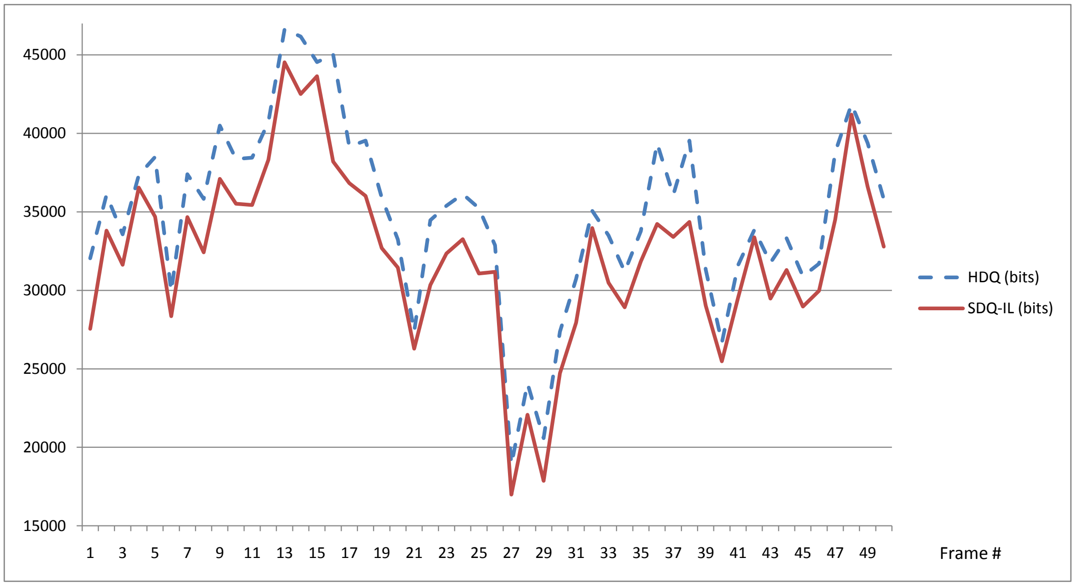

Figure 4.7: Comparison of total number of bits for each frame in the quality scalability scenario. Quality was fixed at 35.6 and 37.8 dB PSNR for base and enhancement layer, respectively.

Table 4.5: Total bitrate comparison for first five frames of “Foreman” sequence. Quality was fixed at 35.6 and 37.8 dB PSNR for base and enhancement layer, respectively.

| **P Frame No.** | **HDQ [bits]** | **SDQ [bits]** | **Bit savings [%]** |
|---|---|---|---|
| 1 | 32040 | 27544 | 14.03 |
| 2 | 36088 | 33808 | 6.32 |
| 3 | 33568 | 31632 | 5.77 |
| 4 | 37336 | 36544 | 2.12 |
| 5 | 38504 | 34696 | 9.89 |
| **Average (50 fr.)** | | | 7.60 |

Table 4.6: Total bitrate comparison for first five frames of “Foreman” sequence. Quality was fixed at 30.1 and 31.8 dB PSNR for base and enhancement layer, respectively.

| **P Frame No.** | **HDQ [bits]** | **SDQ [bits]** | **Bit savings [%]** |
|---|---|---|---|
| 1 | 8336 | 7472 | 10.36 |
| 2 | 10576 | 9760 | 7.72 |
| 3 | 10568 | 9808 | 7.19 |
| 4 | 12416 | 10864 | 12.50 |
| 5 | 11072 | 10392 | 6.14 |
| **Average (50 fr.)** | | | 8.95 |

