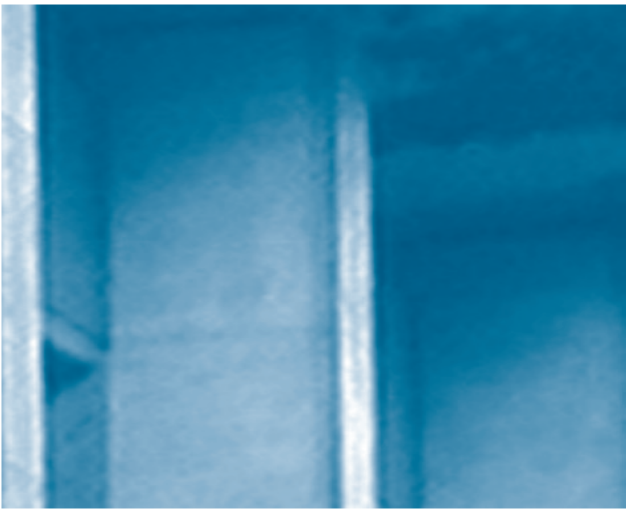


Volume 19

Christian Feldmann

# Low complexity scalable HEVC using single loop decoding

Aachen Series on Multimedia and Communications Engineering

RWTH AACHEN UNIVERSITY

# Low complexity scalable HEVC using single loop decoding

**Von der Fakultät für Elektrotechnik und Informationstechnik der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften genehmigte Dissertation**

vorgelegt von

Diplom-Ingenieur

Christian Feldmann

aus Aachen

Berichter:
Univ.-Prof. Dr.-Ing. Jens-Rainer Ohm
Univ.-Prof. Dr.-Ing. Thomas Sikora

Tag der mündlichen Prüfung: 27.11.2017

Aachen Series on Multimedia and Communications Engineering

Volume 19

**Christian Feldmann**

# Low complexity scalable HEVC using single loop decoding

Shaker Verlag
Aachen 2018

**Bibliographic information published by the Deutsche Nationalbibliothek**
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Zugl.: D 82 (Diss. RWTH Aachen University, 2017)

Printed in Germany.

ISBN 978-3-8440-6285-4
ISSN 1614-7782

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • e-mail: info@shaker.de

# Contents

# Abbreviations

| | |
|---|---|
| AI | All intra (JCT-VC CTC) |
| AMVP | Advanced motion vector prediction |
| AVC | Advanced video coding |
| BD | Bjøntegaard delta measurement |
| BL | Base layer |
| BoG | Break-out group |
| CABAC | Context-based adaptive binary arithmetic coding |
| CGS | Coarse grain scalability |
| CTC | Common testing conditions |
| CTU | Coding tree unit |
| CU | Coding unit |
| DCT | Discrete cosine transform |
| DDR | Double data rate |
| DPB | Decoded picture buffer |
| DPCM | Differential pulse code modulation |
| DST | Discrete sine transform |
| EL | Enhancement layer |
| GOP | Group of pictures |
| HD | High definition |
| HEVC | High efficiency video coding. Also H.265/HEVC. |
| HM | HEVC test model<br>The HEVC reference software implementation. |
| ISO | International organization for standardization |
| ITU | International telecommunication union |
| ITU-T | ITU - Telecommunication Standardization Sector |
| JCT | Joint collaborative team (of ISO and ITU) |
| JCT-VC | Joint Collaborative team on video coding |
| LD | Low delay (JCT-VC CTC) |
| LSB | Least significatn bit |
| LUT | Lookup table |
| MC | Motion compensation |
| MFM | Motion vector field mapping |

| | |
|---|---|
| MGS | Medium-grain quality scalability |
| MPEG | Moving picture experts group |
| MSB | Most significant bit |
| MSE | Mean squared error |
| MV-HEVC | Multiview high efficiency video coding |
| NAL | Network abstraction layer |
| PDF | Probability density function |
| PMF | Probability mass function |
| POC | Picture order count |
| PSNR | Peak signal to noise ratio |
| QP | Quantization parameter |
| RA | Random access (JCT-VC CTC) |
| RAP | Random access point |
| RD | Rate-distortion |
| RDO | Rate-distortion optimization |
| RDOQ | Rate-distortion optimized quantization |
| RExt | Range extensions to HEVC |
| RPL | Reference picture list |
| RPS | Reference picture set |
| SAO | Sample adaptive offset |
| SHM | SHVC test model<br>The SHVC reference software implementation. |
| SHVC | Scalable high efficiency video coding.<br>The scalable extension of H.265/HEVC. |
| SIMD | Single instruction, multiple data |
| SNR | Signal to noise ratio |
| SVC | Scalable video coding<br>The scalable extension of H.264/AVC. |
| TMVP | Temporal motion vector predictor |
| UHD | Ultra high definition |
| VCEG | Visual coding experts group |
| VLC | Variable length code |

# 1 Introduction

The absolute amount of video data that is transmitted over the internet increases each year, while more and more of this data is transmitted over wireless connections which can exhibit highly varying channel conditions. In the area of video coding and transmission, it is a major challenge to provide users with the highest possible video quality even if the available channel bitrate is subject to severe fluctuations. One method in video coding that can help in these situations is scalable video coding in which multiple different versions of the same video sequence are efficiently encoded into a single bitstream. Each layer of the bitstream contains one version of the video which can differ from the other layers in various aspects like spatial resolution, temporal resolution, or quality. Depending on certain criteria like the available bitrate, the decoder can switch between the layers or entire layers can be removed from the bitstream during transmission without an impact on the decodability of the bitstream. Furthermore, the bitstream may offer additional features that aim at a reduction of the decoder complexity or a highly flexible adaption of the bitrate and reconstruction quality.

Among the first and most sophisticated video coding standards to support these types of scalability is the scalable extension to advanced video coding (AVC) called scalable video coding (SVC). For the more recent video coding standard high efficiency video coding (HEVC), such an extension was developed as well. While the scalable high efficiency video coding (SHVC) standard also supports various types of scalability, it was decided in the standardization process to implement scalability without changes to the underlying coding scheme of HEVC. One the one hand, this so called high level syntax only approach is efficient in terms of rate-distortion performance and it can be implemented with only minimal changes to the existing HEVC design. On the other hand, it includes two significant drawbacks: First, while quality scalability is supported, decoding is only possible at very distinct rate points. Unlike in SVC, there is no coding mode available that enables a more flexible adaption to varying channel conditions in terms of quality and bitrate. Second and more importantly, scalability is implemented in such a way that in order to decode the video from a higher layer, all lower layer sequences must be fully reconstructed as well. In turn, the decoder complexity increases almost linearly with the number of layers in the bitstream. For the case of quality scalability, the complexity increase is particularly severe as for every layer, a complete HEVC decoder must operate at the full spatial resolution. This may deem SHVC unattractive for practical applications where the decoder complexity is a limiting factor. Particularly for mobile applications, the strong complexity limits may prohibit the implementation of SHVC altogether. For spatial scalability, the decoder complexity overhead is lower but it may also pose a problem if the resources at the decoder side are limited.

Since this thesis is focused on SVC as well as HEVC and its scalable extension SHVC, the fundamentals on video coding and scalable video coding are detailed in Chapter 2. First, it is shown how scalability for multiple layers is enabled in SVC. Various sophisticated inter layer prediction tools are specified, which allow for an efficient coding of the higher layers by exploiting the lower layer information. With several of the existing lower layer coding tools being altered or newly added, SVC enables decoding of the higher layers at approximately the same complexity as single layer coding using AVC. SHVC, however, enables scalability by essentially running a full HEVC decoder for every quality or spatial layer. Dependencies between the layers are exploited by inter layer prediction where every higher layer is able to access the lower layer reconstruction by means of conventional motion compensation. On the one hand, this approach offers a reasonable rate-distortion performance and can be implemented without any changes to the lower layer coding tools. This "high level syntax only" concept was chosen because it provides certain advantages for practical implementation in hard- and software. In the next part of the chapter, the coding process is detailed and the coding performance of SHVC in comparison to HEVC is analyzed. On the other hand, the multi layer approach has one severe disadvantage: When one of the higher layers is decoded, all lower layers must be fully decoded as well. This results in a significant complexity increase for the multilayer decoder. At the end of the chapter, an analysis of the decoder complexity of SHVC in comparison to HEVC is performed. These results include average measurements as well as a worst case analysis.

In Chapter 3, a flexible inter layer prediction scheme is proposed for SHVC. In combination with some minor changes, it allows for decoding of the higher layer at a significantly reduced complexity compared to conventional SVHC. The basic concept is similar to the one that was introduced in SVC. However, unlike in SVC, the scheme in this thesis is also implemented without any changes to the lower layer syntax or coding tools. Unlike previous coding standards, this retains the "high level syntax only" constraint of SHVC. While this technique increases the coding performance for the higher layer, it also introduces a drift in the lower layer. After the underlying concept is explained in the chapter, it is presented how the coding performance is influenced and that the average as well as the worst case decoder complexity can be considerably reduced. While a certain overhead compared to single layer coding remains, the resulting decoder complexity of the presented multilayer approach is reasonably close to that of single layer coding. For the complexity evaluation, several different values like arithmetic operations and memory access operations are considered. In addition, a visual test is performed in order to further examine the perceptual impact of the drift. The last part of Chapter 3 shows that this approach can achieve a highly flexible adaption of the bitrate and quality.

In SHVC, the higher layer can choose to take the lower layer reconstruction and add a supplemental residual signal using the conventional HEVC coding scheme. For the decoder complexity, this implies that with every layer an additional residual signal may be coded which requires an inverse transform operation at the decoder side. In the arithmetic coding stage, each additional residual signal can require as many context coded bins as a conventional HEVC residual. In general, the transform based approach of SHVC is not well suited for scalability when the higher layers approach a lossless reconstruction quality as the highest layers are mainly comprised of noise. Therefore, an inter layer refinement method is presented in Chapter 4 which allows for a binary mapping between arbitrary quantizer step sizes of two layers. This concept combines an embedded quantization scheme with the quantizer as it is specified in HEVC. In the first part of the chapter it is explained how this mapping is performed and how it can be optimized in a rate-distortion sense with a suitable estimate on the probability distribution of the coefficients. In this way, an existing residual signal from a lower layer can be directly refined in the transform domain. The reconstruction of the residual signal in a higher layer that is being decoded is possible using only one inverse quantization and transformation operation. The refinement information is coded either using conventional context based arithmetic coding or with a low complex entropy coding mode that uses fixed probabilities. The respective probability values are inferred from the mapping and the assumed coefficient distribution. After this method is introduced, the resulting coding performance of the refinement scheme is compared to the conventional SHVC approach of adding another residual signal in the spatial domain. While in SHVC the number of context coded bins per residual signal is constant for every layer, the refinement coding is able to significantly reduce the number of context coded bins per refinement. In addition, with the refinement concept, there is a limit on the maximum number of context coded bins over all layers.

Since the close reloationship of the inter layer prediction method from Chapter 3 and the refinement technique from Chapter 4 suggests a joined approach, they are combined in Chapter 5. As a natural extension of the two individual methods, the new approach is shown to perform well in combination. In addition, it is established that it can further reduce the decoder complexity with regard to the inverse transformations and arithmetic coding.

Chapter 6 concludes this thesis and provides an outlook on potential future research topics in this area.

# 2 Fundamentals

In this chapter, some fundamental topics are explained which lay the foundation for the subsequent chapters. Starting with some basic mathematical principles in Section 2.1, Section 2.2 continues with the fundamentals of video coding. In Section 2.3, some key aspects in the specific video coding standard High Efficiency Video Coding which are relevant for this work are highlighted. Finally, an introduction to scalable video coding (SVC) and Scalable High Efficiency Video Coding is given in Sections 2.4 and 2.5, respectively.

## 2.1 Mathematical Fundamentals

This section highlights some mathematical fundamentals which are relevant in the following work. For a more detailed discussion of these, the reader is referred to [Ohm15; CT91].

### 2.1.1 Random Variables

When dealing with signals in practical applications, the particular values of the input are usually unknown. The input can, however, be regarded as a random process with certain statistical properties. One property that the input in the context of multimedia and video coding usually does not exhibit is stationarity, i.e. the statistical properties of the signal do change over time. While this is not ideal because stationarity of the random variable is often a condition for statistical analysis, statistical considerations can still be beneficial in many situations. One way to describe the statistical properties of a source (of the values from a source), is the probability density function (PDF). The PDF is a function $p(x)$ which represents the probability distribution of the random variable $x$. For a continuous random variable $x \in \mathbb{R}$, the PDF satisfies $p(x) \geq 0$ and $\int_{-\infty}^{\infty} p(x)dx = 1$. For a given PDF, certain values can be derived from it. For example, the probability of the random variable $x$ being smaller than a fixed value $b_0$ is given by

$$P(x < b_0) = \int\limits_{-\infty}^{b_0} p(x)dx. \tag{2.1}$$

The expected probability of the random variable $x$ taking values in the range from $a_0$ to $b_0$

is defined as

$$P(x \in [a_0, b_0]) = \int_{a_0}^{b_0} p(x)dx. \tag{2.2}$$

Finally, the expected value of a random variable with a function $f$ being applied to all values can be calculated by

$$\mathbf{E}(f[p(x)]) = \int_{-\infty}^{\infty} f(x)p(x)dx. \tag{2.3}$$

The mean value of the source can be calculated using the value itself as the function: $f(x) = x$. In digital signal processing, however, the random variables are not continuous but rather quantized to a discrete domain. The discrete equivalent of the PDF is the probability mass function (PMF) which must satisfy the equivalent properties $p(x_i) \geq 0$ and $\sum_i p(x_i) = 1$ where the $x_i$ are the discrete values that can occur. Intuitively, the PMF provides the absolute probability of the random variable $x$ taking the value $x_i$. Discrete variants of the Equations 2.1, 2.2 and 2.3 are similar and can be derived by replacing the integral over the continuous domain with a sum over the discrete values.

### 2.1.2 Entropy

In information theory, entropy is the quantitative representation of the abstract concept of information content. In the context of random variables, the entropy is defined as the average self-information of a random variable. For a discrete random variable $X$ with the symbols $x_i$ and the probability mass function $p(x_i)$, the self information of each symbol depends on its probability of occurrence. The lower the probability of a certain value $x_i$ is, the higher its information content gets, and vice versa. In this case, the entropy is defined by

$$H(X) = -\sum_i p(x_i) \log_2 p(x_i). \tag{2.4}$$

In information theory, the base of the logarithm is commonly chosen as 2, so the entropy is measured in bits. Thus, the entropy represents the minimum number of bits required for lossless coding of symbols from the discrete random variable $X$. One simple situation is the case of a binary alphabet where only $x_0$ and $x_1$ exist. The maximum of the entropy is located at the point of uniform distribution. For the binary alphabet, this corresponds to equal probability ($p = 0.5$) for both symbols which results in an entropy of $H = 1$ bit. This is the point of maximum uncertainty abouct the symbol. In the other extreme ($p = 1$ for one of the symbols), there is no uncertainty about the symbol and the "random" variable can be represented using $H = 0$ bits. This concept of entropy is very intuitive with regard to what the information content of a symbol or message should be: The more unlikely the

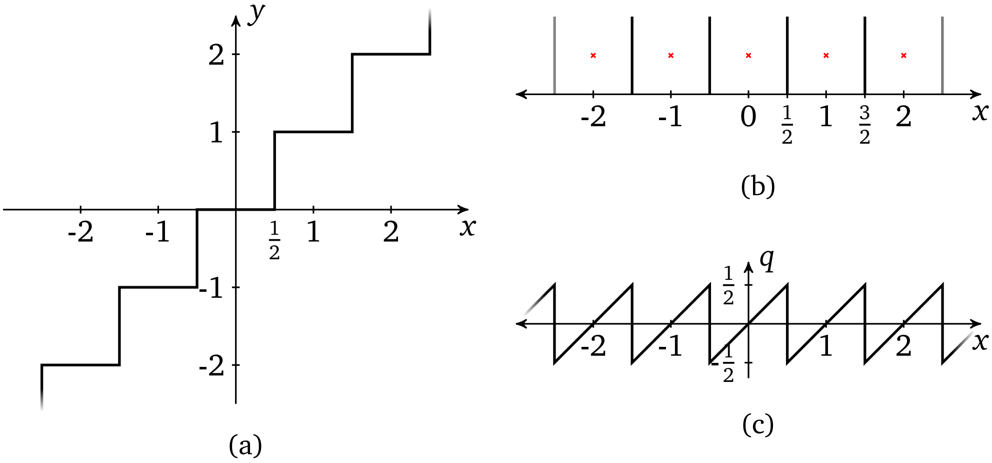


**Figure 2.1** Quantization by the rounding operation. **a** quantization characteristics from the continuous input values $x$ to the discrete output values $y$. **b** an alternative representation of the characteristics. The vertical lines indicate the decisions thresholds and the red crosses the reconstruction values. **c** the quantization error $q$ depending on the actual input value $x$.

occurrence of a symbol is, the more information it carries. Symbols or messages, however, that are very likely, contain only little information.

### 2.1.3 Quantization

Quantization describes a procedure that maps ranges of values to corresponding representation values. The input values to this process can be continuous or discrete, while the set of possible output values is always discrete. One of the most common quantization operations is rounding of real numbers to integer numbers. For all possible integer numbered output values $y$, the input value always lies within the interval from $y-0.5$ to $y+0.5$. Or the other way around, the input values from $y-0.5$ to $y+0.5$ are quantized to the integer value $y$. [1] In Figure 2.1a, the corresponding quantization function from the input values $x$ to the output values $y$ is plotted. In Figure 2.1b, an alternative representation for the same quantization function is given. All input values $x$ between each pair of vertical lines are quantized to the value marked by the red cross within the interval. Obviously, the process of quantization is not invertible and inherently induces a certain amount of error. The amount of this quantization error depends on the value $x$ of the input value. For the rounding example, the quantization error is plotted in Figure 2.1c.

For signal representations in a digital system, quantization of the input signal is inevitably the first step in order to obtain a version of the input signal that can be represented in the digital domain. For example, audio signals need to be quantized in an analog/digital converter in the sound card before they can be handled by a computer. For video signals, the analog

[1] If an input value is precisely on the decision threshold, a ruling has to be made which of the two closest whole numbers is the output. Here, we will round up in these cases (2.5 is rounded to 3). While rounding to integer values as described here is very common, there are also other integer rounding operations that can be used to perform quantization.

to digital conversion is typically performed within the image sensor. Since the number of bits that can be used for the representation of each input value is limited, the supported input range of values is also restricted. This means that for unrestricted input values, the quantization error has no upper bound for values that are outside of the supported input range. This case is also referred to as overload or clipping. In the following, we presume that all values to be quantized are already described by a discrete (digital) representation. Thus, a further quantization of the values reduces the number of bits that are needed to represent the values while some information about the original value is lost.

The uniform rounding example in Figure 2.1 can also be expressed using a quantization step size $\Delta$. The quantized level $k$ and the reconstructed quantized value $y$ are calculated from the input value $x$ as:

$$k = \left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor; \quad y = \Delta \cdot k, \tag{2.5}$$

with the $\Delta$ value set to 1. For uniform quantization, the distance between the reconstruction values is constant. Because of this regular setup, the quantization and reconstruction (inverse quantization) can be implemented very efficiently in hard- and software. [2] In general, the quantization can also be non-uniform. If the input to the quantization is assumed to originate from a stationary process with a non-uniform probability distribution, it can be beneficial to adapt the quantization to the signal characteristics. For example, if the input values exhibit a high concentration around zero, the quantization interval width can be chosen smaller for low amplitudes. Furthermore, a finer quantization can be beneficial for amplitude ranges where a high quantization error has a severe impact on the perceived reconstruction quality. In turn, the implementation of such a non-uniform quantization and reconstruction becomes significantly more complex. In general, the decision thresholds and the corresponding reconstruction values can be chosen arbitrarily depending on the application.

In Figure 2.2, another uniform quantization scheme is presented in which the decision thresholds are shifted away from the reconstruction value 0 while the reconstruction values remain at equidistant positions. In this way, a so called 'dead zone' around zero is formed. The equation 2.5 can be extended for this example by, depending on the sign of $x$, adding or substracting an offset of $\frac{1}{6}$ to the value of $x$. Such a setup is beneficial for signals with a strong concentration around zero. If the probability distribution is Laplacian, then it can be shown that this type of quantization is optimal when the quantization error and the rate needed to encode the quantized values are both considered in a rate-distortion sense [Sul96]. In this case, the quantized values will also exhibit a distribution with a high concentration of values at and around zero. This feature can be exploited in the subsequent entropy coding stage. For a more detailed discussion on quantization please refer to [Ohm15].

[2] While the actual computational complexity depends on the specific design of the quantization, also more complex uniform quantization schemes are usually designed so that they can be efficiently implemented in hard and software. Often only one addition, one multiplication and one shift operation is required.

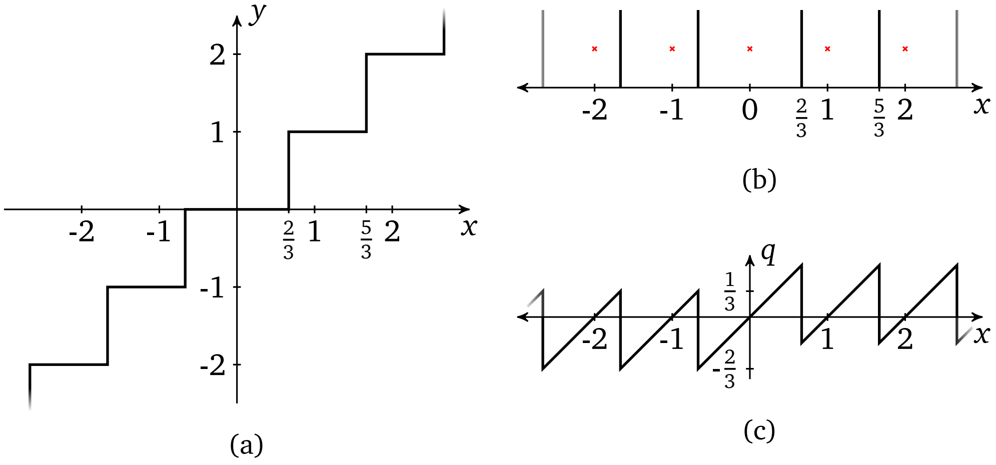


**Figure 2.2** Uniform dead-zone quantization with shifted decision thresholds. **a** quantization characteristics diagram. **b** alternative representation of the characteristics. **c** the quantization error.

## 2.2 Video Coding Fundamentals

Because raw and uncompressed video data consumes immense amounts of bits, the use of video compression is virtually inevitable for practical applications. Typically, video sequences exhibit high redundancy in the spatial and temporal domain which can be exploited in order to significantly reduce the data requirements. While many different approaches were tried, the concept of hybrid video coding became the most commonly used video coding approach for lossy compression. In the following, this approach and the concept of prediction structures are outlined. Furthermore, it is explained how the evaluation of the resulting compression performance is conducted.

### 2.2.1 Hybrid Video Coding

In hybrid video coding, two concepts are combined that compose the basic coding scheme used in most video coding standards since more than 20 years. The two concepts are a DPCM like prediction of the signal in the spatial (or even spatiotemporal) domain and a transformation based coding of the remaining error signal. While the basic concept of hybrid coding is even older, it was first used for video coding in H.261, which was standardized in 1993, and has been refined ever since [Hab74; h.261; h.262; h.263; h.264/AVC]. At first, the entire incoming video signal is fragmented into smaller partitions. Typically, the video is split into frames, where each frame is again split into blocks of a specific (but possibly varying) size. These blocks are then processed in a sequential manner. Figure 2.3 illustrates the underlying process from the encoder point of view. For each block, a prediction block is formed and subtracted from the original input signal. The better this prediction block is (the closer it is to the original block), the lower the energy of the resulting error signal gets. In order to remove the remaining pixel correlations within the block, the error signal is transformed and afterwards quantized to dispose of (possibly irrelevant) information. Finally, the quantized levels are encoded into the bitstream using entropy coding. The quantized levels also go

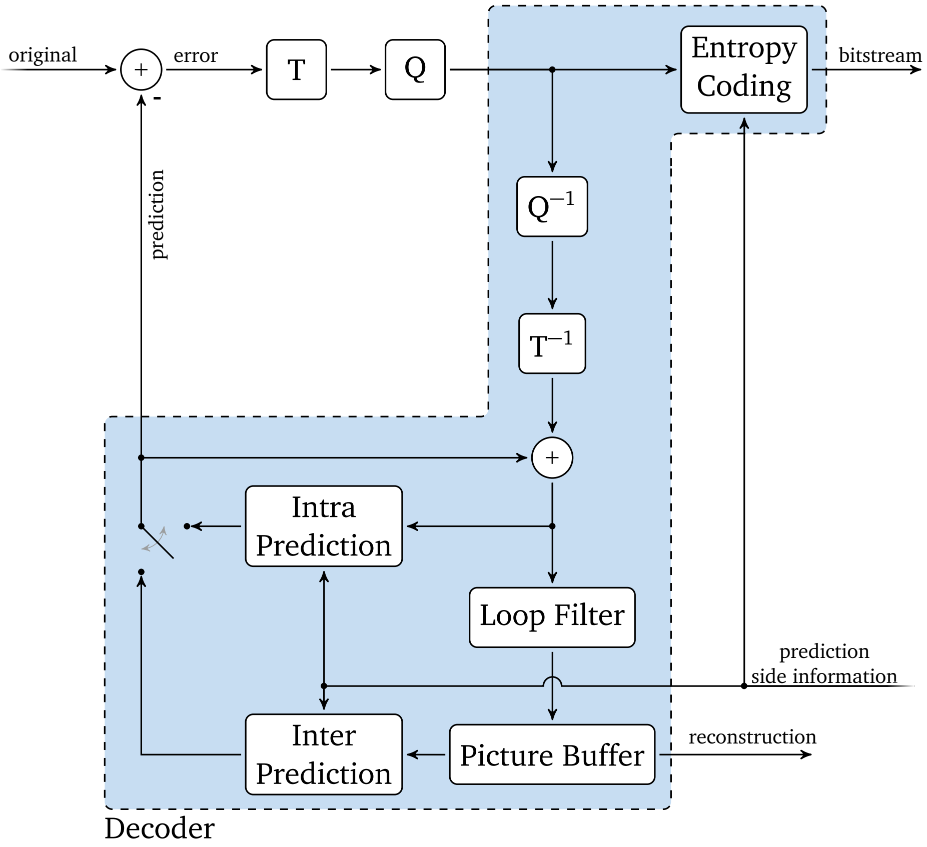


**Figure 2.3** The basic concept of a hybrid video encoder. A prediction is performed either by intra or by inter prediction. T and Q denote the transformation and quantization, while $T^{-1}$ and $Q^{-1}$ denote the corresponding inverse operations. The encoder contains the decoder loop which is highlighted by the dashed box.

through the inverse quantization and transformation steps. The previously used prediction signal is added back such that a reconstruction signal is formed which is identical to the one that the decoder will obtain from the bitstream. Before the entire picture reconstruction is put into the picture buffer, one or several types of loop filters may be applied. Typically, a deblocking filter is applied that reduces the effect of block boundaries that go along with the block based coding approach. However, other filters may be employed as well. While the filters can improve the perceived image quality, they are placed inside of the loop because they also increase the overall coding efficiency. The picture buffer holds a specific set of the filtered, previously coded pictures. Unless there is a transmission error, these pictures are identical to the pictures that the decoder is able to reconstruct from the bitstream.

In video coding, there are usually two fundamental types of prediction: Intra and Inter prediction. For intra prediction, the current block is predicted from the available already reconstructed samples in the neighboring blocks. This neighborhood can be averaged or a directional interpolation can be applied to acquire the intra prediction signal. The averaging technique is effective in homogeneous areas, while a directional prediction is well suited for straight edges and directional structures. For inter prediction, the pictures from the picture buffer are employed. Using a two dimensional vector and a reference index, a certain area from one of the reference pictures is copied and used as the prediction signal. Because this approach is very efficient for static scenes and scenes with motion, it is referred to as motion compensation (MC). One of the two prediction schemes is chosen, and the loop is closed by subtracting this signal from the original.

Both intra and inter prediction commonly require some side information. This side information contains the decision which of the two prediction types to use as well as for example the intra prediction mode for intra prediction, and motion vector and reference picture index for inter prediction. This data is also embedded into the bitstream so that the decoder can perform the same prediction and reproduce the identical prediction signal. The block that was omitted in this figure is the encoder decision block that generates the prediction side information. The inputs to this block are the same as to the intra and inter prediction blocks as well as the original image. The decision block then selects a prediction mode and determines the side information which is used for each block. [3]

Because of the closed loop DPCM approach, the encoder incorporates all building blocks of the corresponding decoder (marked by the shaded area in Figure 2.3). Of course for a pure decoder, the entropy encoding block must be replaced by an entropy decoder which can reproduce the quantized levels and prediction side information from the bitstream. By reversing the quantization and transformation, a reconstruction of the error signal is calculated and added to the prediction, which is again generated from previously decoded information using intra or inter prediction. When the whole frame is decoded and loop filtering was applied, the frame is written into the reference picture buffer and is ready for output to the viewer. The loop in the lower part of the figure (the prediction is formed from previously decoded information which in turn is calculated using a prediction) is also referred to as the coding loop. The coding loop at the encoder and decoder side should always be in sync. Otherwise, the decoding process uses different data for prediction than the encoder: The reconstructions at the encoder and decoder side drift apart. This situation is normally not by design but can occur in case of transmission errors. However, there are cases in scalable coding where a certain amount of drift is introduced intentionally in order to enable a more flexible coding scheme (see Section 2.4).

### 2.2.2 Prediction Structure

As stated above, the input sequence is commonly split into frames, which are processed sequentially in a specific order. Unless a low coding delay is a constraint, the display order and the coding order of frames can be decoupled. The so called prediction structure defines the order by which the frames are processed as well as the sets of reference pictures that each frame can use for inter prediction. Usually, this prediction structure is not fixed by the coding standard, but rather there is a way for the encoder to signal the implemented prediction structure to the decoder. However, the coding standard may define some limitations on the

[3] The encoder decision block is the central core of the encoder. Here it is decided how the available tools from the specific coding standard are used in order to form the output bitstream. The implementation and objective of this block is highly dependent on the application of the encoder. The encoder could for example be optimized to achieve the highest possible coding performance, a certain image quality, a minimum throughput of frames per second, a specific target bitrate, et cetera.

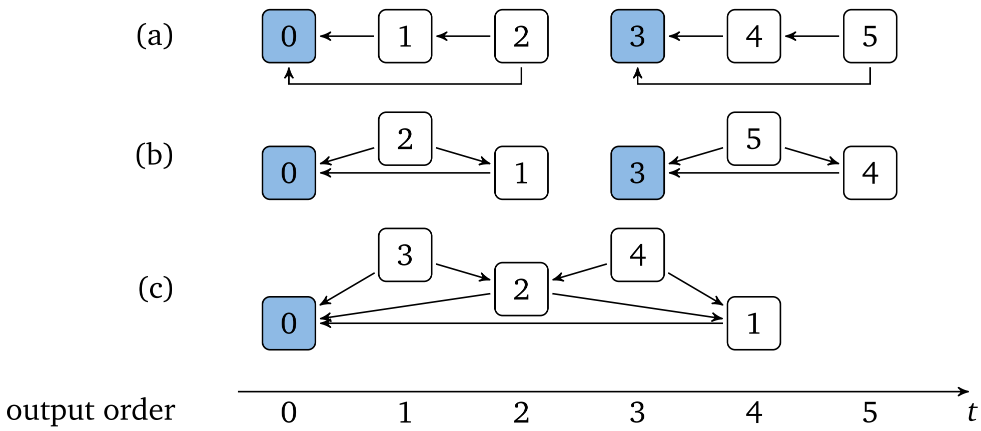


**Figure 2.4** Exemplary prediction structures. The output order on the time axis is indicated on the bottom while the coding order is noted inside of the frames. Frames employed for inter prediction are connected by arrows. The shaded frames do not perform inter prediction and can be decoded independent of other frames.

structure like a maximum number of reference frames that can be used or a maximum size of the picture buffer. Figure 2.4 illustrates some prediction structure examples. In example a), the coding order and the display order are identical. Every third frame is coded without dependencies to previously coded frames and can therefore be decoded independent of other frames. Because decoding can start at these points, they are also referred to as random access points (RAP). Furthermore, any occurring drift is effectively terminated by these intra frames. In example b), the coding order is different from the display order. The frames 2 and 5 can now also predict from frames which are located in the temporal future. Furthermore, this allows for the frame to use bi-prediction where two prediction signals (for example one from the temporal past and one from the temporal future) are weighted together to form the final prediction signal. In the last example c), the concept from b) is extended to a form of hierarchy where even more frames can benefit from references in both temporal directions. For illustrative purposes this hierarchy is pointed out in Figure 2.4 b) and c) by a vertical shift of the frames.

In general, the choice of a prediction structure strongly depends on the application. It can be seen in the figure that the examples b) and c) introduce a structural coding delay. For b), frame 1 in output order can not be output until frame 1 was received and decoded. For example c), this delay is even larger. For real time applications like video conferencing, a configuration without structural delay is usually favored. If the structural delay is not a constraint, a higher compression efficiency can generally be achieved with hierarchical prediction structures. Also the placement of random access points depends on the use case. For broadcast, we want to be able to start decoding at regular points in time if switching to a channel. The same applies if seeking in the video shall be supported. In other applications it might be applicable to actively request a random access point from the transmitter. E.g. in applications where the video is encoded in real time for a limited number of receivers like video conferencing.

### 2.2.3 PSNR and the Bjøntegaard Delta

In case of lossless video compression, the evaluation of the coding performance is very simple. Because the reconstruction is identical to the original input video, only the resulting bitrate must be compared. However, if the compression is lossy, the reconstruction quality must be considered as well. One possibility is to perform a formal subjective test. While this approach yields very relevant results, it is an expensive and tedious process which is impractical to be applied for every performance test. Because of this, the objective measure of peak signal to noise ratio (PSNR) for the reconstruction quality is commonly used. In order to establish the reconstruction quality of an entire video sequence compared to the original, the PSNR is typically calculated per frame followed by an averaging of all values. The PSNR is calculated from the mean squared error (MSE) by

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{A_{max}^2}{\mathrm{MSE}}\right) dB, \tag{2.6}$$

where $A_{max}$ is calculated from the bit depth $b$ of the input by $A_{max} = 2^b - 1$.

The MSE is calculated on a sample by sample basis and can be calculated between arbitrary arrays of samples (e.g. a smaller block within a frame). Because of this, it can also be used at the encoder side to estimate the reconstruction quality of a certain coding mode and to take a decision on which mode to use. Let the original $N$ x $M$ samples be denoted by $o(n,m)$ and the distorted sample array by $d(n,m)$. The MSE is calculated from the sample by sample difference between the distorted signal and the original by

$$\mathrm{MSE} = \frac{1}{NM}\sum_{n=0}^{N}\sum_{m=0}^{M}(o(n,m) - d(n,m))^2. \tag{2.7}$$

While there are more elaborate measures which might better represent the subjective impression of the quality, the simple PSNR measurement is considered to be sufficient in most cases. In the standardization process of HEVC and SHVC, the PSNR was the primary distortion measurement. However, at certain points in time, subjective testing was conducted as well.

The Bjøntegaard delta (BD) was proposed by Gisle Bjøntegaard in 2001 as a solution to a very specific problem in the standardization process [Bjo01]. When a modification to the existing coding scheme was proposed, the recommended simulation conditions stipulated to use 4 points at a different rate and distortion. The resulting points were then plotted in a rate-distortion graph and the figure was reported to the standardization body. The Bjøntegaard delta picks up these 4 points and attempts to calculate an average distance between the two graphs. Figure 2.5 exemplifies how the computation of the delta along the rate axis is performed. Firstly, the bitrate is moved to a logarithmic scale. Then, two third order polynomials are fitted through the two sets of points (4 points of the reference and 4 for the test). The integrals of both polynomial functions along the y-axis is calculated for the overlapping interval $y_0$ to $y_1$. Finally, the results are subtracted and normalized to the

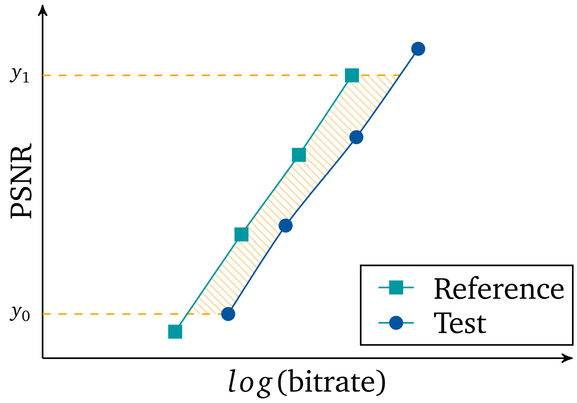


**Figure 2.5** Visualization of the Bjøntegaard rate delta calculation between a reference and a configuration under test. The shaded area between the two polynomial functions is calculated and normalized by the with of the interval.

width of the interval. This value is also referred to as BD-rate. In the same way, the average difference in PSNR along the y-axis can be calculated which is also termed BD-PSNR.

Because of its convenient compact representation and the simple calculation, the Bjøntegaard delta became a quasi standard of calculating and reporting coding performance results. In the following years, the testing conditions were adjusted and it was noticed that the originally proposed polynomial interpolation could be ill fitted in some situations. Because of this, the interpolation was revised to a Piecewise Cubic Hermite Polynomial interpolation that demonstrates a better behavior in these situations. Furthermore, with the piecewise interpolation the delta calculation is more flexible and can be calculated for a variable number of rate-distortion points. However, the basic idea of calculation of an average distance between two data sets remains unchanged [ZSS08].

For color video, there are multiple components that the PSNR can be calculated for. In HEVC, the image is represented in the $YC_bC_r$ color space which includes one luma component which corresponds to the brightness and two chroma components which correspond to the color information. While the PSNR can be calculated for each of these components, the corresponding rate can not be perfectly split for the color channels. Because of this, the Bjøntegaard delta is commonly calculated and reported for all of the three components using the total bitrate. Furthermore, the two chroma components are sub-sampled by a factor of two in horizontal and vertical direction in HEVC version 1. These aspects should be considered for the interpretation of the BD-rate and BD-PSNR results for the luma and chroma components.

## 2.3 High Efficiency Video Coding

High efficiency video coding (HEVC) is a hybrid video coding standard, which was jointly developed by the ITU-T video coding experts group (VCEG) and the ISO/IEC moving picture experts group (MPEG). The name of the joint team is Joint Collaborative Team on Video

Coding or JCT-VC. With the increased popularity of higher resolution applications like ultra high definition (UHD) and a demand for high-quality video on mobile devices, the ambition was to create a new video coding standard with a substantially higher compression efficiency compared to H.264/AVC and a focus on higher resolutions and the capability to utilize parallel processing architectures. The first version of the standard was approved in January 2013. In parallel to the standardization of HEVC, several extensions to HEVC were also developed within JCT-VC. Version 2 of the HEVC standard was subsequently approved in July 2014 and contained three extensions to the first version of HEVC: Multiview video coding (MV-HEVC), range extensions (RExt) and the scalable extension (SHVC).

In order to evaluate the overall coding performance and assess the performance impact of proposed changes to the standard, the standardization body maintains a reference software of encoder and decoder. The decoder is intended to obey the HEVC standard and can be used to test for bitstream conformity to the standard.[4] The included encoder is intended to establish a common basis for the standardization of HEVC and to aid in the evaluation of new coding technologies and their impact on the coding performance. The encoder uses rate-distortion optimized encoding and is designed to provide an HEVC compliant bitstream. However, the encoder is not designed for other applications and lacks some commonly used encoder features such as sophisticated rate control. While it is implemented in C++, it is hardly optimized for speed which already disqualifies it for most practical applications. The software repository is publicly accessible at [HM-Software].

In the following, some aspects of HEVC that are relevant in the context of this work are detailed. However, because of the sheer complexity of HEVC, not all features can be described here. For a more complete overview of HEVC, the reader is referred to [Sul+12; Wie15; SBS14; h.265/HEVC]. Because HEVC is based on the hybrid video coding approach, all the blocks from Figure 2.3 are also present in HEVC. After some general information on the HEVC coding structure, these blocks are further detailed in the subsequent sections.

### 2.3.1 HEVC Coding Structure

A video sequence in HEVC is processed frame by frame. Internally, every color image is represented in the $YC_bC_r$ color space by one luminance and two chrominance signals. The $Y$ component represents the luminance (brightness) component while $C_b$ and $C_r$ are the two chroma components which contain the color information. Due to the fact that the human visual system is less sensitive to the color than to the luminance, the two color components are sub-sampled by a factor of two in both directions. [5] More information on the subject can be found in [Poy12]. Each frame is addressed by a picture order count (POC) value.

[4] Of course the reference software decoder can only give a good indication of bitstream conformance. The HEVC standard text is the valid reference. However, the reference software is implemented in order to comply to the HEVC standard text as closely as possible.

[5] Sub-sampling is always applied for HEVC version 1. In the later version there are also the range extensions to HEVC in which the chroma components may be only sub-sampled in horizontal direction or no sub-sampling may be applied. While in this thesis the sub-sampling with a factor of two in both directions is assumed, a different sub-sampling could be used as well. More information about the range extensions can be found in [Fly+16].

All decoded pictures are put into the decoded picture buffer (DPB). Pictures are removed from the buffer when they were output and are no longer needed as reference pictures. The pictures that can be used as reference pictures are collected in the reference pictures set (RPS). For each frame, two reference picture lists are constructed from the frames in the RPS: list 0 and list 1. Motion compensated prediction can then be performed using an index into these lists. For bi-prediction, one picture from list 0 and one picture from list 1 is used for prediction.

As for many other video coding schemes, each frame in HEVC is split into a grid of square blocks (or units), where all units are processed line by line in a sequential order. [6] The size of these coding tree units (CTU) is signaled at the beginning of the bitstream and can be 16x16, 32x32 or 64x64 pixels. Using a tree structure, each unit can be split into four square subunits of half the width and height. This process can be applied recursively down to a size of 8x8 pixels. The resulting unit which is not further split is referred to as a coding unit (CU). As already mentioned, HEVC uses inter and intra prediction. The choice of prediction mode is signaled per CU. Starting from the CU level and depending on the selected prediction mode, the coding unit can be further split into two or four prediction units (PUs). Following this, another tree for the transform units (TUs) is signaled, again starting at the CU level. HEVC defines transforms for the square block sizes of 4x4, 8x8, 16x16 and 32x32 pixels. Because of the two separate trees, it is possible that a prediction unit contains multiple transform units as well as that transform unit spans multiple prediction units. For all PUs within the CU, a prediction is performed and for each TU in the CU, a transformed and quantized residual signal may be coded. The prediction signals and reconstructed residual signals are then added to form the reconstruction signal for the CU. As it was stated above, the image is composed of a luma- and two chroma components of half the size in vertical and horizontal direction. This also applies to the CUs and TUs. Each CU contains three coding blocks (CBs) for the three color components where the luma CB is of the same size as the CU and the chroma CB has half the width and height. The same applies to the TU which comprises a corresponding group of transform blocks (TBs).

### 2.3.2 Intra and Inter Prediction

For intra prediction, the PU is either of equal size as the CU, or the CU is split into four square PUs. The samples of the current PU are calculated from the available reconstructed samples of neighboring blocks within the same frame. Because of the tree like splitting into CUs, the available samples may not only be left and above of the current block, but also on the lower left and upper right. As in H.264/AVC, different modes of intra prediction can be selected. Firstly, HEVC specifies 33 angular intra prediction modes, which use extrapolation of the reference samples in the current prediction block along a certain direction. Each of

[6] There are some tools in HEVC which alter this basic coding order. Firstly, the grid of units can be split into rectangular sub-sections using tiles. All units within each tile are then processed in a sequential order where the tiles can be processed in parallel. Another technique is so called wavefront parallel processing which allows for a synchronous entropy decoding and reconstruction of multiple rows of units. While these tools are not further detailed in the remainder of this thesis, there are no incompatibilities with the later presented techniques. Further information on tiles and wavefront parallel processing can be found in [Sul+12; Mis+13; Wie15].

the 33 angular modes defines a specific direction. This kind of prediction is particularly efficient for blocks with strong directional structures like edges. Alternatively, two non-angular modes are available, planar- and DC mode. Both modes use different averaging strategies to create more smooth prediction signals. As in H.264/AVC, constrained intra prediction can be enabled by a syntax element in the bitstream. In this mode, neighboring samples that were obtained using inter prediction are regarded as not available for intra prediction. Enabling this feature improves the loss-resiliency by preventing propagation of possibly error prone information from other reference pictures into the intra prediction signal of the current picture [Lai+12].

For inter prediction, the PU can either have equal dimensions as the CU or the CU can be split into two or four PUs. The split into two parts can be performed in horizontal or vertical direction and results in two non-square PUs. In both cases, there are two modes in which the split is asymmetric and the two resulting PUs are not of equal size. For each PU, motion information is signaled, which consists of an index into the reference picture list 0 (list 0) and a motion vector. In case of bi-directional motion compensation, another index into the reference picture list 1 is coded with another motion vector. The prediction signal of the PU is now formed by copying the reconstructed pixel values from the PU area within the indicated reference picture using the given motion vector as a displacement offset. As the motion vectors are indicated with quarter pixel precision, interpolation using an 8-tap filter in horizontal and/or vertical direction is performed if the motion vector is indicating a sub-pixel position. In case of bi-prediction, this process is performed twice and the pixel values are merged together to form the prediction signal. Because motion information in certain areas is commonly quite uniform, motion vectors are always signaled relative to a predicted motion vector which is obtained from spatially or temporally neighboring CUs. There are two methods for derivation of the motion information. In the first mode (merge mode), a list of possible merge candidates is generated from inter predicted CUs in the available neighborhood, and possibly one candidate from the inter predicted CU in one of the reference pictures in the reference picture buffer (temporal merge candidate). An index (merge index) is coded and the motion information from the corresponding candidate is copied to the current PU. In the second prediction mode, a new motion vector is coded differentially using a motion vector predictor. A list of two candidates is generated from spatially and temporally neighboring CUs. If the reference index of the predicted motion vector differs from the reference index of the current PU, the motion vector prediction is scaled according to the POC values of the addressed frames. Afterwards, the x and y components of the motion vector difference are coded. At the decoder side, the motion vector is constructed by adding the prediction and the difference. This also means that alongside the reconstructed pixel values, the motion information for each CU has to be stored in the reference picture buffer. In order to restrict the memory requirement, this information is stored alongside with the reference fames on a 16x16 pixel grid, even if the PU size is smaller [Ugu+13].

### 2.3.3 Transformation and Quantization

After prediction of the coding unit, the residual signal is processed. Firstly, the coding unit is partitioned into transform units using another quadtree partitioning. Each transform unit can be of size 4x4, 8x8, 16x16, or 32x32 pixels. Starting from the size of the CU, a flag (*split_transform_flag*) indicates if the unit is split into four square sub-units. If the unit is further split, the process is invoked recursively for each of the four sub-units. If the minimum transform size of 4x4 pixels is reached or the unit is not split any further, the process is terminated and a transform unit is established. Because the maximum TU size is 32x32 pixels, the first split is applied implicitly if the CU size is 64x64 pixels. This quadtree is also referred to as the transform tree. The transform tree is independent of the partitioning of the coding unit into prediction units so it is possible that a TU extends over a prediction unit boundary. As the minimal transform size is 4x4 pixels and the transform blocks for the chroma components are sub-sampled by a factor of 2, a special case arises if the TU size is 4x4 pixels. In this case, four TBs of size 4x4 for the luma component, but only one 4x4 TB for each of the chroma components are coded.

The residual signal of each transform unit is subsequently transformed using the HEVC core transform. [7] The core transform is an approximation of the discrete cosine transform (DCT) using finite precision integer arithmetic. While specific implementation details differ, the DCT is also used in multiple other video coding standards like H.262/MPEG-2, H.263 and H.264/AVC. It is commonly used because it provides highly uncorrelated coefficients and can at the same time be very efficiently implemented in hardware as well as software. In HEVC, each DCT matrix element is represented using 8 bit precision and each transformed coefficient can be represented using 16 bit. The matrix multiplication can be implemented using 16 bit multipliers and 32 bit accumulators. Furthermore, the coefficients in HEVC were tuned so that the matrix elements for a transform are a subset of the matrix elements for a transform of twice the size. This means that only the DCT matrix elements for the 32x32 pixel transform need to be specified. All smaller matrices can then be derived from the larger matrix.

In a very similar way, an alternative inverse transformation based on the discrete sine transform (DST) is specified in HEVC. This transform is applied for all transform blocks of size 4x4 that use intra prediction. Because it is also defined as a matrix multiplication with similar properties as the DCT, it can be seamlessly integrated into the quantization and transformation process. When it was proposed, it could be shown that for this specific case the DST accomplishes a higher compression efficiency [SF11].

The specified core transform allows for highly optimized implementations of the forward and inverse transformation operation. Particularly, further optimizations are possible if certain areas of transform coefficients are zero (not significant). It is also possible to further exploit the symmetry properties of the transform matrix. Lastly, it is possible to reuse hardware blocks in a unified forward/inverse transform unit. While all of these optimizations can help to save computational complexity, the inverse transform still requires a significant amount of arithmetic operations (multiplications and additions).

Finally, the transform coefficients are quantized to levels. It can be observed, that for image

[7] In the HEVC standard, only the inverse transformation and quantization process is specified. The choice of a forward transformation and quantization is left to the encoder implementer. However, the transformation and quantization were designed and tested in conjunction with their forward counterpart. Because of this, both the forward and inverse processes are described here. The described forward transformation and quantization is used by the reference software encoder [HM-Software].

signals the distribution of the two dimensional array of DCT coefficients is close to a Laplacian distribution [RG83; LG00]. This property is utilized in the quantization process and also for the coding of the transform levels. Firstly, the absolute value of the transform coefficients is handled independent of the sign. Because of the coefficient distribution, positive and negative signs can be assumed to be equally probable while the absolute value of the coefficients is expected to have an exponential distribution. Secondly, the interval boundaries for quantization are shifted so that the reconstruction value of each quantization interval is close to the expected value within each interval.