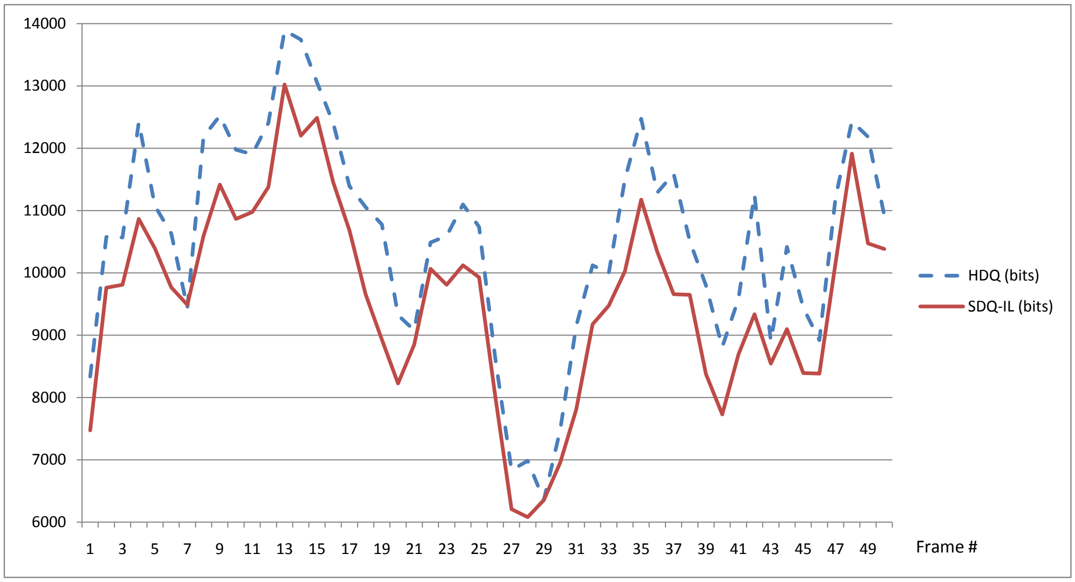

Figure 4.8: Comparison of total number of bits for each frame in the quality scalability scenario. Quality was fixed at 30.1 and 31.8 dB PSNR for base and enhancement layer, respectively.

# Chapter 5

# Conclusions and Future Work

In this Chapter we conclude our discussion of Scalable Video Coding and our contribution towards increasing its performance. We summarize the content of this thesis, provide our final remarks and outline possible future research directions.

## 5.1 Conclusions

In this thesis we first introduce the concept of scalability in video coding and its current position in the industry, especially after the introduction of the scalable extension to the state-of-the-art H.264 standard. We briefly outline the benefits of the new standard and mention the main drawback restricting it from wider adoption.

In Chapter 2, Sections 2.1 and 2.2 we then proceed to describe the H.264/AVC video coding standard and its scalable extension, Scalable Video Coding (SVC). In the latter section, the inter-layer prediction mechanisms present in SVC are brought into highlight and analyzed. It has also been pointed out that even with all those mechanisms enabled there is still a performance gap of roughly 10% separating the SVC encoder from a single-layer H.264 encoder.

Since our work is mainly focused on extending the Soft Decision Quantization concept into the realm of Scalable Video Coding, that revolutionary approach to scalar quantization is covered in detail in Chapter 2, Section 2.3. Its application in the scalable scenario was never considered though, and that fact makes this contribution quite novel. Due to the existing inter-layer prediction mechanisms, there is much larger room for optimization and we have demonstrated that it is indeed a viable solution.

Based on the prior systematic discussion, a joint inter-layer optimization problem is formulated. An iterative, SDQ-based method to solve this problem has been proposed. Particularly, two algorithms for optimizing the spatial scalability and quality scalability scenarios have been outlined.

Experimental results have shown that there is performance gain coming from taking into account the impact that the base layer has on the enhancement layer. Even with targeting only the quantization component of the encoder we have managed to significantly narrow the gap separating SVC from H.264. We have demonstrated that allowing interaction between layers during optimization yields an additional 2-2.5% over independently applying SDQ in each layer. It is therefore our intuition that extending the optimization towards mode, motion vector and quantization step-size selection will yield additional gain which will aid in bringing these two video coding standards almost to parity in terms of coding efficiency.

## 5.2 Future Work

Looking to the future of Scalable Video Coding in general and the application of Soft Decision Quantization in particular, there are several interesting concepts that can be further investigated. Those include, but are not limited to, the inter-layer optimization of motion compensation as well as improved down- and up-sampling method. The following subsections describe these concepts in more detail.

### 5.2.1 Motion Prediction

The algorithm described in this thesis focuses on the quantization stage of the encoding process. We have successfully shown that optimal selection of quantized coefficients leads to improved rate-distortion performance of the entire scalable coding scheme. This concept can be quite easily extended to the motion compensation stage. As Yang and Yu have shown in [30], such joint optimization has resulted in gains of 12% when implemented in the H.264 coding standard and by utilizing the inter-layer correlation present in SVC we strongly believe that potential gains will be more substantial.

Additionally, it is conceivable to improve on the existing inter-layer motion prediction algorithm in the spatial scalability scenario. The current implementation assumes simple motion vector scaling by the factor equal to the ratio of frame resolutions. While straightforward, there is no guarantee that this approach will yield a good approximation of the upper-layer motion vector, especially in cases when the base layer lacks most of the fine detail. Consequently, one can develop an adaptive motion vector up-sampling method which would also take into account the surrounding motion field and attempt at more accurately estimating the upper-layer motion.

### 5.2.2 Down- and up-sampling

During our study of the Scalable Video Coding standard it came to our attention that the performance of the residual inter-layer mechanism is largely dependent on the up-sampling method applied. The currently implemented filter does not take into account the video content nor is it in any other way adaptable to particular coding conditions. It is therefore conceivable to design an up-sampling filter which would perform the residual signal upscaling in an adaptive manner, tailored to the specifics of the video content and provide flexibility during encoding.

# References

[1] Hulu. http://www.hulu.com.

[2] Joint Scalable Video Model - SVC Reference Software. http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm.

[3] Kdevelop. http://www.kdevelop.org/.

[4] The Scalable Video Coding Amendment of the H.264/AVC Standard. http://ip.hhi.de/imagecom_G1/savce/.