Because the quantization induces a loss of information, it impairs the reconstruction quality. However, coding of the quantized levels can be performed with far fewer bits compared to the un-quantized transform coefficients. Therefore, the quantization process allows for a trade-off between the reconstruction quality and the number of bits that are required to encode the quantizer levels. As in H.264/AVC, the quantization step size $\Delta_q$ in HEVC is specified by a quantization parameter (QP) that ranges from 0 to 51. In the HEVC reference encoder, the QP is the primary value to control the reconstruction quality. The QP value converts to a step size by

$$\Delta_q(QP) = 2^{(\frac{1}{6}(QP-4))} \approx 1.1225^{(QP-4)}. \tag{2.8}$$

It can be seen that the quantization step size increases by approximately 12% for a QP increment of 1. For a QP increase of 6, the quantization step size is precisely doubled. For a QP value of 4, the quantization step size is 1. As for the transformation, also the reconstruction process is specified using integer arithmetic. For this, the integer approximation of $\Delta_q$ is expressed using a modulo operation, a table lookup, a binary shift, and an integer division,

$$\Delta_q(QP) = S[QP\%6] \ll \left\lfloor \frac{QP}{6} \right\rfloor, \tag{2.9}$$

where the lookup table $S$ contains the 6 values

$$S = \left\{2^{-\frac{4}{6}}, 2^{-\frac{3}{6}}, 2^{-\frac{2}{6}}, 2^{-\frac{1}{6}}, 2^{-0}, 2^{-\frac{1}{6}}\right\}. \tag{2.10}$$

Instead of a division by the step size $\Delta_q$, the transform coefficient $c$ is multiplied by a scaled integer approximation of the inverse step size followed by a binary right shift. Because of the assumed coefficient distribution, the quantization interval boundaries are shifted. The output level $l$ is calculated from the coefficient $c$ by

$$l = (c \cdot F[QP\%6] + (d \ll s_f)) \gg (s_f + 9), \tag{2.11}$$

where the shift $s_f$ is calculated from the QP value, the sample bit depth $B$ and the transform size $N$ by $s_f = 20 - B - \log_2(N) + \left\lfloor \frac{QP}{6} \right\rfloor$. The value $d$ sets the shift of the quantization interval boundaries. The value of $d$ is 171 for intra slices and 85 otherwise. The lookup table $F$ contains the approximated inverse values of $S$ scaled by $2^{14}$

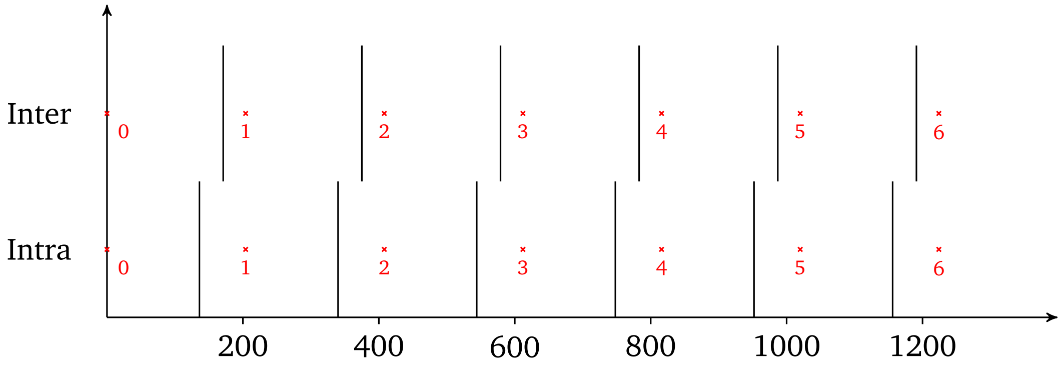


**Figure 2.6** Quantization example for a QP value of 38, a bit depth $B$ of 8, a transform size of 8x8 pixels ($N = 8$) and both intra and inter slices. All values between each pair of quantization boundaries (black lines) are quantized to the same level value (red numbers). The corresponding reconstruction value for each interval is marked by a red cross.

$$\begin{aligned} F &= \{26214, 23302, 20560, 18396, 16384, 14564\} \\ &\approx \left\{2^{\frac{4}{6}+14}, 2^{\frac{3}{6}+14}, 2^{\frac{2}{6}+14}, 2^{\frac{1}{6}+14}, 2^{0+14}, 2^{-\frac{1}{6}+14}\right\}. \end{aligned} \tag{2.12}$$

The reconstruction of the coefficients $c'$ from the level $l$ is then performed by a multiplication with a scaled integer approximation of the step size

$$c' = \left(l \cdot (I[QP\%6] \ll \left\lfloor \frac{QP}{6} \right\rfloor) + (s_i \gg 2)\right) \gg s_i, \tag{2.13}$$

where the shift $s_i$ is calculated from the sample bit depth $B$ and the transform size $N$ by $s_i = log_2(N) + B - 9$. The added offset $s_i \gg 2$ in combination with the binary right shift of $s_i$ implements a rounding operation. The lookup table $I$ contains the approximated values of $S$ scaled by $2^6$

$$\begin{aligned} I &= \{40, 45, 51, 57, 64, 72\} \\ &\approx \left\{2^{-\frac{4}{6}+6}, 2^{-\frac{3}{6}+6}, 2^{-\frac{2}{6}+6}, 2^{-\frac{1}{6}+6}, 2^{6}, 2^{\frac{1}{6}+6}\right\}. \end{aligned} \tag{2.14}$$

The values of the lookup table $I$ can be stored using 7 bits per value. Furthermore, the quantization and reconstruction were designed to approximately result in unity gain. In Figure 2.6, the quantization and reconstruction is showcased for a QP value of 38. The reconstruction values (red crosses) are located at equal distances and are independent of the slice type. The quantization boundaries (black lines) are shifted such that the reconstruction value is at approximately $\frac{1}{3}$ and $\frac{1}{6}$ of the interval width for intra and inter slices, respectively.

It should be noted here, that in the bitstream it can be signaled to disable the quantization and transformation. Firstly, a flag (*cu_transquant_bypass_flag*) can be coded per coding unit

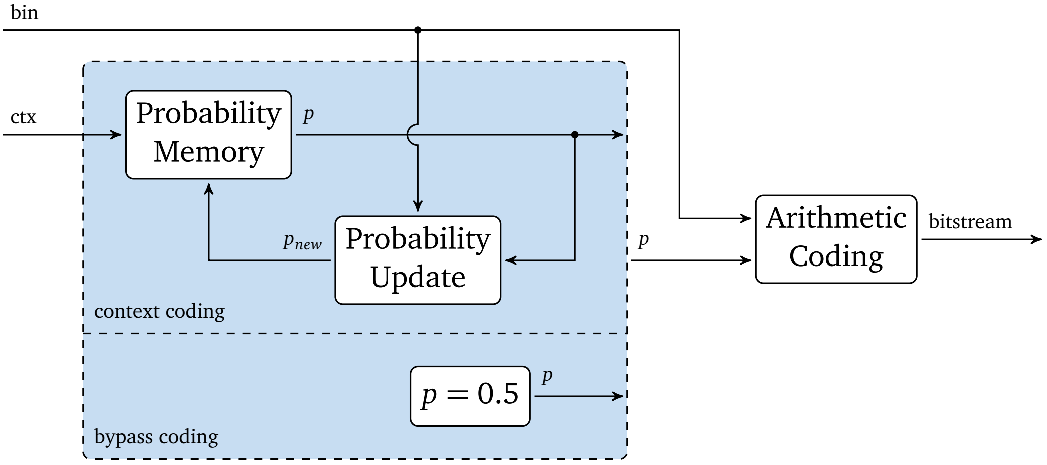


**Figure 2.7** Block diagram of the CABAC coding approach. Every bin is arithmetically coded using a specific probability. The probability estimation (shaded block) can operate in two modes: In a context based backwards adaption mode or in bypass coding mode with a fixed probability of 0.5.

to signal a bypass of both stages. If the flag is set, the residual values are directly coded into the bitstream which allows for perfect reconstruction of the original image signal. Secondly, another flag (*transform_skip_flag*) can be coded per TU. If the flag is set, the transform stage is skipped and replaced by a simple scaling of the sample values. The scaled values are then forwarded to the quantization process and the resulting levels are coded into the bitstream [h.265/HEVC].

It was already mentioned that only the inverse quantization process is defined in the HEVC standard. The implementation of a forward quantization is left to the encoder manufacturer. Typically, an encoder will perform quantization using a rate-distortion trade off between the reconstruction quality and the estimated bitrate. When using this rate-distortion optimal quantization (RDOQ), the encoder will intentionally modify the levels of the quantized coefficients if a higher rate-distortion performance can be achieved by doing so [KYC08]. This encoder only optimization strategy is also implemented in the reference software encoder [HM-Software].

For a more detailed description of the HEVC core transform, quantization and the complexity of the transformation, the reader is referred to [Bud+13].

### 2.3.4 Context Adaptive Binary Arithmetic Coding

The context adaptive binary arithmetic coding engine (CABAC) was first implemented in the H.264/AVC video coding standard, where, alongside a less complex variable length code (VLC), it is one of two selectable entropy coding engines. For HEVC, the basic concept of CABAC remained unchanged. Only the context adaption was revised and the number of used contexts was significantly reduced. Because of its superior performance, CABAC became the only entropy coding engine in HEVC [h.264/AVC; h.265/HEVC].

At its core, CABAC is based on the arithmetic coding concept which is employed widely for state of the art image and video compression because of its excellent performance in this setting. The core of arithmetic coding is an interval, which is split into multiple sub-intervals with one interval per possible input symbol. The width of each interval corresponds to the probability of occurrence of each input symbol. For the arithmetic coding engine in CABAC, the source is always binary. In the context of CABAC, each binary input symbols is referred to as a bin, while the output symbols of the arithmetic coding process are termed bits. The interval for each bin is split into two intervals according to the probability estimation $p$ of the bin. Depending on the bin, one of the two intervals is chosen as the new interval and the process is reiterated. At the end, an arbitrary number from within the interval is transmitted so that the decoder (with knowledge of the probabilities) can reconstruct the input symbols. However, the approach as it is described here would require infinite precision and also has other restrictions which make it impractical for actual implementations using discrete systems. The arithmetic coding engine which is used in CABAC is highly optimized for performance on software as well as hardware platforms. In CABAC, the probability value $p$ is given by a 7 bit index which corresponds to a particular discrete probability value in the range from zero to one. For the bypass coding mode, the probability which is used for coding is 0.5 and one coded bin requires exactly one bit in the bitstream. In combination with various other simplifications and optimizations, the final arithmetic coding engine is multiplication free and can be implemented using only additions, binary shift operations and memory look up operations. For a detailed explanation of arithmetic coding and the arithmetic coding engine in CABAC, the reader is referred to [Ohm15; MW03; MSW03].

Figure 2.7 shows a block diagram of the CABAC coding approach. On the right, the aforementioned arithmetic coding engine encodes every input bin into the bitstream using a certain estimated probability value $p$. The performance of this arithmetic coding step completely depends on this probability estimation. The better the probability estimation, the closer the coding performance can get to the entropy of the source. For the probability estimation (the shaded block), one of two modes can be chosen. In bypass mode, the probability of the bin is assumed to be close to 0.5. Because no adaption is performed and a simplified arithmetic coding step can be used, this mode allows for a low complex coding of bins. The second mode (context coding) uses a backwards adaptive probability estimation with multiple contexts. A context is an index, which is assigned to every bin which is not coded in bypass mode. The value of the index is determined based on the type of bin which is coded as well as other available coding related information. For example, there are three indices (contexts) for coding of the split flag. Depending on the split flags of the neighboring CUs, one of these three contexts is chosen (0 if neither-, 1 if one- and 2 if both of the top and left neighboring CUs are split). The assumption here is, that there is a spatial connection between the flags. If the flags in the neighborhood are set, it is more probable that the flag to code is also set. This enables the context to adapt even better to the actual probability of the flags. Other utilized information can include the current block size, the prediction mode or the position within a transformation block. For all non-binary syntax elements, binarization must be performed prior to arithmetic coding. If multiple bins are used to signal a syntax element, the context for each bin may also be selected by the binarization process. Of course, all of this information must also be available on the decoder side so that the decoder will select the same context and thus use the identical probability estimation for each bin.

For the given context, the corresponding currently assumed probability value $p$ is retrieved from memory and is, in conjunction with the corresponding bin, handed to the arithmetic coding engine. After the bin is coded, the probability value is updated and the new value $p_{new}$ is stored back into memory. The probability memory stores one probability value $p$ for every context. While in H.264/AVC, there were 299 contexts, this number was cut down to 154 in HEVC. Since $p$ is represented using a 7 bit value, the memory must be able to hold 154 7 bit values [Bos+12]. For each context, the update process uses the previously coded bins to calculate a new probability estimation. This is referred to as backward adaption because the probability estimation update is performed after the arithmetic coding step. Therefore, no signaling of probability values is needed. With the decoded bins, the decoder can perform the identical probability update and the probability memory at the encoder and decoder can be kept in sync. For the update itself, various strategies are conceivable. For example, a certain number of coded bins could be kept in memory and be updated with every new bin. The probability estimation could then be performed from this set of previously coded bins. For simplicity, a straightforward table based update process is used in CABAC. Depending on the current value of $p$ and the coded bin, a new value of $p$ is selected from a fixed transition table.

Especially for high bitrates, the entropy coding can become a bottleneck if a high number of bins per image must be encoded or decoded using context based coding. Because of this, the coding process is designed such that only a limited number of bins are coded using context adaption. For large syntax elements, this can be achieved by splitting the value into two parts: A prefix part and a suffix part. The bins for the prefix part are coded using context adaption while the remaining suffix bins are coded using a Golomb-Rice code in combination with bypass coding. For transform coefficient levels in HEVC, the worst case number of context coded bins per coefficient is hereby limited to 3. [8] Furthermore, the context derivation in HEVC (especially for transform coefficients) is optimized for parallel processing as much as possible [Bos+12; SB12].

### 2.3.5 Transform Coding

After transformation and quantization, the transform levels of each transform coefficient in the transform block are coded into the bitstream. In the following, these quantization levels are still referred to as transform coefficients. Each luma transform block can be of size 4x4, 8x8, 16x16 or 32x32 coefficients. For chroma, the available sizes are 4x4, 8x8 and 16x16. Typically, the coefficient energy is highly concentrated in the lowest frequency coefficients while non-zero coefficients are sparse for a higher frequency. The transform coding implementation is optimized for this setting while enabling a high amount of parallelism.

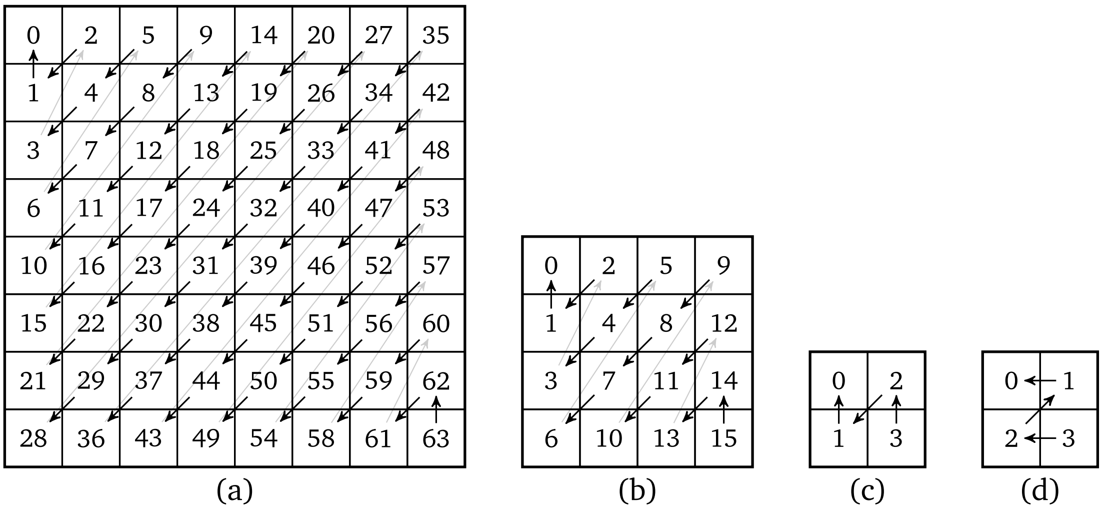


**Figure 2.8** Transform sub-block scan patterns for the different transform sizes. **a** 32x32, **b** 16x16, pixel diagonal scan. For 8x8 transformations two scan patterns for transform sub-blocks are available: One for diagonal and vertical scan (**c**) and one for horizontal scan (**d**).

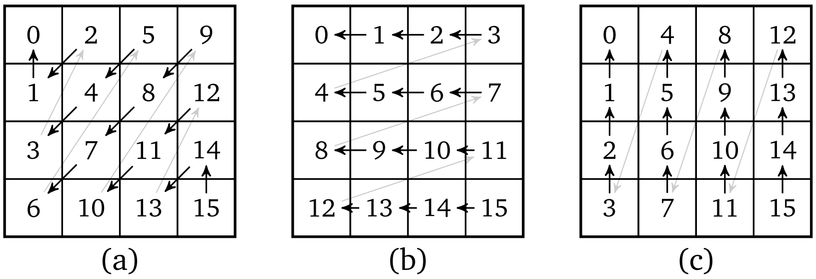


**Figure 2.9** The three scan patterns for the transform coefficients within each transform sub-block. Three scan patterns are available: **a** diagonal scan, **b** horizontal scan, **c** vertical scan.

### Scan Order

The coefficients of each transform block are processed in a specific order. This scan order is sorted from the highest frequency coefficient on the bottom right to the lowest frequency coefficient on the top left. Firstly, if the transform block size is larger than 4x4 coefficients, the transform block is split into transform sub-blocks of 4x4 coefficients each. These transform sub-blocks are then processed from highest to lowest frequency in a diagonal arrangement depending on the transform size (see Figure 2.8 a to c). The 16 transform coefficients within each transform sub-block are in turn scanned using a diagonal order (Figure 2.9 a).

For transform blocks of size 8x8 and 4x4 within a coding block that uses intra prediction, two additional scan orders are defined: Horizontal and vertical scan. One of the three modes is selected according to the applied intra prediction direction. If the intra prediction direction is approximately horizontal or vertical, horizontal or vertical scanning is employed, respec-

[8] Coding of transform coefficient levels becomes more critical at high bitrates because in this case, the coefficient level values tend to be large and thusly most of the bitrate is consumed by them. In H.264/AVC, the worst case number of context coded bins per coefficient is 15.

tively. Otherwise, the default diagonal scan is used. For 8x8 transform blocks, again, the transform block is split into 4 sub-blocks which are scanned according to Figure 2.8 c (vertical) and d (horizontal). In this case, the coefficients within each transform sub-block are also scanned in a horizontal or vertical manner as depicted in Figure 2.9 b and c.

### Last Significant Position

The first syntax element that is coded into the bitstream signals the position of the last non-zero coefficient in scan order. A non-zero coefficient is also referred to as a significant coefficient. Because it can be expected that a lot of the high frequency coefficients are zero, large sequences of not significant high frequency coefficients can efficiently be signaled this way. The position is coded by signaling the x and y position of the last significant coefficient within the transform block.

The x and y positions are coded using the same binarization scheme but separate sets of CABAC contexts. Coding is performed using a unary coded prefix part and a suffix part which uses fixed length coding. The number of bits in each of these parts depends on the coded value and the size of the transform. The maximum number of bits is 9 for the prefix and 3 for the suffix part. Each bit of the prefix part is coded using context adaption while the suffix bits are coded using bypass coding. For details on how the 36 contexts for the last significant position are assigned see Appendix A.1.

### Coefficient Scans

Next, the coefficients are scanned sub-block by sub-block according to the selected scanning order. For every sub-block, the coefficient values are coded using multiple scans of the coefficients within the sub-block. First a coded sub-block flag indicates if the sub-block contains any significant coefficients. The flag is coded using two contexts each for luma and for chroma. Coding of the sub-block flag is omitted for the sub-block that contains the previously coded last significant position because this position already implies that there is at least one significant coefficient in the sub-block. Also, due to the nature of the transformation, it is very likely that the sub-block 0 (containing the DC coefficient) contains at least one significant coefficients. Because of this, the coded sub-block flag for sub-block 0 is also not coded and inferred to one.

If the coded sub-block flag is set, the coefficients within the sub-block are scanned and a significance flag is coded for each of the 16 coefficients. This flag indicates if the coefficient is zero (not significant) or greater than zero (significant). If the last significant coefficient is located within the current sub-block, the scan is started from its known position. A total of 42 contexts are assigned for coding the significance flag (27 for luma and 15 for chroma). This way, up to 16 significance flags per sub-block are coded using context adaption.

In the second scan, all coefficients in the sub-block that were marked as significant in the first scan are revisited in the same selected scan order. For each coefficient that was marked as significant in the first scan, a second flag indicates if the level is greater than 1 or not.

**Table 2.1** The maximum number of context coded bins for each transform block depending on the size of the transform. In the last line, the number is normalized by the number of coefficients in the transform.

| | 4x4 | 8x8 | 16x16 | 32x32 |
|---|---|---|---|---|
| Last Significant Position | 6 | 10 | 14 | 18 |
| Coded Block Flag | 0 | 2 | 14 | 62 |
| Significance Flag | 15 | 63 | 255 | 1023 |
| Greater One Flag | 8 | 32 | 128 | 512 |
| Greater Two Flag | 1 | 4 | 16 | 64 |
| Sum | 30 | 111 | 427 | 1679 |
| Per Coefficient | 1.88 | 1.73 | 1.67 | 1.64 |

For this flag, 24 contexts are assigned (16 for luma and 8 for chroma). In order to limit the number of context coded bins per sub-block, a maximum of 8 coefficients per sub-block are signaled as being greater than one. If more coefficients are greater than one, this information is carried to the later coding of the remaining level.

In the third scan, the last coefficient in this sub-block that was marked as greater one in the previous two scans is revisited again and a flag is coded that indicates if the coefficient level is greater than two or not. Also for complexity reasons, at most one of these flags is coded per sub-block. A total of six contexts are used for coding of this flag (4 for luma and 2 for chroma).

In the worst case, up to 1679 bins need to be coded using context adaption for a transformation of size 32x32. This corresponds to a block in which each sub-block carries at least eight significant coefficients from which one is also greater than one. In addition, the last significant coefficient must be placed in the bottom right of the block. If, however, the worst case number of context coded bins is scaled by the number of transform coefficients, it can be seen that the number of context coded bins per coefficient is similar for all transform sizes and ranges from 1.64 for a transform size of 32x32 to 1.88 for a transform size of 4x4 (see Table 2.1). Because these numbers apply to all three color components, the aggregated worst case number is 2.81 context coded bins per pixel. For a Full HD (1920x1080) video sequence with 24 pictures per second, this adds up to a worst case value of approximately 124 million context coded bins per second for the residual information only. Although this is a significant improvement compared to H.264/AVC where this number is nine times higher, the transform coding can still pose a throughput bottleneck, especially for devices with limited resources and high bitrates. While it is rather unlikely that these worse case scenarios occur, they are very relevant for the decoder because a decoder that claims HEVC compatibility must be designed in a way to handle these situations correctly. Furthermore, the implications can become even worse in scenarios with more than one layer and therefore multiple coded transforms.

For more details on how the contexts for the various above-mentioned symbols are selected, please refer to A.2. For a detailed assessment of the entropy coding implementation in HEVC as well as a close comparison to H.264/AVC, the reader is referred to [SB12].

### Sign Information and Remaining Level

In the final step, the sign information as well as the remaining level of the sub-block coefficients are coded. First, the sign information for all coefficients that were marked as significant in the first scan are signaled. Because positive and negative signs are equally probable, the sign bits are coded without context adaption in bypass mode. Secondly, the remaining level of the coefficients marked as greater than one or greater than two have to be transmitted. For the remaining level, no context based coding is used. Rather, a Golomb-Rice code for small values is combined with an Exp-Golomb code for larger values, where all bits of these codes are coded in bypass mode. This mixed code can be adapted to varying distributions of remaining levels using a parameter $k$. For a low $k$, the code is tuned for a high probability of small values. For higher $k$ values, the code adapts to flatter distributions. For each sub-block the value of $k$ is initialized to 0 and it is increased if a remaining level higher than a certain threshold is coded.

For a more detailed explanation of these codes as well as the general coding of transform coefficients, the user is referred to [Wie15; Sol+12].

## 2.3.6 Loop Filter

As illustrated in Figure 2.3, after the reconstruction of the frame and before it is stored into the picture buffer (DPB), in-loop filtering is applied to the frame. In HEVC, two successive filters are implemented: A deblocking filter and sample adaptive offset (SAO). The deblocking filter is similar to the corresponding filter in H.264/AVC and is designed to reduce the artifacts at block boundaries which occur because of the block based coding approach. All PU and TU boundaries in the frame are processed first in horizontal, then in vertical direction. Depending on the sample values along each edge, weak or strong filtering is applied. In case of weak filtering, only the two boundary pixels are modified while for strong filtering, 3 pixels on each side of the boundary are filtered. Following the deblocking operation, the SAO filter is executed. As the name suggests, it operates on a sample basis and conditionally adds an offset to each sample. For each CTU, one of two modes is selected. For the first mode (edge offset), the offset is selected depending on the surrounding sample values (the edge). In the second mode (band offset), only the magnitude of the sample value itself determines the offset. In smooth areas and around edges, the SAO filter can thus further enhance the reconstruction quality. Control parameters for both filters as well as the SAO offset values are signaled in the bitstream. While both filters are highly parallelizable, they are still quite complex and introduce an additional coding delay [Nor+12; Fu+12].

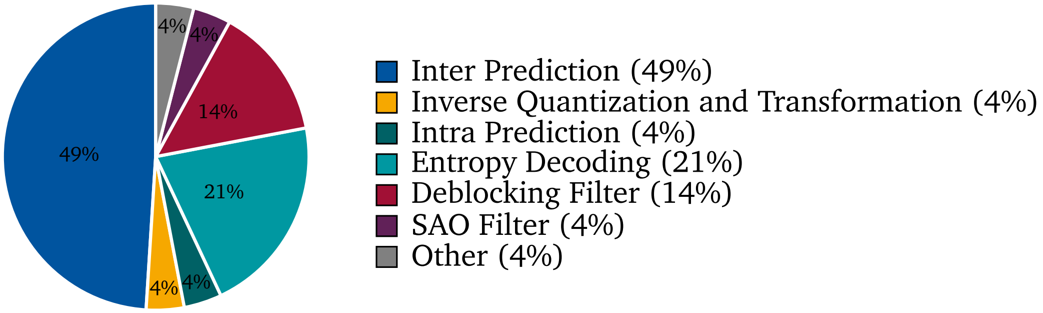


**Figure 2.10** Average relative time spent per decoding operation. Random access decoding for an optimized x86 based software decoder. Source: [Bos+12]

### 2.3.7 Complexity

One concern in the standardization process is the expected complexity with regard to the decoder as well as the encoder. Of course, there is not one universal value that can represent the complexity of HEVC in its entirety. The specific implementation in software or hardware as well as the particular application play a major role for all complexity considerations. For example, memory requirements may be a limiting factor in mobile applications while parallelization is more important in software implementations. One simple measurement that is often employed to give a first indication on the complexity is a measure of the decoding time for a given software decoder. As elaborated in Section 2.3, the HM reference decoder is not well suited for this case. In [Bos+12], an x-86 platform based optimized HEVC decoder is profiled. The decoder does not use multi-threading but heavily utilizes SIMD instructions (single instruction multiple data). The authors do not claim that the decoder could not be further optimized. Especially multi-threading support could further reduce the decoding time of certain operations relative to others. Nevertheless, these profiling values can give a rough idea about the complexity of the necessary decoding operations.

Figure 2.10 shows the average relative time consumption of the aforementioned decoding steps. A set of HD sequences from the common test set with various bitrates and frame rates were used in this test. It can be seen that for the used random access configuration, the motion compensation (inter prediction) operation is the most time consuming, taking up almost half of the entire decoding time. Only 4% of the time is allocated to intra prediction and the inverse quantization and transformation operations, respectively. The two remaining big chunks of the decoding time are used for entropy decoding (21%) and the in loop filtering operations (18%). Depending on the application, the amount of motion compensation operations and the accompanying memory bandwidth can become a bottleneck. At the same time, the motion compensation operation is also highly suitable for implementations using parallel processing. Depending on the bitrate of the stream, also the entropy coding block can be a limiting factor because it only allows for a limited amount of parallelization. The coding complexity becomes even more important for scalable approaches, where not only one video sequence is coded but multiple representations of a video are combined in one bitstream. Depending on the implementation of scalability and the configuration, the decoder complexity may significantly increase in this scenario. In the worst case, each additional representation may add the complexity of another full decoder [9].

## 2.4 Scalable Video Coding

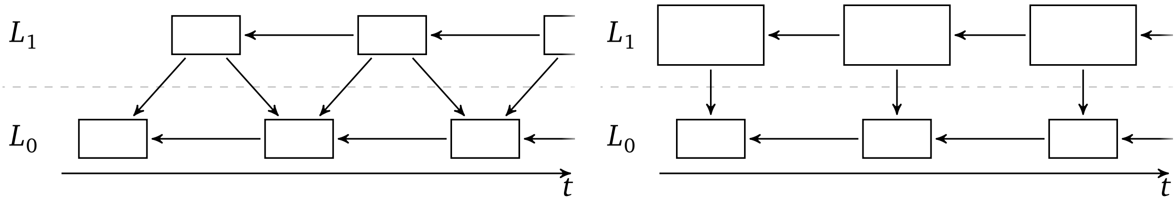
The fundamental goal of scalable video coding is to define a bitstream format that is able to adapt to different application scenarios and varying transmission conditions. Former video coding standards that feature scalable profiles are for example MPEG-2, H.263 and MPEG4-Visual. However, the implementation of the scalability features in these standards comes at the price of significantly increased complexity and reduced coding performance. After the standardization of H.264/AVC was finished in 2003, a scalable extension to this standard was developed as well. It was appended to the H.264/AVC standard in July 2007 and is also known as scalable video coding (SVC). The objective in the standardization of SVC was to add scalability features to H.264/AVC, while the complexity overhead is small and the performance is similar compared to single layer coding of H.264/AVC [h.264/AVC; SMW07; SW08].

In SVC, the bitstream is organized in layers, where each layer corresponds to a possible reconstructed video sequence with specific properties. The lowest layer, which is also referred to as the base layer, is a conventional non-scalable H.264/AVC bitstream which can be decoded by legacy decoders that do not support the scalable extension. From this base layer, the extension allows for the definition of higher layers that can use the base layer information for more efficient coding. In scalable coding, these additional layers contain further refined representations of the same video sequence as the base layer. In this context, they are referred to as enhancement layers. In SVC, several types of scalability are defined. They are specified by the difference between the representations of the layers.

**Temporal Scalability** In temporal scalability, the number of decodable frames per second increases in the enhancement layer. Because of the flexible coding order and prediction structure, the foundation to enable temporal scalability is already established in H.264/AVC. SVC merely extends it by signaling a temporal ID in the header of each frame which corresponds to one reconstruction of the video at a certain frame rate. For example, the enhancement layer could double the frame rate by adding an additional frame between every two frames in the base layer (see Figure 2.11a). SVC further adds the possibility to combine temporal scalability with other types of scalability.

**Spatial Scalability** In Spatial scalability, the spatial resolution of the enhancement layer increases compared to the base layer. For example, the bitstream could contain two versions of the same video, one with a resolution of 1920x1080 pixels and the other with 3840x2160 pixels (see Figure 2.11b). Generally, the reconstruction quality of the enhancement layer is not fixed to the same value as the lower layer and may also be varied. The encoder can freely choose how much bitrate to allocate for each of the layers.

[9] While the worst case performance increase is high, it should be noted here that parallel decoding of the multiple layers could be applied.



(a) Temporal scalability. The enhancement layer doubles the frame rate of the base layer.

(b) Spatial scalability. The width and height of the frame increases for the enhancement layer.

**Figure 2.11** Example prediction structures for temporal and spatial scalability with two layers. The base layer is denoted as $L_0$ and the enhancement layer as $L_1$.

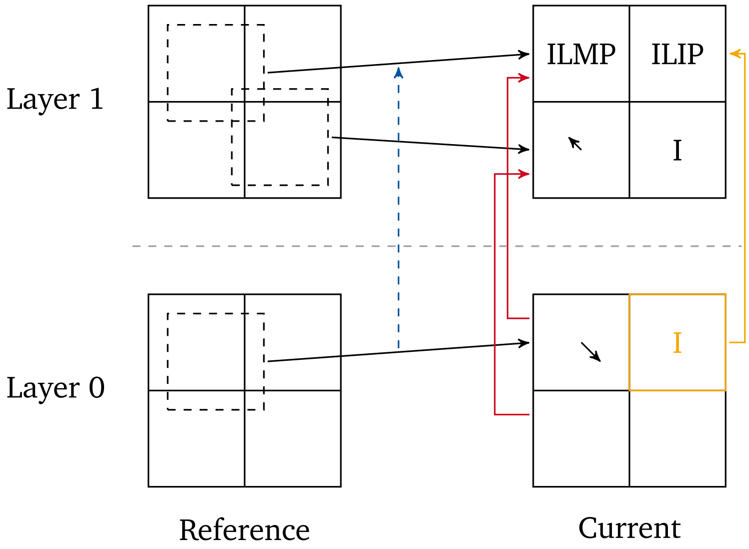


**Figure 2.12** Inter layer prediction in SVC with two layers. The four blocks in the higher layer of the current frame use inter layer motion prediction (ILMP), inter layer intra prediction (ILIP), conventional inter prediction and intra prediction. Optionally, inter layer residual prediction may be employed (red arrows).

**Quality Scalability** Here, the reconstruction quality of the enhancement layer increases compared to the base layer, while the spatial resolution of the layers is identical. While conceptually this type of scalability can be regarded as a special case of spatial scalability with a resolution ratio of 1, it is internally handled very differently to spatial scalability. It is also referred to as signal to noise ratio (SNR) scalability or fidelity scalability. Typically, the enhancement layer quantizer is operated using a smaller QP value (smaller quantization step size), which results in a higher reconstruction quality of the enhancement layer.

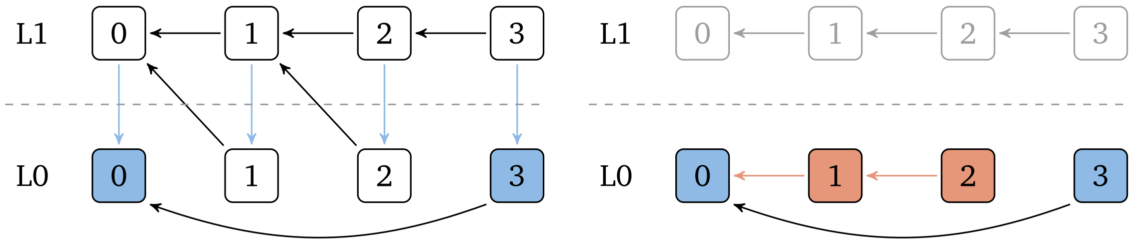
From a compression point of view it depends on the specific application what the optimum layout of the layers should be. For example, depending on the bitrates, it might increase the overall compression performance when the lower layer of a scalable sequence is downsampled instead of using multiple layers with SNR scalability. For high bitrate scenarios, SNR scalability can prove more efficient. This optimal layout, however, even depends on the properties of the sequence itself.

Up to a certain extent, one SVC bitstream can have multiple layers with different types of scalability. For each enhancement layer, the basic macroblock coding process of H.264/AVC is reproduced with certain additions to enable interlayer prediction. An illustration of possible predictions in the higher layer of the current frame are shown in Figure 2.12. There are two types of inter layer prediction, that exploit the lower layer information for an increased overall coding performance. For the top left block of the current layer 1 frame, the corresponding block in the lower layer uses inter prediction. In this case, the motion information can be derived from the lower layer. The motion compensation operation is then performed using the higher layer reference frames. This is also referred to as interlayer motion prediction. For the top right block, the corresponding block in the lower layer uses intra prediction. The intra reconstruction signal from the lower layer is filtered and potentially upsampled to form the interlayer intra prediction signal in the higher layer. In any case, the encoder can choose to disregard the lower layer information and perform conventional inter or intra prediction as depicted for the two lower blocks. Finally, also the residual signal of inter coded blocks in the higher layer can be predicted from the residual signal in the lower layer using interlayer residual prediction.

When, in addition, constrained intra prediction is enabled in the lower layer, the specific design of the inter layer prediction methods in SVC allows for a reconstruction of the higher layer frames without access to any reconstructed inter predicted pixel values in the lower layer. This means that it is not necessary to perform motion compensation in the lower layer. Only blocks that utilize constrained intra prediction in the lower layer require reconstruction. This in turn implies, that there is no need to apply loop filtering and to store the lower layer reconstructed frames in the decoded picture buffer. Only the higher layer pictures must be decoded, filtered and stored. This enables reconstruction of the higher layer video sequence with a significantly reduced complexity because the decoding loop in the lower layer can be suspended. Since only one loop in the higher layer is running, this is also referred to as single loop decoding as opposed to multi loop decoding where one decoding loop is required in each layer. The same applies to scenarios with more than two layers in SVC, where only the loop in the target layer must be operated.

In H.264/AVC, the bitstream is organized in network abstraction layer (NAL) units. Each NAL unit starts with one header byte which signals the type of the NAL unit. In SVC, the header is extended by an additional three bytes, which provides information for the scalable decoder like the layer index that the payload data is associated with. This enables a very simple extraction of valid SVC sub-streams of lower bitrate as well as lower quality and/or spatial or temporal resolution. Since only specific sub-streams at certain bitrates can be extracted using the aforementioned inter layer prediction approach, this is also referred to as coarse grain scalability (CGS). For an even further increased bitrate adaption flexibility and improved error resiliency, SVC features another quality scalability mode which is referred to as medium-grain quality scalability (MGS). For MGS, any NAL unit can be discarded from the bitstream, which can be used for highly accurate adaption to varying channel conditions. [10]

[10] While NAL units can be discarded from the MGS bitstream randomly, it is much more efficient to sort the NAL units according to their impact on the reconstruction quality of the resulting sub-stream and discard them with respect to this "importance". For this purpose, the SVC NAL unit header contains a *quality_id* syntax element [Zha14].



(a) Decoding of the higher layer (L1). All L1 pictures can be correctly decoded.

(b) Decoding of the lower layer (L0). Drift for non-key pictures (1 and 2).

**Figure 2.13** Exemplary prediction structure for MGS coding. Pictures 0 and 3 are key pictures (blue) and only predict from other key pictures. Frames 1 and 2 in the lower layer predict from L1 references. If only the lower layer is decoded, drift occurs (red) for frames 1 and 2.

### 2.4.1 MGS and the Key Picture Concept

For MGS, so called key pictures are introduced. For each frame, a flag is transmitted that indicates if the lower layer or the higher layer reconstructed picture of a reference frame is used for motion compensated prediction (*use_ref_base_pic_flag*). Another flag indicates if the reconstructed frame of the lower layer needs to be stored in the picture buffer (*discardable_flag*). Figure 2.13 illustrates MGS based coding for two layers. For the frames 1 and 2 in 2.13a, prediction is performed from the references in layer 1 (*use_ref_base_pic_flag* is not set) but full reconstruction and storage of these frames in the picture buffer is not required (*discardable_flag* is set). The key pictures 0 and 3 (shaded in blue), however, only use prediction from other key pictures in layer 0 (*use_ref_base_pic_flag* is set) and must be decoded and stored in the picture buffer (*discardable_flag* is false). The prediction from the higher layer frames can further increase the overall prediction performance while the higher layer video can be reconstructed using single loop decoding.

If, however, frames from the higher layer can not be decoded because they were discarded from the bitstream, the lower layer representations of the frames 1 and 2 can only be reconstructed using the available lower layer references. This situation is illustrated in Figure 2.13b. The decoder is not able to reconstruct the signal that was employed for prediction of the successive frames at the encoder side, but only a similar version from the lower layer at a lower quality. The phenomenon is also referred to as drift because the encoder and decoder reconstructions differ and tend to "drift" apart over time. Because key pictures can always be reconstructed correctly, they serve as a boundary for the drift. If only a few higher layer packets are discarded, a decoder can perform a sort of best effort decoding using all the NAL units that are available. The key picture concept as well as the implications of the drift are further detailed in the concept of scalable coding for HEVC in Section 3.1.

SVC was standardized with the goal of a scalable extension to H.264/AVC which has equal or only slightly increased complexity and equal or only slightly lower performance when compared to single layer coding using H.264/AVC. In order to achieve this goal for a variety of different applications, SVC features several different operation modes and a variety of different prediction and coding modes. Using SVC, it is possible to perform spatial and SNR scalability with only a slight complexity overhead while at the same time the bitrate increase can be as low as 10% compared to single layer H.264/AVC [SMW07]. However, the standardization activity for SVC was started after the corresponding work on H.264/AVC was finished. This lead to some drawbacks for the design of SVC because the H.264/AVC specification could not be changed in retrospect. For example, the NAL unit header needed to be extended in order to allow quick access to basic scalable information. However, only specific types of NAL units could be further extended without breaking compatibility with non scalable streams.

## 2.5 Scalable High Efficiency Video Coding (SHVC)

From the very beginning of the development of HEVC, possible (future) extensions were kept in mind for the design. Also, contrary to the scalable extension of AVC (SVC), the standardization activity of the scalable extension to HEVC was initialized in parallel to the still ongoing standardization of HEVC itself.[11] This time overlap allowed for some interaction between the standardization activities. This way, decisions for the scalable extension could be considered in the design of HEVC. The same approach was taken for the other extensions of HEVC, namely the multiview extension (MV-HEVC), the range extensions (RExt) and the 3D extensions (3D-HEVC). Thus, situations as for SVC were avoided, where some suboptimal implementations were required in order to enable scalability while maintaining backwards compatibility to AVC.

The implementation of the scalable extension to HEVC is closely related to the multiview extension of HEVC (MV-HEVC). Both extensions use a layered approach, where the lowest layer is formed by an HEVC main profile compatible bitstream. From this base layer, the extensions allow for the definition of further layers that can use the base layer information for coding. In SHVC, these additional layers contain further representations of the same video sequence as the base layer. In this context, they are referred to as enhancement layers. In MV-HEVC, the layers encode additional views onto the same scene.[12] Because of this close relationship between the extensions, the common specifications for the multi-layer extensions are merged into a separate annex of the standard text (Annex F - Common specifications for multi-layer extensions) [h.265/HEVC].

In Annex G of the HEVC standard text, the SHVC specific modifications are described. In SHVC, several types of scalability are defined. They are specified by the difference between the representations of the layers. However, an SHVC bitstream is not limited to a single specific kind of scalability and all modes of scalability can be combined in one SHVC bitstream.[13] Besides the aforementioned scalability types (spatial scalability, temporal scalabil-

[11] The joint call for proposals on a scalable extension was issued in July 2012, while Version 1 of the HEVC draft was approved in January 2013.

[12] One of the most widely spread applications of multiview coding is Stereoscopic Video, where two views of the same scene are encoded. On the playback side, the two views are then presented to the left and right eyes separately to create a stereoscopic viewing experience. While this is the most common application, the number of layers is not limited to two and various other applications are conceivable.

[13] While there are no limitations to the combination of scalability types by the underlying design, there are some limits specified by the profiles and levels. Also, by design, for hybrid coding scalability only the base layer can be provided by external means.

ity and quality scalability), some additional types are considered:

**Bit Depth and Color Gamut Scalability** The layers can also differ in bit depth and color representation. SHVC adds special inter layer processing tools for these applications in order to increase the overall coding performance compared to a scalable approach that does not consider the change of color space and bit depth between the layers.

**Hybrid Coding Scalability** The SHVC standard allows for the lower layer to be "provided by an external means not specified in [the] Specification" [h.265/HEVC]. This enables scalability scenarios where the lowest layer is coded using a coding standard other than HEVC. The base layer could for example be coded using H.264/AVC, MPEG-2 or any other video coding standard. In particular in broadcast applications, this can provide backwards compatibility to existing transmission systems.

### 2.5.1 Decoder Design

As detailed in Section 2.4, there are different approaches in order to achieve scalability in a hybrid video coding scheme. During the time of the standardization process, several options were discussed and investigated. Although the corresponding complexity implication were known, the final decision was to implement a pure multi-loop approach with some specific inter layer processing tools that only requires high-level modifications to the decoding process. The decoder of a non-base layer in SHVC closely resembles a conventional HEVC decoder and all changes to the HEVC decoder are restricted to the slice header level or above. No modifications to the lower layer coding tools like motion compensation, intra prediction, residual coding or the frame buffer are applied in SHVC. From the decoder implementation point of view, this implies that an SHVC decoder for a higher layer can be implemented with only changes to the higher level decoding implementations of an existing HEVC decoder. However, also with this approach, some changes to the multilayer encoder and decoder are necessary. In the higher layer decoder, the parameter parsing process must be changed, access to the lower layer reconstruction and motion information must be enabled and the presented inter layer processing tools must be added. The notion of this was to simplify the extension of existing hardware and software implementations of HEVC to support SHVC.

As discussed in Section 2.4, such an approach also has severe implications for the decoder complexity. If a specific layer in this multi-loop approach is selected to be decoded, all lower layers that the current layer relies on have to be fully decoded as well. For example in the worst case scenario of SNR scalability, the decoder complexity directly scales with the number of layers. For 2 layers, the decoder complexity is approximately doubled, for 3 layers it is approximately tripled. Furthermore, the specified inter layer processing tools also constitute an additional complexity overhead. While this increase in decoder complexity can be tolerated in some applications, it can also make SHVC unattractive for other use cases. This is further detailed in Section 2.5.4.

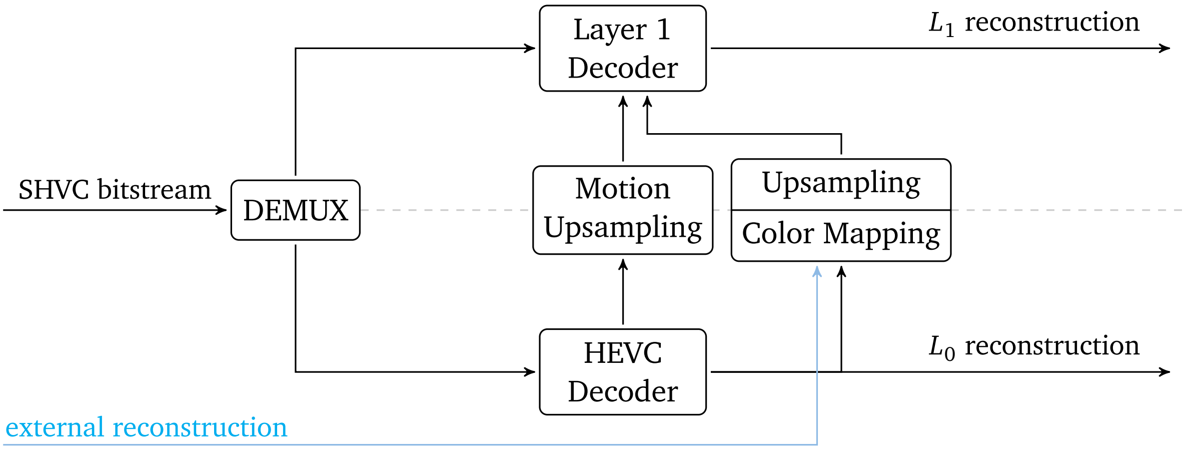


**Figure 2.14** SHVC decoder layout with two layer. $L_0$ is required to be HEVC compliant. Between the layers, upsampling and color mapping of the reconstruction samples can be performed as well as upsampling of the motion information. The $L_0$ reconstruction can also be provided by an external decoder in which case no motion upsampling is performed.

Figure 2.14 illustrates the conceptual layout of the SHVC decoder. While in this example a two layer SHVC decoder is depicted, the concept can be easily extended to more layers. The syntax allows for up to 63 layers, while the number of layers that a certain layer depends on is limited by the profiles. The incoming SHVC bitstream is demuxed and each layer stream is passed to the corresponding layer decoder. The $L_0$ bitstream is compliant to the HEVC main profile, which allows for backwards compatibility. If an SHVC bitstream is passed to an HEVC decoder that adheres to the main profile, the HEVC decoder is required to ignore all enhancement layer NAL-units and decode only the $L_0$ sub-stream.

Three types of inter layer processing are specified in SHVC. For the lower layer reconstruction, color mapping can be applied followed by spatial upsampling. Furthermore, the lower layer motion information can be upsampled by the same ratio as the reconstructed pixel values. The resulting picture at enhancement layer resolution can then be used in the enhancement layer decoder for prediction. In SHVC, this is achieved by simply adding the upsampled picture as an additional reference picture to the list of reference pictures. The higher layer decoder can now use the conventional HEVC prediction tools to take advantage of the lower layer information. In particular, these tools are motion compensated prediction and temporal motion vector prediction (TMVP). The motion vector for prediction from a lower layer reference picture is forced to the zero motion vector (0,0). This restriction is applied in order to limit the additional complexity which is implied by the inter layer prediction process. It also corresponds to the consideration that the lower layer reference is a different representation of the identical picture at the same time instance so, by design, there is no motion between the two layers.[14] The different types of inter layer processing are detailed in the following:

[14] While all motion vectors for inter layer prediction must be zero in order for the bitstream to comply with the SHVC standard, all of these zero motion vectors are still coded into the bitstream. This is done to abide by the high-level syntax approach which dictates that no modifications to the lower layer syntax should be applied. However, the overhead for coding of these zero motion vectors is very small.

**Texture Resampling and Cropping** For resampling of the lower layer reconstruction, SHVC specifies an 8 tap 1D filter to interpolate 15 positions between each two luma pixel positions (1/16 pixel precision). For the chroma components, the corresponding filter has 4 taps. The two dimensional image is processed by filtering in horizontal direction first, followed by filtering in vertical direction. By rounding to the next 1/16 the pixel position, arbitrary ratios between the two layers are supported in each direction. The filter coefficients were designed using the same concept as for the motion compensation filters. This also means that the complexity, specifically the number of memory access and arithmetic operations, are comparable to motion compensation. For instance, the luma interpolation filters for the positions 4/16, 8/16 and 12/16 are identical to the motion compensation filters with a fractional offset of 1/4, 2/4 and 3/4. Furthermore, the resampling process can handle bit-depth scalability and convert 8-bit input to a 10 bit output. Finally, a modification of the horizontal and vertical upsampling filters phase for luma and chroma can be signaled in the bitstream. This is especially beneficial for the use case of scalability from an interlaced lower layer to a progressively scanned enhancement layer and for chroma format scalability (e.g. from a 4:2:0 layer to a 4:4:4 enhancement layer). While only the use of the 4:2:0 chroma format is permitted in the scalable profiles, all tools for chroma format scalability are available so that it could easily be added in the future.

Similar to SVC, SHVC uses cropping parameters to enable flexible inter layer cropping. These parameters are coded in the multi-layer picture parameter set extension of the higher layer. They can be signaled individually for each reference layer and can be updated for each picture. Figure 2.15 illustrates the usage of the cropping parameters. Use cases for this very flexible cropping system include scenarios which use pan and scan in the base layer as well as situations where a dynamic region of interest coding is performed.

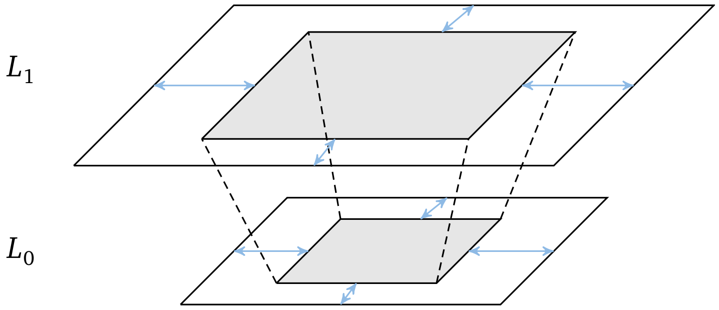


**Figure 2.15** Cropping in SHVC. Four parameters in each layer (blue arrows) specify the number of pixels to crop from each side of the picture.

**Motion Vector Resampling** As described in Section 2.3, HEVC allows the prediction of motion vectors not only from the spatial neighborhood, but also from the motion vector field of a reference picture. Since the lower layer reference picture in SHVC is handled as another reference picture, the TMVP mechanism can be reused for inter layer motion prediction without any changes to the coding process. As in HEVC, the "temporal" predictor is retrieved and scaled according to the POC of the current picture, the POC of the reference picture used for motion prediction, the POC of the lower layer picture and the POC of the picture that the TMVP candidate references.[15]

[15] Technically this is not a temporal prediction but an inter layer prediction. So the POC of the current picture and the POC of the picture used for TMVP are identical (the temporal distance is zero).

In case of SNR scalability, the motion vector field of the lower layer picture can be directly utilized in the enhancement layer. However, if texture resampling and/or cropping between the two layers is performed, the motion vector field has to be resampled and/or cropped accordingly. An inter layer motion vector field mapping (MFM) function is applied to generate the motion vector field that is then used in the enhancement layer TMVP process. As stated in Section 2.3, the motion vector field of all reference pictures is saved in the reference picture buffer in compressed form. A motion field compression algorithm is used to sub-sample the motion field so that one motion vector is saved per block of 16x16 pixels. For backward compatibility reasons, the input to the MFM process is the sub-sampled motion field of the reference picture. Furthermore, also the output of the MFM is a motion vector field with one motion vector per 16x16 pixels in the enhancement layer resolution.

The MFM function considers the spatial relationship between the two layers as well as the cropping parameters which are used for texture resampling. In case of hybrid coding scalability, SHVC specifies no technique to provide a motion field by external means. In this case motion vector resampling is disabled and no motion prediction from the lower layer is employed.

**Color Mapping** If the current layer and the reference layer use different color spaces (color gamut scalability), a color mapping function can significantly increase the inter layer prediction performance [Boy+16]. For this application, SHVC specifies a 3D look up table (LUT) based color mapping approach. The three dimensional YCbCr color space can be split into cuboid partitions. For each of these partitions, the parameters for a linear mapping from the reference layer color space to the color space of the current layer are transmitted. If color mapping is used in combination with texture resampling, the color mapping operation is performed first for reasons of computational complexity. For a more detailed explanation, the reader is referred to [Boy+16; BAZ13].

### 2.5.2 Common Testing Conditions

In order to enable an objective performance evaluation of a specific modification or addition to SHVC and to make the results comparable to other approaches, a well-defined testing environment was established. For SHVC, the "*Common SHM test conditions and software reference configurations*" document [SH14] provides exactly this. The document defines a set of sequences to be used as well as different coding structures, spatial relationships between the layers (spatial ratios) and various settings for the quantization parameter (QP) of each layer. While some minor deviations from these conditions are necessary in the remainder of this thesis, we try to comply to them as closely as possible in order to have a common reference with reports from other experiments. In the following, an overview of these conditions is given.

For all experiments, the number of layers is set to two. The sequences are split into two classes according to their spatial resolution (class A and class B). Table 2.2 provides a list

**Table 2.2** The used test sequences, with their corresponding spatial resolution, frame rate and number of frames.

| Class | Sequence | Resolution | Frame rate (Hz) | Nr Frames |
|---|---|---|---|---|
| A | Traffic | 2560x1600 | 30 | 150 |
| | PeopleOnStreet | 2560x1600 | 30 | 150 |
| B | Kimono | 1920x1080 | 24 | 240 |
| | ParkScene | 1920x1080 | 24 | 240 |
| | Cactus | 1920x1080 | 50 | 500 |
| | BasketballDrive | 1920x1080 | 50 | 500 |
| | BQTerrace | 1920x1080 | 60 | 600 |

of all sequences.[16] Three combinations of spatial resolution for the two layers are defined. These are often referred to as the type of scalability.

- SNR: For SNR scalability, the spatial resolution of both layers is identical. This applies to both class A and B sequences.
- 2x: For 2x scalability, the resolution of the base layer (BL) is halved compared to the resolution of the enhancement layer (EL). The spatial resolution of the BL is 1280x800 pixels for class A sequences and 960x540 pixels for class B.
- 1.5x: For 1.5x scalability, the resolution of the EL is 1.5 times the resolution of the BL. This spatial relationship is only tested for the five class B sequences. The spatial resolution of the BL of the class B sequences is 1280x720 pixels.

The HEVC and SHVC standards do not enforce a specific coding order for the pictures or a fixed scheme to determine the pictures which are used for reference in each picture. While the standard defines certain limitations to the number of references and the size of the reference picture buffer, the decision which picture to code next and which pictures to use as reference is left to the encoder implementation. On the one hand this approach enables the encoder to adapt to specific application scenarios or even to the sequence being coded. On the other hand it also complicates the comparison of results significantly if the two candidates employ different coding structures. That is why the common testing conditions also define fixed prediction structures to be tested. Each prediction structure includes a fixed coding order, as well as a fixed list of reference pictures to be used for each picture. The three coding structures to test were all designed with different applications in mind and are described in the following.

**All Intra (AI)** In the all intra configuration, the slice type of all slices in all pictures is set to intra, and so all coding units are coded using intra prediction. Motion compensated

[16] The original versions of the two class A sequences actually have a resolution of 3840x2048 pixels and contain 300 frames. However, in order to limit the encoding time for these sequences, it was decided to crop and cut them for the SHVC common testing conditions.

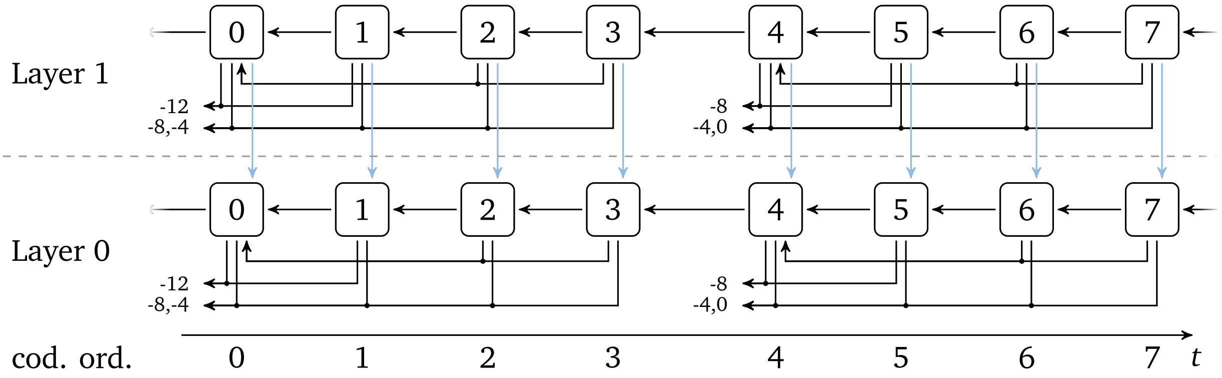


**Figure 2.17** Two GOPs of the low delay prediction structure for 2 layers and a GOP size of 4. The t-axis shows the coding order. Each frame and their references are indexed by POC.

prediction is completely switched off. Prediction from a lower layer, however, can still be employed (see Figure 2.16).

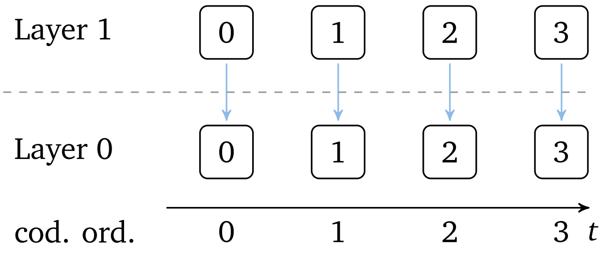


**Figure 2.16** Example of the all intra configuration for 4 frames. Each picture can be decoded independently of previously coded pictures. The t-axis shows the coding order.

On the one hand, since there is no dependency between pictures on the time axis, all pictures can be encoded and decoded independently. In production environments, this allows for frame precise cutting and joining of sequences. For the same reason, there is no error propagation in case of a decoding error, which makes the all intra configuration interesting for applications with high quality requirements (e.g. digital cinema and professional production environments). On the other hand, the overall coding efficiency of the all intra configuration is considerably diminished compared to configurations that utilize motion compensated prediction between frames over the time axis.

**Low Delay (LD)** As the name suggests, the low delay configuration aims to minimize the structural delay of the overall transmission system (encoding and decoding). This is achieved by aligning the temporal order of the pictures and the coding order.[17] Motion compensation is enabled, but due to the coding order, only prediction from pictures in the temporal past is performed. Furthermore, only the first picture of a video sequence is coded as an intra picture so decoding can only start at the very beginning of the sequence.

The most common application for this type of configuration are real time video streaming systems like video telephony or time critical surveillance applications. Figure 2.17 displays the prediction structure as it is specified in the common testing conditions. The sequence

[17] In this context the temporal order refers to the order in which the pictures were recorded and are meant to be displayed on the decoder side.

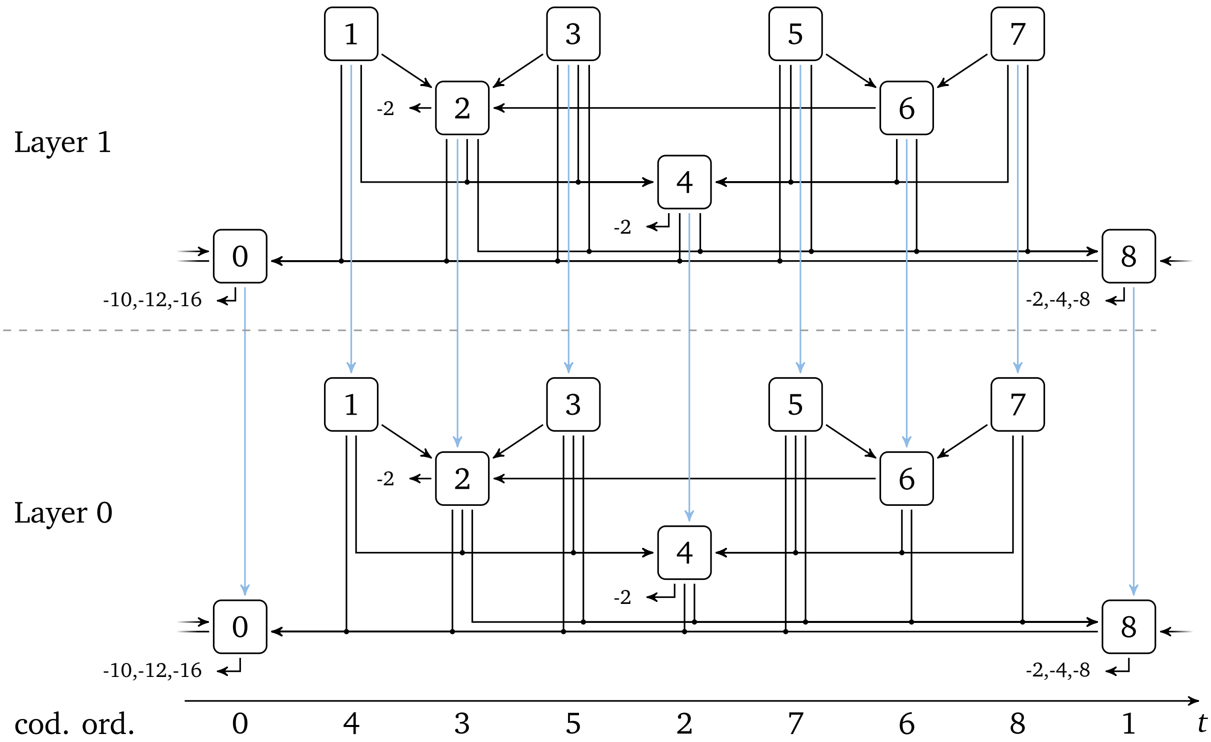


**Figure 2.18** One GOP of the random access configuration with a GOP size of 8 pictures. The t-axis shows the coding order. The y-offset illustrates the hierarchical level of each frame within the coding structure. References to frames in the previous GOP's are denoted by negative POC numbers.

is split into groups of 4 pictures (GOP) with the coding order and the display order being equal. Each picture in a GOP predicts from its previous picture and from the first picture of the previously coded two GOPs. The first two pictures of each GOP additionally predict from the first picture of the GOP which was coded 3 GOPs before the current one.

**Random Access (RA)** For the random access configuration, a hierarchical coding approach is taken where each additional hierarchical level doubles the number of frames. The sequence is split into GOP's of eight pictures each. The GOP's are coded sequentially, but the coding order within each GOP is decoupled from the display order. Figure 2.18 depicts one GOP of the random access configuration for two layers as it is specified in the common testing conditions. The coding order is plotted on the t-axis and the y-offset of each frame within the GOP illustrates the hierarchy level of the frame.

Compared to the previous coding structures, the random access configuration exhibits a higher coding performance. This is due to the reason that all pictures within a GOP can utilize references from the temporal "past" and "future" which allows for a very efficient bidirectional prediction. However, in order to enable this bidirectional prediction, the configuration introduces a structural delay of 7 frames to the coding system.

This coding structure is meant to simulate applications with a strong requirement for coding efficiency but without the necessity for real time capabilities. This includes broadcast and streaming as well as storage applications.

**Table 2.3** Quantization parameters for each type of scalability as defined in the common testing conditions.

| | $QP_{BL}$ | $\Delta QP_1$ | $\Delta QP_2$ |
|---|---|---|---|
| 2x | 22,26,30,34 | 0 | 2 |
| 1.5x | 22,26,30,34 | 0 | 2 |
| SNR | 26,30,34,38 | -6 | -4 |

In addition to the scalability ratios and the coding structure, also the quantization parameter (QP) of each layer can be modified. The QP controls the quantization step size of the transform coefficients. In the SHVC reference encoder software it is also used to control the rate-distortion decision between the reconstruction quality and the required bitrate. Hereby, the QP is directly linked to the reconstruction quality and the required bitrate. A high QP parameter implies coarse quantization, low bitrate and low reconstruction quality, while a low QP parameter implies fine quantization, high bitrate and a high reconstruction quality. The relationship of the QP values between the layers is defined using a delta value where:

$$QP_{EL} = QP_{BL} + \Delta QP \tag{2.15}$$

In the common testing configuration, 4 QP values for the base layer and 2 $\Delta QP$ values are tested, which are depending on the type of scalability between the layers. Table 2.3 provides a full list of the defined quantization parameters.

In order to provide full results according to the common testing conditions, all the test sequences have to be tested using all the described scalability ratios (2x, 1.5x, SNR), coding structures (AI, LD, RA) and QP values. However, there are two exceptions:

- For the all intra configuration, SNR scalability is excluded from the mandatory test set.
- For 1.5x scalability only the class B sequences are tested.

While there are numerous more parameters that can be set for the reference encoder, they are not all detailed here. For a full list of all used parameters, the reader is referred to the common testing conditions [SH14] and to the encoder configuration files, which are located in the software repository of the SHVC test model (SHM) [SHM-11].

### 2.5.3 Performance

Now that the testing conditions are established, a performance analysis of SHVC according to these testing conditions can be performed. In this section, the coding performance of SHVC is compared to the non-scalable coding alternatives using HEVC. Let's assume we are testing a certain SHVC configuration with multiple layers of different fidelity.[18] There are two main

[18] In this context, the word "fidelity" refers to all possible parameters that can enhance in the higher layer (increased spatial resolution, enhanced reconstruction quality, increased frame rate, etc.). See Section 2.5.

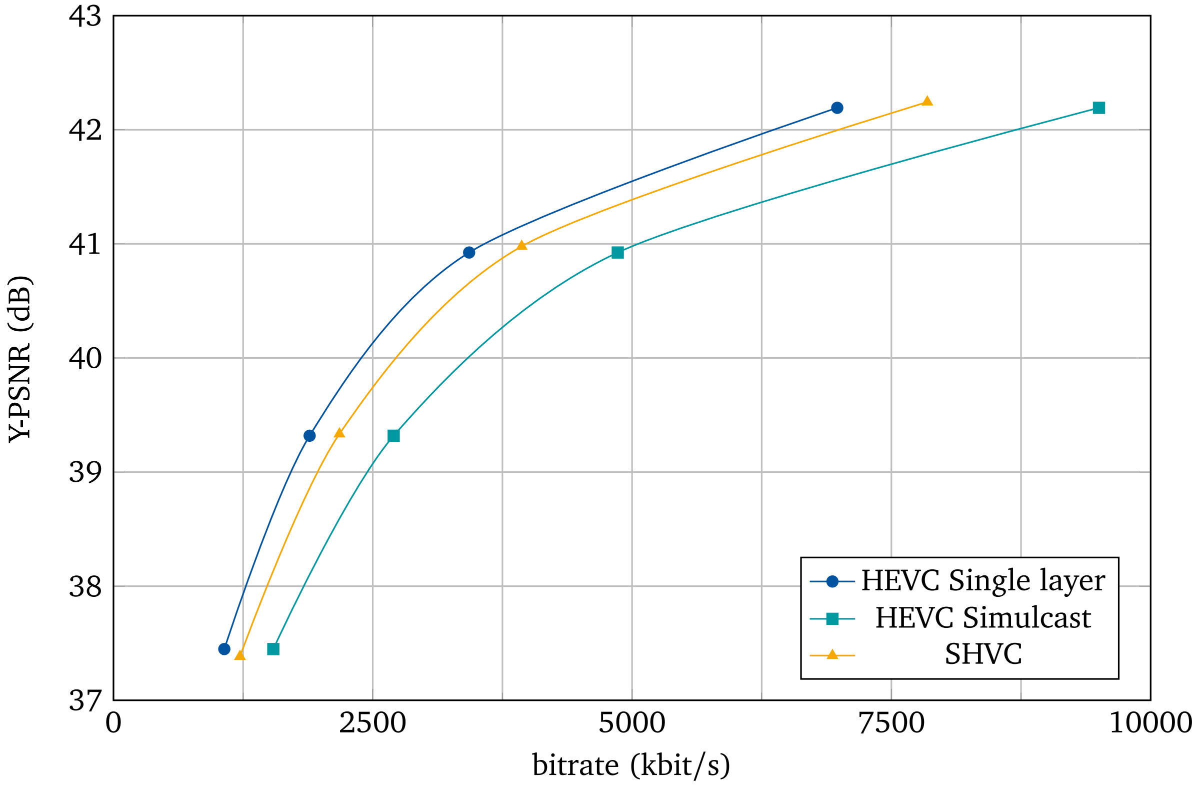


**Figure 2.19** Rate-distortion plot for the sequence Kimono according to the common testing conditions. The $\Delta QP$ value is set to -6 and SNR scalability is used.

alternatives to the scalable approach which only use single layer coding with HEVC. The performance of SHVC is then compared to these alternatives.

**Single Layer Coding:** This is the comparison to the conventional coding approach. Instead of coding a scalable bitstream with multiple fidelities, we only transmit one layer using conventional HEVC. For the comparison to SHVC, the fidelity of the single layer stream is comparable to that of the highest layer of the SHVC bitstream. Naturally, this single layer stream lacks all features of scalable coding.

The comparison to single layer coding allows to draw conclusions on the rate increase which is induced by the coding of multiple layers: "*If we are switching from a system where only one layer at high quality is coded to a system where additionally lower quality versions of the same video can be decoded, what is the additional rate that we have to spend for the features of scalable coding?*"

**Simulcast:** Similar to SHVC, multiple versions of the same video are coded in the simulcast scenario. The fidelity of each version is aligned with the fidelity of each of the SHVC layers. The resulting bitstreams provide features similar to SHVC while, unlike SHVC, the correlation between the multiple versions of the same video are not exploited. Each version is coded independently using HEVC.

*If we have multiple versions of the same video, how much bitrate can we save if we use SHVC compared to using HEVC for each version independently?*"

**Table 2.4** SHVC coding performance compared to HEVC single layer coding (Y BD-rate)

| | | 2x | 1.5x | SNR |
|---|---|---|---|---|
| All intra | $\Delta QP_1$ | 11.87% | 11.10% | |
| | $\Delta QP_2$ | 13.68% | 9.94% | |
| | **Avg** | **12.78%** | **10.52%** | |
| Low delay | $\Delta QP_1$ | 25.54% | 26.13% | 22.12% |
| | $\Delta QP_2$ | 31.12% | 23.40% | 26.55% |
| | **Avg** | **28.33%** | **24.77%** | **24.33%** |
| Random access | $\Delta QP_1$ | 17.01% | 16.14% | 13.03% |
| | $\Delta QP_2$ | 20.96% | 15.86% | 15.59% |
| | **Avg** | **18.98%** | **16.00%** | **14.31%** |

Figure 2.19 shows an exemplary rate-distortion graph from the test set. The coded sequence is Kimono and the scalability type is SNR scalability. As defined in the common testing conditions, the base layer QPs are 26, 30, 34 and 38. The $\Delta QP$ value is -6, which yields the values 20, 24, 28 and 32 for the enhancement layer QP. For the "HEVC Single Layer" curve (—●—), the sequence is encoded using HEVC and the enhancement layer QPs. For "HEVC Simulcast" (—■—), an additional lower layer is coded using HEVC and the corresponding base layer QP values. The Y-PSNR values are identical to "HEVC single layer" (—●—) but the bitrate increases because of the additional base layer stream. For the "SHVC" curve (—▲—), also two layers are encoded but here the dependency between the layers is exploited. It can be observed that the performance of SHVC is somewhat diminished compared to single layer coding but increased when it is compared to the simulcast reference.

In the following, the Bjøntegaard delta rate results for the luma component (Y BD-rate) for the test set as defined in Section 2.5.2 are presented. The results were obtained using the SHVC reference software (SHM) version 11.0 and the HEVC reference software (HM) version 16.7.[19] For each sequence, the 4 QP values were used to calculate a Bjøntegaard delta rate value. These values were then averaged over all tested sequences.

Table 2.4 shows the luma BD-rate increase of SHVC compared to single layer coding with HEVC. Depending on the coding structure and the scalability type, a different BD-rate increase can be observed. While for the all intra configuration the BD-rate increase is relatively low (between 10.52% and 12.78%), it increases for the random access configuration (between 14.31% and 18.98%) and is highest for the low delay configuration (between 24.33% and 28.33%). For the scalability types (2x, 1.5x and SNR) a much lower change in BD-rate increase can be found. For the average BD-rate results, the BD-rate values increase with the spatial difference between the layers. For example for the random access configuration the BD-rate values increase from 14.31% for SNR scalability to 16% for 1.5x scalability to

[19] The SHVC and the HEVC software can be obtained from [SHM-11] and [HM-16.7], respectively.

**Table 2.5** SHVC coding performance compared to HEVC simulcast (Y BD-rate)

| | | 2x | 1.5x | SNR |
|---|---|---|---|---|
| All intra | $\Delta QP_1$ | -19.43% | -27.81% | |
| | $\Delta QP_2$ | -24.44% | -35.76% | |
| | **Avg** | **-21.93%** | **-31.78%** | |
| Low delay | $\Delta QP_1$ | -7.51% | -15.22% | -9.24% |
| | $\Delta QP_2$ | -13.09% | -27.88% | -15.69% |
| | **Avg** | **-10.30%** | **-21.55%** | **-12.46%** |
| Random access | $\Delta QP_1$ | -13.84% | -22.21% | -17.58% |
| | $\Delta QP_2$ | -19.15% | -31.82% | -24.33% |
| | **Avg** | **-16.49%** | **-27.01%** | **-20.95%** |

18.98% for 2x scalability.

In Table 2.5, the luma BD-rate reduction for SHVC compared to the HEVC simulcast scenario is presented. It can be seen that in this scenario, depending on the coding structure and scalability type, SHVC can significantly reduce the required bitrate. It can further be observed, that the BD-rate reduction is higher for $\Delta QP_2$ compared to $\Delta QP_1$. This can be explained by the relation of the $\Delta QP$ to the reconstruction quality of the two layers. For $\Delta QP_2$ the reconstruction quality difference between the base layer and the enhancement layer is lower compared to $\Delta QP_1$ and if this difference is lower, SHVC can exploit this similarity much better. Depending on the configuration, the BD-rate reduction ranges from -10.30% to -31.78%.

While the coding performance of SHVC using the multi loop scheme is good, it could be further increased if the high level syntax only approach was abandoned. In this case, more advanced inter layer tools like inter layer residual prediction or inter layer mode prediction could be implemented. Already during the standardization it was demonstrated that these tools could further increase the overall coding performance [Fra+13; Ser+13].

### 2.5.4 Complexity Analysis

During the standardization phase of SHVC, the question on the decoder complexity of SHVC compared to single layer coding using HEVC was put forward. However, since the decoder complexity strongly depends on the decoder implementation and the implicit 'cost' for certain operations, there is no universal answer to this question. In JCT-VC, a break-out group (BoG) was formed to work out a way to form an estimate on the decoder complexity. The recommendation of the BoG includes a worst case analysis using the SHVC specification and practical measurements using the reference software (SHM) decoder [FTA13].

In the following, a number of values are presented that can be measured at the decoder side and allow to draw some conclusions on the decoder complexity. These include the values recommended in [FTA13] as well as additional values. As for the evaluation of the performance, the measured complexity indicators are always compared to the single layer scenario. For the complexity analysis, the simulcast reference only has a very limited relevance because of its practical implications: The assumption for the simulcast complexity is that for multiple layers, all available single layer streams are always fully decoded at the decoder side and only one of them is presented to the viewer. While this would enable highly flexible switching between the steams, it is also impractical. In a real world simulcast implementation of multiple streams, it is likely that at the receiver side only the best available stream is decoded while the other streams are discarded. Thereby, there is no complexity overhead. If necessary, the decoding can be switched to another stream without any decoding artifacts at any random access point. For completeness, the corresponding complexity results compared to this hypothetical simulcast scenario can be found in Appendix B.2. All results in this section were obtained using the SHVC reference software version 11.0 and following the corresponding common testing conditions [SHM-11; SH14]. The presented measurements are also employed in the later chapters in order to evaluate the complexity of the proposed methods.

### Decoding Time

One thing that can easily be measured using the reference software is the time that the software decoder requires to decode a specific stream on a specific machine. If all streams (the SHVC streams as well as the single layer streams) are decoded on an identical machine, the complexity increase can be inferred from the runtime difference.

While these measurements are simple to conduct, they should be interpreted with caution. Firstly, the reference software is a very specific implementation of an SHVC decoder that is running on a conventional PC. This PC is also running an operating system and it may be that other tasks are active in the background. In addition, the SHVC reference software is not designed to be optimized for speed and an optimized decoder may produce highly differing decoding times.[20] For these reasons, the measured decoding time increase or decrease should be reviewed with caution.

Table 2.6 illustrates the average decoding time differences of SHVC compared to the single layer coding approach. In order to avoid the inaccuracy of comparing different software, the same software was used to decode the scalable and non-scalable streams (SHM 11.0). Each decoding job was run independently on the same machine.

It can be seen that compared to single layer coding, SHVC induces a significant increase in decoding time for certain configurations. For the low delay and random access configuration,

[20] For example, when decoding an enhancement layer, the SHVC decoder does always upsample the entire lower layer reference picture even if the lower layer reference is not or only sporadically utilized in the enhancement layer. An optimized decoder may exploit this and only perform upsampling for the blocks that require it. Of course, such decoder optimizations can only reduce the average decoding complexity and do not have any influence on the worst case.

**Table 2.6** Average decoding time difference of the SHVC reference decoder compared to single layer coding. The corresponding simulcast results can be found in Table B.3.

| | 2x | 1.5x | SNR |
|---|---|---|---|
| All intra | 204% | 341% | |
| Low delay | 136% | 211% | 176% |
| Random access | 143% | 223% | 187% |

comparable results can be observed: While the increase for 2x scalability compared to single layer coding is still moderate at roughly 140%, it is significantly higher for 1.5x and SNR scalability where the decoder runtime is approximately doubled. Interestingly, the decoding time increase is higher for 1.5x scalability than for SNR scalability. This can be explained by the upsampling process: While decoding of the lower layer is less complex in 1.5x scalability, no upsampling of the lower layer reconstruction needs to be performed for SNR scalability. For the all intra configuration, the decoding time of SHVC increases to 204% for 2x scalability and 341% for 1.5x scalability. At first, such a high increase may seem implausible. However, this can be explained if we consider that the simulcast reference uses only intra prediction and for the scalable version, upsampling and inter layer prediction is added. This can be interpreted as another indication that the upsampling process and inter layer prediction in SHVC are relatively complex compared to intra prediction.

### Memory Access Operations

For hardware- as well as software decoders, random read operations from the reference picture buffer memory are relatively costly. Because of this, the number of bytes read from memory can give another indication for the decoder complexity. The counted memory read operations include all motion compensation operations where a prediction block has to be fetched from the reference picture buffer as well as the upsampling operations for which the lower layer reconstruction has to be read from memory. These memory access measurements were also used in the complexity analysis of [SF11].

With motion compensation, this process is straightforward. For each motion compensation operation within the same layer, the number of bytes that are read from memory are counted. This number depends on the size of the PU. Because there is no upsampling involved, the same process applies for SNR scalability. It is assumed that for an inter layer prediction operation, the lower layer reference picture can be directly accessed and the number of bytes read are identical as with a motion compensation operation of the same size and no subsample interpolation.

If upsampling is applied, the number of bytes accessed depends on how the upsampling process in the decoder is implemented. Two possible implementations of the upsampling process are considered:

**Picture based upsampling** Before the actual decoding process of a picture in the enhancement layer is started, the entire lower layer picture that requires upsampling is

**Table 2.7** Average memory access as well as arithmetic (multiplication and addition) operations used for filtering relative to the single layer reference. Upsampling is performed per prediction block. The corresponding simulcast results can be found in Table B.4

| | | Block upsampling | | | Picture upsampling | | |
|---|---|---|---|---|---|---|---|
| | | 2x | 1.5x | SNR | 2x | 1.5x | SNR |
| Memory Access | Low delay | 109% | 104% | 180% | 129% | 129% | 180% |
| | Random access | 118% | 110% | 193% | 137% | 132% | 193% |
| Arithmetic | Low delay | 111% | 220% | 169% | 195% | 323% | 169% |
| | Random access | 118% | 224% | 183% | 199% | 326% | 183% |

upsampled to the enhancement layer resolution. For the memory access operations, this implies that the entire lower layer picture has to be read from memory. It is then upsampled and again stored in memory. For every block in the enhancement layer that uses the lower layer reference, another memory access operation is added to account for the read operation from this upsampled reference picture.

**Block based upsampling** For each prediction block in the enhancement layer that uses the lower layer picture as a reference, the equivalent area in the lower layer reference picture is upsampled. For this, the corresponding area in the lower layer picture plus an additional area around it has to be read from memory and is accounted for in the memory access calculations.[21] A possible reading overlap with the neighboring block is not considered in this calculation.

These two models can give an indication on the memory access operations for upsampling. While the second option seems to be the less complex approach, there are other possible implementations of the upsampling process and the actual numbers are strongly dependent on a specific realization. Additionally, the underlying memory architecture also affects the required memory bandwidth. Three different memory architectures were considered: A so called pure memory architecture as well as a DDR2 (Double Data Rate) and DDR3 based memory architecture. Regarding the bandwidth, they all differ in how bytes in memory can be accessed. For the pure memory architecture, it is assumed that every byte in memory can be accessed individually. For DDR2 and DDR3, the premise is that memory can only be read in blocks which are aligned on an 8 byte raster and are of size 32 and 64 bytes, respectively. Independent of the memory architecture, all results show similar behavior. Because of that, only the results for the pure memory architecture are presented here. The corresponding results for DDR2 and DDR3 can be found in Tables B.6 and B.7 in Appendix B.3.

The upper section of Table 2.7 shows the relative number of memory access operations of SHVC for block and picture based upsampling, compared to the single layer reference. There are no results for the all intra configuration because no motion compensation is applied for all

[21] Since SHVC uses an 8-tap upsampling filter, additional values at the borders of the block have to be read from the lower layer reference picture. The size of the block in the lower layer increases by 7 pixels in width and height.

**Table 2.8** Worst case memory access and arithmetic operations compared to the worst case results using single layer coding at the enhancement layer resolution. These results are further broken down in Appendix B.1, Tables B.1 and B.2.

| | 2x | 1.5x | SNR |
|---|---|---|---|
| Memory Access | 129% | 151% | 215% |
| Arithmetic Operations | 141% | 168% | 200% |

intra. Only absolute numbers of memory access operations for the upsampling process can be provided since the single layer reference does not apply motion compensation. It can, however, be said that in terms of memory access operations SHVC is more complex than the single layer and simulcast references, since for SHVC, memory access from a reference picture is necessary.

It can be seen that in terms of memory access operations it is preferable to use a block based upsampling implementation.[22] For SNR scalability, the type of upsampling is irrelevant since in this scenario, no upsampling is applied. From the results it can be further inferred that compared to single layer coding, the number of memory access operations increases. Particularly for SNR scalability the number almost doubles, which might be unacceptable for certain applications. For 1.5x and 2x scalability the increase is much lower and could be tolerated in order to benefit from the scalable features of SHVC. Especially for the block upsampling approach, the memory access overhead compared to single layer coding is small.

It should be also noted here that these are average results that were measured for the specified test set. In this paragraph, the worst case complexity increase for the memory access operations is examined. Again, a two layer scenario with the different types of scalability is considered. In the worst case, bidirectional prediction is performed in both layers where all PUs are of size 8x8 and use fractional motion compensation. In case of spatial scalability, the relative number of read operations in the lower layer is reduced since the spatial resolution is smaller. For the upsampling process, the utilization of an efficient picture based algorithm is presumed which only reads all bytes from the lower layer reconstruction once. Also in the SNR case it is assumed that the lower layer reconstruction is copied to the higher layer. The upsampled lower layer reference, however, is not used in the higher layer. In Table 2.8, it can be seen that also for the worst case considerations, there is a severe overhead for the memory access operations in SHVC. This further increase is even higher than the spatial ratio between the layers would suggest and is evoked by the additional inter layer processing step which is counted here as an additional memory access operation. It should be noted that this worst case examination is a theoretical scenario and the results strongly depend on the underlying assumptions. When a specific coding structure or a special implementation is considered, the results could differ.

[22] While this statement is true for the average results that are tested here using the common testing conditions, it might not hold for other configurations with more layers or a different relationship between the layers. Also for the worst case considerations this does not apply. A specific application might also require upsampling of the entire lower layer picture, in which case a picture based upsampling implementation may be preferable.

### Multiplications and Additions for Filtering

Another value that can help to assess the decoder complexity is the number of arithmetic operations (multiplications and additions) that are executed by the filters for motion compensation as well as upsampling. Just as for the memory access operations these values are counted and then compared to the values of the single layer reference. While the absolute number of additions and multiplications are counted separately and are different, the relative numbers compared to the reference are identical for multiplications and additions. Because of this, the results in the lower section of Table 2.7 apply to both operations.

As for the memory access operations, it seems to be advantageous to implement a block based upsampling scheme. The increase in multiplication and addition operations is notably increased for 1.5x scalability. Like the other measurements, the informative value of the presented results differs strongly with the specific implementation. For example, in a hardware decoder, the number of filtering operations might be non-critical if they are executed in a dedicated hardware unit. The results indicated that the upsampling process has a high impact on the number of arithmetic operations in relation to the motion compensation. It can be seen that, compared to single layer coding, the number of arithmetic operations for 2x scalability doubles for the picture upsampling scheme and even more than triples for 1.5x scalability. At the same time, it is much lower for SNR scalability in which no upsampling is applied. While the overhead is lower for the block upsampling approach, is still highest for 1.5x scalability. For SNR scalability, in which no upsampling is applied, the overhead is lower.

As for the memory access operations the presented average results are based on experiments using the specified test set. However, a worst case analysis for the arithmetic operations is performed as well. As for the previous worst case investigation, it is assumed that all PUs in both layers use bi-directional inter prediction with a PU size of 8x8 pixels. For the upsampling process, again, an efficient algorithm is considered that respects the specific resolution relationship between the layers and processes the entire image in one pass. In case of SNR scalability, no upsampling as well as no arithmetic operations are necessary. In Table 2.8, the corresponding results are shown. It can be seen that in case of SNR scalability the worst case complexity increase in terms of arithmetic operations is equal to a factor of two. This is lower than the corresponding value for the memory access operations. For spatial scalability, however, the overhead is even higher than that of the memory access operations. Of course, also these results are theoretical and could differ with different assumptions.

### Inverse Transform Operations

While the HEVC core transform was designed with practical software and hardware realizations in mind and allows for a very efficient implementation, it still requires a significant amount of multiplication and addition operations (see Section 2.3.3). So in addition to the recommendations in [FTA13], some additional higher layer measurements were performed.

Firstly, the number of pixels for which an inverse transformation is performed are recorded. Table 2.9 shows that on average the highest increase compared to single layer coding occurs

**Table 2.9** Average difference in number of pixels that an inverse transform is applied for, that are considered by the deblocking process and that are considered by the sample adaptive offset (SAO) process. All values are relative to the single layer reference. The corresponding simulcast results can be found in Table B.5

| | | 2x | 1.5x | SNR |
|---|---|---|---|---|
| Inverse Transform | All intra | 112% | 81% | |
| | Low delay | 100% | 106% | 161% |
| | Random access | 99% | 96% | 165% |
| Deblocking | All intra | 84% | 67% | |
| | Low delay | 109% | 92% | 185% |
| | Random access | 103% | 88% | 181% |
| SAO | All intra | 160% | 134% | |
| | Low delay | 101% | 97% | 144% |
| | Random access | 107% | 100% | 161% |

for SNR scalability (165% to 161%). For the other scalability types there is essentially no change for the low delay and random access configurations as well as only modest change for all intra. The worst case analysis is very straightforward for the number of inverse transformations. For single layer coding, an inverse transformation is performed for every pixel. For SHVC, this applies to all layers while the lower layers may use a lower spatial resolution. Therefore, the worst case numbers can be directly inferred from the spatial relationship: For SNR, the value is 200%, for 1.5x it is 144% and for 2x scalability it is 125%. So compared to the average values, the worst case implications are much more severe.

### Deblocking and SAO Operations

As described in Section 2.3, all pictures are filtered before they are displayed and put into the reference picture buffer. For SHVC, this filtering process is applied in each layer before the pictures are upsampled and used for inter layer prediction. Two filters are applied: A deblocking filter and a sample adaptive offset (SAO) filter [h.265/HEVC; Wie15; Sul+12]. For each of these filters, we count the number of pixels that are tested for a possible modification. Table 2.9 also contains the results for deblocking and SAO. For the random access and the low delay configuration the same conclusions can be drawn: Compared to single layer coding, the highest increase occurs for SNR scalability (144% to 161%), while for 2x and 1.5x scalability the values are close to 100%. For all intra, however, there is also a significant increase compared to single layer coding. In the worst case, the total amount of deblocking and SAO operations directly scales with the spatial resolution of the layers. This implies that the relative worst case value is 200% for SNR, 144% for 1.5x and 125% for 2x scalability.

### Conclusion

In the above section an analysis of the SHVC decoder complexity compared to single layer coding and simulcast is provided. Because there is not one unique criterion to represent the general decoder complexity and it is also greatly dependent on the concrete implementation, a multitude of values were measured at the decoder side using the common testing conditions. These values can then be analyzed to give an idea about the decoder complexity.

While the different measured values are not identical and they vary depending on the used configuration and the type of scalability, some general conclusions about the decoder complexity can be drawn. Compared to single layer coding using HEVC, the average results indicate that the complexity of SHVC increases substantially. The complexity increase is particularly high for SNR scalability, where the values are nearly doubled. However, depending on the measured value, the complexity impact of SHVC is also considerable in case of spatial scalability. Similar conclusions can be drawn from the worst case results which also indicate a large increase in decoder complexity compared to single layer coding. In the worst case, this increase can be even higher than the pure added complexity of the two decoders of the layers because in addition to these, inter layer processing must be applied. So the overall complexity increase of SHVC results from the multilayer coding design as well as the necessary inter layer processing.

With regard to real applications of SHVC, a certain amount of decoder complexity increase may be acceptable. However, it is improbable that SHVC will be employed if it comes with such severe ramifications for the decoder complexity as is was shown in this chapter. While there are practical applications in which more than two layers could be beneficial, only a two layer scenario is considered here. If the number of layers is further increased, it can be expected that the decoder complexity increase of SHVC compared to single layer coding is further enlarged, making SHVC even more unattractive.

For these reasons, it is essential to reduce the SHVC decoder complexity in order to make it competitive with non-scalable alternatives like simulcast. In the following, multiple techniques are presented and analyzed, which all focus on a reduction of the decoder complexity for the SNR scalability scheme. By modifying the prediction structure and changing some encoder settings, so called single loop decoding can be enabled which significantly reduces the decoder complexity. Like SHVC, this scheme also satisfies the high level syntax only approach and no changes to the lower layer coding tools are necessary. In addition, the complexity can be further reduced by an inter layer refinement technique that operates in the transform domain.

# 3 Flexible Inter Layer Prediction in SHVC

In the previous chapter, the SHVC decoder complexity was evaluated and it could be seen that in particular for SNR scalability, the SHVC decoder is almost twice as complex when compared to a single layer HEVC decoder. In this chapter, the prediction structure is modified and the so called key picture concept is introduced. This is analogous to the key picture concept which was employed in scalable video coding (SVC). However in this novel approach, following the high-level syntax only design, the prediction structure is changed but no changes to the lower layer coding tools are applied. This is the most significant difference to SVC in which several low layer coding tools were newly introduced or modified. The coding scheme which is presented here can be implemented using only high level syntax changes and can therefore be considered a high level syntax only approach.

This approach can help to address two main topics: Firstly in Section 3.1, it is described how the modified prediction structure allows for reconstruction of the higher layer video without the requirement of full reconstruction of the lower layer video. Thus, the decoder complexity overhead for decoding of the higher layer can be significantly reduced. At the same time, however, a drift is introduced when only the lower layer is reconstructed. The decoder complexity is analyzed in relation to the complexity of SHVC as it was detailed in Section 2.5.4 and the coding performance is evaluated. In addition to these objective measurements, a visual test for the drift was performed. Secondly in Section 3.2, it is described how the key picture concept can be combined with temporal scalability in order to construct a bitstream that enables highly flexible adaption to varying channel conditions with so called medium-granular quality scalability.

On the software side, only few minor changes to the reference encoder and decoder are necessary to enable this flexible inter layer prediction scheme. While the integration of the complexity reduction feature into the decoder is somewhat more demanding, it is still manageable.

## 3.1 Key Picture Concept

### 3.1.1 SHVC Prediction Structure

In Section 2.5.2, the prediction structures from the common testing conditions were detailed. As previously mentioned, these representative fixed coding structures were chosen with a certain application in mind. Specifically, these are: The low delay configuration for real time applications with a requirement for low structural delay and the random access configuration with a focus on coding efficiency and periodic points at which the decoding process can be started.

Additionally, the common testing conditions define an all intra (AI) configuration. As the name suggests, no motion compensation is employed in this configuration and all frames are coded independently using intra prediction only. Because there is no temporal dependency between the frames, a decoding error cannot propagate to other frames. Furthermore, an all intra bitstream can be cut and joined at any frame within the sequence. Therefore, this configuration is most commonly used in professional applications, where error resiliency and frame accurate cutting is of high priority. Since the key picture concept is utilizing the temporal prediction structure to reduce the decoder complexity, it cannot be used to reduce the decoder complexity of the all intra configuration. While there are possible ways to also reduce the decoder complexity for the all intra configuration, such methods are not handled here and the all intra configuration is not further discussed in this chapter.[1]

As previously presented, the inter layer dependencies are exploited in SHVC by adding the lower layer reconstruction as an additional reference picture to the reference picture list of the enhancement layer frame. If needed, the lower layer reconstruction is upsampled and color mapping is applied. There are various advantages and disadvantages to this approach which are detailed in the following:

**Standardization and Implementation Complexity** One appealing property of the reference picture approach is the limited amount of necessary additional standardization work. In fact, when considering the syntax of HEVC, all additions of SHVC are defined at the slice header level or higher. No changes to lower layer tools like prediction, transformation or quantization are necessary.

**Implementation reusability** This high-level syntax approach is very beneficial for the actual implementation of encoders and decoders. For an existing single layer hardware or software HEVC decoder, only few changes and additions are required to add support for decoding of multiple layers (SHVC).

**Coding performance** Although only high-level syntax modifications are included, the overhead of scalability in the SHVC approach is quite low compared to SVC (See Section 2.5.3). While during the standardization process multiple lower layer modifications were proposed that could further improve the coding performance, it was decided to disregard these gains and opt for the high-level syntax only approach.[2]

**Decoder Complexity** The main drawback of SHVC is the significantly increased decoder complexity (See Section 2.5.4). This complexity increase was also identified in the

[1] A possible technique for reduction of the all intra configuration's decoder complexity is prediction from the lower layer reconstruction signal before loop filtering is applied. This way, the higher layer decoder can skip the lower layer loop filtering step. Also, for scalability with more than two layers a more sophisticated inter layer residual prediction scheme could help to further decrease the number of necessary inverse transform operations at the decoder side.

[2] The additional tools that were proposed and tested in the context of SHVC include methods on inter layer residual prediction, inter layer mode prediction as well as various other proposals to improve the exploitation of inter layer dependencies. Some proposals were evaluated in so called Tool Experiments [Fra+13; Ser+13]. All proposals can be found in the JCT-VC document management system [JCTVC-Docs].

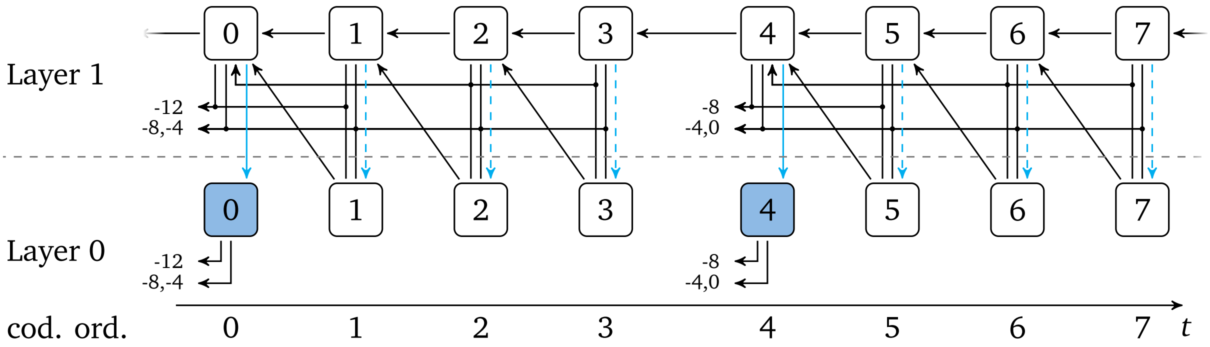


**Figure 3.1** Two GOPs of the low delay configuration with a GOP size of 4 pictures using the key picture concept. Every 4th picture is defined to be a key picture (blue). All non-key pictures in layer 0 employ only layer 1 pictures for prediction.

> standardization process and it was accepted in favor of the high-level syntax approach. However, while a moderate increase of the decoder complexity as it arises for spatial scalability might be tolerable, the immense increase for SNR scalability is certainly undesirable in practical implementations.

### 3.1.2 Modified Prediction Structure

The complexity increase of SHVC results from the used prediction structure and the implemented multi-loop design of SHVC. In Section 2.2.1, the concept of a coding loop was introduced. If we consider the prediction structure of SHVC (see Figures 2.17 and 2.18), one can see that for SHVC, one loop has to be closed per layer. In other words, all pictures of all layers have to be reconstructed and put into the reference picture buffer because they may be used for prediction by a higher layer or by other frames in the same layer.

As shown before, this is particularly complex for SNR scalability. However, for SNR scalability there is a way to use the key picture concept to enable so called single loop decoding which allows for a decoder implementation with a significantly reduced complexity. By adding a downsampling filter for prediction from higher layers, this concept could in principle also be extended to spatial scalability. However, as it was detailed in Section 2.5.4, the complexity overhead in case of spatial scalability is much lower and the potential for improvement is very limited. Furthermore, the downsampling filter adds an additional complexity for the prediction between the layers. Because of this, only the case of quality scalability (SNR scalability) is discussed in the following. The main focus of this modified prediction structure lies on the higher layer as the desired decoding point. While the lower layer should still be decodable at a lower quality, it is considered as being of a lower priority and a certain amount of loss in this lower layer can be tolerated.

Figure 3.1 illustrates the modified prediction structure of the key picture concept for the low delay configuration (the unmodified prediction structure can be found in Figure 2.17). In regular intervals, base layer frames are defined as key pictures (marked in blue). While in this example the key picture distance is set to 4 pictures, it could be freely selected by the encoder and in the following we also evaluate other fixed key pictures distances (8, 12 and

16). Every key picture is restricted to only utilize other key pictures from the same layer as references for motion compensated prediction. In this example, picture 4 in layer 0 now uses the pictures 0, -4 and -8 for prediction while in the unmodified prediction structure it could also predict from picture 3. Otherwise, there are no modifications to the coding of key pictures. For the key picture reconstruction in the lower layer, loop filtering and SAO is applied and the result is put into the reference picture buffer. As in SHVC, inter layer prediction is employed from this filtered lower layer reconstruction. This is indicated by a solid blue line in the figure.