[5] Youtube. http://www.youtube.com.

[6] *ISO/IEC 13818-2:2000 Generic coding of moving pictures and associated audio information: Video*, November 1994.

[7] *ITU-T H.263 : Video coding for low bit rate communication*, March 1996.

[8] *ISO/IEC 14496-2:1999 Coding of audio-visual objects - Part 2: Visual*, 1999.

[9] *ISO/IEC 14496-10:2003 Coding of Audio-Visual Objects  Part 10: Advanced Video Coding*, July 2003.

[10] Ken Cabeen and Peter Gent. Image Compression and the Discrete Cosine Transform.

[11] Rakesh Dugad and Narendra Ahuja. A Fast Scheme for Downsampling and Upsampling in the DCT domain. In *Proceedings. 1999 International Conference on Image Processing*, October 1999.

[12] En hui Yang and Xiang Yu. On Joint Optimization of Motion Compensation, Quantization and Baseline Entropy Coding in H.264 with Complete Decoder Compatibility. In *Proceedings of ICASSP*, 2005.

[13] En hui Yang and Xiang Yu. Soft Decision Quantization for H.264 With Main Profile Compatibility. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(1):122 –127, January 2009.

[14] En hui Yang and Xiang Yu. Soft Decision Quantization for H.264 With Main Profile Compatibility. *IEEE Transactions On Circuits and Systems for Video Technology*, 19(1), January 2009.

[15] En hui Yang, Zhen Zhang, and T. Berger. Fixed-slope universal lossy data compression. *Information Theory, IEEE Transactions on*, 43(5):1465 –1476, September 1997.

[16] Jens-Rainer Ohm. Standardization in JVT: Scalable Video Coding. Presentation at ITU-T VICA Workshop, July 2005.

[17] Iain Richardson. H.264 / MPEG-4 Part 10 White Paper: Variable-Length Coding. Technical report, VCodex, October 2002.

[18] Iain Richardson. H.264 / MPEG-4 Part 10 White Paper: Transform and Quantization. Technical report, VCodex, March 2003.

[19] Iain Richardson. White Paper: 4x4 Transform and Quantization in H.264/AVC. Technical report, VCodex, April 2009.

[20] Ralf Schaefer, ThomasWiegand, and Heiko Schwartz. The Emerging H.264/AVC Standard. *EBU Technical Review*, January 2003.

[21] Heiko Schwarz, Tobias Hinz, Detlev Marpe, and Thomas Wiegand. Constrained Inter-Layer Prediction for Single-Loop Decoding in Spatial Scalability. In *ICIP 2005. IEEE International Conference on Image Processing*, September 2005.

[22] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. Overview of the Scalable H.264/MPEG4-AVC Extension. In *Proceedings of the IEEE International Conference on Image Processing, ICIP 06*, pages 161–164, 2006.

[23] Gary J. Sullivan, Pankaj Topiwala, and Ajay Luthra. The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions. In *SPIE conference on Applications of Digital Image Processing XXVII*, pages 454–474, 2004.

[24] ThomasWiegand, Gary J. Sullivan, Gisle Bjntegaard, and Ajay Luthra. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions On Circuits and Systems for Video Technology*, 13(7), July 2003.

[25] Thomas Wiegand, Heiko Schwarz, Anthony Joch, Faouzi Kossentini, and Gary J. Sullivan. Rate-Constrained Coder Control and Comparison of Video Coding Standards. *IEEE Transactions on Circuits and Systems for Video Technology*, July 2003.

[26] Thomas Wiegand, Gary Sullivan, Julien Reichel, Heiko Schwarz, and Mathias Wien. Joint Draft ITU-T Rec. H.264 — ISO/IEC 14496-10 / Amd.3 Scalable Video Coding. Technical report, JVT, July 2007.

[27] Koen De Wolf. Scalable Video Coding: Prediction of Residual Information. In *Sixth FirW Phd Symposium*, November 2005.

[28] Koen De Wolf, Davy De Schrijver, Jan De Cock, Wesley De Neve, and Rik Van de Walle. Performance Evaluation of Adaptive Residual Interpolation, a Tool for Interlayer Prediction in H.264/AVC Scalable Video Coding. *Lecture Notes in Computer Science*, 1(4522):740–749, 2007.

[29] En-Hui Yang and Xiang Yu. Rate Distortion Optimization in H.264. *Information Theory and its Applications Workshop*, February 2006.

[30] En-Hui Yang and Xiang Yu. Rate Distortion Optimization for H.264 Interframe Coding: A General Framework and Algorithms. *IEEE Transactions on Image Processing*, 16(7):1774 –1784, July 2007.

[31] Xiang Yu, En hui Yang, and Haiquan Wang. Down-Sampling Design in DCT Domain With Arbitrary Ratio for Image/Video Transcoding. *IEEE Transactions On Image Processing*, 18(1), January 2009.