For the remaining frames in layer 0 (the non-key pictures), several modifications are applied. While these adjustments require a modification for the encoder as well as the multilayer decoder, following the high level syntax only approach they do not involve any changes to the lower layer coding tools like intra prediction, motion compensation or entropy coding. Furthermore, the coding process of the symbols is also not changed so that all changes are compatible with the unmodified single layer decoding process.

1. For prediction, the reconstruction signal of the reference pictures is replaced by the reconstruction signal of the corresponding enhancement layer picture. In Figure 3.1, this is indicated by the arrows now pointing to the layer 1 pictures. Furthermore, the prediction of the non-key pictures is restricted to the frame range which is delimited by the previous and next key picture and to the enhancement layer representation of key pictures. For example, picture 2 can predict from picture 1 (the closest key pictures are 0 and 4) and pictures -4 and -8 (they are marked as key pictures); However, a prediction to other non-key picture before picture 0 (E.g. picture -2) is not permitted.[3]

2. For inter layer prediction, only the unfiltered (loop filter, SAO) lower layer reconstruction is accessed by the enhancement layer. This is indicated by dashed lines in Figure 3.1. Depending on the encoder implementation this may require some changes. However, deblocking as well as SAO can be disabled in HEVC using flags in the parameter sets. Therefore, an existing decoder implementation must contain the capability to skip these operations. Please note that this only applies for the inter layer prediction process. Before a picture is output, loop filtering as well as SAO are always applied in the highest layer that is decoded. The signaling in the bitstream is also not changed and each layer has individual SAO parameters.

3. For the non-key pictures in layer 0, constrained intra prediction is enabled. By doing this, intra areas in layer 0 can be correctly reconstructed independently of the inter predicted areas (See Section 2.3).

4. While the prediction is performed using the reconstructed pixel values from layer 1, the temporal motion vector predictor candidate (TMVP) is still obtained from the lower layer motion vector field. This change is applied so that the correct motion information in the lower layer can be reconstructed even if the higher layer is not available. [4]

[3] While this rule does not induce any changes for the low delay configuration and a key picture distance of 4, it is of importance for the low delay configuration with a larger key picture distance and for the random access configuration.

[4] While a drift in the lower layer motion information could be permitted, the implications for the reconstruction quality of this drift are so severe that this situation should always be avoided. As an alternative, TMVP can

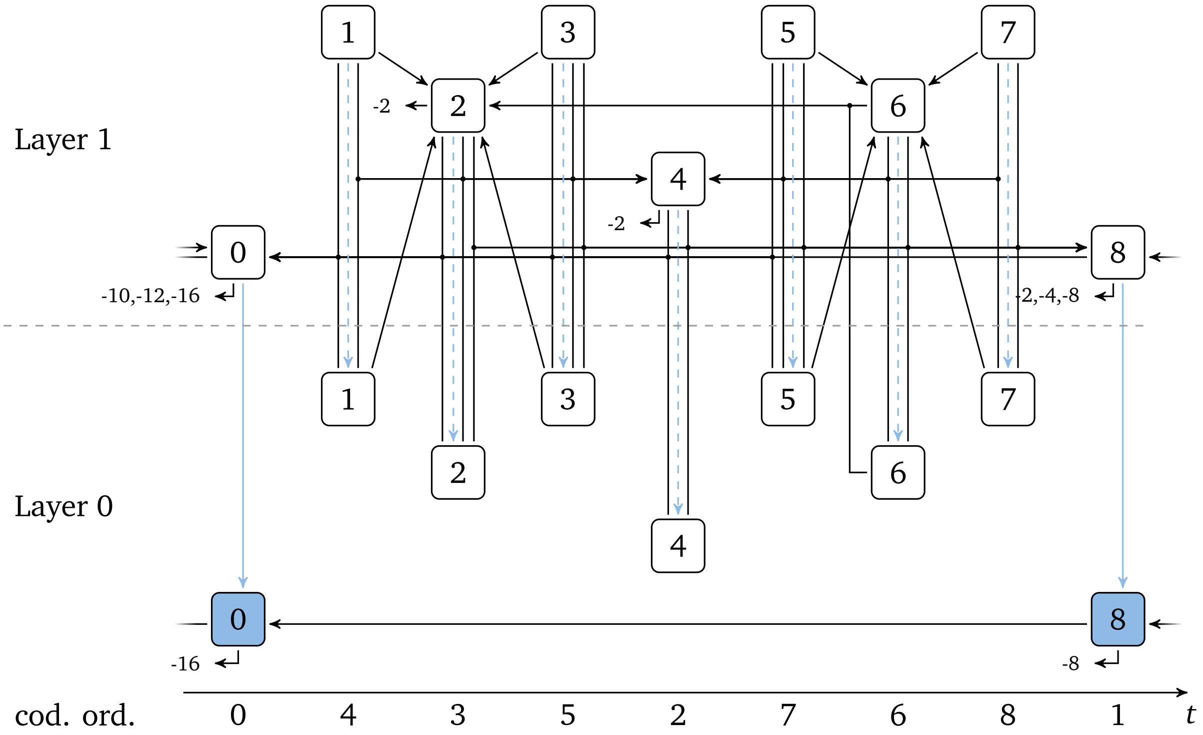


**Figure 3.2** One GOP of the random access configuration with a GOP size of 8 pictures using the key picture concept. Every 8th picture is defined to be a key picture (blue). All non-key pictures in layer 0 employ only layer 1 pictures for prediction.

In Figure 3.2, the same rules are applied to the random access configuration with a key picture distance of 8 pictures (compare with Figure 2.18). The key pictures in the base layer now only predict from the previous two key pictures and prediction for the non-key pictures is performed using the layer 1 pictures. It can also be observed that the prediction of non-key pictures is now limited by the key pictures as described in the aforementioned modification 1 (the pictures 2 and 4 no longer perform prediction from picture -2 because the intermediate picture 0 is defined as a key picture).

In the enhancement layer, no modifications to the prediction structure or other coding options are applied. From the two examples it can already be seen, that the non-key pictures in layer 0 are not used as a reference for any other picture in layer 0, so it is not necessary to put them into the reference picture buffer. This fact can now be exploited to significantly reduce the complexity of the layer 0 loop if the higher layer is decoded.

### 3.1.3 Single Loop Decoding

From the prediction structure in Figures 3.1 and 3.2, it can be inferred that full decoding of the non-key pictures in layer 0 is no longer necessary. As these pictures are only used as a reference by their corresponding enhancement layer picture, only those fragments of

be deactivated for the non-key pictures. Naturally, deactivation of TMVP has a negative effect on the coding efficiency.

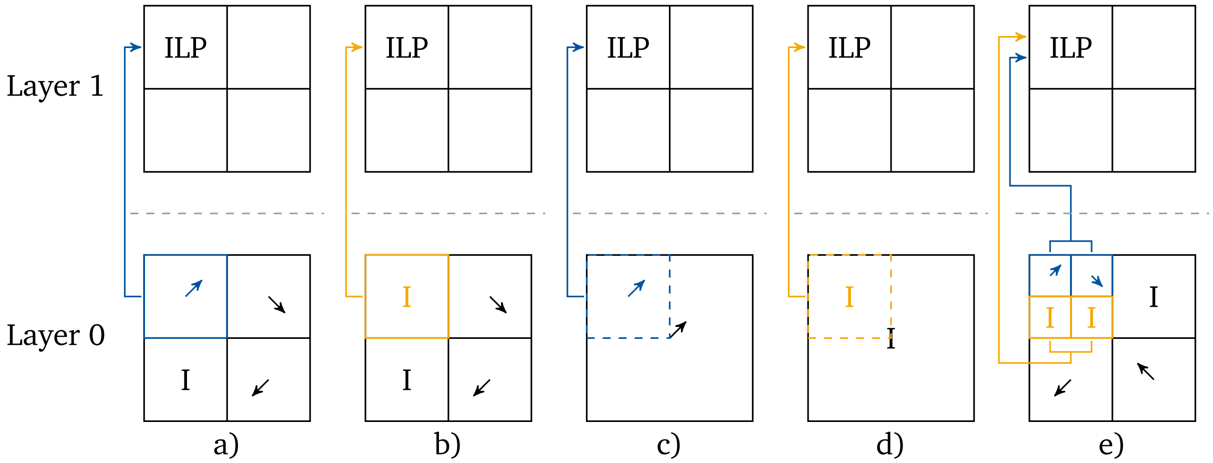


**Figure 3.3** Possible constellations for inter layer prediction (ILP). Only the top left block in layer 1 uses ILP. For layer 0, several possible prediction structures are presented. For motion compensated prediction in layer 0, only the motion information is utilized in the enhancement layer (blue arrows). No reconstructed pixel values are copied. For intra prediction (I) in the base layer, the reconstructed pixels are copied to the higher layer (orange arrows).

the layer 0 picture which are effectively used for inter layer prediction by the higher layer require reconstruction in layer 0. Furthermore, due to the modified prediction structure, the non-key pictures of both layers use the identical frames for motion compensated prediction. This means that if a layer 1 block uses inter layer prediction and the corresponding block in the lower layer uses motion compensated prediction, the prediction can be directly executed in the higher layer.

Figure 3.3 showcases some possible situations for a prediction block in layer 1 that uses inter layer prediction. In this example, the enhancement layer block is split into four sub-blocks where only the top left block employs inter layer prediction. The other three blocks use intra prediction or inter prediction from another enhancement layer picture so they can be reconstructed without access to the lower layer reconstruction. Because the HEVC coding tree partitioning for the two layers is not necessarily aligned, different constellations can occur. If a block in layer 1 is using inter layer prediction, the corresponding lower layer block can be of equal size, be larger or it can be split further into multiple smaller blocks. In the following, these cases are detailed with the help of Figure 3.3:

**a)** The block sizes of the ILP block in layer 1 matches the lower layer block size. The upper left block of Layer 0 uses inter prediction from another layer 1 picture. In this case, the motion information is copied to layer 1 and the motion compensation operation is performed in the enhancement layer (indicated by the blue arrow). The right two blocks in the lower layer also use inter prediction, but the corresponding layer 1 blocks do not use inter layer prediction. Therefore, the motion compensation operation in layer 0 can be skipped. The remaining lower left block uses intra prediction and is therefore reconstructed.[5]

[5] The intra block in layer 0 is reconstructed because a neighboring intra block in the base layer might use the reconstruction of this intra block in turn for intra prediction. The reconstruction of one of these linked intra blocks might then be used for inter layer prediction in the enhancement layer. For the decoder implemen-

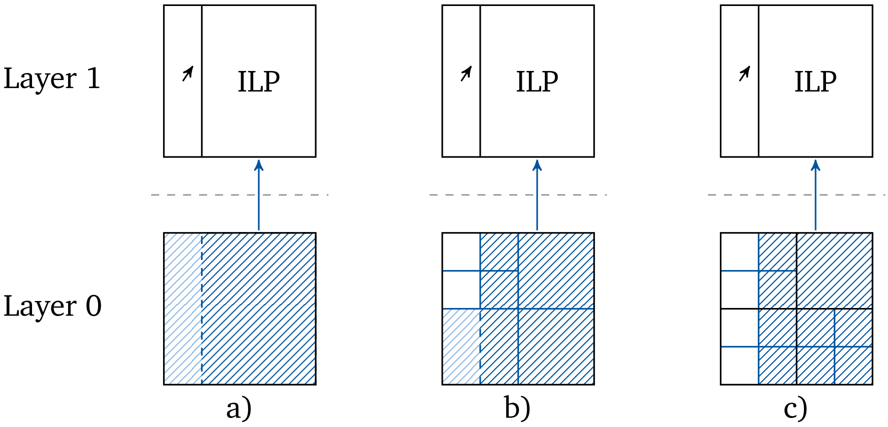


**Figure 3.4** Possible constellations for inter layer transfer of the inversely transformed lower layer residual data. Depending on the lower layer TU structure and the selected prediction mode in the enhancement layer, some inverse transform operations in the lower layer can be skipped (marked in white). The residual signal marked in dark blue is copied to layer 1. The residual data in the light blue area is reconstructed in the lower layer, but not transferred to the higher layer.

**b)** Here, the top left block uses intra prediction. It is therefore reconstructed in layer 0 and the resulting pixel values are copied to the enhancement layer for inter layer prediction (indicated by an orange arrow). As for a), the right two motion compensated blocks do not require reconstruction. The lower left intra block, however, is reconstructed.

**c) and d)** In this case, the size of the corresponding block in the base layer is larger than the block in the enhancement layer using inter layer prediction. In c), the motion information of the block is copied to the smaller block in layer 1 and motion compensation is carried out in layer 1. No motion compensation is performed in the lower layer. In d), the larger lower layer block uses intra prediction. Thus, the whole lower layer block is reconstructed but only the required top left part is copied to layer 1.

**e)** The corresponding lower layer block is further split and contains both intra and inter predicted sections. All parts have to be considered individually. For inter parts, the motion information is replicated in layer 1 and motion compensation is performed there. For intra parts, reconstruction is performed in layer 0 and inter layer prediction is performed by copying the reconstructed pixel values to the enhancement layer. Reconstruction of the lower layer intra parts is possible independent of the inter parts because constrained intra prediction is employed.

When these rules are applied during decoding, it can be assured that a majority of lower layer prediction operations can be skipped. However, it depends on the actually coded prediction structure how many decoder operations can be skipped. On average, however, the potential here is very high (see Section 3.1.5).

In a similar fashion, inverse transform operations for inter coded blocks in the lower layer can be avoided if the reconstruction values of the inverse transform are not utilized in the

tation that is evaluated here, it was chosen to reconstruct all intra blocks of non-key pictures in layer 0. A more sophisticated decoder could analyze this data dependency further and skip decoding of layer 0 intra blocks that are never used by the inter layer prediction process.

enhancement layer. As for the prediction operation, different situations can occur which result in varying amounts of operations that can be skipped. Figure 3.4 shows some examples. In layer 1, the block is split asymmetrically, where only the right half uses inter layer prediction. In all cases, only the residual signal of the corresponding right half in the lower layer (dashed in dark blue) is copied to the higher layer (blue arrows). The three blocks in layer 0 all hold different transform unit (TU) trees. For case a), the lower layer block contains one TU. Because the right half of the inverse transformation will be copied to the higher layer, the inverse transformation has to be performed for the entire TU. No transformation operation can be skipped in this case. For the cases b) and c), the lower layer block contains TUs that are entirely unused in layer 1. No inverse transform for these TUs is required in order to reconstruct the enhancement layer block, so the decoder can opt to skip these operations. In theory, it is possible that all residual signals in the lower layer need to be reconstructed because they are required in the higher layer. As for the prediction it depends on the actually coded modes how many inverse transform operations can be skipped (see Section 3.1.5 for the average results).

Using the described methods, a decoder can be implemented that only performs reconstruction of the information in the base layer that is effectively used in the higher layer. As mentioned before, it is also not required to filter the non-key pictures in the lower layer and put them into the reference picture buffer. Because of this, the described decoder operation mode is mostly referred to as single loop decoding. While it is true that the lower layer loop is suspended for non-key frames it should be noted that the key pictures in the lower layer are decoded as in SHVC. They are fully reconstructed, loop filtering is applied and they are put into the lower layer reference picture buffer. Hence, the lower layer loop is still running. It is merely running with a drastically reduced number of frames which is depending on the key picture distance.

It should further be noted here, that it is not obligatory to implement a decoder with the described complexity reductions. It is also feasible for a decoder to disregard any complexity reduction and fully reconstruct all pictures in the lower layer. This might actually be a requirement in some applications, where the reconstruction of all layers is displayed or further processed in any other way. Furthermore, the base layer still conforms to HEVC and can be decoded by the unmodified single layer HEVC decoder. While, as described above, some changes are required for the multilayer encoder and decoder, these changes are minimal. For the higher layer, no changes were applied to any lower layer coding tools so that the proposed key picture concept still follows the high-level syntax only approach.

For the encoder implementation which is used here, the implementation is for the most part unchanged. Especially for the order in which decisions are taken the bottom up approach of the reference software is inherited in which all layers are encoded one after the other from the lowest to the highest layer. This also means that for inter layer prediction in the higher layer, the lower layer information is fixed. In a more advanced encoder implementation optimization could be performed based on a higher layer which could further increase the coding performance with regard to the coded motion vectors and other information in the lower layer.

As it was already mentioned, a similar concept of key pictures was introduced with scalable video coding (SVC). However, unlike in SVC, the presented scheme adheres to the high level

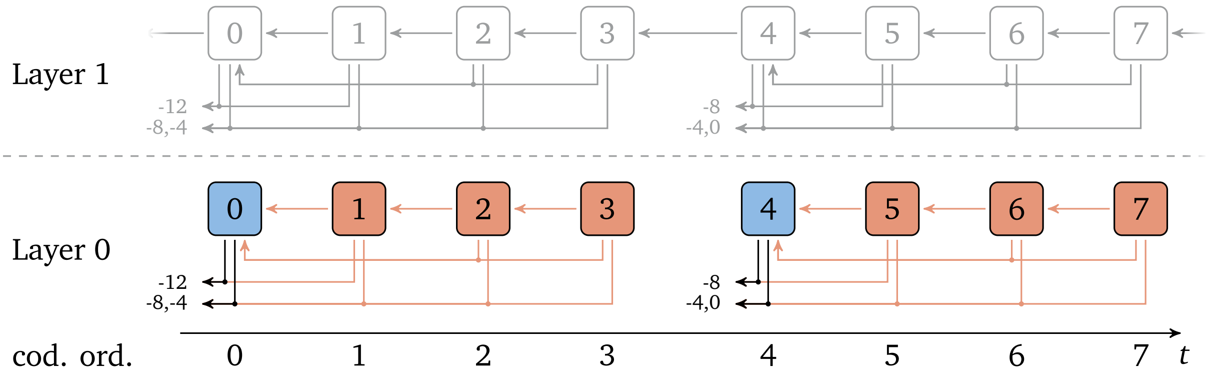


**Figure 3.5** Decoding only the lower layer for the low delay configuration. The enhancement layer references for the non-key pictures in layer 0 are no longer available and have to be replaced by the corresponding reference frames in layer 0. The red arrows indicate the modified prediction from reference pictures that were not used at the encoder side.

only approach and no changes to the lower layer coding tools are applied in the higher layers. It can therefore be regarded as a high level syntax only approach. Also in SVC, the concept was used to reduce the decoder complexity when a higher layer is decoded. A detailed comparison to SVC is discussed in Section 3.2.2.

### 3.1.4 Drift

So far, we only considered decoding of the enhancement layer quality. But of course, the idea of scalable coding is that a compliant decoder can also decode only the lower layer of the bitstream. [6] For the key pictures in the lower layer this is straightforward. Since they are restricted to only use other key pictures for inter prediction, they can be reconstructed without any error as before.

However, at the decoder side, the non-key pictures in the lower layer employ the enhancement layer frames as references for motion compensated prediction. While this can be beneficial when decoding the enhancement layer quality, the reconstructed enhancement layer frames are not available when decoding only the base layer. In this case, the decoding process of the lower layer is reverted to the conventional decoding scheme for one layer. The loop is closed in the lower layer and all pictures are fully reconstructed and put into the reference picture buffer. The references of the non-key pictures in layer 0 are replaced by the corresponding reconstructed frames from layer 0. While the layer 0 reconstructions are very similar to the layer 1 frames (which were used as a reference at the encoder side), they are not identical. Hence, the motion compensation operations at the encoder and decoder yield different results. This difference is referred to as drift.

Figure 3.5 shows the modified low delay prediction structure when decoding only layer 0. For the key pictures in layer 0 (pictures 0 and 4), all reference frames that were used at

[6] Depending on the application, the decoder can choose to only decode the lower layer or it might be the only option because the enhancement layer has been removed from the bitstream at some point. For the decoding process it is irrelevant why only the lower layer is decoded.

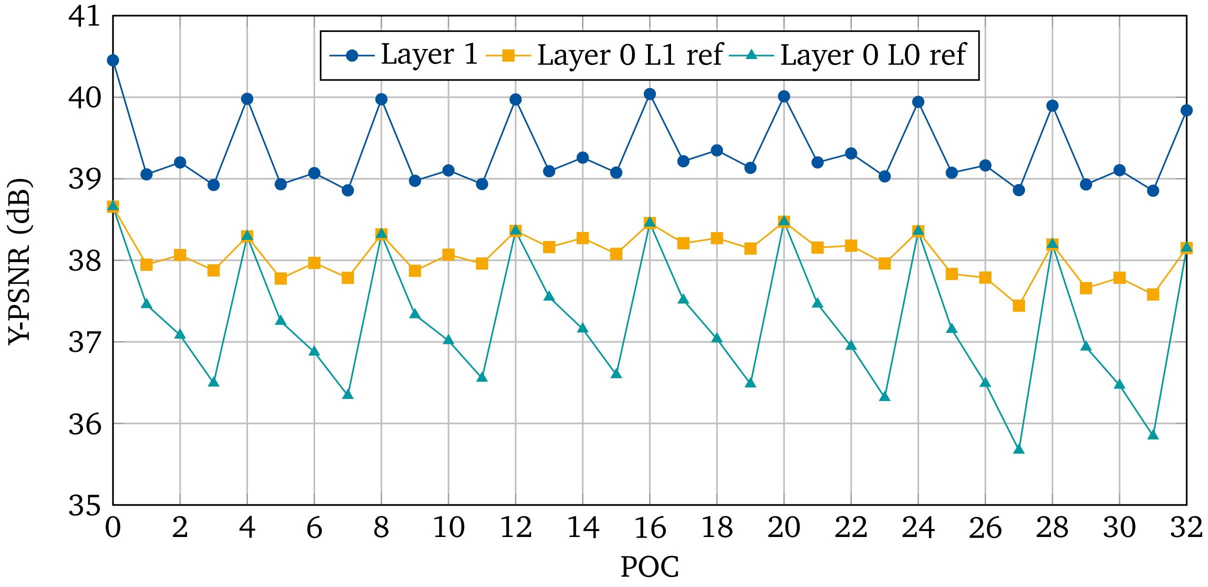


**Figure 3.6** The Y-PSNR of the first 33 frames of the sequence BasketballDrive with a base layer QP of 30, an enhancement layer QP of 24, the low delay configuration and a key picture distance of 4. For the decoding of layer 0, either the layer 1 (Layer 0 L1 ref) or the layer 0 references (Layer 0 L0 ref) are used.

the encoder are still available and they can be decoded without modification and without any drift (compare to Figure 3.1). However, for the non-key picture in layer 0 (marked in red), the enhancement layer references which were used when encoding the sequence are not available. The unavailable references are substituted by the corresponding lower layer frames (red arrows). Since these are not the references that were used when encoding the sequence, the reconstruction of the non-key pictures contains some drift.

In Figure 3.6, the Y-PSNR of the first 33 frames of the sequence BasketballDrive using the low delay configuration are plotted. The upper blue curve plots the Y-PSNR values of the enhancement layer reconstruction (—●—). It can be observed that the reconstruction quality varies over time. The first picture of each GOP (the GOP size is 4), has a higher reconstruction quality than the remaining 3 pictures. This is caused by a slightly increased QP value for these remaining 3 pictures, which is specified in this manner in the common testing conditions.

The lower two curves display the layer 0 reconstruction Y-PSNR results. For the yellow curve (—■—), the enhancement layer reconstructions are assumed to be available for decoding. This is the drift free layer 0 reconstruction signal that is employed at the encoder side. The decoder could also obtain this signal if the higher layer is available and decoded. It should be noted that this theoretical drift free lower layer reconstruction is included as an additional reference for clarity. In most practical applications, this reconstruction would never be decoded or displayed when the higher layer reconstruction is available. For the other curve (—▲—), only the lower layer is decoded and the missing layer 1 references are replaced by the corresponding lower layer pictures as described. Because the same QP offsets as for the enhancement layer are applied, a similar uneven distribution of the reconstruction quality over time can be examined. Furthermore, it can be inspected how the reconstructions at the encoder and decoder drift apart over time. While the drift is still small for the first non-key picture in each GOP (1, 5, ...), it increases for the following two frames.

As described in Section 3.1.2, the prediction structure is modified such that non-key pictures in the lower layer do not predict from other non-key pictures beyond the next key picture. It can now be seen that this restriction is applied in order to limit the drift to the interval between two key pictures. E.g. in Figure 3.6 the frames 1, 2 and 3 contain drift; however, since the following frames do not refer to these erroneous frames, the drift is not propagated beyond frame 3. The next key picture (4) can be reconstructed error-free and the following frames (5, 6 and 7) again introduce drift. In this way, the drift is effectively confined to the frames within each GOP of 4 frames.

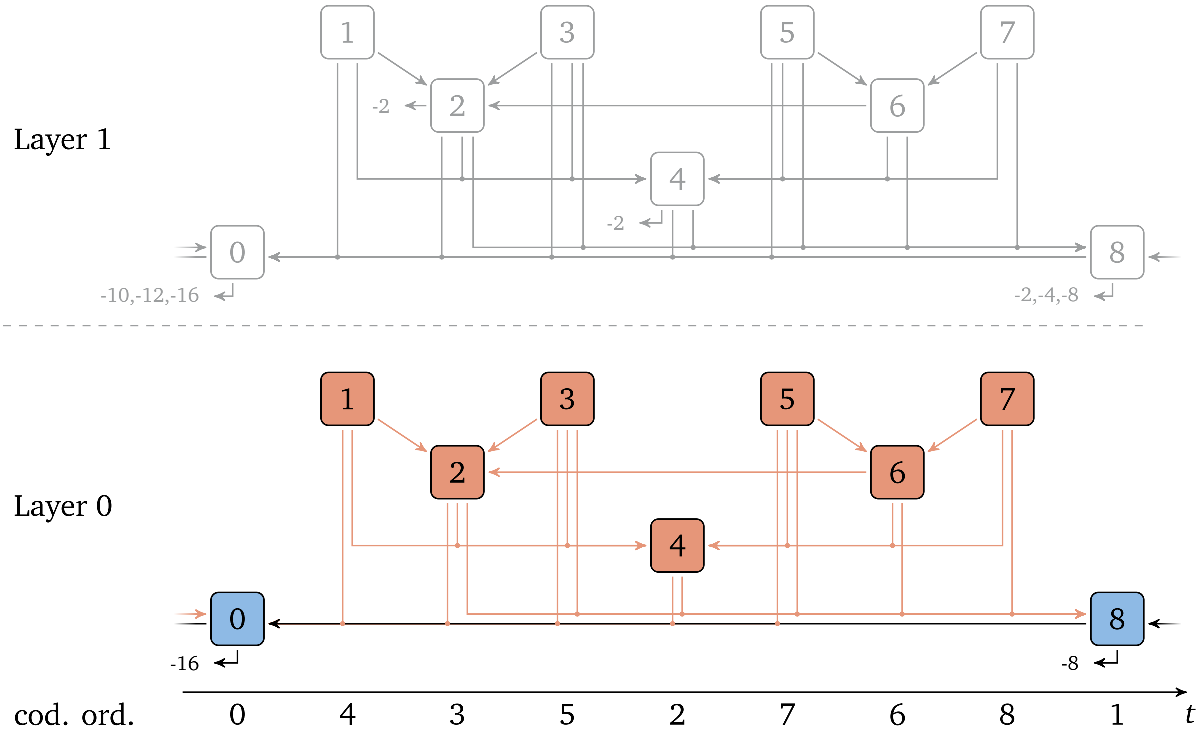


**Figure 3.7** Decoding only the lower layer for the random access configuration. The enhancement layer references for the non-key pictures in layer 0 are no longer available and have to be replaced by the corresponding reference frames in layer 0 (red arrows).

Figure 3.7 demonstrates the decoding of layer 0 for the random access configuration. Also here, the lower layer pictures take the place of the missing enhancement layer references (red arrows). The key pictures (0 and 8) are correctly decoded while the non-key pictures (1 to 7) hold some drift (indicated in red). The corresponding Y-PSNR results for the first 33 frames of the sequence BasketballDrive using the random access configuration are shown in Figure 3.8. Also for the random access configuration, an uneven distribution of the reconstruction quality over time can be observed which is defined by a QP variation in the common testing conditions. Because the coding order and the display order in the random access configuration are decoupled, there is no increase of the drift over time as there was for the low delay configuration; However, it can be seen that there is some change in the amount of drift depending on the position in the GOP. E.g. for picture 4 there is a certain

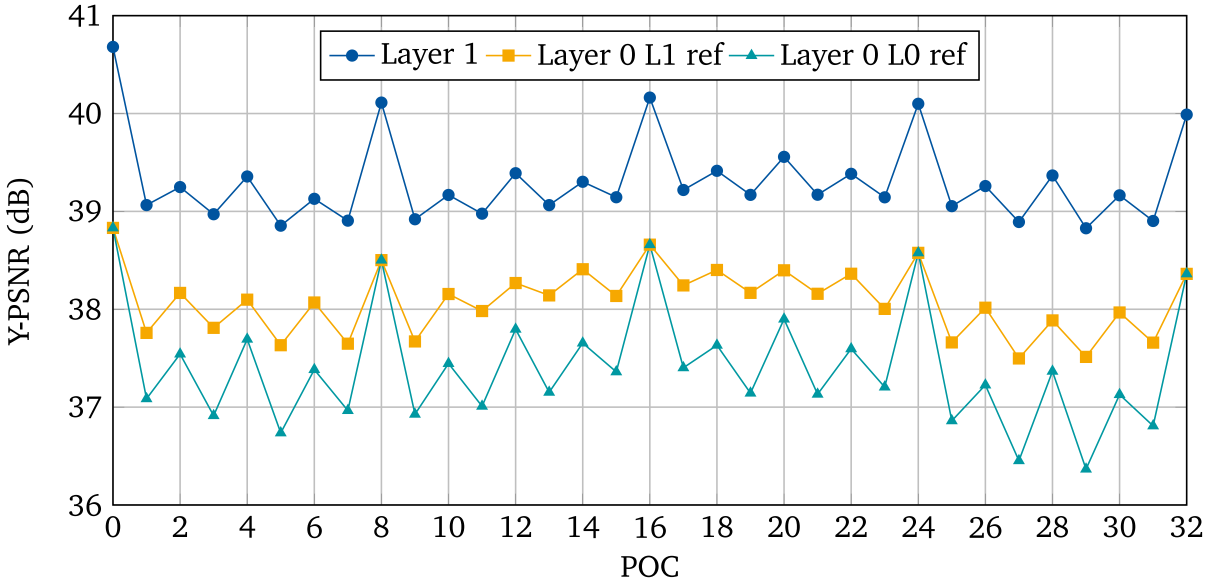


**Figure 3.8** The Y-PSNR of the first 33 frames of the sequence BasketballDrive with a base layer QP of 30, an enhancement layer QP of 24, the random access configuration and a key picture distance of 8. For the decoding of layer 0, either the layer 1 (Layer 0 L1 ref) or the layer 0 references (Layer 0 L0 ref) are used.

amount of drift. For the pictures 2 and 6 the drift increases and again for the pictures 1,3,5 and 7 the drift is highest. So for the random access configuration the drift does not increase over time but with the hierarchical level in the coding structure. In certain cases an extreme and regular variation of the reconstruction quality can be perceived and is often described as quality "pumping". While there is some variation in the quality for the drift when using the random access configuration, the variation is well below the level where it is perceivable in this manner (see also Figure 3.8). For the low delay configuration and long key picture distances, however, the quality fluctuation gets noticeable.

For the presented key picture concept, the key picture distance has a strong correlation to the amount of drift in the lower layer. In Figure 3.9 and 3.10 the Y-PSNR reconstruction values for the same sequence with a larger key picture distance is shown (8 for the low delay, and 16 for the random access configuration). In particular for the low delay configuration, a substantial drift increase can be monitored and the regular quality jumps also get noticeable.

In the software implementation which is used here, the encoder is not aware of the drift which occurs at the decoder side. The key picture distance is fixed and for the non-key pictures in the lower layer, the modifications as described in Section 3.1.2 are applied. The drift at the decoder side can in this case only be controlled by the key picture distance because by design, the drift is limited to the interval between two key pictures. For an improved drift control, an optimized encoder could be designed that is aware of the drift in the base layer that occurs if only the lower layer is decoded. The placement of key pictures in the stream as well as the layout of the general prediction structure could then be left to the encoder. Also a general ban on the non-key pictures to predict only from within the same key picture interval is not required in this case. The drift aware encoder could use the information

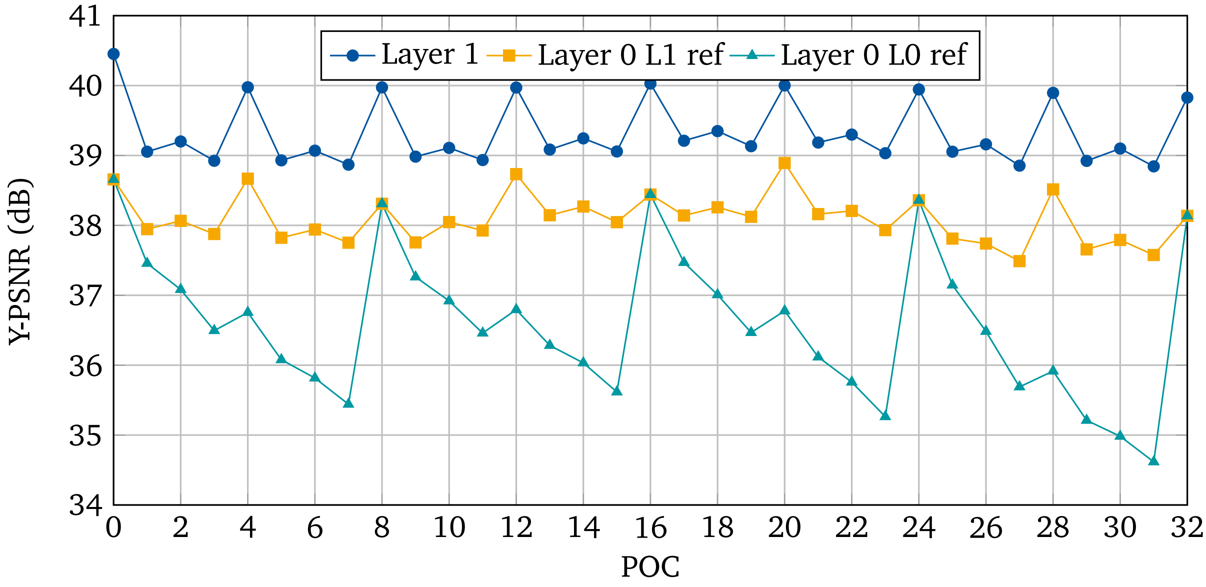


**Figure 3.9** The Y-PSNR of the first 33 frames of the sequence BasketballDrive with a base layer QP of 30, an enhancement layer QP of 24, the low delay configuration and a key picture distance of 8. Decoding of layer 0 is performed using either the layer 1 (Layer 0 L1 ref) or the layer 0 references (Layer 0 L0 ref) are used.

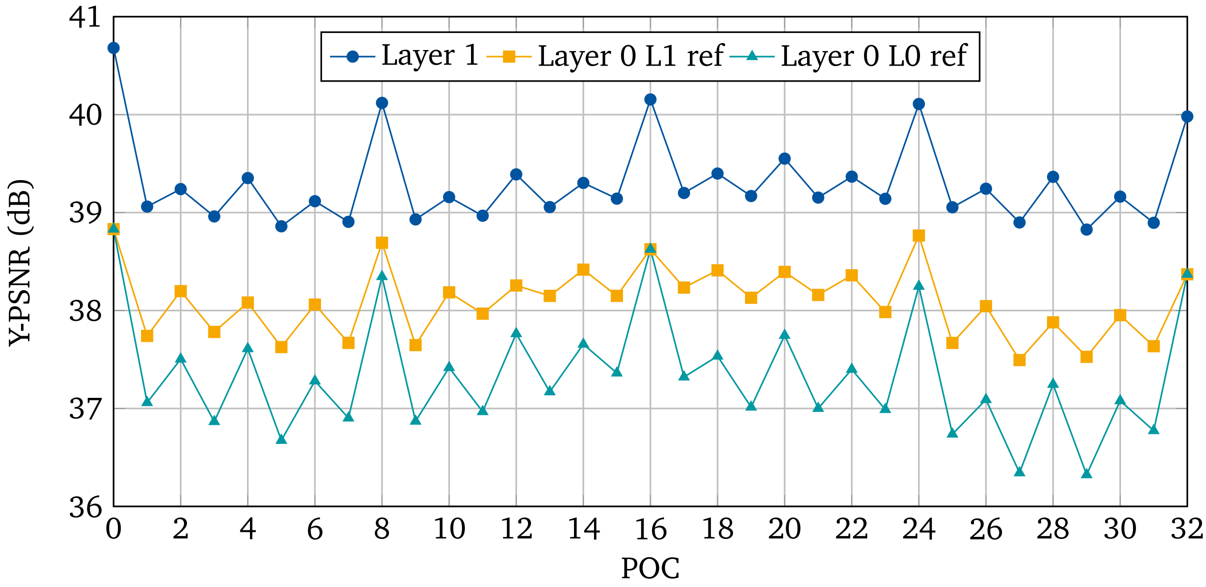


**Figure 3.10** The Y-PSNR of the first 33 frames of the sequence BasketballDrive with a base layer QP of 30, an enhancement layer QP of 24, the random access configuration, a GOP size of 8 and a key picture distance of 16. Decoding of layer 0 is performed using either the layer 1 (Layer 0 L1 ref) or the layer 0 references (Layer 0 L0 ref).

on the drift and perform prediction from reference pictures in such a ways that the drift is limited to a certain amount. The encoder could further perform an application dependent trade off between the decoder complexity, the coding performance and the drift in the lower layer. While the implementation of such an encoder and the evaluation of the achievable

**Table 3.1** Coding performance of the presented key picture concept compared to SHVC for the low delay configuration and different key picture distances (KPD) of 4, 8 and 12 pictures. 'Layer 1' corresponds to the higher layer quality and the bitrate of both layers.

| | | BD-rate (%) | | | BD-PSNR (dB) | | |
|---|---|---|---|---|---|---|---|
| | KPD | Y | U | V | Y | U | V |
| Layer 1 | 4 | -1.13% | -6.27% | -7.59% | 0.0303 | 0.1081 | 0.1280 |
| | 8 | -2.20% | -8.20% | -9.92% | 0.0624 | 0.1428 | 0.1706 |
| | 12 | -3.54% | -9.89% | -11.50% | 0.1024 | 0.1747 | 0.2027 |
| Layer 0 | 4 | 12.72% | 2.17% | 1.15% | -0.3922 | -0.0371 | -0.0440 |
| | 8 | 38.16% | 18.08% | 15.55% | -1.0207 | -0.2446 | -0.2535 |
| | 12 | 56.64% | 30.24% | 25.88% | -1.4207 | -0.3832 | -0.3879 |

performance is an intriguing field of further research, it is outside of the scope of this work.

### 3.1.5 Performance and Complexity

In this section, the coding performance and the decoder complexity of the presented single loop decoding scheme is compared to the coding performance and the decoder complexity of the reference SHVC coding approach. For both alternatives, the common testing conditions as described in Section 2.5.2 are followed. For the presented key picture approach, the modifications as described in Section 3.1.2 are implemented and various key picture distances are tested. For the random access configuration it is also tested to not use key pictures in the lower layer at all. In this case, the drift is only limited by the periodic intra pictures.

#### Coding Performance and Drift

First, the coding performance of the presented key picture coding approach compared to SHVC is evaluated. Tables 3.1 and 3.2 provide the rate and PSNR Bjontegaard Delta results for the key picture concept compared to the SHVC reference for the low delay and random access configurations, respectively (see Section 2.2.3). Both tables present the results for two decoding scenarios. In the upper part of the table (Layer 1), both layers of the scalable stream are fully decoded. The BD-rate and BD-PSNR values are then calculated from the enhancement layer reconstruction PSNR and the bitrate of both layers. In the lower part (Layer 0), only the base layer of the scalable stream is decoded for the key picture concept as well as for the SHVC reference. For the key picture concept, this reconstruction contains the aforementioned drift, while the SHVC reference is drift free. The PSNR results and the bitrate of only the lower layer are then used to calculate the BD-rate and BD-PSNR values.

From the low delay results in Table 3.1, it can be concluded that there is an overall performance increase when using the key picture concept. Depending on the key picture distance,

**Table 3.2** Coding performance of the presented key picture concept compared to SHVC for the random access configuration and different key picture distances (KPD) of 4, 8 and 16 pictures. 'Layer 1' corresponds to the higher layer quality and the bitrate of both layers. For the key picture distance marked with '-' there are no key pictures in the lower layer.

| | | BD-rate (%) | | | BD-PSNR (dB) | | |
|---|---|---|---|---|---|---|---|
| | KPD | Y | U | V | Y | U | V |
| Layer 1 | 4 | -1.08% | -4.75% | -5.26% | 0.0310 | 0.0813 | 0.0907 |
| | 8 | -1.23% | -7.41% | -8.49% | 0.0334 | 0.1297 | 0.1514 |
| | 16 | -0.13% | -7.71% | -9.37% | 0.0026 | 0.1344 | 0.1638 |
| | - | -1.65% | -10.51% | -12.94% | 0.0381 | 0.1809 | 0.2253 |
| Layer 0 | 4 | 3.17% | 1.46% | 1.66% | -0.0949 | -0.0236 | -0.0325 |
| | 8 | 5.16% | 2.75% | 2.90% | -0.1538 | -0.0433 | -0.0548 |
| | 16 | 13.25% | 11.01% | 10.53% | -0.3611 | -0.1597 | -0.1742 |
| | - | 21.76% | 17.93% | 15.14% | -0.5401 | -0.2487 | -0.2475 |

the bitrate reduction ranges from 1.13% to 3.54% BD-rate for luma and from 6.27% to 11.5% for chroma. At the same time, the lower section of the table exposes the drift when only the lower layer is decoded. Since the drift induces a decrease in reconstruction quality, the loss in quality (BD-PSNR) is considered here. Depending on the key picture distance, the BD-PSNR loss for luma ranges from 0.39 dB to 1.42 dB while the loss for chroma ranges from 0.03 to 0.39 dB. It can be inferred that with a larger key picture distance the overall coding performance as well as the drift in the lower layer increases. It can be observed that, compared to the luma component, the overall performance increase for the chroma components is higher while at the same time the BD-PSNR reduction in the drift prone lower layer is lower. As it was explained in Section 2.2.3, it must be considered that for the calculation of the delta values, the same overall bitrate is used for all three components. Because of this connection, it might be possible at the encoder side to transfer some of the higher gains from the chroma components to the luma component. At the same time, however, it must also be considered that the chroma components are sub-sampled by a factor of two in both horizontal and vertical direction so that the results of this are not directly predictable.

For the lower layer results, a combination of two effects must be considered. For one, the non-key pictures in the lower layer can utilize a reconstruction of higher quality which changes the allocated bitrate as well as the reconstruction quality. Secondly, there is a certain amount of drift which degrades the reconstruction quality. When the allocated bitrate for the lower layer of the key picture concept is compared to the lower layer bitrate in SHVC, it can be seen that the average bitrate allocation slightly increases. While this is counterintuitive, it can be explained by the implemented encoder decision of the software. On the encoder side, various coding options are tested and the one with the best rate-distortion tradeoff is chosen. No direct control on the bitrate is applied. So, also with better reference pictures, the rate-distortion decision may select a mode with a higher bitrate. In addition to this slight increase in rate, the reconstruction quality compared to SHVC is lower because of the drift. In the enhancement layer, the overall reconstruction quality is very similar while the necessary bitrate is decreased.

For the random access configuration (Table 3.2), a similar relationship between the key picture distance, the overall performance gain and the drift in the lower layer can be noted. For the key picture distance labeled '-', no pictures in the lower layer are defined as key pictures. In this case the drift is only confined by the periodic intra pictures in the lower layer. Compared to the low delay configuration, the random access configuration exhibits a much lower amount of drift. E.g. for a key picture distance of 8 and the random access configuration there is 1.23% overall coding gain and 0.15 dB BD-PSNR reduction in the lower layer. For a key picture distance of 4, the low delay configuration has a similar overall coding gain of 1.13% while the lower layer demonstrates a BD-PSNR reduction of 0.39 dB.

For the random access configuration a special case occurs when the key picture distance increases from 8 to 16. While the drift in the lower layer increases as anticipated, the overall coding performance decreases. This can be explained with the GOP size for the random access configuration which is set to 8 frames in both cases. In case of a key picture distance of 8, each 16th frame can predict from the two previous key picture -8 and -16; However, if the key picture distance is increased to 16, each 16th picture can now only predict from picture -16. By excluding the more efficient prediction from picture -8, the coding performance is noticeably impaired. Possibly, this loss could be compensated by adding another hierarchical layer to the prediction structure and extending the GOP size to 16 pictures. However, such variations on the prediction structure were not further investigated here.

### Average Decoder Complexity Reduction

As described before, with the key picture concept in place, it is possible to implement a decoder with a reduced complexity. This is because for the non-key pictures in the lower layer, the motion compensation operations and, for the most part, also other reconstruction operations can be skipped. In this section the achievable decoder complexity reduction is evaluated. In order to gather information on the complexity, the same decoder measurements as in Section 2.5.4 are considered. Also the same sequences and testing conditions were used and the results were averaged over the whole test set. All values are presented relative to the values of the SHVC reference, which is analogous to the coding performance results which were presented in the previous section. As an additional reference point, the complexity values for the single layer scenario relative to SHVC are also included here. [7]

Figures 3.11 and 3.12 show the measured complexity values for different key picture distances relative to SHVC for the low delay and random access configurations, respectively. As it was already concluded in Section 2.5.4, it can also be seen here that SHVC is approximately twice as complex when compared to the single layer coding scenario in which no scalability features are available and only one single layer bitstream at high quality is transmitted. Depending on the key picture distance and the employed measurement, varying complexity

[7] The values for single layer coding are identical to the values presented in Section 2.5.4. The only difference is that here, SHVC is used as another reference while in Section 2.5.4 SHVC is compared to single layer coding as a reference.

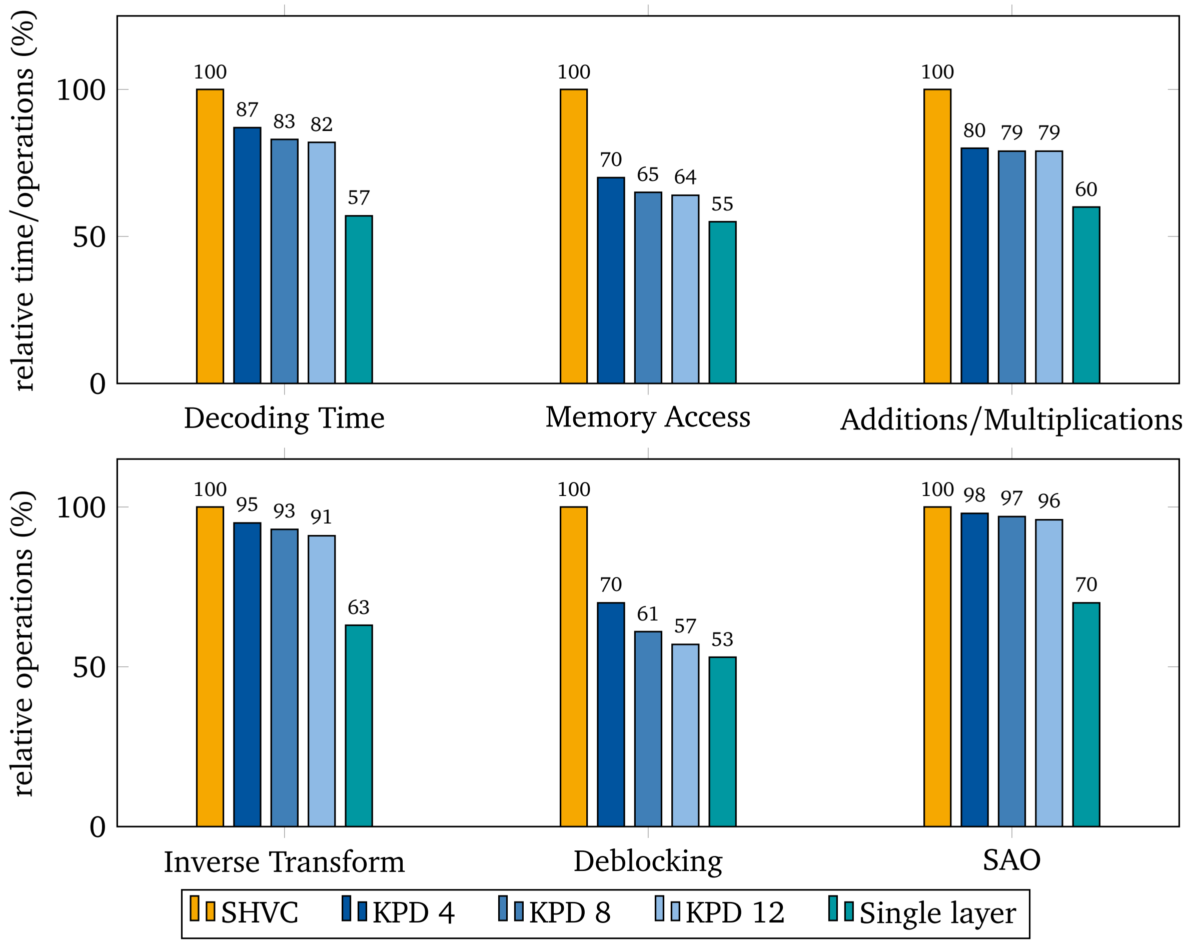


**Figure 3.11** Number of operations relative to the SHVC reference for the low delay configuration and varying key picture distance (KPD).

reduction results can be observed for the key picture concept. The first general indication of a complexity reduction is given by a drop in the decoder runtime. As it was noted before, these results should be interpreted with caution.

Especially for the memory access and deblocking operations, a significant reduction can be detected. For the longer key picture distances, the values for memory access and deblocking are almost reduced to the level of single layer coding. These results are reasonable since the prevalent mode of prediction for the non-key pictures in the lower layer is motion compensation and all motion compensation and deblocking operations for non-key pictures in layer 0 can be skipped with single loop decoding. Also the longer the key picture distance is, the more non-key pictures there are to be decoded with a reduced complexity.

The number of addition and multiplication operations used for motion compensation are also reduced, although not to the same extent. This can be traced back to the inter layer prediction process for SNR scalability. In SHVC, the motion vector for inter layer prediction is set to the zero vector, so that no filtering operation has to be applied. When single layer decoding is applied in this case, the memory access operations can be reduced but there are no filtering operations to skip. The number of memory access operations reduces, but the

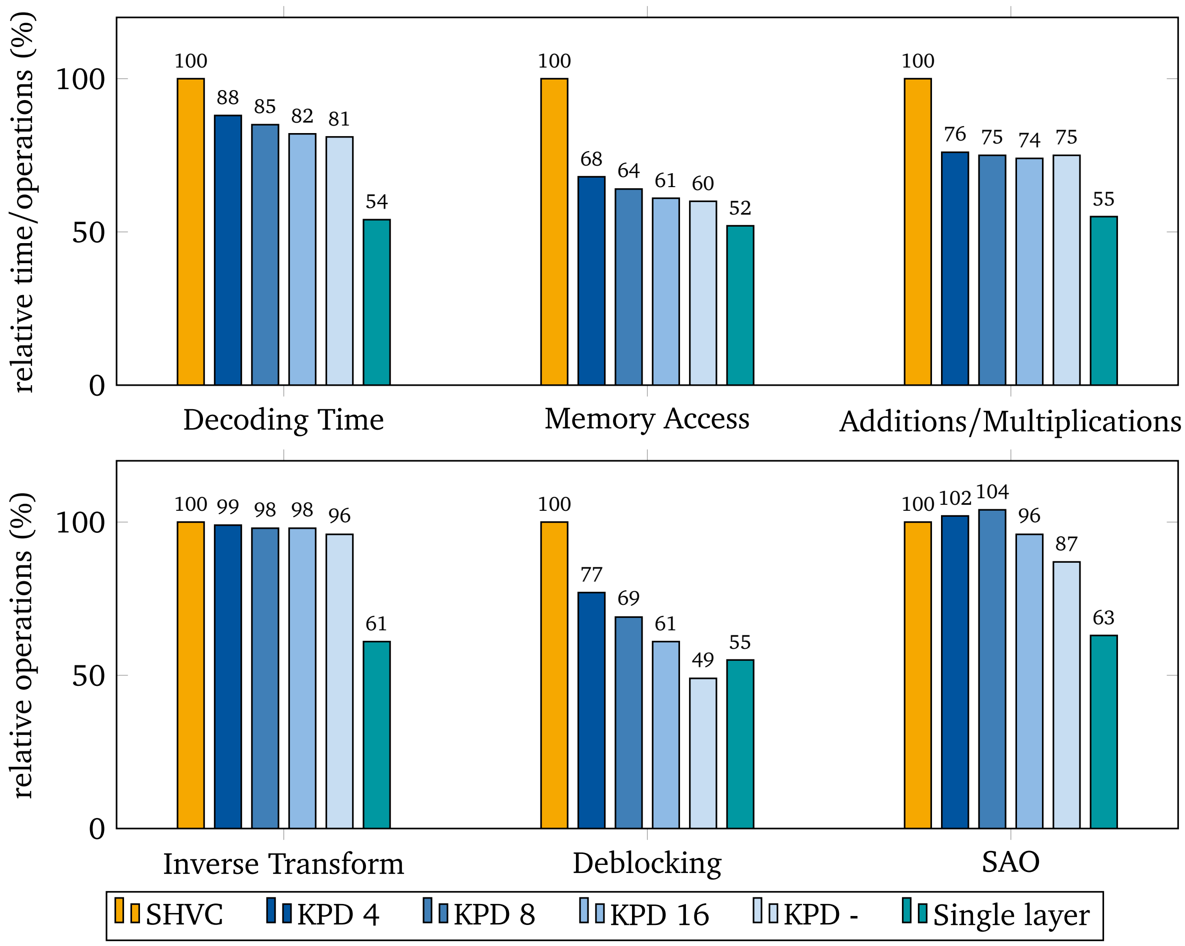


**Figure 3.12** Number of operations relative to the SHVC reference for the random access configuration and varying key picture distance (KPD).

number of additions and multiplications is unchanged.

For the number of inverse transformation and SAO filtering operations, there is only small to no complexity reduction with the single layer decoding approach. This can be explained as follows: As described in Section 3.1.3, the lower layer inverse transform operation can only be skipped if it is not used in the higher layer by inter layer prediction. This, however, occurs very rarely. For non-key pictures, both layers use the identical reference pictures from layer 1 for prediction. Because of this it is almost always beneficial for the higher layer to use the lower layer block (which uses motion compensation and an additional residual signal) instead of performing motion compensation from the same reference pictures and coding a new residual signal. So in the majority of cases, the enhancement layer encoder will opt to use a lower layer block with a coded residual and no inverse transform operations can be saved. For SAO, the issue is that SAO filtering is primarily applied for the frames in the higher layer and for the key pictures in the lower layer. As full reconstruction of these pictures is mandatory for single loop decoding as well, there is no reduction of SAO operations. The slight increase might then result from the higher quality reference that is available from the lower layer.

**Worst Case Complexity Reduction**

Another important aspect that ought to be considered is the worst case complexity. Worst case considerations are so essential because they provide an upper limit for the performance of a decoder. While it is highly unlikely to ever occur, a decoder which complies to the standard must be able to handle a worst case bitstream. Therefore, every reduction in the worst case complexity lowers the demand on the decoder.

Because in the common testing conditions the frequency of intra frames varies depending on the sequence, the impact of the intra frames on the worst case complexity considerations are neglected here. For the memory access and the additions/multiplications operations, similar worst case conclusions can be drawn. For SHVC, the worst case of memory access occurs when bi-directional prediction is used for every CU in the lower and higher layer. At the same time, using the key picture concept, the higher layer can be reconstructed without motion compensation being processed in the lower layer. However, this only applies to all non-key pictures in the coding structure. For all key pictures, full reconstruction of both layers is required just as it is in SHVC. For the random access configuration this is every 8th- and for the low delay configuration every 4th frame. This results in a worst case reduction of memory access operations for the key picture concept to 56.25% and 62.5% compared to SHVC for the random access and low delay configuration, respectively. For the additions and multiplication operations, the same relative numbers apply. In the worst case, bi-directional motion compensation is performed using an 8-tap filter in both directions. While this must be performed for both layers in SHVC, motion compensation can be skipped in the lower layer for non-key pictures by using the key picture concept.

As previously mentioned, the inverse transformation in the lower layer can only be skipped if the higher layer does not perform inter layer prediction from the lower layer. So unfortunately, in the worst case in which inter layer prediction is used for every CU, inverse transformation must be performed for both layers as it is for SHVC. So in terms of worst case complexity, there is no reduction in the number of inverse transform operations. However, this issue is addressed in the following chapters where the inter layer prediction process is changed in a way that allows for reconstruction of the higher layer with only one inverse transform operation.

### 3.1.6 Visual Test

While the drift is constrained to the pictures within one GOP, it is not controlled spatially inside of the picture or temporally for prediction from within the GOP. [8] Inside each picture, the drift could be concentrated in a specific region where it is perceived as visually more annoying than the measured PSNR result (which gives an average over the whole image) suggests. In order to better judge the subjective influence of the drift on the visual quality,

[8] The encoder implementation that is used here is not aware of the drift that occurs when only the lower layer is reconstructed. However, it is absolutely possible to design an encoder that is able to measure the anticipated decoder drift. This information about the drift could then be used to consider the drift in the encoder decision and to limit it to a certain level within each frame.

**Table 3.3** Coding performance of the presented key picture concept compared to SHVC for the configuration used in the visual test: random access configuration, key picture distance 16, $\Delta QP$ -6, 1080p sequences only.

| | BD-rate (%) | | | BD-PSNR (dB) | | |
|---|---|---|---|---|---|---|
| | Y | U | V | Y | U | V |
| Layer 1 | -0.07% | -6.98% | -8.75% | 0.0049 | 0.0983 | 0.1457 |
| Layer 0 | 12.85% | 10.53% | 9.93% | -0.2985 | -0.1314 | -0.1568 |

a visual test was conducted between the lower layer reconstruction of SHVC and the lower layer reconstruction of the key picture concept which contains drift. The test was designed using the stimulus-comparison method and a pair comparison. Where possible, the ITU recommendations on subjective visual tests were followed [ITU-T B.910; ITU-R BT.500; ITU-R BT.710]. The tests were conducted in a room with controlled lighting settings. A group of three out of the total 13 participants per test run were seated in front of the screen at a distance corresponding to three times the height of the screen.

Following the ITU recommendations, the stimulus of each A/B comparison consists of 10 seconds of sequence A followed by a 2 second break using a gray screen and another 10 seconds of sequence B. Finally, there is a 5 second time window for voting. The order of A and B is chosen randomly. The tested sequences were taken from the common testing conditions. Because no 4k screen was available at the time of testing, only the 5 sequences with a resolution of 1920x1080 pixels were tested for the 4 specified QP values. Additionally, a hidden reference was added for every sequence and at the beginning of the test 5 additional stabilization stimuli were prepended that are ignored for the evaluation. This results in a total of 30 A/B comparisons per test. For each pair, the participants were asked to rate the visual relation of the two representations on a scale from -4 to 4, where -4 implies that A was perceived as much better than B and 4 indicates the opposite. The center rating of 0 implies that no difference could be noticed.

For the SHVC reference as well as for the key picture concept, the random access configuration was used and the common testing conditions were obeyed. The key picture distance was set to 16 frames and the $\Delta QP$ to -6. As it was shown in Table 3.2, there is a considerable BD-rate and BD-PSNR reduction for the lower layer with drift in this case. The rather large key picture distance of 16 was chosen in order to test an extreme situation where the drift might get so significant that it is visible in certain regions of the image. Because there is no drift for the higher layer but rather a small coding performance increase, this visual test is only performed for the lower layer. The bitrates of the evaluated base layer representations were approximately the same. After all, a total of 12 valid result sheets were obtained from the visual tests.

Table 3.3 depicts the average BD-rate and BD-PSNR values that are measured for the conditions that were chosen for the visual test. It can be seen that for the lower layer the table indicates a luma BD-PSNR reduction for the key picture concept of 0.2985 dB. Figure 3.13 illustrates the corresponding results of the visual test. It can be concluded that, except for

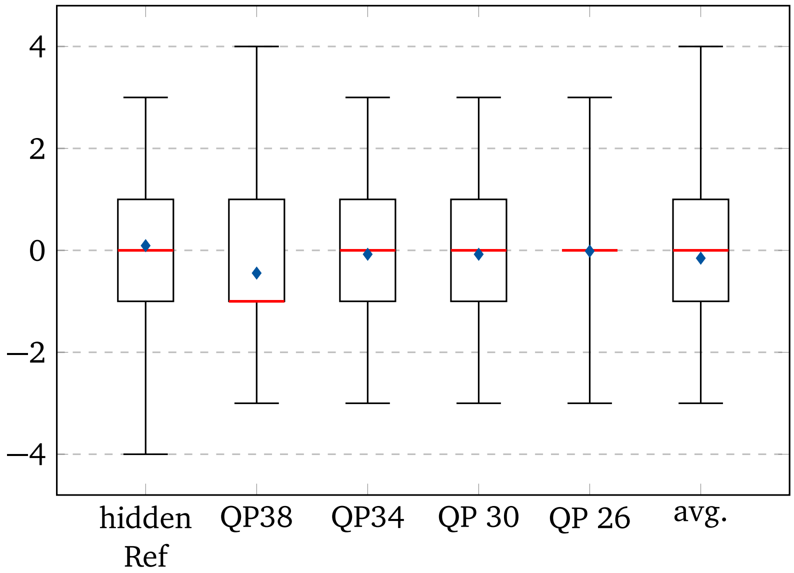


**Figure 3.13** The box plot of the visual test results. The box marks the interquartile range and the whiskers show the minimum and maximum values. The red line and the blue diamond indicate the median and the average value, respectively. Results are shown for the hidden references, per QP value and averaged over all QP values (excluding the hidden references). The value -4 implies that the SHVC reconstruction was perceived as highly superior over the reconstruction of the key picture concept. A value of 4 represents the opposite situation.

the QP value of 38, there is no tendency towards either of the two tested reconstructions and presumably no visual difference is perceivable here. For QP 38, the distribution is somewhat skewed towards the negative values. The median is at -1 and also the average value is negative (-0.15). While this indicates a slight tendency towards the SHVC reconstruction without drift, it cannot be considered conclusive evidence of the visibility of the drift.

Overall, the visual test indicates that while there is a certain amount of drift, the drift is not or only hardly visible. In particular, it is evident that the drift does not produce highly visible artifacts in the lower layer reconstruction which would severely affect the visual impression. In this visual test, only the random access configuration was evaluated. As it was pointed out in section 3.1.4, the drift in the lower layer, especially for a long key picture distance, is noticeably higher and shows jumps in quality over time. As a result the drift might be much more visible for the low delay configuration.

## 3.2 Decoding of Intermediate Layers

The approach that SHVC employs for quality scalability can also be considered a special case of spatial scalability with two or more layers of identical spatial resolution. The only difference is that no upsampling between the layers is performed. Conceptually, only full layers can be decoded. When considering the bitrate of each layer, the number of rate points is equivalent to the number of layers in the stream. In our two layer scenario, only two bitrate versions of the stream can be decoded. In addition, it is only possible to start decoding of the higher layer (switch to the higher layer) at random access points in the enhancement layer. Therefore, this concept of scalable coding is also referred to as coarse-grain quality scalability (CGS).

However, in some applications, a more flexible way of bitrate adaption is desired. In streaming applications for example, a more granular adjustment to changing channel conditions can benefit the average reconstruction quality and thereby the overall quality of the service. In another scenario, finer bitrate steps can be combined with error protection algorithms to increase the error robustness of a transmission. In this section, different approaches to enable a finer granularity of the scalable coding scheme are presented. While the first one is based on a combination of SNR and temporal scalability using SHVC, the second one employs the key picture concept and is related to medium-grain quality scalability (MGS) similar to the SVC coding design.

### 3.2.1 Temporal Scalability in SHVC

As described in Section 2.5, temporal scalability is already implemented in conventional HEVC. SHVC inherits this feature, adding the ability to combine it with the other forms of scalability. In temporal scalability, every layer has a different temporal resolution and each higher layer increases the frame rate of the reconstructed video. It was also mentioned that in SHVC all types of scalability can be combined in a single bitstream. So when joining SNR and temporal scalability, the additional prediction constraint for temporal scalability has to be obeyed. So for every temporal enhancement layer, only pictures from lower temporal layers can be used for prediction. Considering Figure 2.18, it can be seen that the hierarchical structure of the random access configuration only requires few changes to enable temporal scalability.

Figure 3.14 illustrates a possible prediction structure for the combination of SNR and temporal scalability if the hierarchical level of each picture is used to define the temporal layer and the temporal prediction restrictions are applied. It can be seen that there are now four temporal sub-layers ($T_{id}0$ to $T_{id}3$) in each layer, where every temporal sub-layer doubles the frame rate of the previous sub-layer. As intended by temporal scalability, each temporal sub-layer can be discarded without preventing the lower temporal layers from being decodable.

It is now feasible to skip decoding of temporal sub-layers in the enhancement layer without affecting the decodability of the base layer or the lower temporal sub-layers in layer 1. Figure 3.15 illustrates a scenario in which the highest two temporal layers in the enhancement layer ($T_{id}3$ and $T_{id}2$) are discarded. The layer 1 pictures 0, 4, and 8, however, can still be decoded because they reside in lower temporal layers. From all reconstructed pictures, an output video signal could be compiled which contains the pictures 0, 4, and 8 from layer 1 and the pictures 1, 2, 3, 5, 6 and 7 from layer 0 (marked in blue). [9]

Using this approach, we can subdivide the enhancement layer into four sub-layers, increasing the number of rate points of the stream from 2 to 5. Figure 3.16 displays the bitrate of each

[9] While a reconstruction of such a video with pictures from both SNR layers is not specified by the SHVC standard, it is technically possible and can be used as a step towards a scalability with finer granularity.

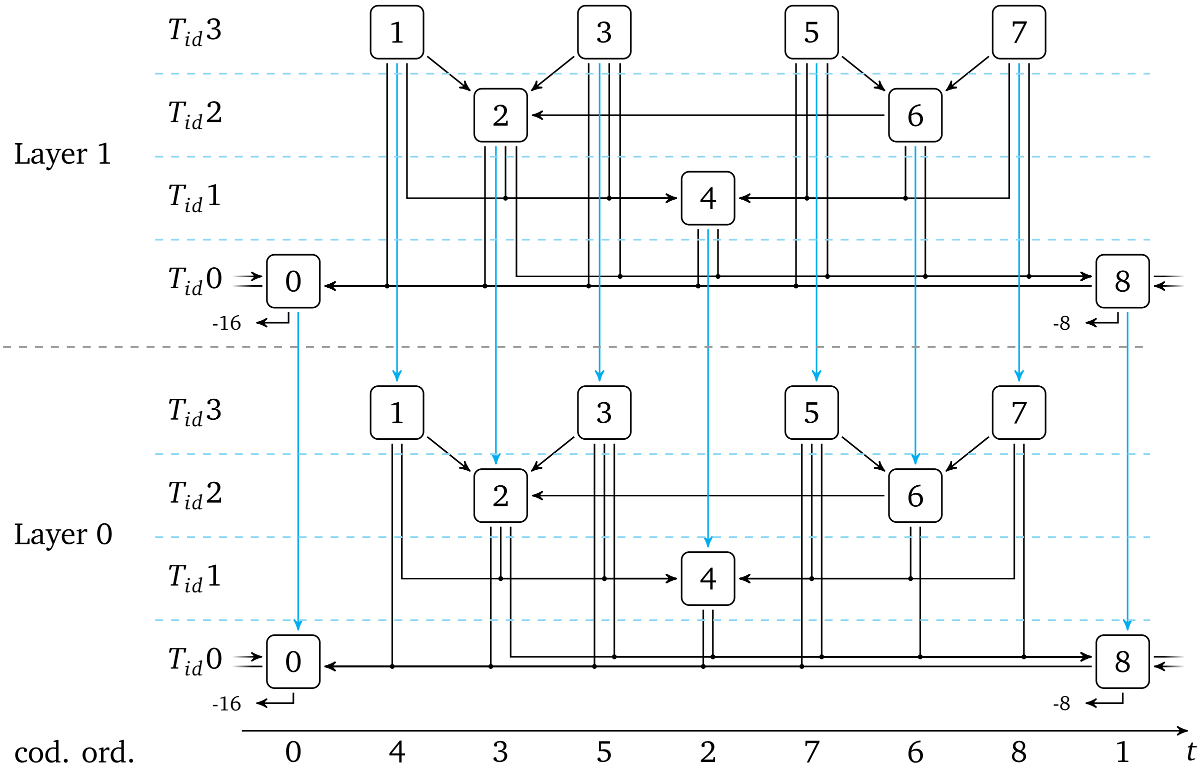


**Figure 3.14** One GOP of the random access configuration with a GOP size of 8 pictures and two layers. Each layer uses temporal scalability with 4 temporal sub-layers ($T_{id}0$ to $T_{id}3$).

temporal sub-layer for the sequence Cactus and a $\Delta QP$ of -6 for different lower layer QP values. For each plot, the left bar shows the entire bitrate of the two layers as they can be decoded in SHVC. For the corresponding right bar, the layer 1 bitrate is broken down into the bitrates of each temporal sub-layer ($T_{id}0$ to $T_{id}3$). The lowest rate point only contains layer 0 (with all its temporal sub-layers). For the succeeding rate points the temporal sub-layers of layer 1 are added one by one. For a layer 0 QP of 30, for example, decodable bitstreams at bitrates of 3.6, 8.1, 9.4, 10.8 and 11.9 Mbps can be extracted. It can be seen that for the first bitrate step from 3.6 to 8.1 Mbps, the allocated bitrate more than doubles while the bitrate increase for each additional step is much smaller. This situation is most severe for the base layer QP of 38. It should be noted that the bitrate allocation here is controlled by the QP settings which are defined in the common testing conditions. For practical applications, a finer step size might be preferred which could be realized by alteration of the encoder configuration or a different type of rate control.

As can be seen from Figure 3.16, the bitrate per layer and the distribution of bitrate amongst the temporal layers depend on the base layer QP. Furthermore, these are also dependent on the $\Delta QP$ value and on the sequence. This results from the common testing conditions, which specifies fixed QP values and the chosen prediction structure that defines a QP offset per temporal layer. [10] Considering these fixed settings, we have no influence on the result-

[10] While the prediction structure was somewhat modified to enable temporal scalability, we are still complying with the common testing conditions as closely as possible. No changes were applied to the sequences, QP values or to the hierarchical QP offset which is defined in the common testing conditions.

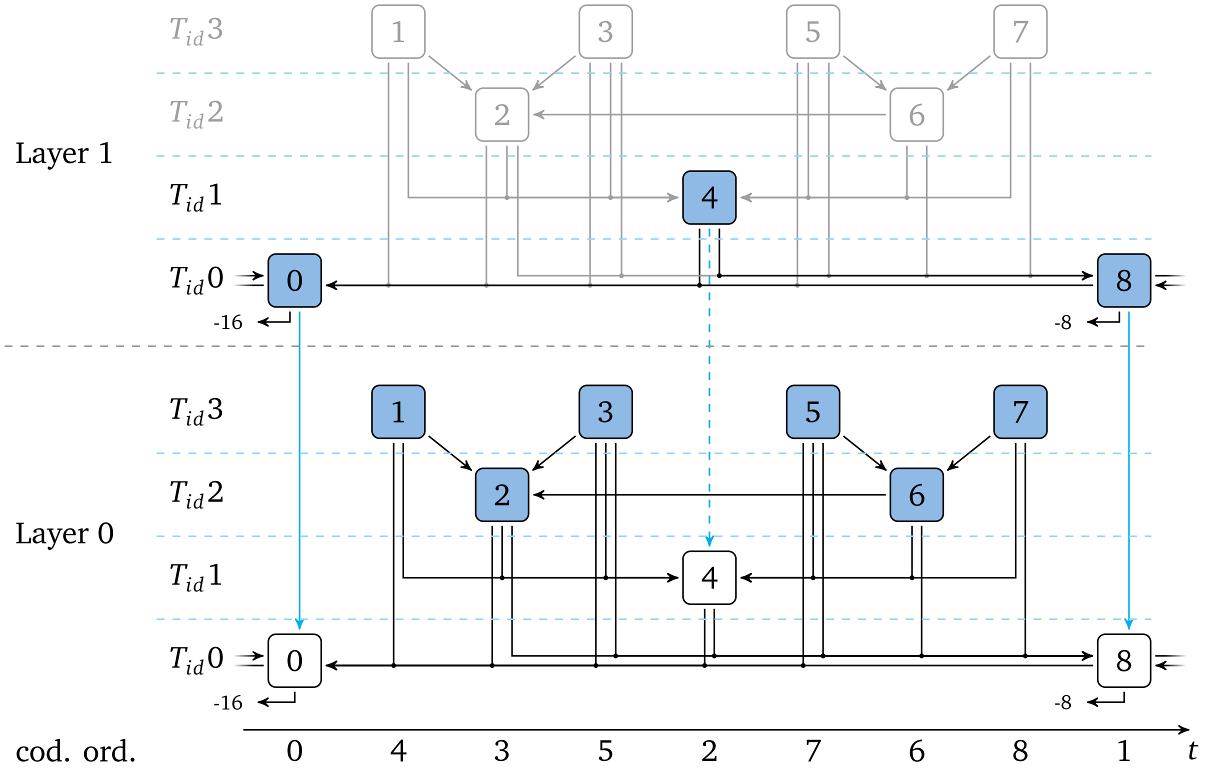


**Figure 3.15** One GOP of the random access configuration with a GOP size of 8 pictures, two layers and four temporal sub-layers. While all temporal layers of layer 0 are decoded, only the temporal layers $T_{id}0$ and $T_{id}1$ of layer 1 are decoded. The pictures that are output are marked in blue.

ing allocation of bitrate per layer. A more practical encoder implementation, however, may choose a very different allocation of bitrate per temporal sub-layer. For example in a streaming application, it might be beneficial to allocate an equal amount of bitrate per temporal sub-layer so that an improved adaption to varying channel conditions is within one's means.

Figure 3.17 displays the corresponding bitrate and Y-PSNR results for the sequence Cactus at a $\Delta QP$ of -6. It can be observed that with each additional temporal sub-layer that is decoded, the overall bitrate of the stream increases along with the average reconstruction quality. While this appears to be the solution to our initial request for a more granular type of scalability, there are some drawbacks that render the presented approach unsuitable for practical applications. Firstly, it can be seen from Figure 3.17 that when adding the temporal sub-layers, the average Y-PSNR gain compared to the additional bitrate is very low for the sub-layers. While this relation increases per temporal sub-layer, the coding performance is significantly impaired when the temporal sub-layers are only partly decoded. [11]

Another problem becomes obvious when we take a look at the reconstruction quality over time. Figure 3.18 illustrates the Y-PSNR of the first 33 frames of the Cactus sequence. While it can be seen in Figure 3.17 that the average reconstruction quality increases with each additional temporal layer that is decoded, it becomes obvious in Figure 3.18 that only the

[11] In Figure 3.17, the two main layers of SHVC are plotted and connected by a straight line (-●-). This line, however, does not represent an optimal achievable performance and the results for the temporal sub-layers should not be compared to it.

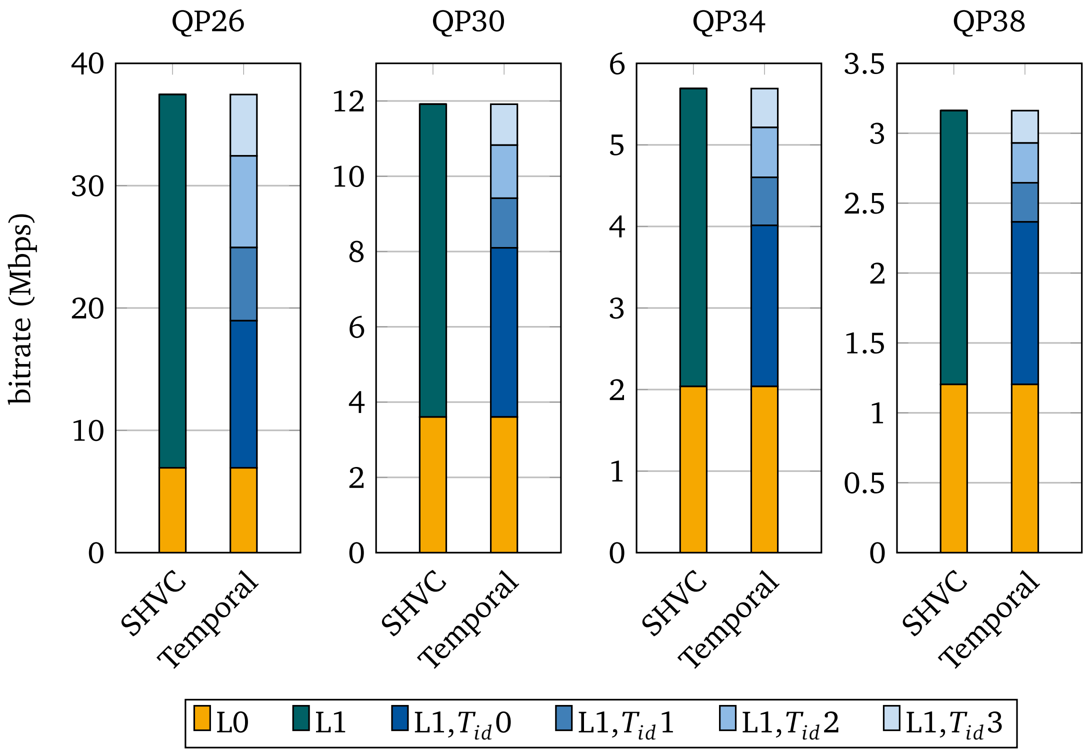


**Figure 3.16** Bitrates of the base layer (L0) and the enhancement layer (L1) of SHVC for the sequence Cactus and a $\Delta QP$ of -6 using the random access configuration and temporal scalability with four layers. In the respective right graph the bitrate of the enhancement layer is split into the four temporal sub-layers ($T_{id}0$ to $T_{id}3$).

reconstruction quality of discrete pictures increases. In this example, only the temporal layers $T_{id}0$ and $T_{id}1$ of the enhancement layer are decoded. This results in a reconstruction quality increase to the layer 1 quality for every fourth picture. The reconstruction quality of all other frames remains at layer 0 quality. When this sequence of frames is played back, a visually very disturbing flicker can be perceived when there is a sudden increase or decrease in reconstruction quality. This is caused by the fact that the additional bitrate of a temporal sub-layer only increases the reconstruction quality of the pictures in the sub-layer. The left-over lower layer frames cannot benefit from the additional information and remain at layer 0 quality.

In the following section, it is evaluated how the key picture concept can be incorporated in order to mitigate the aforementioned disadvantages and to enhance the performance of gradual scalable decoding.

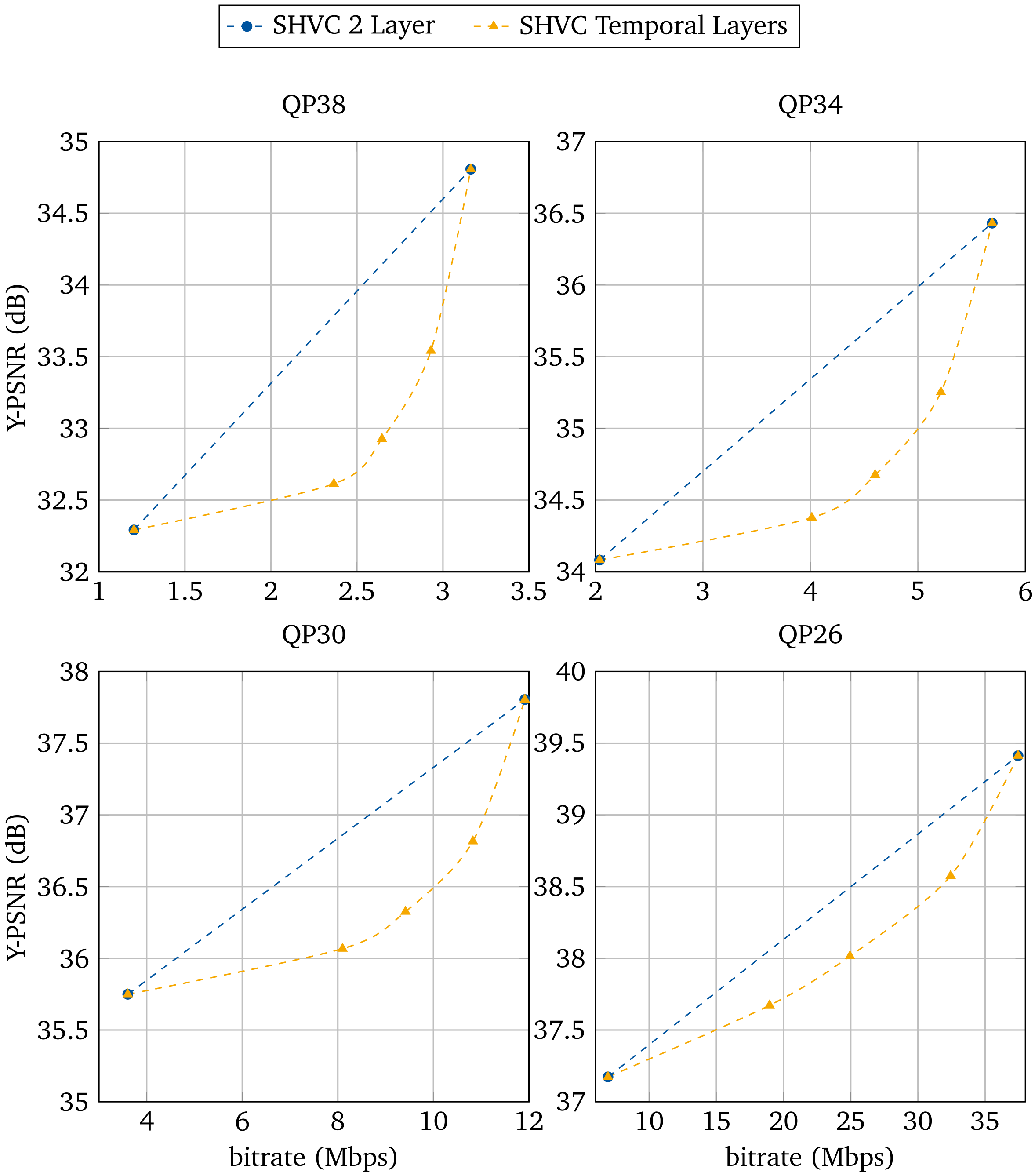


**Figure 3.17** Bitrate and Y-PSNR values for the sequence Cactus and a $\Delta QP$ of -6 as well as various lower layer QP values using the random access configuration and temporal scalability with four layers. For the blue curve (-●-) only the two SHVC layers are plotted by two points and connected by a straight line. For the orange curve (-▲-) also the four temporal sub-layers are shown.

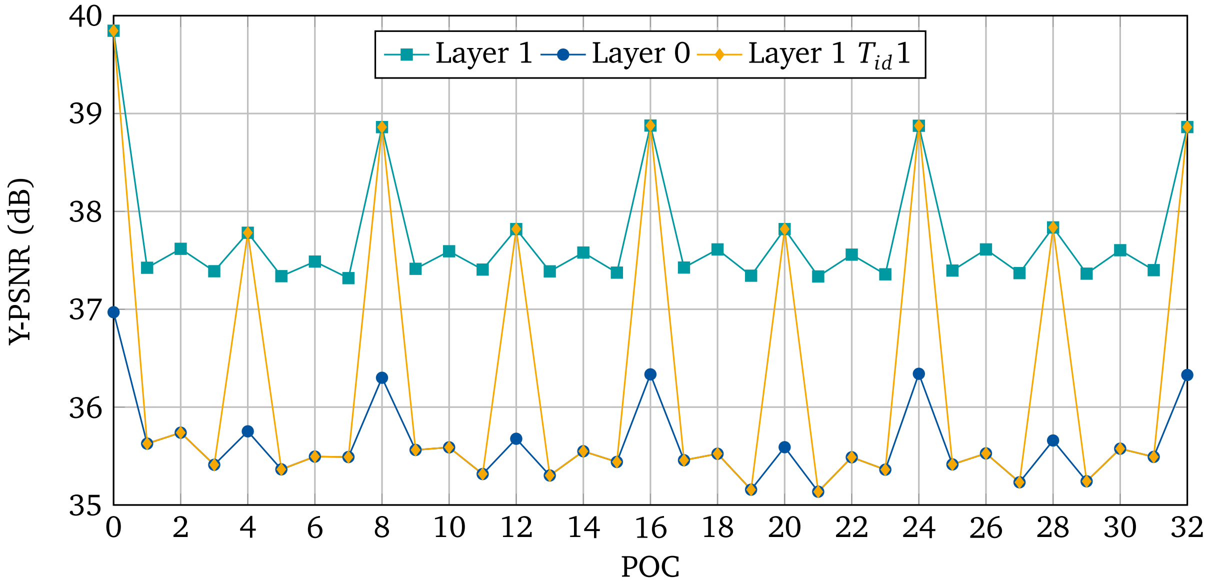


**Figure 3.18** The Y-PSNR of the first 33 frames of the sequence Cactus with a base layer QP of 30, an enhancement layer QP of 24, the random access configuration and 4 temporal sub layers. For the orange curve (—◆—) only the lower two temporal layers of the enhancement layer ($T_{id}0$ and $T_{id}1$) are decoded.

### 3.2.2 Medium-granular Quality Scalability

In this section, we apply the technique for a finer granular scalability which was described in the previous section, but use it in combination with the key picture concept which was described in Section 3.1. Here, we start off with the prediction structure which was used for the key picture concept (see Figure 3.2) and add four temporal sub-layers to both layers. Figure 3.19 illustrates the modified prediction structure of the random access configuration for a key picture distance of 8 pictures.

As in the previous section, the hierarchical structure of the random access configuration is used to assign pictures to the temporal sub-layers. The decodability of each temporal sub-layer depends on its lower temporal sub-layers. For the key picture concept, again, periodic pictures in the lower layer are defined as key pictures (marked in blue in Figure 3.19). For these pictures, only prediction from the same temporal sub-layer in layer 0 is permitted. All remaining non-key pictures use the higher layer reconstructed pictures for prediction while obeying the restrictions from the key picture concept and from the temporal scalability at the same time: No prediction from beyond the limit set by the key pictures and no prediction from higher temporal sub-layers.

As before, temporal sub-layers can now be discarded from the bitstream without impeding the lower temporal layers from being decoded. In Figure 3.20, the same example as in the previous section is depicted: The highest two temporal layers of layer 1 are removed from the bitstream and only the pictures marked in blue compose the final reconstruction of the video sequence. While this process is very similar to the approach described in the previous section, there is one key difference which is induced by the key picture concept:

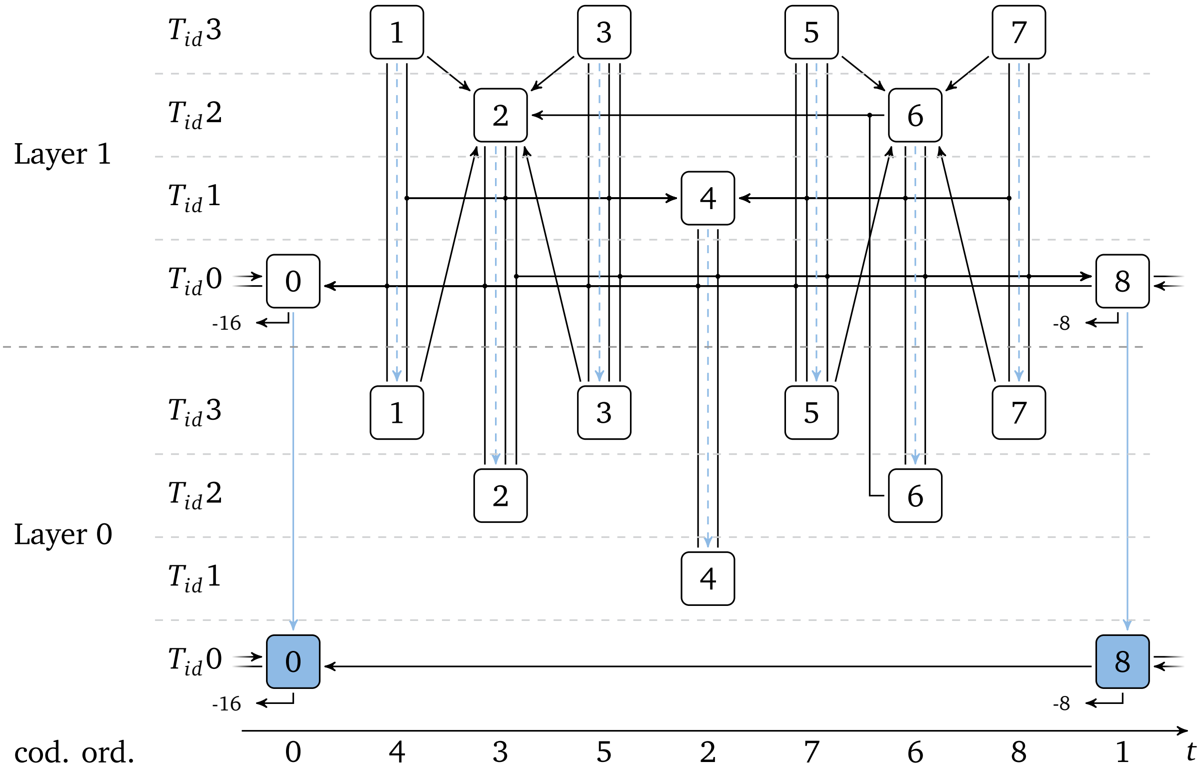


**Figure 3.19** One GOP of the random access configuration with a GOP size of 8 pictures and two layers using the key picture concept. Each layer uses temporal scalability with 4 temporal sub-layers ($T_{id}0$ to $T_{id}3$). The key pictures in the lower layer are marked in blue.

Since the lower layer non-key pictures utilize the higher layer pictures as references (whenever available), all pictures can benefit from an increased reconstruction quality of the higher layer pictures. In our example, the temporal sub-layers $T_{id}0$ and $T_{id}1$ of layer 1 are decoded, while $T_{id}2$ and $T_{id}3$ are discarded. This results in an enhanced reconstruction of the pictures 0, 4 and 8. However, unlike before, the reconstruction quality of the pictures 1, 2, 3, 5, 6 and 7 also increases because they reference the enhanced higher layer pictures for motion compensated prediction. Figure 3.21 shows the corresponding reconstruction quality over time. When comparing to Figure 3.18, the improved reconstruction of the lower layer non-key pictures becomes apparent.

In Table 3.4, the coding performance of the described decoding of intermediate layers is compared to the coding performance of the approach using only temporal layers in SHVC (see Section 3.2.1). In the bottom section of the table (L0), only the lower layer is decoded and the entire enhancement layer is discarded. From here upwards, the temporal sub-layers of layer 1 are added successively: For "L1,$T_{id}0$" the lowest temporal sub-layer $Tid0$ of layer 1 is also decoded and so forth until for "L1,$T_{id}3$" no temporal sub-layers are discarded and the entire higher layer is decoded.

It can be seen that for decoding of the base layer only (L0), there is a performance loss compared to the SHVC reference, while for decoding of the complete enhancement layer (L1,$T_{id}3$) there is a slight performance gain. The explanation is that at these two points we are again comparing the key picture concept to the conventional SHVC prediction structure.

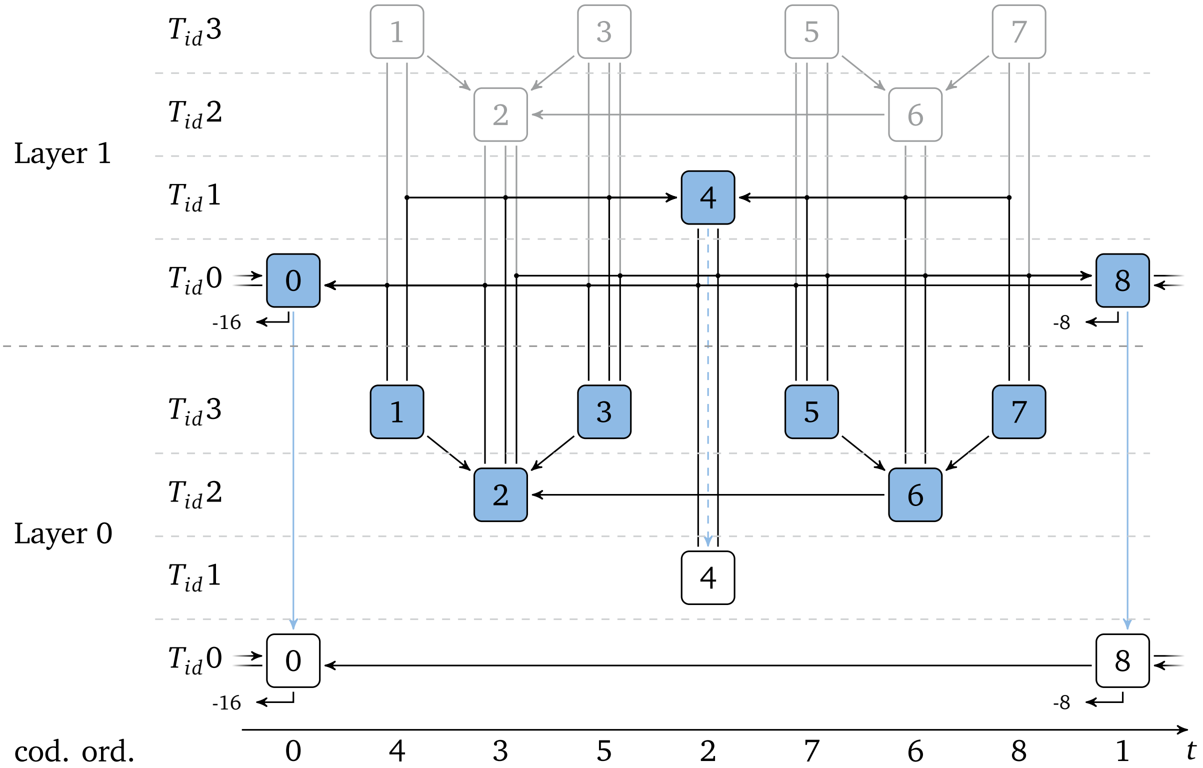


**Figure 3.20** One GOP of the random access configuration with a GOP size of 8 pictures, two layers, four temporal sub-layers and the key picture concept with a key picture distance of 8. While all temporal sub-layers of layer 0 are decoded, only the temporal sub-layers $T_{id}0$ and $T_{id}1$ of layer 1 are decoded. The pictures that are output are marked in blue.

It was already explained in Section 3.1.5 that the key picture concept generates a certain amount of drift at the lower layer while the overall coding performance is moderately increased. In Table 3.4 we can observe the same effect. [12] While they are not identical, the distribution of the bitrates amongst the temporal layers is very similar to those which were shown in Figure 3.16.

For the three test points at which only a subset of the temporal layers in the higher layer are decoded ("L1,$T_{id}0$", "L1,$T_{id}1$" and "L1,$T_{id}2$"), a significant coding performance increase compared to the decoding of temporal sub-layers using SHVC of 0.61 to 0.73 dB BD-PSNR is evident. This results from the fact that with the key picture concept, each additional temporal sub-layer increases the reconstruction quality of all pictures in the sequence instead of just some discrete pictures as the SHVC reference does. Another point that can be observed is, that the performance increase for the decoding of these intermediate layers is higher for a $\Delta QP$ of -6 than for a $\Delta QP$ of -4. This can be traced to the quality difference between the layers which is set by the QP delta. For a higher delta, the difference is higher and the additional temporal layer can improve the reconstruction even further.

In Figure 3.22, the rate-distortion graphs for the presented hierarchical decoding approach

[12] Note that the results are not identical to the ones that were collected in Section 3.1.5 since the prediction structure of the two tests are slightly different. For the test in the previous section, no temporal scalability was used.

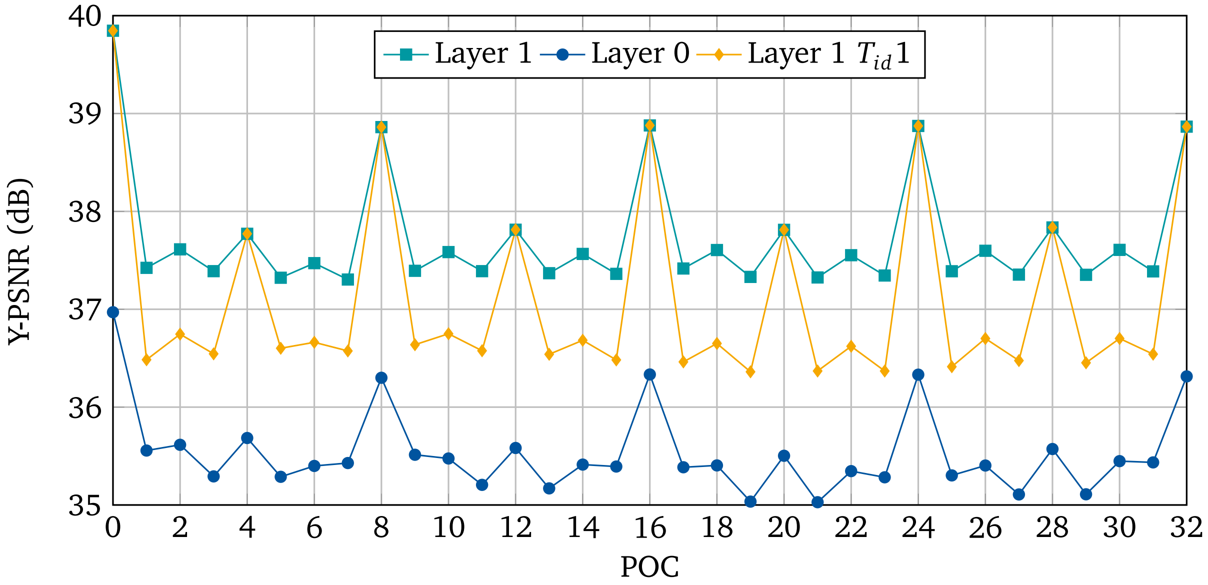


**Figure 3.21** The Y-PSNR of the first 33 frames of the sequence Cactus with a base layer QP of 30, an enhancement layer QP of 24, the random access configuration, 4 temporal sub-layers and the key picture concept. For the orange curve (—◆—) only the lower two temporal sub-layers of the enhancement layer ($T_{id}0$ and $T_{id}1$) are decoded.

are plotted with the aforementioned method using temporal layers in SHVC and SHVC using no temporal scalability. Similar conclusions can be drawn from these graphs. While there is a small loss at the lowest bitrate point, there also is a performance increase at the highest decodable rate point (except for QP 26, where the rate at the highest rate point is slightly decreased for this sequence). For the three additional intermediate bitrate points, however, the coding performance of the key picture concept significantly increases compared to the SHVC approach using temporal layers.

While only results for the random access configuration are presented here, comparable results are obtained when the described methods are applied to the low delay configuration. The corresponding prediction structure modifications and performance results for the low delay configuration can be found in Appendix C.

### Comparison to SVC

As noted before, the concept of medium grain quality scalability (MGS) was already included in scalable video coding (SVC) [h.264/AVC; SMW07; SW08]. For MGS coding, a similar approach to the one described above was taken. In SVC, each frame carries two flags which indicate if the lower layer representation of a picture needs to be decoded and saved in the reference picture buffer and if the higher or lower quality reconstruction of the reference pictures are used for motion compensated prediction (See Section 2.4.1). In the implementation of the key picture concept which is presented here, no additional flags are transmitted. The information if the lower layer reconstruction needs to be saved and which reference to use is explicitly set by the fixed key picture distance and can be derived from the POC of the

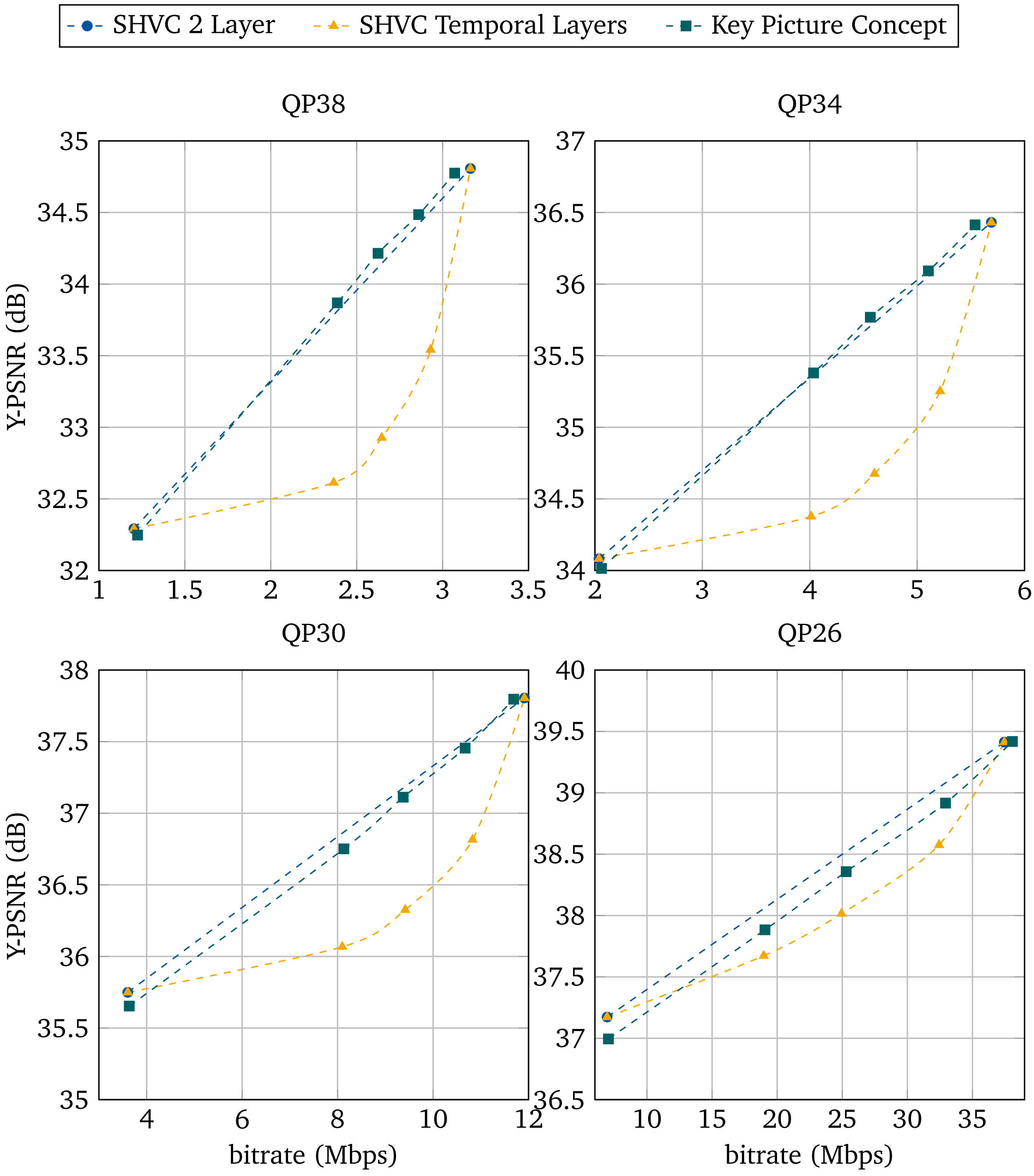


**Figure 3.22** Bitrate and Y-PSNR values for the sequence Cactus and a $\Delta QP$ of -6 using the random access configuration and temporal scalability with four layers. For the blue curve (-●-) only the two SHVC layers are plotted. For the other curves (-▲- and -■-) also the intermediate layers are plotted, where for -■-, additionally the key picture concept is used.

**Table 3.4** Coding performance of decoding intermediate layers using the higher layer temporal layers. For each temporal layer that is discarded in layer 1, we compare the performance using the key picture concept to conventional SHVC using temporal layers. The random access configuration is used and a key picture distances (KPD) of 8 pictures for the key picture concept.

| | | BD-rate (%) | | | BD-PSNR (dB) | | |
|---|---|---|---|---|---|---|---|
| | $\Delta QP$ | Y | U | V | Y | U | V |
| L1, $T_{id}3$ | -4 | -1.48% | -7.68% | -8.50% | 0.0438 | 0.1357 | 0.1553 |
| | -6 | -0.98% | -7.36% | -8.63% | 0.0286 | 0.1287 | 0.1505 |
| | Avg. | -1.23% | -7.52% | -8.57% | 0.0362 | 0.1322 | 0.1529 |
| L1, $T_{id}2$ | -4 | -16.89% | -25.06% | -25.23% | 0.4970 | 0.4456 | 0.4891 |
| | -6 | -24.14% | -34.46% | -35.55% | 0.7252 | 0.6756 | 0.7631 |
| | Avg. | -20.52% | -29.76% | -30.39% | 0.6111 | 0.5606 | 0.6261 |
| L1, $T_{id}1$ | -4 | -19.35% | -30.08% | -30.02% | 0.5872 | 0.5559 | 0.6071 |
| | -6 | -27.66% | -41.73% | -42.31% | 0.8751 | 0.8747 | 0.9857 |
| | Avg. | -23.51% | -35.91% | -36.17% | 0.7312 | 0.7153 | 0.7964 |
| L1, $T_{id}0$ | -4 | -16.61% | -30.20% | -30.09% | 0.5111 | 0.5689 | 0.6234 |
| | -6 | -23.45% | -41.21% | -41.82% | 0.7575 | 0.8813 | 1.0079 |
| | Avg. | -20.03% | -35.70% | -35.95% | 0.6343 | 0.7251 | 0.8157 |
| L0 | -4 | 3.82% | 2.02% | 2.17% | -0.1168 | -0.0309 | -0.0416 |
| | -6 | 5.07% | 2.65% | 2.85% | -0.1529 | -0.0407 | -0.0540 |
| | Avg. | 4.45% | 2.33% | 2.51% | -0.1348 | -0.0358 | -0.0478 |

picture. [13] This approach using a locked prediction structure is certainly not optimal for real world applications. Analog to SVC, a flag could be added to the slice header of each picture to indicate if the picture is a key picture or not. In order to not conflict with the existing standard, this flag could be added in a slice header segment extension which is ignored by standard compliant SHVC decoders. Another option is to enforce the use of temporal layers and signal the temporal ID of the layers that are defined to only contain key pictures (E.g. in the sequence parameter set extension). While a flexible implementation using some signaling is certainly desirable, the specific signaling scheme has only a minimal impact on the presented results. While some possible signaling schemes were discussed, we will not specify a concrete implementation here.

Another difference is the employed layer handling. In SVC, every NAL unit header extension contains a quality ID (*quality_id*). This allows for a bitstream to hold up to 15 quality enhancements per picture, where every enhancement is packaged in an individual NAL unit.

[13] A key picture, as it is used in the presented implementation, is then equivalent to a picture in SVC for which it is indicated to decode and save the lower layer representation of the picture as well as to use the lower layer representations of the reference pictures.

There is even a mode in which the quality enhancement is performed in the transform domain and the additional transform coefficients are encoded in separate quality enhancement NAL units [Kir+07]. With the signaling in the NAL unit header extension, it is now possible to discard any enhancement layer NAL units from the bitstream. On the receiver side, the SVC decoder will attempt to decode the reconstruction of each picture with the highest reconstruction quality possible. This reconstruction is then used by pictures referencing the higher quality reconstruction. Because any enhancement NAL unit can be discarded, this scheme is also referred to as packet-based fidelity scalability (PFS).

Depending on the application, the discarding process can be conducted at any point during transport. While random discarding of enhancement NAL units is feasible, a much better approach is to discard NAL units based on their impact on the overall reconstruction quality of the stream. For this purpose, additionally to the aforementioned quality ID, the encoder can signal an optional priority id (*priority_id*) per NAL unit to assist the discarding process. It is, however, a non-trivial task to determine the impact of the loss of a certain quality enhancement package at the encoder side. For a detailed evaluation of this, the reader is referred to [Zha14].

In the presented approach, we use a fixed number of 2 layers with temporal sub-layers. The discarding of these temporal sub-layers (that can be considered quality enhancement layers) is assumed to always constitute complete temporal sub-layers, with the sub-layers having higher temporal IDs being discarded first. Also here, an approach similar to SVC could increase the flexibility of the medium granular quality scalability approach. For one, a similar best effort decoding approach can be applied at the decoder side. This way, discarding of NAL units on a much finer granular level can be enabled. Also the use of temporal layers could be removed and instead an indication on the importance of a NAL unit could be signaled by the encoder. Furthermore, the number of quality enhancements per picture is not required to be limited to one. While this is a very interesting field of future research, a detailed evaluation of these options in SHVC are outside of the scope of this work.

The most significant difference to SVC, however, is the extent of the implemented changes. For scalable coding in SVC, several of the existing coding tools from single layer AVC were modified as well as several new coding tools were added. Firstly, the arithmetic coded syntax is changed and new flags on the macroblock level are introduced which signal new coding modes. For the inter layer prediction process, several modes can be used to take advantage of the lower layer information: If the collocated reference block uses intra prediction, inter layer intra prediction may be used. Otherwise, the higher layer may perform inter prediction using inter layer motion compensation. Furthermore, inter layer residual prediction may be employed for coding of the higher layer residual signal. Contrary to SVC, the presented coding scheme complies to the high level syntax only approach and does not apply any changes to the lower layer coding tools.

## 3.3 Conclusion

In this chapter, two main points were presented and discussed:

**SHVC decoder complexity** In the first part, it was shown that especially for quality scalability, the complexity increase of SHVC compared to single layer coding with HEVC is quite significant. Depending on the measured value, the complexity of SHVC increased up to twice the complexity of HEVC. It was further demonstrated how the key picture concept can be employed in SHVC to significantly reduce the decoder complexity down to a level much closer to single layer HEVC. While an extension to spatial scalability may be feasible, the focus here lies on SNR scalability because the most severe complexity overhead occurs in this scenario. Similar to previous implementations of the key picture concept, a drift is introduced when the enhancement layer is discarded from the bitstream, while at the same time a small overall performance increase can be observed when all layers are decoded. A visual test was conducted which revealed that the drift had no measurable impact on the perceived visual quality. It was shown that there is a trade-off between the complexity reduction, the overall gain and the drift in the lower layer. An encoder could perform this tradeoff dynamically by intelligent placement of the key pictures depending on the application and the sequence. It could hereby also control the drift in the lower layer. It was shown how, unlike in SVC, the key picture concept can be applied without changes to the lower layer syntax or coding tools. While some modifications to the encoder as well as the multilayer decoder must be applied, they are all limited to the higher layer syntax and the scheme can thusly be considered a high level syntax only approach. While a compliant HEVC decoder can be used to decode the lower layer, the multilayer decoder requires some slight changes. [14] A higher layer decoder could furthermore not make use of the possible complexity reduction and opt to fully decode all layers.

**Medium-granular quality scalability** In the second part, it was demonstrated how the key picture concept can be combined with temporal scalability to enable highly efficient decoding of intermediate layers. These additional decodable rate points can then be used to enable a much more flexible adaption to changing channel conditions or optimize the error robustness of the stream. On top, decoding with reduced complexity as enabled by the key picture concept is also applicable. While the method as it is implemented here is quite rigid and provides only a limited number of additional rate points, it could be easily be extended to support more flexibility similar to packet-based fidelity scalability in SVC.

As previously noted, the current version of SHVC is not well suited for certain applications in which very flexible bitrate adaption is required or the decoder complexity is a limiting factor. For these applications, the presented flexible inter layer prediction approach can minimize the drawbacks of SHVC with only minor impact on the SHVC coding performance. While this approach was also proposed in the standardization of SHVC, there were various concerns about the drift as well as possible issues with existing hardware implementations. In the end, it was not adopted and the current, significantly more complex, approach was adopted. In the future, a more flexible high level syntax approach could benefit the acceptance of scalable coding schemes in general.

[14] Also an unmodified SHVC decoder could decode the higher layer bitstream (although because of the modified prediction structure, the reconstruction for the higher layer would exhibit a high amount of drift in this case).

# 4 Residual Refinement

In the scalable coding approach of SHVC, the bitstream is organized in layers. The lowest layer is compatible to HEVC and establishes the basis for all higher layers (it is also referred to as the base layer). As described in Section 2.3, the reconstruction signal is obtained on a block basis in two steps. Firstly, a prediction from previously coded frames or from already decoded neighboring blocks within the same frame is computed. In the second step, a residual for the block can be signaled. If a residual signal is present, the quantized transform coefficients are extracted from the bitstream and the residual signal is reconstructed by inverse quantization and inverse transformation. Finally, the reconstruction is obtained by adding the prediction and the residual signal in the spatial domain.

Higher layers of an SHVC bitstream add information that enhances the reconstruction quality compared to the lower layers. Therefore, they are also referred to as enhancement layers. The coding of higher layers is very similar to conventional coding with HEVC. The key difference is that each higher layer can make use of the information from the lower layer by inter layer prediction. For inter layer prediction, the lower layer reconstruction of the corresponding block is copied into the higher layer reference picture buffer and subsequently employed for prediction in the higher layer. As for HEVC, the encoder may choose to signal a residual signal after the prediction. The reconstruction of the block is then again formed by adding the prediction (which is the reconstruction from the lower layer) and the residual signal. This addition is also performed in the spatial domain (See Section 2.5).

So altogether, when using inter layer prediction, the reconstruction in the first enhancement layer is composed by adding the prediction in the lower layer and up to two residual signals. For a scenario with more than two layers, each additional enhancement layer may add a further residual signal. So in order to obtain the higher layer reconstruction, the entire residual reconstruction process is possibly performed multiple times. This involves multiple scans of the coefficients, as well as multiple inverse transformation operations per residual signal.

It depends on the sequence and on the encoder settings how much inter layer prediction is used. Figure 4.1 shows the average relative utilization of intra, inter and inter layer prediction for two SNR layers in SHVC. [1] It can be seen that for the lower layer, mostly inter prediction is used. However, for low QP values, the relative amount of intra prediction rises to up to nearly 40%. Naturally, no inter layer prediction is available in the lower layer. In the higher layer, the amount of intra prediction is negligible. For inter and inter layer prediction, the values are highly dependent on the QP. For a high QP, inter layer prediction is only used for a low percentage of image samples. With a decreasing QP, however, this value grows and inter layer prediction becomes the dominating type of prediction. In Figure 4.2, it is

[1] The results were obtained using the SHVC reference software version 6.0. Except for the extended QP value range, the common testing conditions for the random access configuration were followed.

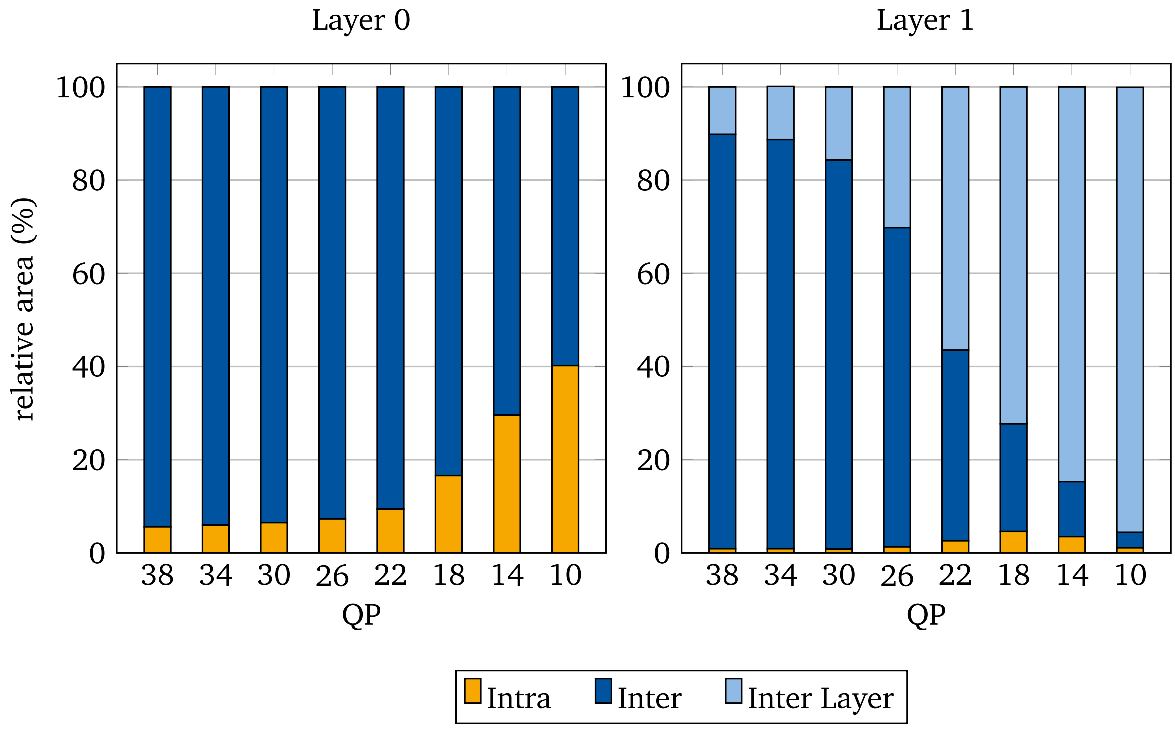


**Figure 4.1** Average relative utilization of prediction modes for two layers in the random access configuration depending on the lower layer QP. The delta QP for the higher layer is -6.

shown how much of the output bitrate is used for which part of the coding. The first part, the high-level syntax, includes all parameter sets as well as slice headers. From the figure it is evident that this high-level syntax only plays a minor role for the bitrate. The second part contains all symbols that are coded on a coding unit (CU) level including the SAO information, CU tree and prediction information. In the last component, all transform information like the TU tree, coded block flags and transform levels are pooled together. From the figure it becomes evident that, for both layers, most of the bitrate is used for coding of the transform levels. Depending on the QP, the amount of bitrate that is used for the transform levels ranges from around 40% (QP 38) to over 90% (QP 10). From these results it becomes evident that especially for high bitrates where a high amount of inter layer prediction is used and the higher layer mostly codes additional residual information, the coding overhead per layer gets even more severe.

In the following, a method is presented that refines the lower layer residual signal directly in the transform domain. This way, only one inverse transformation is required at the layer that is being reconstructed. Furthermore, only one scan of the coefficients is needed. However, the approach also comes with some drawbacks compared to the SHVC coding approach. While in SHVC, the block size for inter layer prediction is independent of the lower layer block structure and can span multiple blocks from the lower layer, refinement can only be performed for each TU in the lower layer while the size of the TU must remain unchanged. This loss of flexibility certainly comes at the expense of some loss in coding performance.

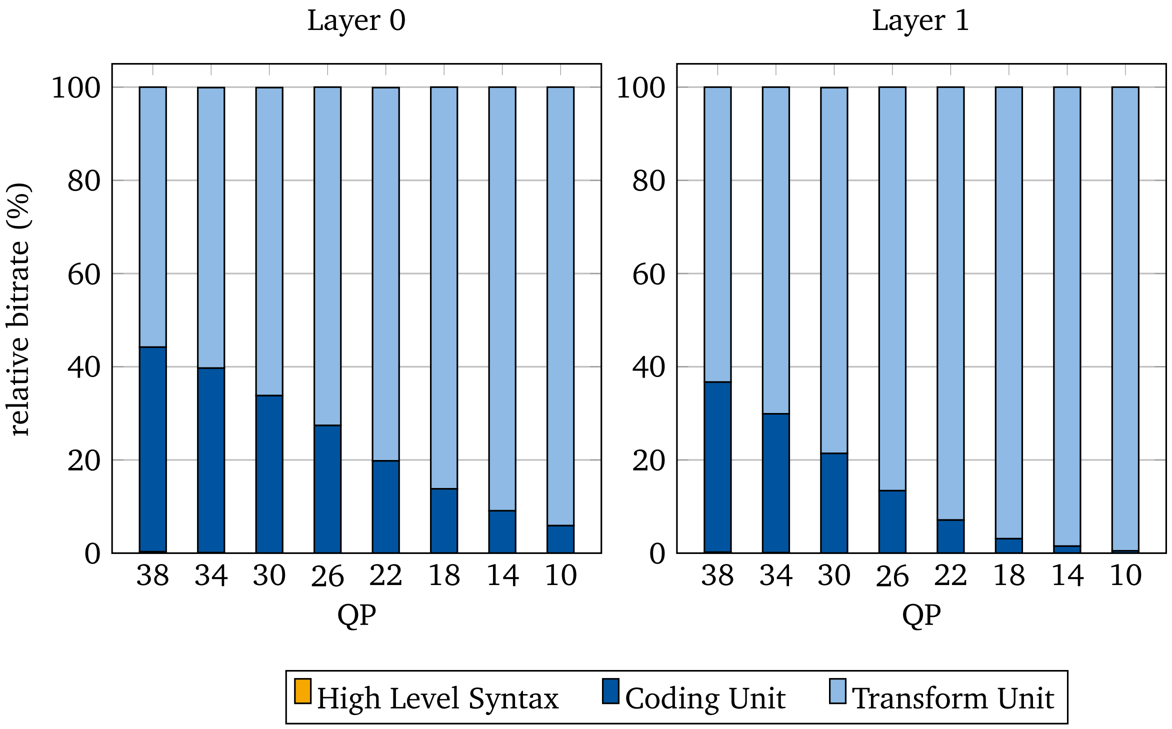


**Figure 4.2** Average relative allocation of rate in lower and higher layer for the random access configuration depending on the lower layer QP. The delta QP for the higher layer is -6.

## 4.1 Inter Layer Refinement in the Transform Domain

The basic idea of residual refinement is to add information to the transform coefficients directly in the quantized transform domain. A related concept to achieve this goal is the concept of embedded quantization. Figure 4.3 illustrates an example for embedded quantization. The input values to the quantization are unsigned and range from 0 to MAX. If a quantization to 2 bits is applied to the input values, the value range is split into four intervals (indicated by the black lines). Following the concept of quantization, all input values within each range are quantized to the same level and reconstructed by the same reconstruction value (indicated by the red crosses) [Ohm15].

When another bit is added to the quantization output, each quantization interval is further split into 2 subintervals. The additional bit then indicates which of the two new reconstruction values to use for the input value. With each additional bit, the accuracy of the reconstruction value increases which lowers the expected quantization error. [2] The process can be continued until the desired bit depth is attained. This is referred to as embedded quantization because all quantizers using less than $N$ bits are a subset of the $N$ bit quantizer (they are embedded in the $N$ bit quantizer).

Embedded quantization can be used to create a scalable representation of values like the

[2] The reduction in quantization error strongly depends on the signal statistics. If a uniform distribution for the values from 0 to MAX is assumed for the given quantization example, each additional bit lowers the expected squared error by a factor of $\frac{1}{8}$. See 2.1.3.

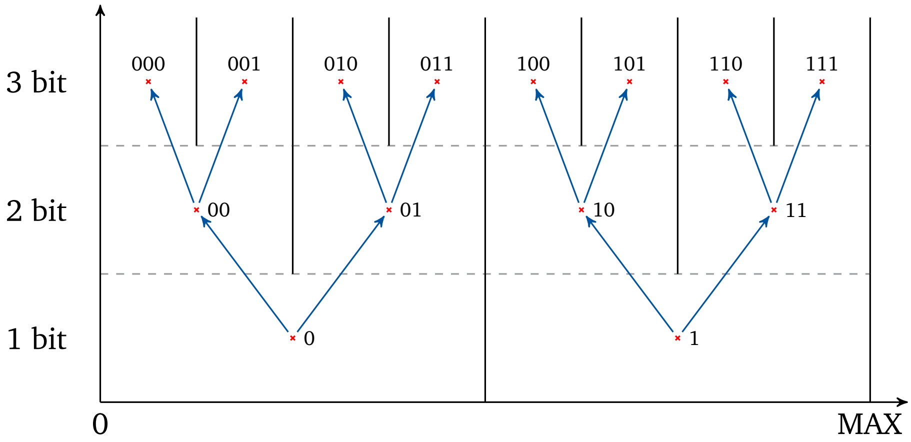


**Figure 4.3** Embedded quantization scheme with coded bits for unsigned values with a value range from 0 to MAX. The black lines are the decision lines while the reconstruction values are plotted with red crosses. With each additional bit, the quantization interval is split into two and the bit indicates which one of the two new reconstruction values is chosen (arrows).

transform coefficients. Firstly, the transform coefficients are quantized using $N$ bits. The bits are arranged from most to least significant. The most significant bit (MSB) corresponds to the bottom of Figure 4.3 and each additional bit is less and less significant until the last and least significant bit (LSB) is reached. The bits of all coefficients are then rearranged bit-plane by bit-plane. Firstly, the most significant bit of each coefficient is signaled followed by the next less significant bit of each coefficient until all bits are processed. The entire bitstream is now sorted by significance and can be truncated at any point to obtain a sub-bitstream with a lower bitrate and reconstruction quality.

Similarly, embedded quantization can be performed for signed values using a dead zone quantizer. Figure 4.4 depicts an embedded dead zone quantization scheme using a varying number of bits. The input values are signed and range from -MAX to MAX. The sign and the magnitude are handled independently. For the coarsest quantization, one or two bits are coded. The first bit signals if the quantization level is 0 or not. If it is greater than 0, one more bit (S) indicates the sign of the value. When going to a finer quantization, the number of additional bits depends on the reconstruction value in the lower layer. For values that were not quantized to zero in the coarser quantization stage, one additional bit indicates which of the two possibilities is to be used as the reconstruction value in the finer quantization stage. For values that are quantized to zero in the lower quantization stage, one additional bit indicates if the reconstruction stays at zero or not. If the value in the finer quantization is not zero anymore, the sign bit (S) has to be signaled as well.

In JPEG 2000 the embedded quantization concept is used to devise a bitstream which is scalable in multiple ways. By using wavelet transformation with a tree based coding algorithm (EBCOT), a truncated JPEG 2000 stream represents a valid bitstream at a lower reconstruction quality and/or spatial resolution. In this way, a version of the image with higher compression ratio can be obtained without re-compression. Another application is progressive transmission of an image where a low quality image can be decoded while the transmission of the less significant data is still in progress. For a detailed explanation on

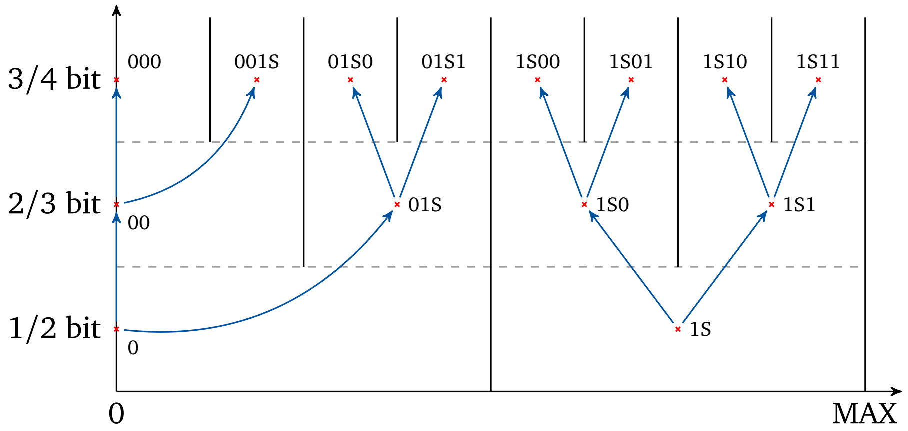


**Figure 4.4** Embedded quantization with dead zone for values ranging from -MAX to MAX. The sign and the magnitude of the values are coded independently. On the x-axis the magnitude values from 0 to MAX are plotted. The sign bit (S) only needs to be transmitted if the reconstruction value is not zero.

JPEG 2000 the reader is referred to [Mar+02; TM02].

The concept of gradual information refinement using embedded quantization can also be seen as a mapping from one quantizer to the next finer quantizer with a lower quantization step size. For the illustrated examples in Figure 4.3 and Figure 4.4, one additional bit is used to map to the next finer quantizer [3]. In both cases, the reconstruction values are placed in the center of each quantization interval and the decision thresholds of all quantizers are aligned. Therefore, there are always exactly two possible values to map to and there is no better reconstruction than one of these two.

When the quantization process of HEVC is considered, these assumptions do no longer hold. While a scalar quantization is used, it is non-uniform. The reconstruction values are spaced equally, but the quantization thresholds are not placed in the center between each pair of reconstruction values. They are rather shifted away from the zero reconstruction value, creating a dead-zone around the zero value. The extent of this dead-zone depends on the prediction type but is smaller than the dead-zone in the embedded quantization approach. For multiple layers, the number of reconstruction values changes in the higher layer depending on the QP settings in the layers and is not necessarily doubled for each additional layer. As a result, the decision thresholds between the layers as well as the quantization intervals of the layers are not aligned. In this case, a different approach has to be taken.

## 4.1.1 Quantization Mapping

Figure 4.5 shows two quantizers as they were detailed in Section 2.3.3. In this example, the slice type is Intra and the QP difference between the quantizers ($\Delta QP$) is -6. With regard to scalable coding, these are the quantizers of two quality layers. For the lower layer, a

[3] For the second example, the separate sign bit also has to be transmitted when mapping from the zero reconstruction value to a non-zero reconstruction value.

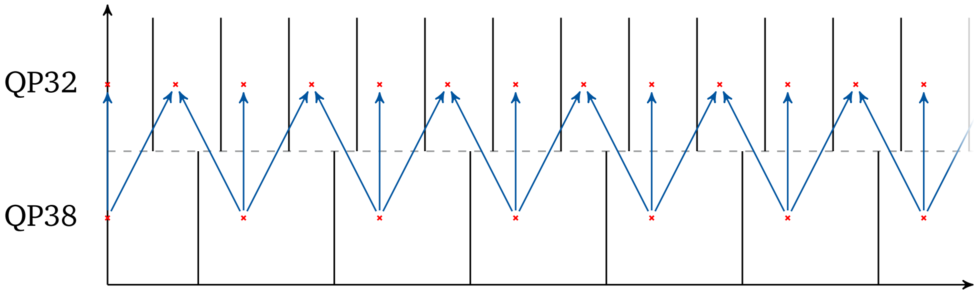
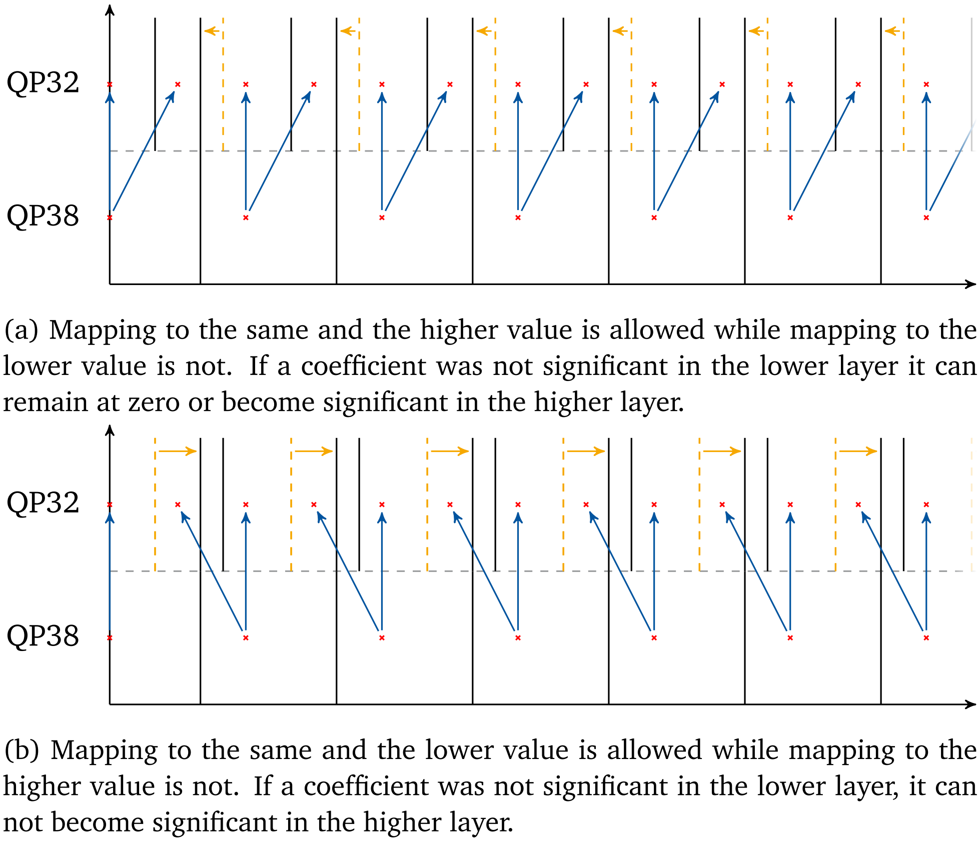


**Figure 4.5** Quantization mapping for the QP values 38 and 32 ($\Delta QP$ = -6) for an intra slice. For every reconstruction value in the lower layer, there are 3 possible values in the finer quantizer.

quantizer with a QP of 38 is used and the higher layer uses a quantizer with a QP of 32. For the case of inter layer prediction in the higher layer, the objective is now to perform direct refinement of the established lower layer reconstruction values. For the QP difference of -6, the quantization step size is halved and the number of reconstruction values is doubled in the finer quantizer. It can be seen in the figure that for every reconstruction value in the lower layer, there are two reconstruction values in the higher layer. One of these two is identical to the lower layer reconstruction value while the other one is placed precisely in between two lower layer reconstruction values.

It was already brought up in Section 2.3.3 that only the position of the reconstruction values are defined in the HEVC standard. The quantization decision is not standardized and the corresponding thresholds can be placed arbitrarily by the encoder. However, if it is assumed that the coefficients exhibit a Laplacian distribution, it is optimal to place the decision thresholds so that the reconstruction value is not centered within the interval. The optimal placement depends on the slope of the coefficient distribution. In the HEVC reference software encoder two placement options are used depending on the slice type [HM-16.7]. The quantization interval is placed such that the reconstruction value is shifted further towards lower values within the interval. For slices using only Intra prediction, a flatter distribution is assumed and the shift towards lower values is smaller compared to slices which also use inter prediction (compare Figure 2.6).

As previously noted, the decision thresholds of the quantizers do not line up. The lower layer quantization interval is not split into two sub-intervals as it is for embedded quantization. If the reconstruction value is 0 in the lower layer (not significant), the situation is the same as for embedded quantization. In the higher layer, the value can stay at 0 or it can be mapped to the first significant (not zero) reconstruction value in the higher layer. If it becomes significant, the sign of the coefficient has to be signaled. However, if the coefficient was quantized to a non-zero level in the lower layer, there are now 3 possible reconstruction values in the higher layer that the level can be mapped to. As one can see, it is possible in this example for a mapping to map to the exact same reconstruction value in the enhancement layer. While a mapping to the identical reconstruction value does not change the reconstruction, it still carries some information about the quantized value and should be encoded. The coding of the information could be adapted to this situation by signaling one bit which indicates if the reconstruction value changes and if it does, a second bit could indicate if a mapping to the higher or lower value is performed. However, it might not be optimal if an enhancement



(a) Mapping to the same and the higher value is allowed while mapping to the lower value is not. If a coefficient was not significant in the lower layer it can remain at zero or become significant in the higher layer.

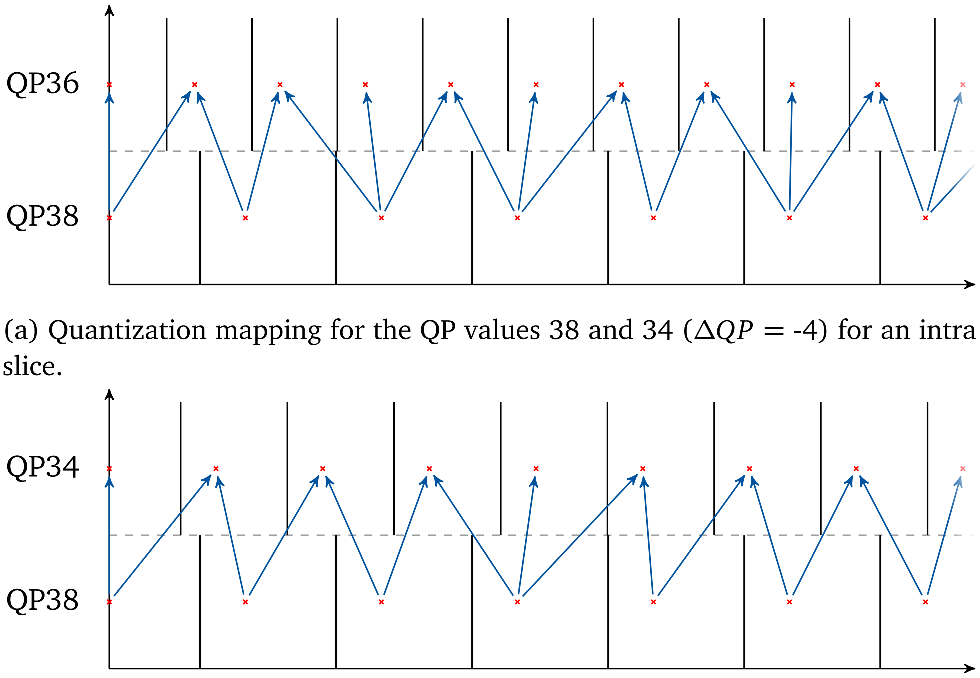
(b) Mapping to the same and the lower value is allowed while mapping to the higher value is not. If a coefficient was not significant in the lower layer, it can not become significant in the higher layer.

**Figure 4.6** Quantization mapping for a $\Delta QP$ of -6 in intra slices when only two of the tree potential mapping options are permitted. The former decision thresholds and their movement is indicated in orange.

layer level is reachable from two different levels in the lower layer. Also from the coding efficiency point of view it is preferable to only allow two mappings per lower layer level so that the mapping information can be coded using a single bit.

Figure 4.6 depicts two options of mapping to the finer quantizer using only a two way mapping. In this case, one bit is sufficient to encode the mapping decision. For the first option (Figure 4.6a), mapping is only performed to the higher two of the three possible enhancement layer reconstruction values while the lower one is ignored. With respect to the higher layer quantizer, effectively the right decision threshold for this value is shifted left to the corresponding lower layer decision line. The original higher layer thresholds are indicated by the dashed orange line and the shift is marked by an orange arrow. For the mapping of the non-significant coefficients in the lower layer nothing changes. If the coefficient was mapped to zero in the lower layer, it can remain zero or become significant in the higher layer. This is signaled by one bit. If the coefficient becomes significant in the higher layer, as for embedded quantization, the sign of the coefficient must be transmitted as well.

For the second option in Figure 4.6b, mapping is only allowed to the lower two of the three respective higher layer reconstruction values. Here, the left decision threshold of the ignored higher layer reconstruction value is shifted right to the value of the lower layer decision line as indicated by the dashed orange line and the orange arrow. It can be seen that the lower



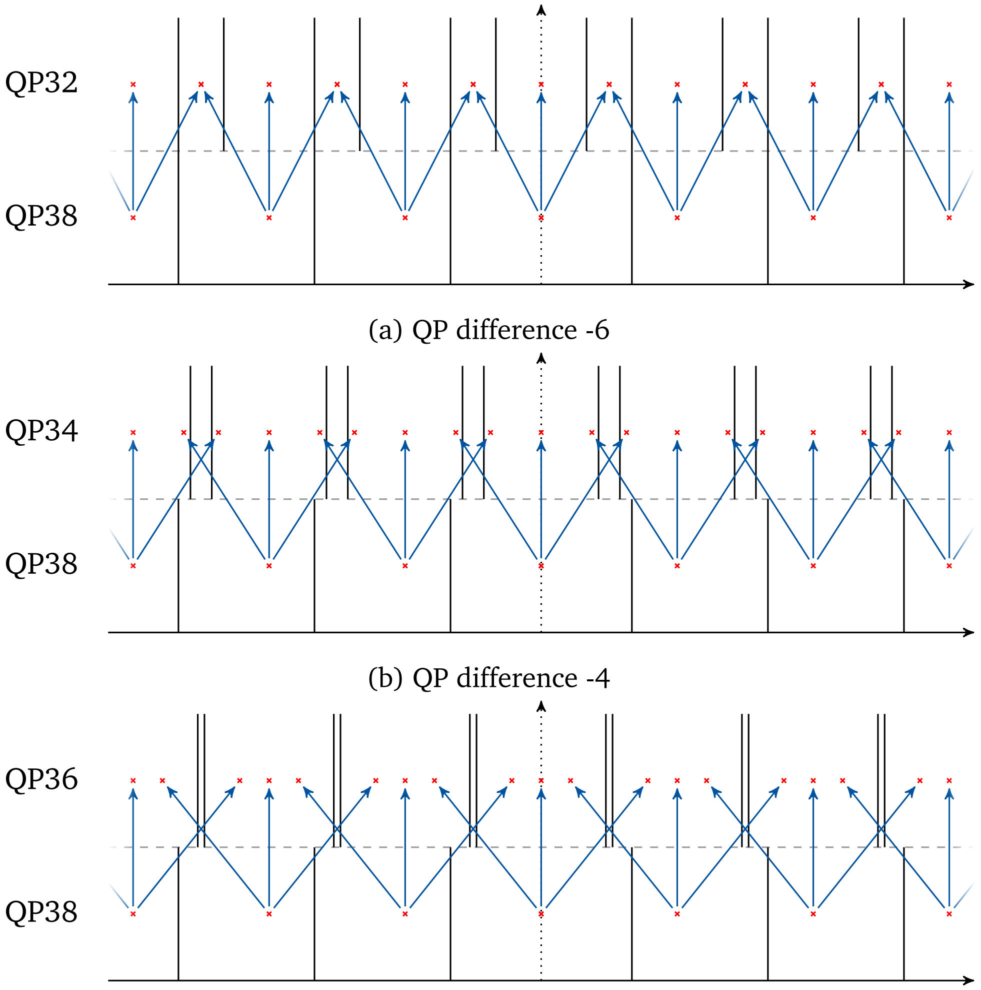
(a) Quantization mapping for the QP values 38 and 34 ($\Delta QP$ = -4) for an intra slice.

(b) Quantization mapping for the QP values 38 and 36 ($\Delta QP$ = -2) for an intra slice.

**Figure 4.7** Quantization mapping for a $\Delta QP$ of -4 and -2. Depending on the lower layer reconstruction value, there are two or three reconstruction values that the value can be mapped to in the higher layer quantizer.

layer quantization interval is split into two intervals in the higher layer. However, the split is performed in such a manner that the reconstruction value of the smaller left interval does not lie within the quantization thresholds of the interval. For a zero level in the lower layer, only a mapping to zero is specified in this example. On the one hand this means that no additional information (new significance and sign) has to be transmitted for non-significant coefficients in the lower layer. On the other hand this also implies that non-significant coefficients in the lower layer can not become significant in the higher layer which in turn can have a negative impact on the performance of the refinement scheme. This might be even worse when more than two layers are used and mapping from these outlying reconstruction values could be particularly inefficient.

For a QP difference of -6, the number of reconstruction values is exactly doubled in the higher layer. For this case, the mapping is fully regular because the location of the higher layer reconstruction values relative to the lower layer values is fixed for all lower layer quantization intervals. However, if the QP difference between the layers is not a multiple of 6, the mapping becomes irregular. Figure 4.7 demonstrates the mapping for a QP delta of -4 (QP38 and QP34) and -2 (QP38 and QP36). It can be observed that, depending on the value in the lower layer, there are two or three possible values that the lower layer level can be mapped to in the higher layer. The arrangement of the reconstruction values in the two layers is not regular and the overlap of the quantization intervals across the layers depends on the lower layer reconstruction value.



(a) QP difference -6

(b) QP difference -4

(c) QP difference -2

**Figure 4.8** Representation of the addition of the higher layer residual signal in SHVC in the transform domain. The intermediate inverse and forward transformation is disregarded in this illustration. The zero reconstruction value is located on the dashed line in the center.

As noted before, the quantization intervals for inter slices are shifted further to the right while the reconstruction values are unchanged. Thus, the mapping between quantizers has very similar but slightly different characteristics. The respective mapping figures for inter slices can be found in Appendix D.1.

Finally, we want to compare the mapping process to the conventional SHVC approach of performing inter layer prediction and coding of another additional residual in the spatial domain. Of course, the additional residual signal in the higher layer is not added in the transform domain but inverse transformation is applied to the residual signal of both layers before they are added in the spatial domain. For a perfectly orthonormal transformation, the resulting signal is independent of the domain that the addition is performed in. Because the forward and inverse transformation in HEVC are only discrete approximations to the DCT and additional clipping may be applied, this does not perfectly apply to the transformation in HEVC. However, if we disregard this for now we can also represent the addition of the higher layer residual signal in the transform domain. Figure 4.8 depicts this for different QP differences between two layers. From each lower layer level, three arrows point to the higher layer reconstruction values corresponding to a higher layer level of -1, 0 and 1. While there is no specific restriction of the higher layer levels to only these three values, larger values are not plotted here for clarity. When the coefficient distribution is taken into account, it can be seen that the coding scheme is not optimal. Already for the QP difference of -6, it can be seen that some of the plotted mapping options which can be signaled, can in practice never occur (e.g. the lower layer reconstruction index 1 can map to the higher layer index 1). For the smaller QP differences the situation becomes even more pronounced. Also the coding of the information is flawed from this point of view: For every higher layer coefficient that is significant (which is not mapping to the exact same value in the higher layer), a sign bit is coded followed by the remaining level. However, because of the coefficient distribution, the bypass coded sign bit should reveal a clear bias towards one direction depending on the sign of the reconstruction value in the lower layer. In addition, a remaining level greater than zero is highly implausible. While the entropy coding engine can learn that this greater 1 bit is never set and can therefore encode it very efficiently, it would be even better to skip it altogether. It should be further noted that this analysis only applies under the assumption that inter layer prediction is used and the transform size in the two layers is identical. If a different prediction is used in the higher layer, this mapping idea does not apply and a mapping to more than the three values in the higher layer is possible. While this is just a theoretical consideration, it should highlight some of the flaws of conventional coding using an additional residual signal as it is performed in SHVC.

A similar mapping approach between scalar quantizers is also described in [EF04] where it is referred to as conditional hierarchical mapping. While the underlying idea is analogous to the presented refinement mapping, the concept is further expanded in the following. In the next section, it is investigated how the mapping from one to the next layer can be optimized. With an assumption on the coefficient distribution, the probability of each mapping can be estimated. From this, an approximation of the expected error and entropy can be performed which then allows for a rate-distortion optimization of the mapping. If it yields an improvement in a rate-distortion sense, the optimization process may even discard some of the mappings which is also contrary to the conditional hierarchical mapping concept.

### 4.1.2 Probability and Error Analysis

First, we establish an estimation on the coefficient distribution. In the literature, the coefficients are commonly assumed to exhibit a Laplacian distribution with the mean value at zero [RG83; LG00]. The corresponding PDF is given by

$$p(x) = \frac{1}{\sqrt{2\sigma^2}} e^{-\frac{\sqrt{2}|x|}{\sigma}}, x \in (-\infty, \infty). \tag{4.1}$$

The probability of positive and negative values is equal in this case so we can also model the distribution using a sign and the amplitude of the value. The sign has equal probability and

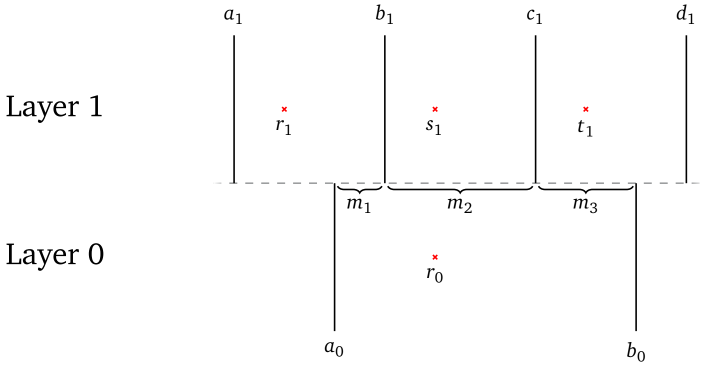


**Figure 4.9** Exemplary mapping for the lower layer quantization interval from $a_0$ to $b_0$ with the reconstruction value at $r_0$. In the higher layer, the decision lines are at $a_1$, $b_1$, $c_1$ and $d_1$. The three corresponding reconstruction values that the value $r_0$ can be mapped to are $r_1$, $s_1$ and $t_1$.

the amplitude value can be modeled by an exponential distribution with the PDF

$$p(x) = \alpha e^{-\alpha x}, x \in [0, \infty), \alpha > 0. \tag{4.2}$$

Using the probability function and the mapping layout, we can perform an estimate on the probabilities for the potential mappings. Figure 4.9 shows an exemplary mapping for one lower layer quantization interval. In the lower layer, all values from $a_0$ to $b_0$ are quantized to the same level and will be reconstructed by the value $r_0$. In the higher layer, there are three quantization intervals, $a_1$ to $b_1$, $b_1$ to $c_1$ and $c_1$ to $d_1$, with the corresponding reconstruction values $r_1$, $s_1$ and $t_1$. With the assumption of an exponential coefficient distribution, the probability of the coefficient being within the interval $[a_0, b_0]$ can be calculated by

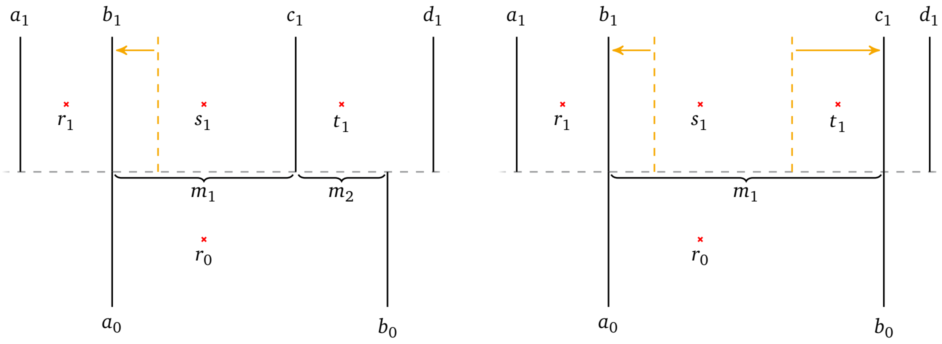
$$P(x \in [a_0, b_0]) = \int_{a_0}^{b_0} \alpha e^{-\alpha x} \mathrm{d}x = e^{-\alpha a_0} - e^{-\alpha b_0}. \tag{4.3}$$

If it is now assumed that the value lies within a specific interval $[a_0, b_0]$, the probability distribution of the values within this interval is also exponential. The PDF of the coefficient value, on the condition that the value lies within this interval, is

$$p(x | x \in [a_0, b_0]) = \frac{\alpha e^{-\alpha x}}{e^{-\alpha a_0} - e^{-\alpha b_0}}, x \in [a_0, b_0]. \tag{4.4}$$

From this, it is possible to calculate the probability of the coefficient value being in one of the intervals $m_1$, $m_2$ and $m_3$ provided that the value lies within the lower layer interval $[a_0, b_0]$. For the interval $m_3$, the probability is calculated by

$$P(x \in m_3 | x \in [a_0, b_0]) = \frac{e^{-\alpha c_1} - e^{-\alpha b_0}}{e^{-\alpha a_0} - e^{-\alpha b_0}}. \tag{4.5}$$



(a) Mapping from $r_0$ to $r_1$ is disallowed.

(b) Single mapping from $r_0$ to $s_1$ only.

**Figure 4.10** Mapping from $r_0$ is only enabled to some of the higher layer reconstruction values. The original quantization threshold and where it moves is indicated the dashed orange line and the orange arrow, respectively.

For the other intervals, the calculation is analog using the respective interval boundaries. From these probabilities, we can in turn calculate the entropy $H$ of the mapping in the interval. This can then be interpreted as the amount of information (the rate) that is required to code the mapping. [4] For the present example, the entropy in bits is calculated from the probabilities by

$$H = \sum_{i=1}^{3} H(m_i) = -\sum_{i=1}^{3} P(x \in m_i | x \in [a_0, b_0]) \, log_2 \, (P(x \in m_i | x \in [a_0, b_0])) \, . \tag{4.6}$$

Finally, we can also use the probability distribution from 4.4, the quantization interval boundaries and the reconstruction values to calculate the expected value of the mean square error (MSE) for each of the mapping options. For example, the MSE for the interval $m_3$ is calculated by

$$\mathrm{MSE}(m_3) = \int_{c_1}^{b_0} \frac{(t_1 - x)^2}{e^{-\alpha a_0} - e^{-\alpha b_0}} \alpha e^{-\alpha x} \mathrm{d}x \, . \tag{4.7}$$

The expected value of the MSE for the other intervals is calculated by inserting the corresponding interval boundaries and reconstruction values. By adding these values for all available mappings, the MSE for the complete mapping is obtained. This expected MSE can then be interpreted as a measure for the distortion of the particular mapping.

In the example of Figure 4.9, the lower layer interval overlaps with 3 quantization intervals in the higher layer. Of course, the presented estimation of the rate and distortion can also be performed for cases where there are more or less corresponding quantization intervals in the higher layer. Furthermore, the values can also be calculated for cases where only some of the possible mappings are allowed. Figure 4.10a demonstrates a potential mapping where the mapping from $r_0$ in the lower layer to $r_1$ in the higher layer is excluded.

[4] Please note that this is only the entropy contribution of this one interval. For the absolute entropy, all lower layer values and reconstructions would have to be considered.

Effectively, the former decision line (dashed orange line) is moved left to the value of the lower layer decision line $a_0$. The values that were originally reconstructed by $r_1$ are now appended to the quantization interval that corresponds to the reconstruction value $s_1$. In this case, the expected bitrate and error is calculated from the two intervals $m_1$ and $m_2$ and the corresponding reconstruction values $s_1$ and $t_1$. Even if only one mapping is considered, the expected distortion can be calculated. Such a scenario is depicted in Figure 4.10b. All values within the interval $[b_1, c_1]$ (which in this case is equivalent to the lower layer interval $[a_0, b_0]$) are reconstructed by the value $s_1$ in the higher layer. In this special case with only one possible mapping, no rate has to be spent to encode this information. [5]

In the following section, we use the obtained error and probability values to optimize the mapping. The whole process is related to similar optimization strategies like the Lloyd max quantizer. While for the Lloyd max algorithm, the quantization boundaries and reconstruction values are adapted to a given distribution of values, we use the distribution to optimize the mapping while the reconstruction values are fixed.

### 4.1.3 Mapping Optimization

It was already established, that the coefficient distribution can be modeled using an exponential function (see Equation 4.2). However, in order to calculate the values for the distortion and rate of a certain mapping, an approximation of the $\alpha$ value that models the slope of the distribution is necessary. For the following calculations, we obtain the $\alpha$ from the lower layer quantization. For a given QP value in the lower layer, the position of the decision thresholds and the reconstruction values are fixed and can be calculated (see Section 2.3.3). Let's consider one quantization interval in the lower layer with the decision thresholds at $a_0$ and $b_0$ and the corresponding reconstruction value $r_0$ (compare Figure 4.9). The values within this interval are exponentially distributed as defined in Equation 4.4. For this exponential distribution, the optimum reconstruction value (which minimizes the mean squared error) must be placed at the mean value of the distribution within this interval. The mean value for the assumed distribution within the interval $[a_0, b_0]$ can be calculated by

$$m(\alpha, a_0, b_0) = \int\limits_{a_0}^{b_0} x\alpha e^{-\alpha x} \mathrm{d}x = \ldots = \frac{1}{\alpha} + \frac{b_0 - a_0}{1 - e^{\alpha(b_0 - a_0)}} + a_0. \tag{4.8}$$

If we now assume that the reconstruction value within the interval is placed at this optimum position, we can apply the reverse process and compute the $\alpha$ value from the known values $a_0$, $b_0$ and $r_0$. Because of the properties of the exponential distribution, the relative position of the expected value within the interval is independent of the position of the interval. Therefore, the computation of the $\alpha$ value is independent of the fact which of the quantization intervals in the lower layer is selected.

[5] While it is obvious that no rate is needed if there is only one possible mapping target in the higher layer, it can also be derived from Equation 4.5 and 4.6. Because the intervals are identical, $p(x \in m_1 | x \in [a_0, b_0])$ is 1. Hence, the entropy $H$ becomes $H(m_1) = 0$.

With the estimation of the distortion and the rate, it is now possible to apply Lagrangian multipliers in order to obtain a cost value for every possible mapping to the higher layer [Ohm15]. This cost value is then used to compare the mappings and look for the optimum mapping option. The distortion of a certain mapping $M$ shall be denoted by $D(M)$ and the bitrate by $R(M)$. The cost function $C(M)$ using the Lagrangian multiplier $\lambda$ is then

$$C(M) = D(M) + \lambda R(M). \tag{4.9}$$

We now assemble a list of all possible mappings from the lower layer interval to the higher layer intervals. This list contains the option of mapping to all corresponding higher layer intervals as well as options where certain mappings are disallowed. For the example in Figure 4.9, this list contains the following mapping options from $r_0$:

- to all three corresponding higher layer values $r_1$, $s_1$ and $t_1$ using more than one bit
- to two of the higher layer values using one bit (to $r_1$ and $s_1$ or $s_1$ and $t_1$)
- to only one of the higher layer values $r_1$, $s_1$ or $t_1$ using no additional bits

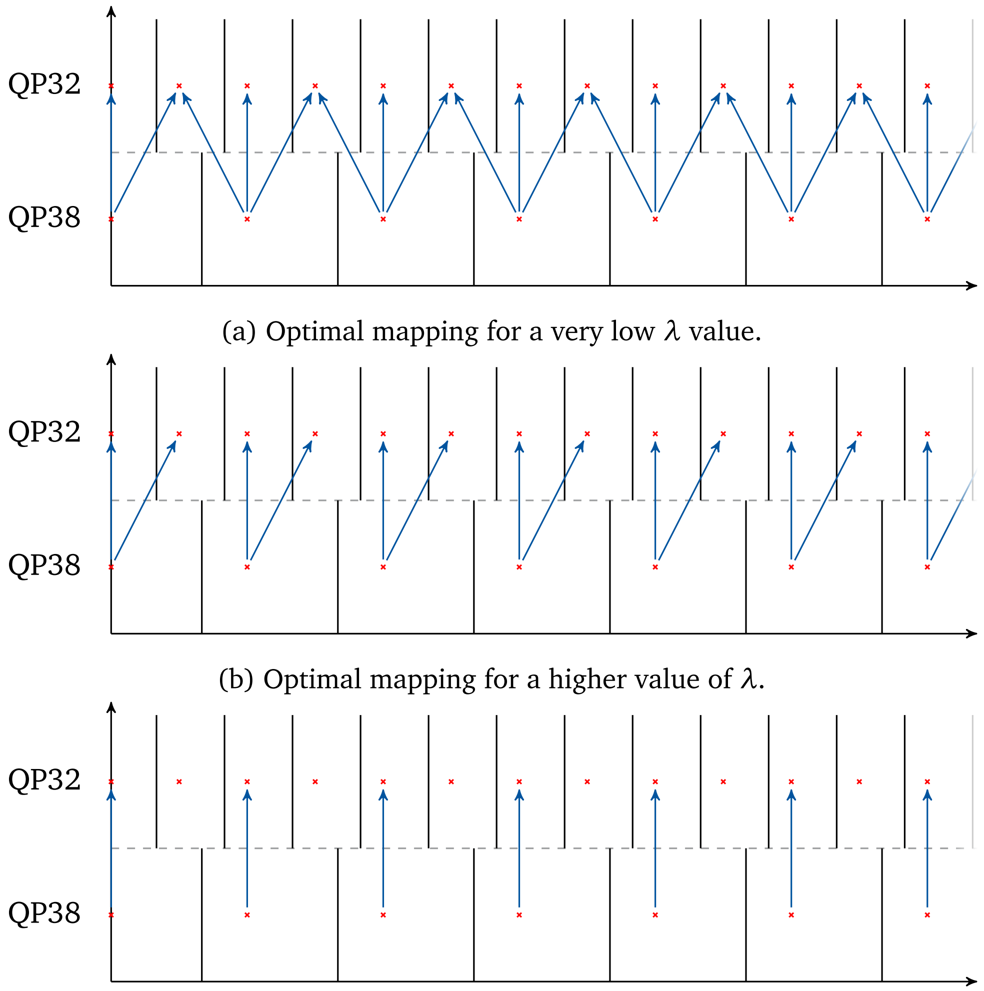
For all of these possible mappings, we are now looking for the mapping that minimizes the cost function:

$$M_{opt} = \arg\min_{M} \{D(M) + \lambda R(M)\} \tag{4.10}$$

The Lagrangian multiplier $\lambda$ performs a trade-off between the expected rate and the distortion. For a low $\lambda$, the influence of the distortion on the cost is high while the impact of the bitrate is minor. For the extreme case of $\lambda = 0$, the bitrate is completely disregarded for the optimization. In the opposite case (a very high value of $\lambda$), the rate is the dominating parameter for the cost term. If $\lambda$ is approaching infinity, the expected distortion is completely ignored.

### Two layers

In the following, we will focus on the delta QP values -6, -4 and -2 as a representative set of QP deltas and on a scenario with two layers. For a QP difference of -6, the situation is relatively simple because the mapping is regular and with the assumed model, the probability values for mapping to the higher layer reconstruction values do not depend on the lower layer level. For intra prediction and a low value of $\lambda$, the optimal mapping uses all 3 available mappings (as depicted in Figure 4.11a). When the Lagrangian multiplier is increased, the mapping to the left (lower) higher layer level is discarded and only mapping to the same and the next higher reconstruction value in the higher layer is permitted (see Figure 4.11b). Increasing the $\lambda$ value even further will finally result in a mapping where only the same value in the higher layer can be mapped to (Figure 4.11c). In case of inter prediction, the situation changes slightly: Even for very low values of $\lambda$ the mapping to all 3 corresponding higher layer values is never optimum. Illustrations for all optimum mappings for the QP difference of -6 and inter prediction can be found in Figure D.2.



(a) Optimal mapping for a very low $\lambda$ value.

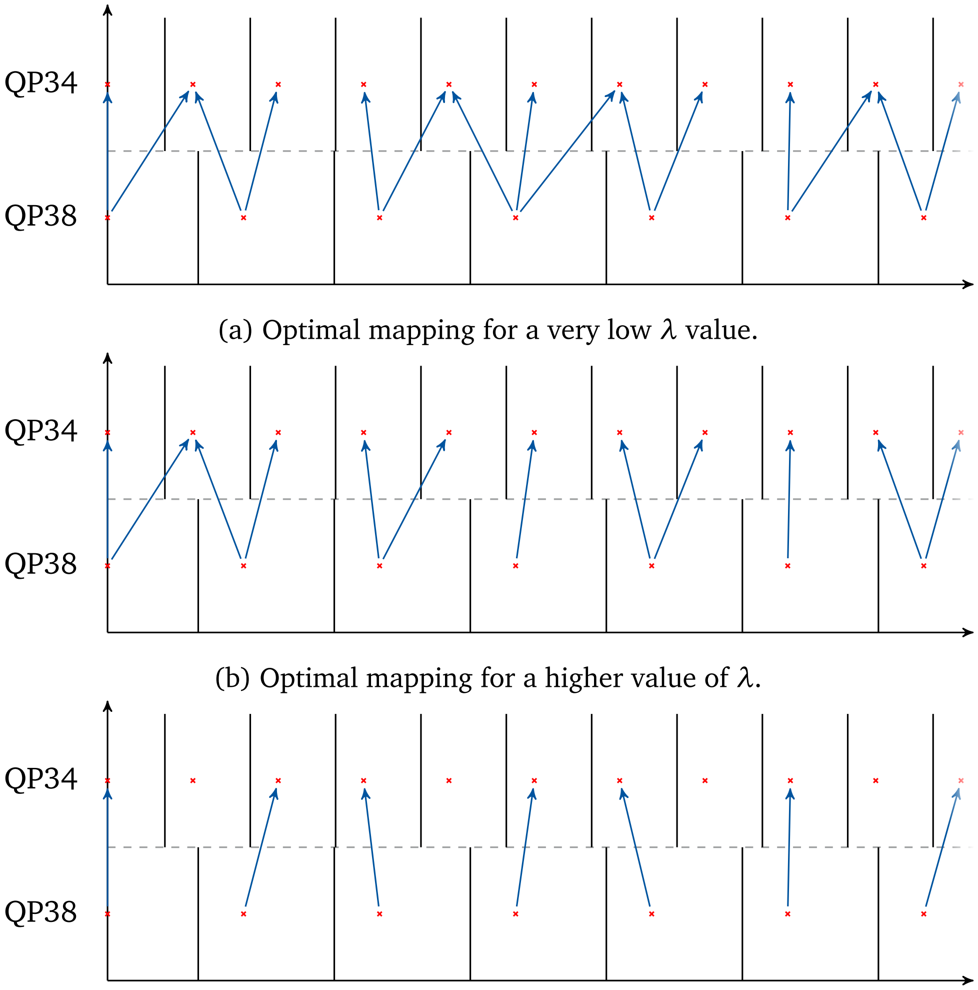
(b) Optimal mapping for a higher value of $\lambda$.

(c) Optimal mapping for a very high value of $\lambda$.

**Figure 4.11** Optimal mappings for different values of $\lambda$, a QP delta of -6 and intra prediction.

For the other two QP difference values -4 and -2, the mapping is not regular and the optimal mapping arrangement depends on the lower layer level. Figure 4.12 demonstrates the optimized mapping for a QP difference of -4 and intra prediction. For a low value of the Lagrangian multiplier (4.12a), the levels are mapped to either two or three reconstruction values in the higher layer, depending on the lower layer level. Some of the higher layer levels are accessible from several levels in the lower layer. As the $\lambda$ is increased, more and more of the mappings are eliminated. At a certain point (4.12b), all higher layer levels are reachable by exactly one path from the lower layer levels. The only exception is the first significant coefficient in the higher layer, which is also accessible from the zero level in the lower layer. Finally, for a very high $\lambda$, the mapping transforms into a pure one to one mapping in which no bits need to be coded and no additional information is added in the higher layer. All levels are mapped to precisely one level in the higher layer. For inter prediction, the corresponding results are depicted in Figure D.3.

For a QP difference of -2, similar observations can be made. For a low value of $\lambda$ (4.13a) the majority of higher layer levels are accessible from more than one level in the lower



(a) Optimal mapping for a very low $\lambda$ value.

(b) Optimal mapping for a higher value of $\lambda$.

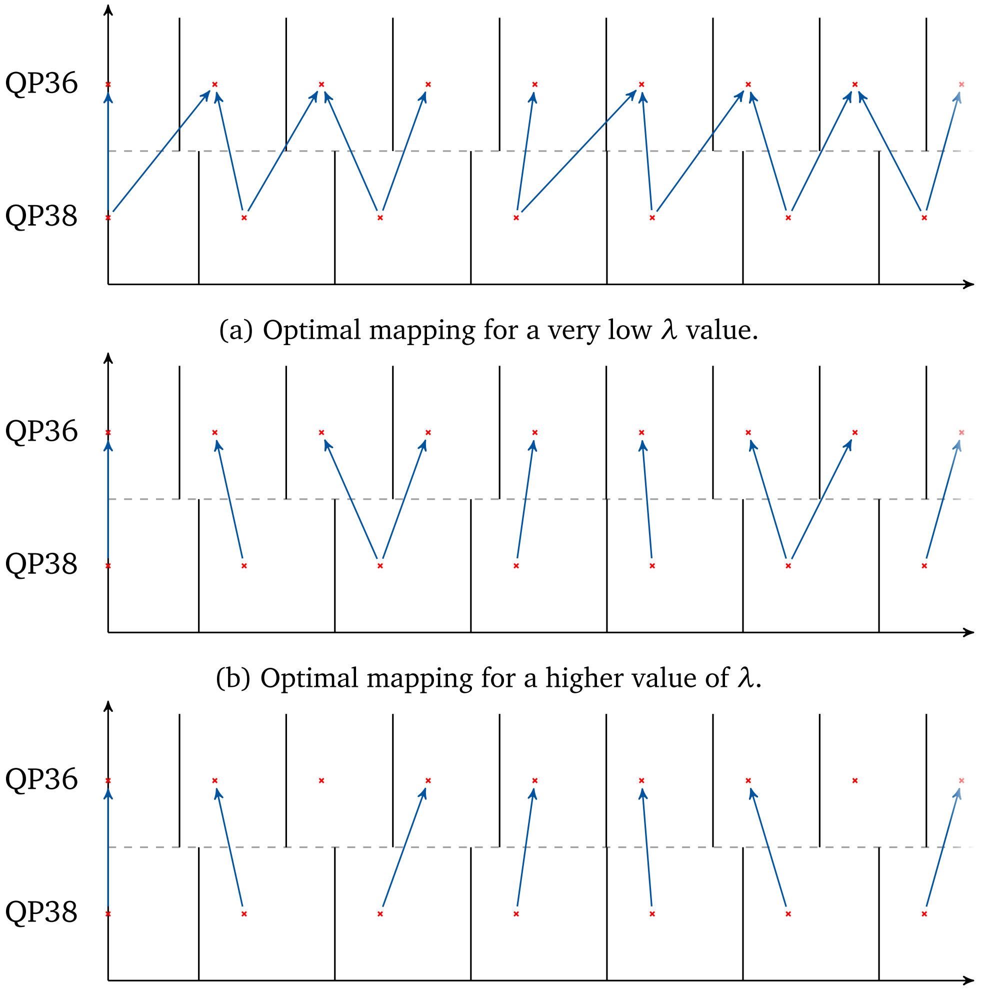
(c) Optimal mapping for a further increased value of $\lambda$

**Figure 4.12** The optimal mapping for different values of $\lambda$, a QP delta of -4 and intra prediction.

layer. For the second example (4.13b), the Lagrangian multiplier was chosen so that all higher layer reconstruction values are accessible from exactly one lower layer value. It can further be observed in this case that if a coefficient is not significant in the lower layer it is optimal to always map it to the same zero level in the higher layer. Finally, for a very high $\lambda$ value (4.13c), the mapping again degenerates to a pure one to one mapping with no added information in the higher layer. The corresponding optimization results for inter prediction can be found in D.4.

### More than two layers

In the following, we will extend the optimization scheme to cases with more than two layers. If there are more than two layers, the optimization can have different targets. First, the optimization for two layers can simply be applied multiple times for each step between two layers. In this scenario, each mapping is optimized individually without knowledge of the complete mapping scheme. In the second scenario, the mapping from the lowest to the



(a) Optimal mapping for a very low $\lambda$ value.

(b) Optimal mapping for a higher value of $\lambda$.

(c) Optimal mapping for a further increased value of $\lambda$

**Figure 4.13** The optimal mapping for different values of $\lambda$, a QP delta of -2 and intra prediction.

highest layer is optimized first and in a second step the mapping through all intermediate layers is considered.

In figure 4.14, an example for the optimization per layer step is shown. In this example, three layers are used and the QP delta between the layers is -4 and -4. For the optimization there now is a $\lambda$ value for each mapping between each two layers which can be selected independently. Here, the $\lambda$ values were chosen so that each coefficient in the higher layer is reachable by one path from the lower layer. For each layer, mapping is possible from each reconstruction value to one or two reconstruction values in the higher layer. Except for the lower three coefficients, each coefficient in the highest layer is reachable from exactly one reconstruction value in the lowest layer. It can already be seen that some paths from the lowest to the highest layer might not be optimal because optimization was performed layer per layer. E.g. the highlighted paths from the lowest to the highest layer are highly unlikely if the coefficient distribution is taken into account. For more results using differen values of $\lambda$ as well as more different QP deltas please see Appendix D.2. Since optimization is performed on a per layer base, it is trivial to extend it to an arbitrary number of layers.

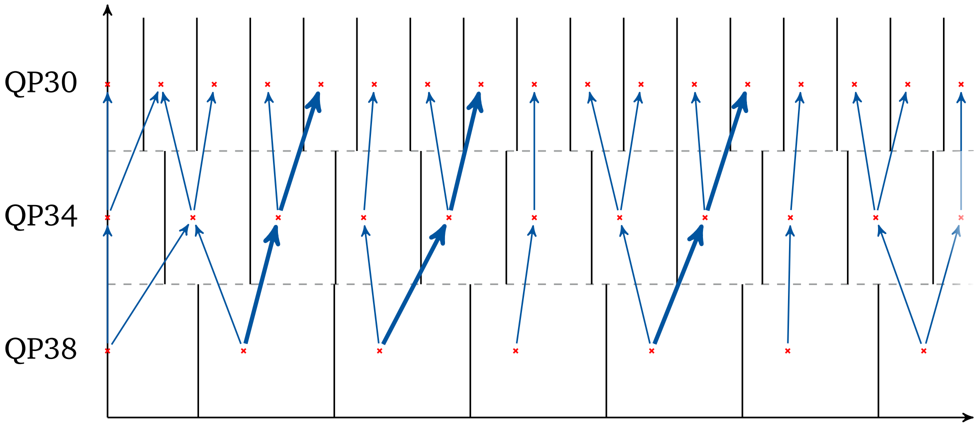


**Figure 4.14** Per layer optimized mapping for a delta QP value of -4 between the layers.

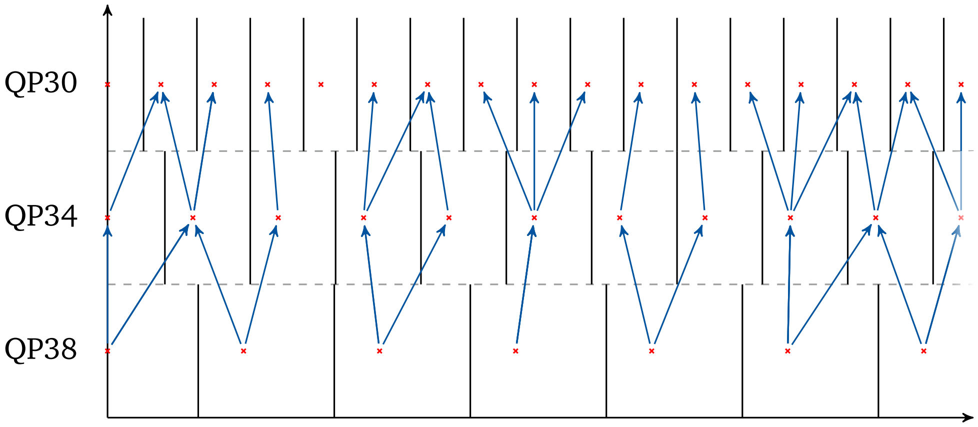
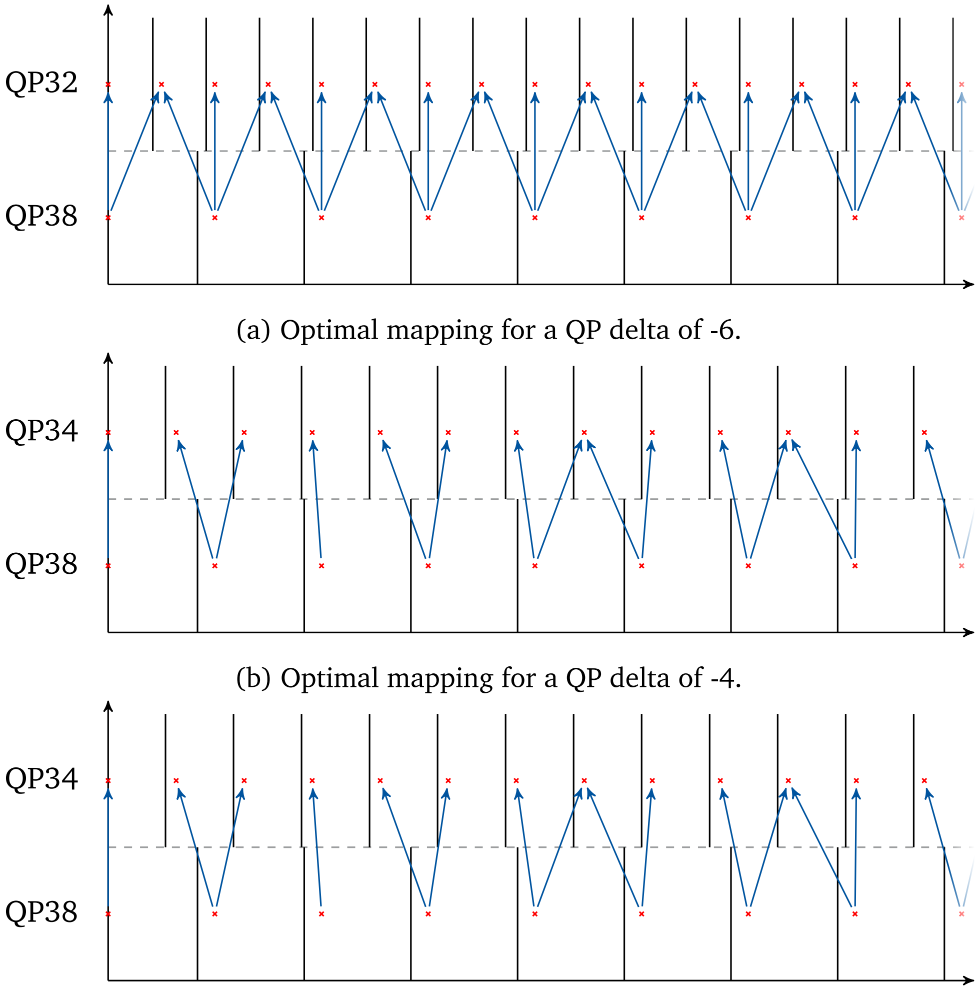


**Figure 4.15** Overall optimized mapping for a delta QP value of -4 between the layers.

In the alternative approach. the optimization for the mapping from the lowest to the highest layer is performed first. For this first step, the same optimization technique is used which was also used for only two layers. This also means that for the rate calculation the mapping through intermediate layers is disregarded for this optimization step. In the second step, for each mapping from the lowest layer to the highest layer the mapping through the intermediate layers is optimized. For each possible path, the rate for the path as well as the distortin is calculated and used for the optimization. With this approach there is one $\lambda$ value for the optimization from the lowest to the highest layer as well as one value for the mapping through each intermediate layer.

In figure 4.15, an example is shown for the same three layers with a QP delta of -4 between the layers. It can be seen that the optimization result differs significantly from the per layer optimized scheme in Figure 4.14. Mapping is possible from each reconstruction values in the lowest layer to two or three reconstruction values in the highest layer. From one layer to the next, there are mappings to one, two as well as three coefficients. Also, some coefficients in the highest layer are not reachable at all. The highlightes paths from Figure 4.14 are all not present using this kind of optimization. More results are available in Appendix D.2. While it is not as straightforward, this scheme can also be extended to any number of layers.



(a) Optimal mapping for a QP delta of -6.

(b) Optimal mapping for a QP delta of -4.

(c) Optimal mapping for a QP delta of -2.

**Figure 4.16** Optimization from the higher layer based on the mean square error for different QP deltas and intra prediction.

### Optimization from higher layers

The optimization schemes which were presented so far all perform optimization from the lower to the higher layers. In principle, an optimization could also be performed starting from the quantization of the higher layer deciding which of the possible lower layer reconstruction values to map from. One problem which arises is with the calculation of the rate. Since in the higher layer we can not say which of the mappings from the lower layer are actually used, we can not perform an estimation of the associated rate for the mappings. However, using the probabilities we can perform an optimization using the expected mean square error when mapping from all possible lower layer reconstruction values is performed.

In Figure 4.16, an example for different QP delta values is shown. Since the optimization is performed from the higher layer reconstruction values, only the mean square error is considered for the optimization in this case. For a QP difference of -6, the optimization results in a full layout where all possible mappings are allowed. This is due to placement of the reconstruction values. For a QP delta value of -6, one additional reconstruction value is placed exactly in between each pair or reconstruction values in the lower layer. Since the distance to these two neighboring coefficients is maximized, a mapping from both of them minimizes the mean square error. For the delta QP values -4 and -2, mapping from the lower layer is performed to one or two reconstruction values in the higher layer. The coefficients in the higher layer are reachable from one or two reconstruction values in the lower layer. One of the issues with the optimization from the higher layer is the performance drop in the lower layer. While this drop might be acceptable in the case of refinement coding in favor of the performance in the higher layer, it should be avoided if refinement coding is not chosen in the higher layer. However, in order to adaptively choose the layer of the initial quantization a multilayer encoder is required which optimizes all layers simultaneously.

Ultimately, several things should be considered in order to select a suitable refinement mapping. For the optimization schemes from the lower to the higher layers the Lagrange multiplier plays a vital role. It was shown that the mapping degenerates to a pure one to one mapping if a very high value is chosen. Except for the QP difference of -6, the reconstruction values in the higher layer are only shifted while no additional information is added. When the distribution of the coefficients is considered, this means that the expected distortion in the higher layer increases. Essentially, this kind of mapping is counterproductive and can only result in a lower RD performance compared to the lower layer. In SHVC, this situation can not arise because there is always the option to leave a value unchanged. In a practical encoder implementation such a mapping would only very rarely be chosen and the encoder would rather opt to not encode a refinement at all. In the other extreme (a very low Lagrange multiplier), it can be suboptimal if a reconstruction value in the higher layer is reachable from multiple values in the lower layer. Also from a coding poitn of view, it is not ideal if multiple bits are needed in order to encode the refinement information. Therefore, the Lagrange multiplier in this thesis was chose in a way so that one to one mappings as well as three way mappings are avoided. Since the optimization from the higher layer only considers the mean square error, no tradeoff between rate and distortion is possible.

At last, the limits of the presented model shall be detailed. It should be noted that the calculated probability values as well as the subsequent mapping optimization are both based on the hypothesis that the coefficient values are exponentially distributed with a certain slope $\alpha$. While this is a commonly used assumption ([RG83; LG00]), it was also shown that this premise might not hold for all coefficients. For example in [AK04] it was shown that the distribution of the lowest frequency coefficients is closer to a Cauchy than to an exponential distribution. This also means that some of the presented optimal mapping layouts may not be actually optimal. Nevertheless, a detailed mathematical analysis of the real distribution of the transform coefficients is outside of the scope of this work. But also without definite knowledge about the coefficient distribution, the probabilities of the mappings could be approximated by counting at the encoder or decoder side. Another assumption for the optimization is, that no rate-distortion optimal quantization (RDOQ) or sign data hiding is performed for the lower layer. However, this does not prohibit the use of RDOQ or sign data hiding in general. In a more advanced implementation, the optimization may be combined with rate-distortion optimization at the encoder side. Nevertheless, such an advanced encoder that optimizes over multiple layers is out of the scope of this thesis. In any case, the presented model gives some valuable insight into the mapping process and allows for a first optimization of the mapping.

## 4.2 Refinement Coding

For the refinement information, a coding scheme was established, that allows for efficient coding of the refined residual signal. Essentially, the coefficient scan of the lower layer coefficients is performed again, where there are two classes of information:

**New significance:** For all coefficients that were not significant in the lower layer, a new significance has to be coded. This is performed similar to signaling of significant coefficients in HEVC, where first the x-y position of the last significant coefficient is coded followed by a scan of the coefficients starting at this position. Here, however, the location of the last significant coefficient in the lower layer is known so that a differencing value to that position can be coded. All the non-significant positions are rescanned and a new significance flag is coded. If the coefficient gets significant in the higher layer, a sign flag must be signaled as well.

**Refinement:** For all coefficients that were significant in the lower layer, the coding depends on the mapping. In case of two possible reconstruction values in the higher layer, the refinement information is coded in the form of a single flag. If only one corresponding reconstruction value is present, nothing needs to be signaled. For the evaluation, the mapping was optimized so that a mapping to 3 or more coefficients in the higher layer does not occur.

In the following, it is detailed how the update to the last significant position, the newly significant coefficients and the refinement information is coded.

### 4.2.1 Last Significant Position

In Section 2.3.5 it was detailed how the coefficient coding is performed in HEVC. The process is split into several steps. In the first step, before a scan is performed, the position of the last significant coefficient in scan order is signaled using the x- and y coordinates of this position within the transform unit. For the refinement of the existing signal, this 2D position is already known. As for the lower layer, it can be expected that the signal energy is concentrated in the lower frequency coefficients. However, because we are now using a finer quantization on the same transform coefficients, the last significant position in the higher layer is likely located at a higher frequency position. Therefore, we take this premise into account and use a similar approach by transmitting an update to the x- and y-coordinates first.

For the example in Figure 4.17,the last significant coefficient in the lower layer (indicated by the cross) is located at $x = 6$ and $y = 5$. The shaded coefficients were subsequently scanned in the lower layer and all coefficients that were significant must lie within this area; all other coefficients were not significant in the lower layer. The new last significant position in the higher layer is located at x=10 and y=1. Coding of the position update with the knowledge about the scanned positions in the lower layer is now performed in two steps. The space

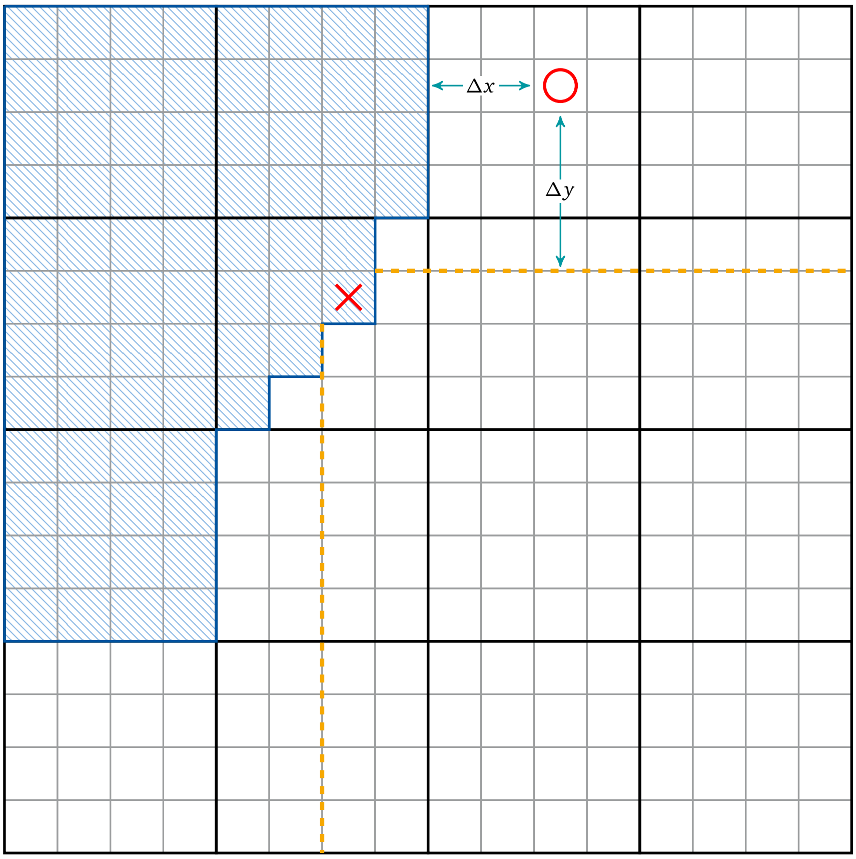


**Figure 4.17** Example for coding of the last significant position in a 16x16 pixel transform. In the lower layer, the last significant position is located at $x = 6$ and $y = 5$ (red cross). The highlighted coefficients were consequently scanned in the lower layer. The last significant position in the higher layer is located at $x = 10$ and $y = 1$ (red circle).

of possible positions is split into three regions as indicated by the two dashed orange lines. Firstly, it is transmitted which of these three regions is occupied by the updated position. For this, the first flag indicates if it is the lower right region. If not, a second flag indicates which of the remaining two areas is the correct one. Coding of these flags is omitted if the new position can not lie within one or both of these regions. In this example, the first flag is false and the second flag will indicate the upper right partition. In the next step, the delta values for the $x$ and $y$ component are transmitted. These delta values are calculated relative to the coefficients that were scanned in the lower layer and to the lines that divide the space of possible new values. In this example, these delta values are $\Delta x = 2$ and $\Delta y = 3$. The value and the possible value range of one delta component depends on the value of the other component. Except for the upper right region where the $y$ value is coded first, the $x$ value is coded first. For the example, if the $\Delta y$ value is 0 (the last significant position in the higher layer moves down by three coefficients to $x = 10$ and $y = 4$), the $\Delta x$ value is 3. The delta values are coded the same way the original position of the last significant coefficient is coded using a prefix and a suffix along with context based entropy coding (see Section 2.3.5). There are some special cases where there is a slight deviation from this coding scheme (e.g. when the x or y component of the last significant position in the lower layer is zero). While there is no dedicated flag that indicates if the position changes at all, signaling of an unmodified last position is still quite efficient. The first flag will signal the lower right region and the two remaining zero valued deltas are coded using one false flag for each delta.

### 4.2.2 Newly Significant Coefficients

As in conventional HEVC, the transform coefficients are arranged in sub-blocks of 4 by 4 coefficients. Starting with the sub-block that contains the updated last significant position, the sub-blocks are scanned in a specific order (scan order). For each sub-block, a coded block flag (CBF) indicates if any of the coefficients that were not significant in the lower layer become significant in the higher layer. This approach allows for a very efficient signaling of sub-blocks with no new significant coefficients. As in HEVC, coding of the CBF is omitted for the sub-block that contains the new last significant coefficient and the last sub-block in scanning order (the one that contains the DC coefficient). The CBFs are either coded using contexts depending on the value of the neighboring CBFs as in HEVC or using a fixed probability which is also determined depending on the neighboring CBFs.

If the CBF of a sub-block is set, the 16 coefficients within the sub-block are scanned. For every coefficient that was not significant in the lower layer, a new significance flag is coded in this scan. Two options for coding of this information are presented here: For the first option, coding is performed just like for the significance flags in the lower layer using contexts depending on the significance of the surrounding coefficients (this was detailed in Section 2.3.5). However, the new significance of the higher layer coefficients does not carry the same information as in the lower layer. In the lower layer, the significance flag indicates if the coefficient is zero or any value higher than zero. In the higher layer, the flag of new significance only carries the information if the coefficient is zero or one. So the new significance can also be interpreted as another refinement flag. For the second option, the new significance flag is coded using a certain fixed probability value that is only dependent on the QP. Of course, as for the significance flags in the lower layer, an additional sign bit is coded using equal probability coding if the coefficient becomes significant in the higher layer.

### 4.2.3 Refinement

As it was already explained, it depends on the chosen mapping how many bits have to be coded for the refinement of coefficients that already were significant in the lower layer. If mapping to more than two coefficients in the higher layer is possible, more than one bit needs to be coded. If there is only one possible mapping, no information needs to be transmitted. The implemented mappings come from the optimization approach explained in Section 4.1.3. In the following, the optimization parameter $\lambda$ was chosen so that mapping is always performed to one or two of the coefficients in the higher layer so that at most one refinement bit has to be coded. The two additional objectives were to avoid cases where one higher layer coefficient is reachable from multiple coefficients in the lower layer and to prevent coefficients in the higher layer that are not reachable at all.

As for the newly significant coefficients, the actual coding of the binary flags is performed using two different approaches: For the first method, context based coding with multiple

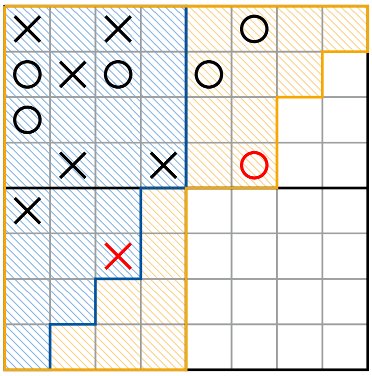


**Figure 4.18** Refinement coding example for an 8x8 transformation. The lower layer coefficients are marked with a cross and the coefficients that become significant in the higher layer with a circle. The respective last significant positions are indicated in red.

contexts is used. The selection of the context depends on the level of the coefficient in the lower layer. For the second option, the flag is coded using a fixed probability, eliminating the context update step.

A full coding example is illustrated in Figure 4.18. The significant coefficients in the lower and higher layer are indicated by crosses and circles, respectively. All unmarked coefficients are significant in neither of the layers. The update to the last significant coefficient is coded first by two false flags, where the first flag indicates that the bottom right area is not chosen and the second one that neither the bottom left region is chosen (leaving only the top right region). The subsequent delta values for the y and x component are both 1. Next, the coefficients are scanned sub-block by sub-block starting with the top right sub-block. From the last significant coefficient (red circle), all coefficients in the sub-block are scanned in scan order and a flag for the new significance is coded (in this example these are: 0000000110). For the last significant coefficient position no flag is coded because the new significance is already contained in the update of the position. In the next sub-block in the bottom left, no previously non-significant coefficient becomes significant in the higher layer. This is indicated by a false coded block flag (CBF). The only additional information that is coded for this sub-block are two refinement flags for the two coefficients which were already significant in the lower layer (indicated by crosses). For the last sub-block in the top left, all coefficients are scanned again and 16 new significance and refinement flags are coded. For this example: r0000r0100rr101r (the new significance flags are indicated by 0/1 and the refinement flags, which are not explicitly given here, by r).

### 4.2.4 Fixed Probability Coding

As detailed in Section 2.3.4, the entropy coding engine of HEVC (CABAC) supports coding of bits either using backwards adaptive context based arithmetic coding or a special mode called bypass mode in which the bit is assumed to have equal probability of being 0 or 1. However, in Section 4.1, it was also established that in theory, the mapping probability only depends on the coefficient distribution and possibly the level that the coefficient in the lower layer was quantized to. Therefore, a suitable way of coding these refinement bits would be using a fixed probability mode that uses efficient arithmetic coding without the complexity implications of context adaption. Such a fixed coding option was added to the CABAC scheme. Analog to the context value, which is an input to the context coding block in Figure 2.7, a fixed probability value is provided alongside with the bit value. Both values are then directly passed on to the arithmetic coding engine. In the implementation which is used here, the probability values are only dependent on the lower layer level and on the QP difference between the layers. While this option is simple to implement, there are certainly more advanced options that could be used to select the probability values. For example, the position of the coefficient within the transform unit or the values of already decoded coefficients could also be taken into account. Furthermore, the probability values are fixed for all frames and sequences in this experiment. In a more practical approach, the values could be determined by the encoder and somehow be signaled within the bitstream.

Particularly at the decoder side, CABAC can become a considerable throughput bottleneck. If none of the optional parallelization tools are activated, the entire payload of each frame must be piped through one instance of CABAC in which parallel processing of multiple bins is difficult and often very costly. The reason for this are data dependencies that, on the one hand increase the coding efficiency but on the other hand hinder fast CABAC decoding. Such dependencies occur because, for example, the context update depends on the decoded bit or because for some symbols, the context selection depends on the value of a previously coded symbol. During the standardization process of HEVC, high throughput CABAC implementations in hardware and software were already considered in order to minimize these dependencies. Some of the applied techniques are a reduction of the number of context coded bits, grouping of bits coded in bypass mode or with the same context and context derivation with as little data dependency as possible. While these techniques somewhat defuse the situation compared to the CABAC implementation in H.264/AVC, coding dependencies are not completely eliminated and the CABAC throughput remains a limiting factor. For the case of fixed probability coding, there is no context update and the selection of the probability does not depend on previously decoded values within the higher layer picture. [6] More information on optimized CABAC decoders can be found in [SB12; YH05; Hab+15]. In the presented implementation, the probability is determined using a table lookup. The index for the lookup is determined from the lower layer coefficient level only. However, there might be other dependencies that could be exploited; For example it is reasonable to assume that the probability changes depending on the position of the coefficient within the transform. Also, compared to the context based coding, the complexity of the arithmetic coding step is unchanged. Only the values of the internal lookup table are changed.

[6] While using the lower layer coefficient level together with the mapping setup to determine the probability value that is used in the entropy coding step is a straightforward approach, it also generates a parsing dependency between the layers, meaning that if, because of a transmission error, the coefficient levels in the lower layer were decoded incorrectly or not at all, parsing of the higher layer bitstream might also fail. The same applies if the context for coding is selected depending on the lower layer coefficient level. Furthermore, there might be memory considerations as well. If there is a dependency on the reconstruction values of the lower layer transform coefficients, they must either be saved in memory or decoding of both layers must be performed simultaneously. Because of this, such coding dependencies between frames are usually avoided.

## 4.3 Performance Evaluation

In this section, the performance of the conventional SHVC approach is compared to the proposed refinement coding scheme where the existing residual signal is refined in the transform domain. In this test, a very specific test setup is chosen in order to enable an explicit comparison to the SHVC approach of adding another residual signal in the spatial domain. While SHVC has several coding options within the higher layer, these are constrained for this test. Firstly, the setup is explained in 4.3.1 and the results are subsequently presented in 4.3.2.

### 4.3.1 Coding Scheme

The complete encoding process is performed in two steps. In the first step, the lower layer is encoded using the unmodified reference software HM14.0 [HM-14.0]. This bitstream is then used to add an enhancement layer. The lower layer is encoded without any knowledge of the higher layer and the higher layer has no influence on the lower layer encoding. This is the equivalent approach as implemented in the SHVC reference encoder software. For the encoder configuration, the common testing conditions from Section 2.5.2 are used with two modifications: For the entire coding scheme, first, rate-distortion optimal quantization (RDOQ), and second, sign data hiding is disabled. For RDOQ, the quantization is performed in a rate-distortion optimal way. Essentially, the quantized levels are modified intentionally if the modification yields a gain in a rate-distortion optimal sense. Also for sign data hiding the quantized level of one coefficient is changed in order to skip encoding of one of the sign bits. On average, both of these techniques result in an overall performance gain. While these tools are disabled here, it should be noted that this does not imply that the presented refinement scheme is incompatible with RDOQ or sign data hiding. However, because the mapping is affected by these techniques, the rate-distortion optimal decisions of both methods can not be performed for both layers independently but rather needs to consider the mapping implications. While it can be expected that such adapted variants of RDOQ and sign bit hiding would also increase the overall coding performance, the implementation and evaluation of such a method would require an effort which is outside the scope of this work. It should also be noted that because of these modifications, the results can not directly be compared to those in Chapter 3. For further information on RDOQ and sign data hiding, the reader is referred to [KYC08; CHJ11; Wie15].

In the second step, the higher layer is encoded using the fixed lower layer information. This approach is similar to the design of the SHVC reference encoder, which also codes all layers successively without any dependencies between them. In fact, the lower layer encoder is completely unaware of any subsequent higher layer. There are two separate fundamental coding configurations in the higher layer which are compared in the following: Coding a new residual signal on top of the existing one from the lower layer (similar to SHVC) and refining the existing lower layer residual signal. For both configurations, the coding structure of the lower layer is inherited so that the coding units and transform units of the lower layer are processed again in the higher layer. Of course this also implies that no coding of a coding structure is performed in the higher layer. As in SHVC, the luma and chroma color components are processed successively. For both configurations, no additional coding of the higher layer TU is performed if the corresponding lower layer TU carries no residual data. In this case, the reconstruction of the lower layer is copied and adopted as the reconstruction of the higher layer. If, however, a residual signal is coded in the lower layer, several options for coding the higher layer TU are tested in each of the two configurations. For each of the two configurations, a separate set of options are tested:

**Coding another residual** Similar to conventional SHVC, a new residual signal in the spatial domain can be added in this configuration. For this, the unfiltered lower layer reconstruction signal is subtracted from the original signal to form the residual in the higher layer. Next, the conventional residual coding scheme is applied. The signal is transformed and then quantized using the higher layer quantization parameter (QP). The level information is then encoded to obtain the estimated bit cost of this option. After the respective inverse process, the reconstruction of the residual is used to calculate the associated distortion. Alternatively, the encoder can choose to just set the coded block flag (CBF) to zero, which indicates that all transform coefficients are zero. The higher layer reconstruction signal in this case is equal to the one from the lower layer. The required bits and the resulting distortion are calculated as well. Finally, a rate-distortion optimal decision is made to determine which of the two options to choose for each TU. One special case arises if all transform levels after the transformation and quantization of the higher layer residual signal are already zero. In this case, the CBF is set to zero and there is no decision to make. [7]

**Refining the existing residual** For the second configuration, the existing lower layer residual signal is refined in the transform domain rather than adding another residual signal in the spatial domain. The refinement and coding process were detailed in Section 4.1 and 4.2. The refinement information is coded to obtain the implied bitrate cost. Also here, the inverse process is performed next in order to determine the associated distortion. For the refinement configuration, the encoder can also choose to set the CBF to zero which indicates that the higher layer reconstruction is copied from the lower layer and no refinement is applied. Finally, the rate-distortion optimal decision is made and one of the two options (refining the lower layer residual signal or leaving it unchanged) is chosen. Other than the coding of another residual, the CBF is not set to zero in any special case of the refinement. Thus, the CBF is never implicitly set and the RD decision is always performed.

If the transform size is 4x4, another option is also tested and compared to the other options using the same RD-decision: transform skip. As in conventional HEVC, the transformation is skipped and the quantization is directly applied to the scaled residual values which are then coded into the bitstream. Also for the inverse operation, only inverse quantization and scaling is required. This can also be interpreted as a way of refinement in the quantized spatial domain. It is tested for both configurations: Coding another residual and refining of the existing residual.

[7] The whole process is very similar to the decision process in the reference software encoders HM and SHM. Various options of encoding, all with different bit cost and distortion, are tested and an optimal decision based on a trade-off between the rate and the distortion is performed. The same is applied here. The lambda for the decision is calculated as it is for the higher layer in SHM.

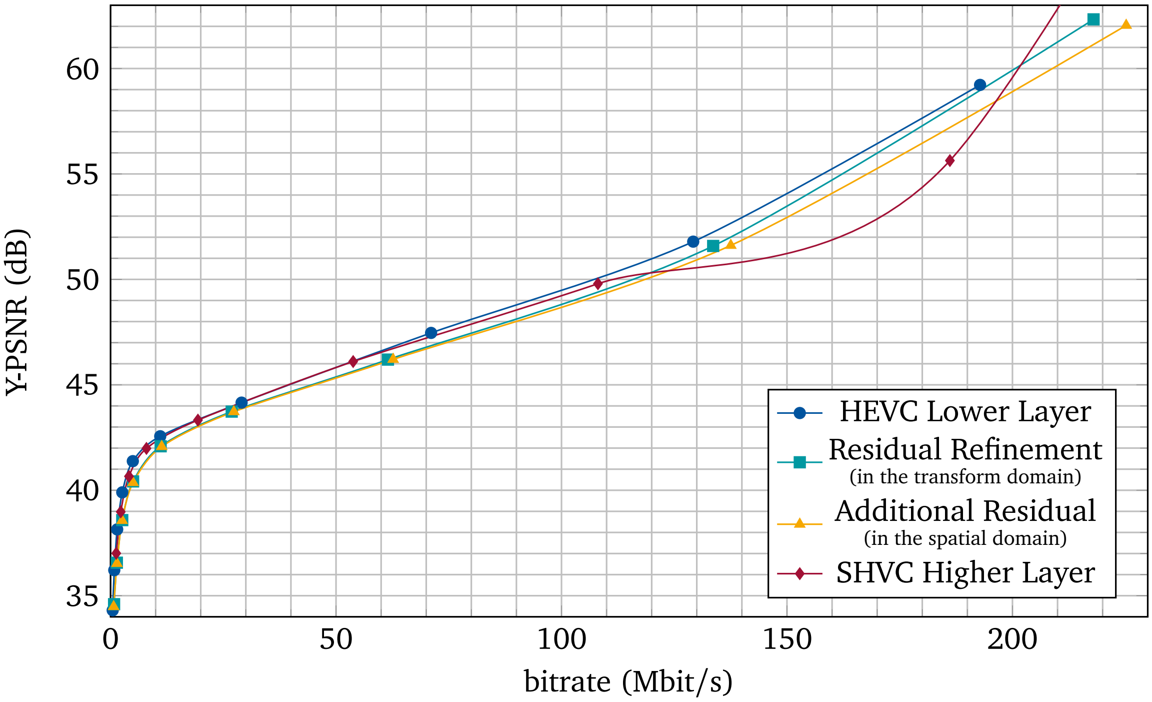
Certainly, the described coding scheme is not particularly suitable for real applications because some of the features that make SHVC so efficient are disabled. For instance, only inter layer prediction is enabled in the higher layer. Neither intra prediction within the higher layer frame nor inter prediction from other higher layer reference pictures is performed. Furthermore, both configurations (coding another residual and refining the existing residual) are only tested if the lower layer carries a residual signal. If, however, only prediction is performed in the lower layer and no residual signal is coded, the higher layer could certainly choose to copy the lower layer prediction and add another residual signal. The coding concept was deliberately trimmed in this way in order to achieve one very specific comparison: If the lower layer carries a residual signal, what is the performance of coding an additional residual signal in the higher layer similar to how it is done in SHVC compared to the performance of refining the existing lower layer residual signal. While the disabled features would significantly increase the overall coding performance, they would do so in both configurations. At the same time, the comparison of the two configurations would become more difficult. [8]

### 4.3.2 Coding Performance

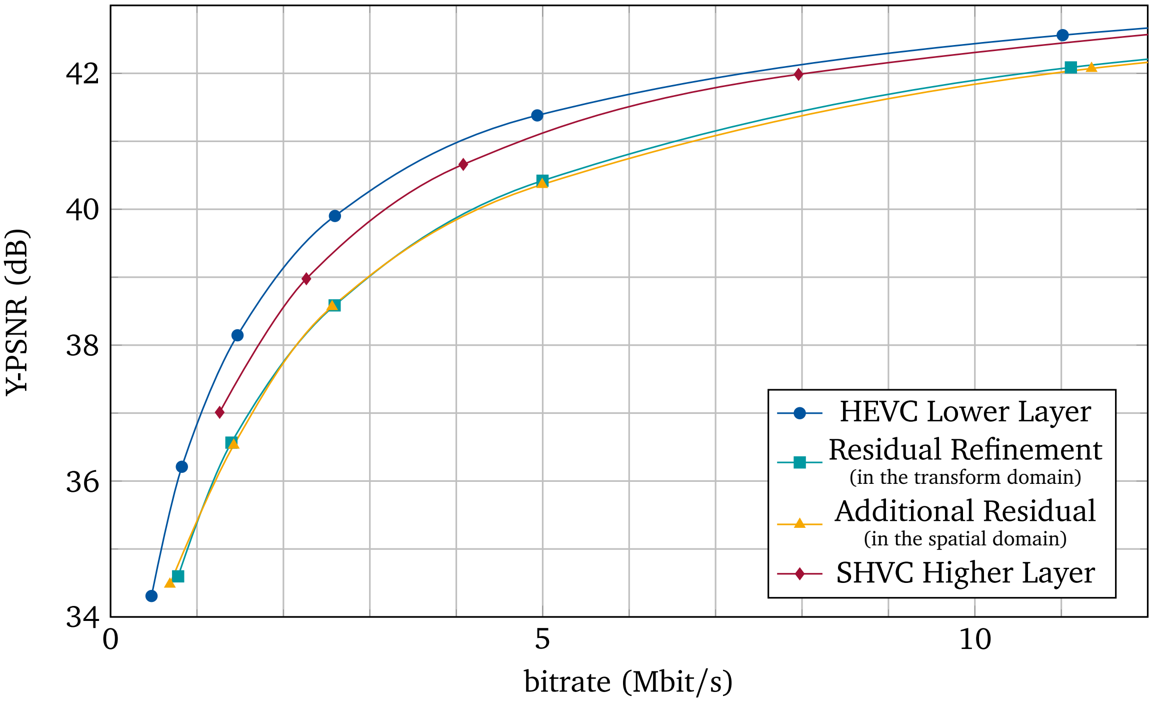
The two described configurations were tested using the random access configuration from the common testing conditions. Because the refinement technique may become particularly helpful for low QP values where the reconstruction quality is approaching lossless coding, the QP range for the lower layer was extended from the four values from 38 to 26 which are specified in the common conditions to the 9 values from 38 to 6 in steps of 4. Figure 4.19 shows the results for the QP difference of -6. The 'HEVC Lower Layer' graph shows the performance of the, except for the disabled RDOQ and sign data hiding tools, unmodified lower layer HEVC streams. On top of this lower layer bitstream, the following two graphs represent the two aforementioned scenarios in which either the existing lower layer residual is refined in the transform domain ('Residual Refinement') or another new residual signal is added in the spatial domain ('Additional Residual'). As a supplemental reference, the graph for the conventional 'SHVC Higher Layer' is presented as well. For SHVC, 'HEVC Lower Layer' also corresponds to the lower layer of the SHVC bitstream. For the higher QP values (detailed in 4.19b), the performance of the two configurations of coding are very similar. When comparing to the single layer coding of the lower layer, a quite significant drop in performance can be observed. For example, at around 5 MBit/s the Y-PSNR drops by roughly 1 dB. For the higher QPs this reduction is even higher. As already mentioned, however, the performance of both configurations could be further increased using various additional techniques. For the lowest two QP values, the residual refinement shows a distinct increase in performance. With the 'SHVC Higher Layer' reference, the reduction becomes more apparent.

For the very low QP points, some unusual behavior for SHVC becomes apparent. At these points, where the PSNR is very high and the difference to the original is limited to a low number of pixel values, the SHVC inter layer coding approach seems to not behave as expected.

[8] Since the two configurations employ a different set of coding tools, the performance increase for the two configurations could somewhat vary.



(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.



(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22. For 'HEVC Lower Layer' also QP 18 is visible.

**Figure 4.19** Coding performance results of the lower layer for the sequence Kimono using single layer coding and the two coding configurations for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -6.

**Table 4.1** Average BD-Rate results of the different refinement coding options compared to the SHVC approach of coding another residual signal for the QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6 as well as a delta QP of -6. A negative sign indicates a BD-rate reduction.

| | Y | U | V |
|---|---|---|---|
| Equiprobable | -1.47% | -1.69% | -1.46% |
| Context Coding | -2.81% | -2.82% | -2.65% |
| Fixed Refinement | -2.87% | -2.90% | -2.71% |
| Fixed Refinement and Significance | -2.44% | -2.52% | -2.31% |

It should be mentioned that the conventional transformation based coding approach is used for SHVC. However, at such high quality where the residual signal energy is very low and mainly consists of noise data, the transform based coding approach is suboptimal. Skipping the transform operation could benefit SHVC in these scenarios. While a mode for skipping the transformation operation is defined in HEVC, it is only enabled for the smallest transform size of 4x4 pixels. [9] Furthermore, a flag per CU can signal to bypass the quantization as well as the transformation stage. However, this feature is disabled in the common testing conditions. Nevertheless, the upwards trend can indicate that SHVC is very close to lossless coding.

The corresponding BD-rate results for the residual refinement results compared to the coding of another residual with a QP difference of -6 between the layers are shown in the 'Context Coding' column of Table 4.1. The coding of the reference, which is the coding of another residual signal, is unchanged and is using a context based coding as specified in conventional HEVC. It can be seen, that in terms of BD-rate over the whole QP range, the refinement has a BD-rate reduction of 2.81% for the Y component; similar values can be observed for the chroma components. The table furthermore explores some of the different entropy coding options of the refinement data. For 'Equiprobable', the refinement bits are coded using the bypass coding mode where essentially, the probability modeling is bypassed and each bit that is input into the entropy coding engine results in one bit being written to the bitstream (see Section 2.3.4). These results are provided as a worst case reference: If the probability of the refinement bits is entirely ignored for entropy coding, this is the resulting performance. For the remaining two columns ('Fixed Refinement' and 'Fixed Refinement and Significance'), the refinement bits are not coded using context adaption, but rather use entropy coding with a fixed probability as described in Section 4.2.4. The probability values are selected depending on the level of the lower layer coefficients and the used mapping. The values themselves were obtained by counting of the mapping decisions at the decoder side. For the column 'Fixed Refinement and Significance', in addition to the refinement bits, also the new significance flags for coefficients that were not significant in the lower layer as well as the coded block flags use coding with fixed probabilities. It is apparent that the performance when using a fixed probability coding of the refinement information is virtually identical to the context based coding. Also, if the new significance information is coded using fixed probabilities, only a slight deterioration in performance can be observed. This slight loss can

[9] In the range extensions to HEVC, skipping the transform operation for larger transform sizes is defined.

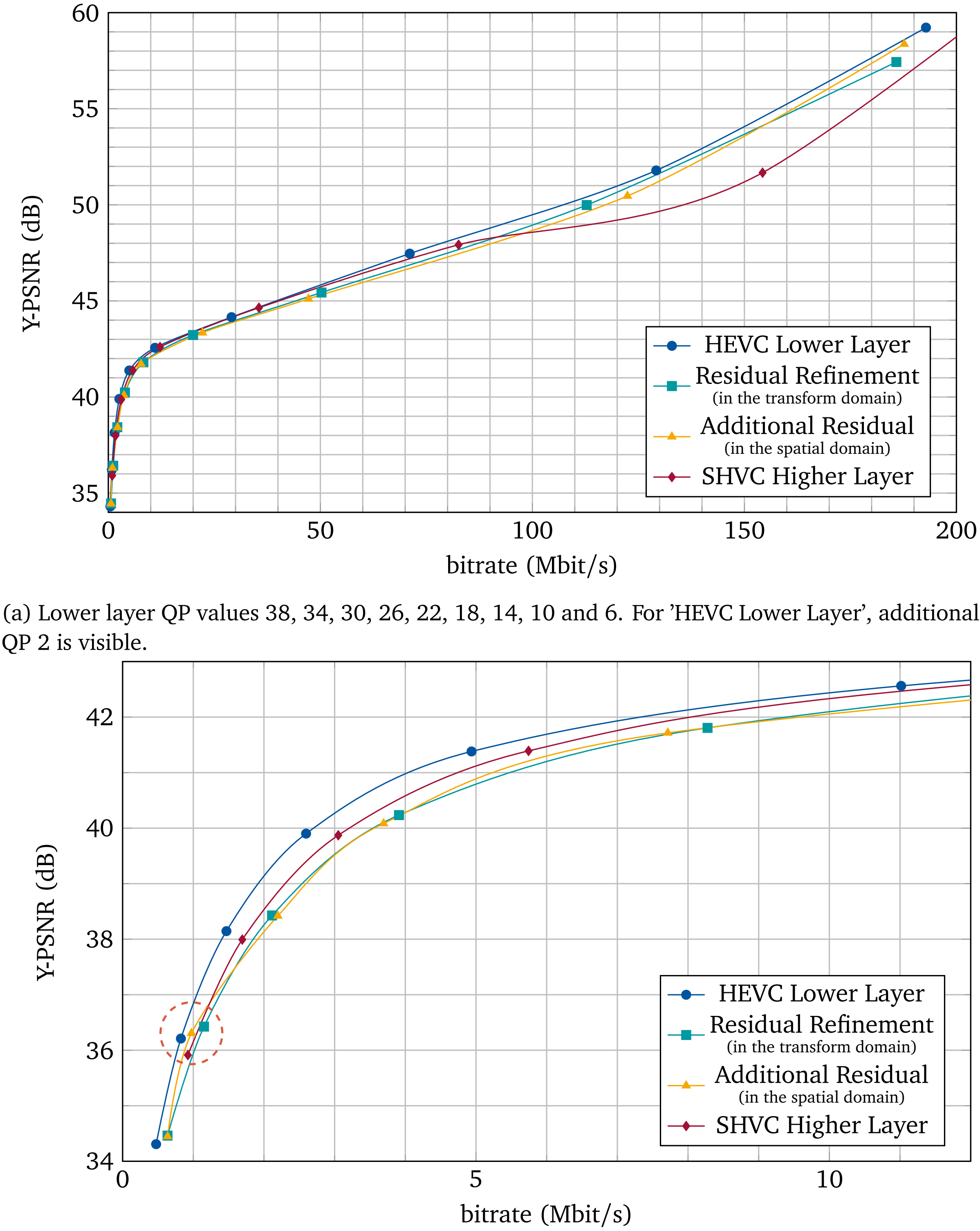
**Table 4.2** Average BD-Rate results of the different refinement coding options compared to the SHVC approach of coding another residual signal for the QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6 as well as a delta QP of -4. A negative sign indicates a BD-rate reduction.

| | Y | U | V |
|---|---|---|---|
| Equiprobable | 2.53% | 0.63% | 0.83% |
| Context Coding | -0.83% | -2.26% | -1.89% |
| Fixed Refinement | -0.29% | -1.58% | -1.25% |
| Fixed Refinement and Significance | -0.14% | -1.66% | -1.30% |

be very well acceptable because it allows coding of the refinement in the higher layer almost without context coded bits. The implications of this are further detailed in Section 4.3.3.

The matching results for the QP delta of -4 between the layers can be found in Table 4.2 and Figure 4.20. While the results look mostly consistent with the results for the QP delta of -6, there is an anomaly for the lower layer QP value of 34 which is highlighted in Figure 4.20b. For this QP, the point for the additional residual diverges from the 'Residual Refinement' curve towards the QP 38 point on the 'HEVC Lower Layer' curve. While the figure only shows the results for the sequence Kimono, this behavior can be observed in all test sequences. The corresponding graphs for the other sequences can be found in Appendix D.3. As it was explained, the first thing that is coded for every transform unit is a coded block flag (CBF) which signals if a residual signal is coded at all. On the encoder side, this flag is set to zero if after transformation and quantization, no transform coefficient is significant or if the encoder chooses to not encode a residual signal. Upon closer inspection of this QP point, it was revealed that, compared to the residual refinement, a higher number of blocks were coded with the coded block flag set to zero so that no actual additional residual signal is coded. The reason for this is in most cases not a decision of the encoder but the fact that after transformation of the higher layer residual signal, all coefficients were quantized to zero values, forcing the CBF to be set to zero. The encoder, in this case, has no possibility to add information in the higher layer using the given higher layer quantizer; So the reconstruction of the higher layer is identical to the reconstruction of the lower layer. Because the encoder is unable to add information and enhance the lower layer reconstruction, the 'Additional Residual' point is shifted towards the corresponding 'HEVC Lower Layer' point. At the same time, the refinement of the existing residual signal allows for coding of an improved residual signal so that this mode can be (and is) chosen by the encoder.

The BD-rate results in Table 4.2 show an average rate reduction of 0.83% over the entire QP range for the residual refinement using context based coding for the refinement information. When entropy coding is disabled for the refinement information, the performance significantly drops and results show a BD-rate increase of 2.53% compared to coding another residual signal. It can be concluded that the probability of the refinement bits is not equiprobable and that the refinement coding can benefit considerably from entropy coding. As for the QP difference -6, the performance of the refinement coding drops when the refinement bits and also the new significance information is coded using fixed probabilities. While the relative drop is higher for the QP difference -4, the overall performance of the



(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.

(b) Detailed view for the lower layer QP values 38, 34 (highlighted), 30, 26 and 22. For 'HEVC Lower Layer' also QP 18 is visible.

**Figure 4.20** Coding performance results of the lower layer for the sequence Kimono using single layer coding and the two coding configurations for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -4.

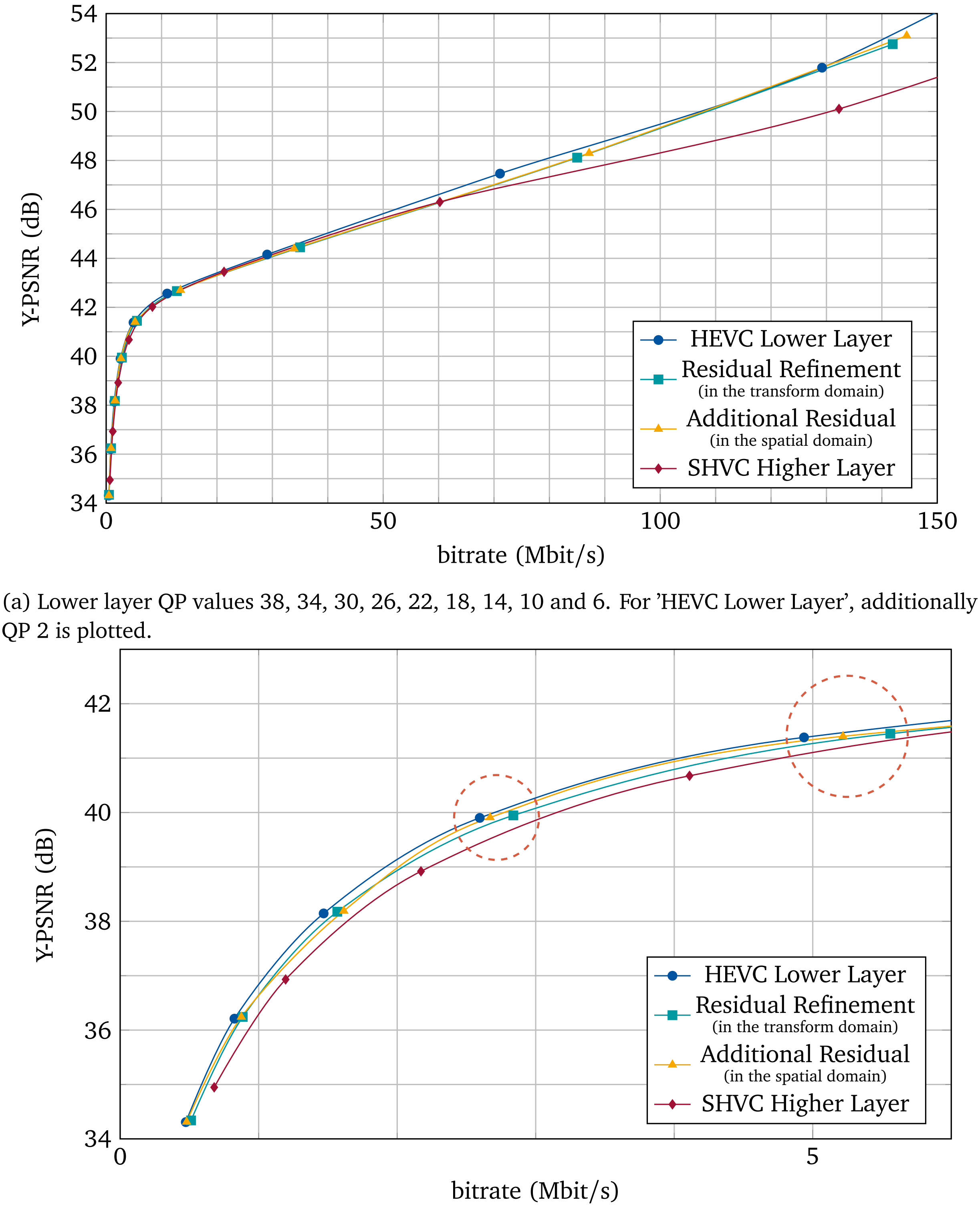
**Table 4.3** Average BD-Rate results of the different refinement coding options compared to the SHVC approach of coding another residual signal for the QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6 as well as a delta QP of -2. A negative sign indicates a BD-rate reduction.

| | Y | U | V |
|---|---|---|---|
| Equiprobable | 0.31% | -0.29% | -0.65% |
| Context Coding | 0.75% | 0.49% | 0.27% |
| Fixed Refinement | 0.72% | 0.49% | 0.24% |
| Fixed Refinement and Significance | 0.83% | 0.63% | 0.43% |

refinement coding with fixed probabilities is still comparable to the performance of adding another residual signal. Overall, the gains for a delta QP of -4 are smaller than for the delta QP of -6.

Finally, the results for the QP difference of -2 can be found in Table 4.3, while Figure 4.21 plots the corresponding results for the sequence Kimono. In the figure, it can be seen how small the distance between the two coding configurations and the single layer stream of the lower layer is. Because of the specific coding configuration and the very small QP difference, only a limited amount additional information is coded especially for the higher QP values which are detailed in Figure 4.21b. As for the QP difference -4, the performance of the highlighted QP points of the 'Additional Residual' are shifted towards the 'HEVC Lower Layer' curve due to the fact that with the approach of coding of an additional residual signal, it is not possible to add more information in a majority of cases. For the higher QP values it further seems that the performance of conventional SHVC ('SHVC Higher Layer') is lower compared to the two tested configurations. However, it should also be noted that compared to SHVC, hardly any information is added for the tested configurations. Thereby, a direct comparison to SHVC should be performed carefully and the practical usability is quite limited.

The BD-rate results in the table show a slight performance loss of the refinement coding compared to coding of another residual signal. With context coded refinement, a BD-rate reduction of 0.75% for the luma component and 0.49% and 0.27% for the chroma components can be observed. When the refinement is coded using fixed probabilities, the performance is approximately identical. When in addition the new significance is coded with fixed coding, the performance drops slightly. The results for equiprobable coding exhibit an unusual behavior and show a higher performance than the other modes using entropy coding. In order to interpret this, it should be recalled that when the refinement is coded, the encoder can choose to set the coded block flag to false and not refine the existing residual. This decision is taken based on a rate-distortion tradeoff. When the residual coding is now using a worse coding approach (like equiprobable coding), the encoder will preferably choose not to refine the existing residual. In the test scheme which is used here, this will move the rate-distortion curve closer to the 'HEVC Lower Layer' curve. In the worst case, the encoder will choose to not refine any residual and the coding performance is nearly identical to the single layer performance. While this will result in higher BD values compared to coding an additional residual signal, the performance is not effectively better. In general, because of the very limited amount of additional information that is added in the higher layer for the



(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is plotted.

(b) Detailed view for the lower layer QP values 38, 34, 30, 26 (highlighted) and 22 (highlighted).

**Figure 4.21** Coding performance results of the lower layer for the sequence Kimono using single layer coding and the two coding configurations for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -2.

QP difference of -2, there is only a limited potential for practical applications where such a small quality increase is required. One application of the QP difference of -2 is in combination with another mapping operation to form a multistage binary mapping. E.g. a mapping with a delta of -6 and -2 could be combined to form a mapping with a delta of -8.

Also from the results it can be seen that the presented coding scheme is not particularly useful for real world applications. While there is a bitrate overhead for the SHVC higher layer, this overhead is much larger for the presented residual refinement and additional residual scheme. This is particularly severe for the lower bitrates at a higher delta QP value. For example, in Figure 4.19b the overhead for SHVC is in the range of 10-20% while for the two presented schemes this overhead is close to 60-70%. It should however also be noted that this looks different for low delta QP values where the quality approaches lossless and the overhead compared to SHVC is much lower. As it was already mentioned, the coding concept was deliberately trimmed in this way in order to enable the specific comparison between the refinement in the transform domain and the addition of a conventional additional residual.

### 4.3.3 Worst Case Analysis of Context Coded Bits

In Table 2.1, the worst case number of context coded bins for coding of the transform coefficients in HEVC for different block sizes was presented. It could be seen that the transform coding was designed such that the maximum number of context coded bins per coefficient is between 1.88 and 1.64. As already explained in Section 2.3, the reconstruction is made up of the luma component and two chroma components that are sub-sampled by a factor of two in horizontal and vertical direction. So, when all three components are considered for the worst case investigation, the result is a number of 2.81 context coded bins per pixel for the residual coding in each coding unit (CU). A compliant decoder must be designed to handle this amount of context coded bins correctly.

When we consider quality scalability in SHVC, each additional layer may add another residual signal to the reconstruction with the same worst case number of 2.81 context coded bins per pixel. While CABAC is already considered a bottleneck in HEVC, this will only exacerbate the situation for each additional layer in SHVC. For refinement of the lower layer residual signal, the worst case context coding considerations depend on the coefficient levels in the lower layer. If, as in the prior worst case scenario, all coefficients in the lower layer are scanned, all coefficients will be scanned again in the higher layer refinement scan. However, for each coefficient, only one additional bit is coded: If the lower layer coefficient is not significant, a new significance flag and, if it was significant, one refinement flag. For multiple layers, the worst case scenario is the situation in which all coefficients are scanned in the lower layer and each higher layer adds a refinement using 1 context coded bin per coefficient. If these bins are coded using context based entropy coding and all color components are considered, this results in approximately 1.5 context coded bins per pixel and layer compared to the 2.81 context coded bins per pixel of an additional SHVC residual.

When the refinement flags are coded using fixed probability coding, the worst case number of context coded bins drops slightly because context coding is only used for coefficients which were not significant in the lower layer. Depending on the number of layers, this results in decreasing values which are always lower than the 1.5 additional context coded bins mentioned before. This can be made clear when one of the 4x4 sub-blocks of the transform is considered. For the worst case in the lowest layer, only one of the coefficients within the sub-block is significant but a significance flag is coded for all 16 values. For refinement coding with multiple layers, the worst case occurs if exactly one of the non-significant coefficients in the lower layer becomes significant in the higher layer. In this case, 15 flags for the new significance information are coded using context adaption. For the already significant coefficient in the lower layer, one refinement flag is coded using a fixed probability. In the next layer, the worst case number of context coded bins is reduced to 14 because two of the coefficients are already significant. This can be continued until all coefficients are significant and only refinement flags are coded.

In the best case, the refinement bins, the new significance and the coded block flags are coded using fixed probabilities. In this case, only the signaling of the last significant coefficient update is performed using context based adaption. Relative to the number of pixels, these few bits have only a very low impact on the number of context coded bins per pixel. In this case, the number of context coded bins per pixel is very close to zero for each additional layer.

## 4.4 Conclusion

In this chapter, an inter layer coding tool for scalable video coding is presented, that operates directly in the transform domain and refines an existing residual signal. This concept is an alternative approach to the coding of another residual signal in the spatial domain as it is performed for scalable coding in HEVC. Basically, the existing lower layer prediction is copied and the coefficients from the lower layer residual signal are mapped from the quantizer in the lower layer to the higher layer quantizer with a smaller quantization step size. For reconstruction, inverse quantization and transformation is then applied to the refined residual signal before it is merged with the prediction signal. For every coefficient from the lower layer, the information to which of the possible higher layer levels the mapping is performed is then encoded into the bitstream using entropy coding. For the coding of the refinement, the knowledge about the coefficient distribution can be exploited in order to realize an efficient and low complex coding scheme.

While a similar concept of mapping between scalable quantizers is also presented in [EF04], the underlying idea is further expanded here. With a suitable assumption on the distribution of the transform coefficients, a rate-distortion optimization of the mapping between two scalar quantizers is performed which may even result in discarding of mapping options. Hereby an efficient embedded quantization scheme with overlapping intervals is compiled.

The performance of the refinement scheme compared to the conventional approach of coding another residual signal and adding it in the spatial domain was subsequently investigated using a simplified SHVC coder as a reference. It should again be noted that this is a very specific test setup which artificially confines the SHVC scheme in order to enable an explicit comparison between the binary refinement scheme and the SHVC approach of adding another residual signal in the spatial domain. The most important difference to normal SHVC is that no motion compensated prediction between the higher layers is allowed. Compared to normal SHVC, this introduces a significant rate overhead especially for the lower rates. For the highest rate points this overhead is much lower. Again this approach is not meant for practical applications but just to enable a direct comparison between refinement coding and the signaling of another residual signal in the spatial domain.

For the direct comparison which is performed here, it could be seen that, depending on the QP difference between the layers, the presented refinement coding scheme demonstrates a comparable performance. While a slight loss in BD-rate performance can be observed for a QP delta of -2, there are BD-rate gains for a QP delta of -4 and -6. It was further detailed how the refinement affects the number of context coded bins per pixel. While these numbers naturally depend on the encoder settings as well as the coded content, the number of context coded bins in a worst case scenario is significantly reduced compared to coding another residual signal. Furthermore, if a slight additional drop in performance is acceptable, the refinement information can be coded using fixed probabilities. With this method, the refinement information can be coded almost without context coding. In either case, the refinement coding approach is an efficient alternative to the coding of another residual signal which can help to lessen the bottleneck of CABAC. Because only one inverse transform operation in the higher layer is required for the reconstruction, it is also highly suitable to be combined with the single loop coding approach presented in Chapter 3. The combination of these two techniques is elaborated in the following Chapter 5.

Finally, there are some interesting aspects for further investigation in the context of the residual refinement. While quite some effort was put into the coding of the refinement information, there is possibly still room for improvement. There might be further dependencies between the flags that can be utilized in the context selection phase of the refinement flags and the new significance flags. The same applies to the probability values that are used for coding with a fixed probability. These probabilities might depend on other factors like the position within the transformation which could also be exploited for entropy coding. As mentioned before, two tools which are enabled in the common testing conditions were disabled here: rate-distortion optimal quantization (RDOQ) and sign data hiding. While the refinement coding approach is fully compatible with these tools, they both require a new multi layer encoder design. This problem arises because the residual refinement coding introduces a very close connection between the residual signals which are manipulated by both tools. In this context, the rate-distortion optimal quantization could also be considered for the mapping optimization. The mapping must also not be fixed but could rather be signaled in the bitstream. As a final goal, the optimization process at the encoder side could consider all layers as well as the mapping between the layers and optimize them jointly. Hereby, the performance loss that arises from discarding some of the possible mapping options could also be shifted to a lower layer, making the mapping process more efficient for the higher layers. However, this also requires an encoder that performs decisions considering all used layers. On another final note, the fixed probability coding could also be employed in conventional non-scalable video coding with CABAC in order to further decrease the number of context coded bits.

# 5 Combining Refinement Coding and the Key Picture Concept

As it was already mentioned, the very distinct refinement approach that was presented and tested in chapter 4 is not generally suited for practical applications. In this setting, the higher layer encoder could only add information to the lower layer reconstruction but was not able to perform any other type of prediction. So in this chapter, the residual refinement technique is combined with the flexible inter layer prediction approach from chapter 3. In combination with the key picture concept, a far more realistic approach is constructed. In order to combine refinement coding with the flexible inter layer prediction approach, some restrictions need to be applied to the basic coding structure. This modified coding structure is then also applied for the SHVC reference in order to enable a reasonable comparison of the two in a more realistic scenario. In section 5.1, the specific coding method is detailed and the corresponding results are shown in section 5.2.

## 5.1 Combined Coding Approach

The basic prediction structure which is used for the combined approach is the same as it was used in Section 3.2.2 and is depicted in Figure 3.19. For the combination of the two techniques, the key picture concept with a key picture distance of 8 frames is applied. For every key picture in the lower layer, only prediction from the lower layer reconstruction of other key pictures is performed. In contrast, the non-key pictures in the lower layer employ the higher layer reconstruction pictures for inter prediction. In order to limit the potential amount of drift when only the lower layer is reconstructed, the prediction of these frames is further constrained to the interval of frames between two consecutive key pictures (including the key pictures themselves). In order to allow for single loop decoding to be performed in the higher layer, constrained intra prediction is applied for all non-key pictures in the lower layer. In the higher layer, there are three types of prediction: conventional intra prediction from within the same frame, inter prediction from another frame in the higher layer and lastly, inter layer prediction from the lower layer reconstruction.

Depending on the frame, two different types of inter layer prediction are employed. For the key pictures in the higher layer, conventional SHVC inter layer prediction is performed. After loop filtering is applied to the lower layer reconstruction, the higher layer can choose to perform prediction from the lower layer by copying the corresponding block from this filtered lower layer reconstruction. This is indicated in Figure 3.19 by the solid arrows between the layers. For the 7 non-key pictures within the GOP, however, residual refinement based prediction can be performed. First of all, the lower layer prediction signal is copied to the

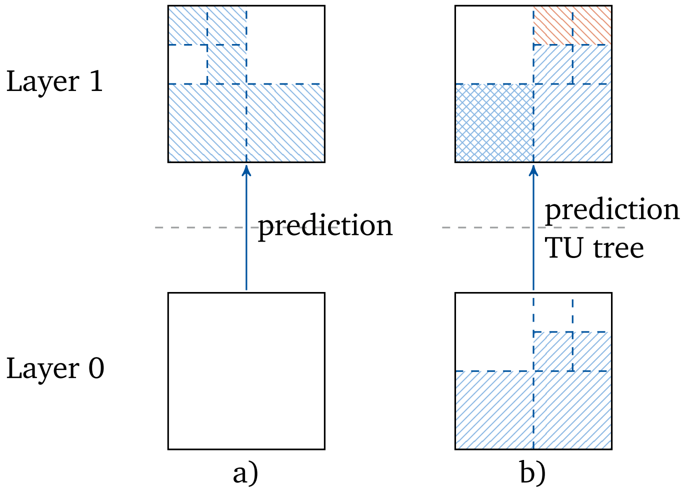


**Figure 5.1** Possible inter layer refinement options. **a** If the lower layer carries no residual, the prediction is copied and a new TU tree and residual can be signaled in the higher layer. **b** If the lower layer carries a residual, the prediction and the TU tree are inherited. If the lower layer TU has a residual it can be refined, otherwise a new residual can be coded.

higher layer. In the higher layer, one of several coding options can be signaled. An initial flag indicates, if any refinement- or new residual information is coded. If the flag is not set, the existing lower layer residual is copied to the higher layer and added to the prediction signal. In this case, the reconstruction signal in the two layers is identical. In the other case (the flag is set), a refinement signal is coded next. This case is illustrated in Figure 5.1. First of all, it is checked if the corresponding block in the lower layer carries a residual signal. If it does not (Figure 5.1a), the prediction signal from the lower layer is inherited and a conventional residual signal can be signaled as it is done in HEVC. This includes a new transform tree, coded block flags and the transform coefficients. If there is a residual signal in the lower layer (Figure 5.1b), the prediction signal as well as the transform tree of the lower layer are inherited. For every TU that carries a residual signal in the lower layer (marked by diagonal lines in the figure), the encoder can choose to refine the existing residual signal or leave it unchanged. In the example, the lower left TU is refined in the higher layer while the residual of the other three blocks is unchanged. If the TU carries no residual, the encoder can choose to encode a new residual signal for the TU using conventional HEVC coding. In Figure 5.1b, this applies to the top right two blocks which are marked by diagonal red lines in layer 1. If no new residual signal is coded and the lower layer has no residual signal, the reconstruction is equivalent to the lower layer prediction signal. In Figure 3.19, this refinement process is illustrated by the dashed arrows between the layers.

Because of the inter layer refinement process, some additional restrictions do apply. For the non-key pictures in layer 1, a new CU structure is signaled as for the key pictures. However, a refinement of the lower layer residual can only be performed on a TU basis which is in the current implementation only possible if the CU size in both layers match. This way, the encoder has the flexibility to signal a new CU tree if this yields a higher coding performance than inter layer prediction with residual refinement. In order to employ inter layer prediction for a CU, the higher layer must signal a CU of the same size as in the lower layer. The encoder will choose this option whenever it makes sense in a rate-distortion sense. While this implementation imposes a slight penalty for the inter layer refinement, it also enables highly flexible inter and intra prediction within the higher layer. For the key pictures, this restriction does no apply because for these pictures, conventional SHVC like inter layer prediction is performed. Also for the SHVC reference, no restrictions are imposed on the coding tree. Another constraint is required with regard to rate-distortion optimal quantization (RDOQ) and sign data hiding. As already detailed in Chapter 4, the current refinement implementation does not support these techniques. While they are not inherently incompatible with the refinement coding approach, they require a joint decision for all layers at the encoder side, which is outside of the scope of this work. Because of this, RDOQ and sign data hiding are disabled for the refinement tests as well as for the SHVC references.

For the SHVC reference, the same basic prediction structure is applied, but of course, only references from within the same layer are used. The same approach was taken in Section 3.2.2 and the corresponding prediction structure is illustrated in Figure 3.14. As defined in SHVC, inter layer prediction is always performed from the loop filtered lower layer reconstruction as indicated by the solid arrows between the layers. The intention is to have an identical set of reference pictures for the refinement coding approach as well as for the SHVC based reference. The last modification which was applied is related to the signaling of inter layer prediction. For refinement coding, inter layer prediction is signaled as an additional coding mode as follows: The first flag indicates if the inter layer coding is used or not. If not, a second flag indicates if inter or intra prediction from within the same layer is to be used. If, due to the slice type, inter prediction is not available, coding of the second flag is omitted. In conventional SHVC, the lower layer reference is placed in the higher layer reference picture list in order enable inter layer prediction. However, because of the dissimilar coding of the inter layer coding mode, these two approaches are not directly comparable. Therefore, the coding for the SHVC reference was modified to also code inter layer prediction as an individual prediction mode.

Besides these modifications, the common testing conditions for SNR scalability with the random access configuration are still obeyed (compare Section 2.5.2). Two layers are used and the QP values for the lower layer are set to 26, 30, 34 and 38. Also, the test sequences and the coding hierarchy including the QP offsets are unchanged. As an additional reference, the results for the key picture concept without residual refinement coding are provided as well. These correspond to the key picture concept which was detailed in Section 3.1. For this test point, the key picture concept is enabled and inter layer prediction in the non-key pictures is performed from the unfiltered lower layer reference. In the higher layer, inter layer prediction is also signaled using a third prediction mode, but no restrictions are enforced on the CU or TU tree for inter layer prediction. This test point shall highlight the results of the key picture concept individually, while the refinement coding setup uses it in combination with the refinement of the lower layer residual signal.

Compared to conventional SHVC, both methods feature a significant reduction of the decoder complexity. For the key picture concept, reconstruction of the higher layer can be performed while at the same time, full reconstruction of the non-key pictures in the lower layer is not required. In particular, only the blocks using constrained intra prediction need to be reconstructed while all motion compensation operations can be omitted. For a detailed

**Table 5.1** BD-rate results for the key picture concept and refinement coding compared to conventional SHVC for the random access configuration and a QP difference of -6.

| | key picture concept | | | combination | | |
|---|---|---|---|---|---|---|
| | Y | U | V | Y | U | V |
| Traffic | -2.04% | -2.77% | -3.31% | -0.62% | -1.73% | -2.30% |
| PeopleOnStreet | -1.85% | -14.79% | -13.10% | 1.84% | -13.21% | -11.47% |
| Kimono | -2.66% | -6.46% | -9.23% | -1.31% | -5.37% | -8.39% |
| ParkScene | -1.68% | -3.01% | -3.80% | -0.38% | -2.15% | -2.96% |
| Cactus | -2.02% | -4.15% | -6.74% | 0.10% | -2.83% | -5.76% |
| BasketballDrive | -0.28% | -7.35% | -6.64% | 2.56% | -6.28% | -5.48% |
| BQTerrace | -0.30% | -7.08% | -8.29% | 0.84% | -6.61% | -7.90% |
| Average | -1.55% | -6.52% | -7.30% | 0.43% | -5.45% | -6.32% |

explanation of the single loop decoding scheme as well as the corresponding complexity reduction results, please refer to Section 3.1. With the residual refinement scheme, the reconstruction complexity of non-key pictures in the higher layer is further reduced. As detailed in Chapter 4, only a single inverse quantization and transformation operation is required in the higher layer. Furthermore, the entropy coding complexity for the refinement is significantly lower compared to the conventional coding of an additional residual signal. The coding performance- as well as the complexity results are detailed in the following section.

## 5.2 Performance Results

In this section, we present the performance results of the key picture concept in combination with the residual refinement approach compared to conventional SHVC. As an additional reference point, the results for the key picture concept only are also provided. In Table 5.1, the BD-rate results for the random access configuration and a QP difference of -6 are shown. For the key picture concept, it can be observed that there is an average BD-rate reduction compared to SHVC of 1.55% in the luma component, as well as 6.25% and 7.3% in the two chroma components. The amount of BD-rate reduction varies amongst the tested sequences. While there is a reduction of 2.66% for luma in the sequence Kimono, the sequences BasketballDrive and BQTerrace exhibit much lower values of 0.28% and 0.30%. These results agree with the results in Section 3.1. When refinement coding is combined with the key picture concept, the performance compared to SHVC drops to a BD-rate increase of 0.43% for the luma component and a BD-rate reduction of 5.45% and 6.32% for the chroma components. Also here, the results vary with the tested sequence. While for some sequences there is a loss for the luma component, there are also sequences that exhibit a BD-rate reduction compared to conventional SHVC. For the lower layer, there is a slight loss compared to conventional SHVC due to the drift which is also present in this coding scheme. With an average loss of

**Table 5.2** BD-rate results for the key picture concept and refinement coding compared to conventional SHVC for the low delay configuration and a QP difference of -6.

| | key picture concept | | | combination | | |
|---|---|---|---|---|---|---|
| | Y | U | V | Y | U | V |
| Traffic | -2.91% | -3.61% | -3.91% | -1.58% | -2.71% | -3.07% |
| PeopleOnStreet | -3.47% | -12.29% | -10.63% | -0.54% | -11.46% | -9.95% |
| Kimono | -4.50% | -6.70% | -8.79% | -3.17% | -5.95% | -8.30% |
| ParkScene | -2.63% | -3.35% | -3.84% | -1.75% | -3.06% | -3.55% |
| Cactus | -3.62% | -5.41% | -7.14% | -1.81% | -4.24% | -6.10% |
| BasketballDrive | -2.18% | -7.16% | -6.69% | 0.34% | -6.09% | -5.08% |
| BQTerrace | -0.95% | -5.03% | -5.76% | 0.08% | -5.61% | -6.60% |
| Average | -2.89% | -6.22% | -6.68% | -1.23% | -5.59% | -6.09% |

-0.14 dB BD-PSNR for the luma and -0.04 and -0.06 for the chroma components, this loss is very close to the loss that was measured in the results in Section 3.1.

The next table (Table 5.2) contains the corresponding results for the low delay configuration and a QP difference of -6. Because of the high drift in the lower layer, the key picture distance for the low delay configuration is set to 4 frames [1]. As defined in the common testing conditions, there is no intra frame in the lower layer except for the first one. Otherwise, all restrictions for the prediction structure that were detailed before do also apply here. As it was already shown, the key picture concept results in an increased coding performance for both layers. In this test, the BD-rate reduction is 2.89% for the luma-, and 6.22% and 6.68% for the chroma components. As for the random access configuration, the overall coding performance increase compared to conventional SHVC is lower when in addition to the key picture concept the residual refinement coding is used. However, on average, the BD-rate results using the low delay configuration show a reduction compared to SHVC by 1.23% for luma and 5.59% and 6.09% for chroma. As for the random access results, there are sequences in the test set that benefit more from the key picture concept than others. This also applies to the combined approach where some sequences even exhibit a slight increase in BD-rate compared to conventional SHVC. Compared to the key picture concept, which was presented in Section 3.1, the drift in the lower layer is reduced for low delay configuration. Compared to SHVC, the average loss for the lower layer is -0.19 dB BD-PSNR for the luma and 0.04 and 0.03 for the chroma components.

For a QP difference of -4, a similar outcome can be observed. The corresponding Tables for the random access as well as the low delay configuration can be found in Appendix E. Also here, using the key picture concept increases the overall coding performance while adding the residual refinement concept reduces some of this gain. Overall, the key picture based coding in combination with the residual refinement shows a BD-rate reduction for both

[1] The tradeoff between the coding performance and the drift was performed in Section 3.1.5. The relatively high drift for the low delay configuration is analyzed here as well.

**Table 5.3** Inverse transform operations normalized by size and the number of context coded bins relative to the SHVC reference for a QP difference of 4 and 6 as well as the random access (RA) and low delay (LD) configurations. The tested configurations are the key picture concept (Key Pic) as well as the combination with the refinement (Comb) and additionally using fixed probability coding.

| | | inverse transformation | | Context Coded Bins | | |
|---|---|---|---|---|---|---|
| | | Key Pic | Comb | Key Pic | Comb | Comb Fixed |
| Δ QP 6 | RA | -0.40% | -8.95% | -0.07% | -4.79% | -9.70% |
| | LD | -2.01% | -6.53% | -1.23% | -3.05% | -5.53% |
| Δ QP 4 | RA | 0.26% | -4.17% | -0.18% | -2.51% | -4.85% |
| | LD | -1.29% | -3.64% | -1.81% | -3.11% | -4.45% |

configurations when compared to the SHVC reference. For the random access configuration the reduction is 0.44%, 7.02% and 7.5% in the luma- and the two chroma components. For the low delay configuration, the BD-rate reduction for the three components are 2.42%, 7.43% and 7.77%.

## 5.3 Complexity Reduction

For the same test set as for the performance results, the complexity of the combined approach is compared to conventional SHVC coding in this section. Table 5.3 summarizes the average results for the QP differences 4 and 6 as well as the random access and low delay configuration. As it was already explained, the combined coding approach aims at a reduction of the number of inverse transform operations and the amount of context coded bins. Therefore, these are the two complexity measurements presented here. Besides the combined coding approach, the table also contains results for the key picture concept as an additional reference. First, we address the inverse transform operations. These are normalized by the size of the transform and compared to the number of the unmodified SHVC coder. It can be seen that for the key picture concept only (Key Pic), there is virtually no change for the amount of inverse transform operations. This corresponds to the results from Section 3.1.5 where there was also only a very slight reduction in inverse transform operations (See Figure 3.12). If, however, the key picture concept is combined with the residual refinement approach (Comb), the value decreases and compared to SHVC, there is a reduction of inverse transform operations of up to 8.95% for a QP difference of -6 in the random access configuration. For the QP difference -4, the results are lower. This can be explained by the inter layer prediction: For the lower QP values, the reconstruction quality of the two layers is closer together and the encoder more frequently chooses to not refine the existing lower layer residual but just copy the lower layer reconstruction because adding of additional information for the small quality difference does not make sense in an RD trade-off. Since inverse transform operations can only be saved if the lower layer residual is refined, the complexity benefit is reduced. Also for the low delay configuration the results of the combined approach are somewhat lower compared to random access.

For the context coded bins, a similar effect can be observed. While there is almost no change for the key picture concept alone (Key Pic), there is a reduction of up to 4.79% for the combination of the key picture concept with residual refinement coding (Comb). It should be noted that for this combination, the refinement information is coded using context adaption. In the last column (Comb Fixed), the refinement is coded using fixed probabilities so that no context coded bins are required. In this case, the number of context coded bins compared to SHVC can be further reduced up to 9.7%. As for the inverse transformation, the values are lower for a delta QP of -4 because less inter layer prediction is utilized.

It should be noted that these results were measured and averaged over the whole test set. In a worst case scenario, the complexity reduction compared to SHVC is significantly higher. In the worst case in SHVC, a conventional residual is coded for both the lower and the higher layer. Inverse transformation must be performed for both of these layers in order to reconstruct the higher layer. However, in the combined approach, a maximum of one inverse transformation is required for the non-key pictures in the higher layer: If prediction from within the higher layer is performed, the lower layer does not have to be reconstructed and if inter layer prediction is used, a new residual may only be added in case the lower layer does not hold a residual. For the context coded bins the situation is different because decoding of bins in the lower layer can not be skipped for the combined coding approach. So in the worst case, the lower layer is not refined and both layers perform a different prediction in conjunction with a separate residual signal. This corresponds to the situation in SHVC where one residual signal can be coded per layer. In conclusion, the amount of context coded bins can only be reduced compared to SHVC if inter layer residual refinement is utilized. [2]

## 5.4 Conclusion

In this chapter, the key picture concept and residual refinement coding are combined and the performance is evaluated in comparison to the conventional SHVC coding approach.

In Chapter 3, it was already shown how the application of the key picture concept increases the overall coding performance compared to conventional SHVC. While there is a gain for all sequences in the test set, the amount of BD-rate reduction varies for the individual sequences. Furthermore, it was also explained and evaluated how the flexible inter layer prediction enables decoding of the higher layer at a significantly reduced computational complexity compared to the regular SHVC approach. This coding performance increase and complexity reduction comes at the cost of a certain amount of drift when only the lower layer is reconstructed. In this chapter, the key picture concept of chapter 3 is combined with the residual refinement coding which was introduced in Chapter 4. In order to enable refinement of the lower layer residual signal, some restrictions need to be applied to the coding

[2] If there is a strong limitation for the complexity of context coding, the higher layer may be forced to perform inter layer prediction. While this would result in a significantly lower worst case number of context coded bins, it would also have a considerable impact on the overall coding performance.

structure in cases where the lower layer residual is refined. Firstly, the coding performance of the combined scheme relative to SHVC is evaluated. It can be observed that independent of the configuration or the QP difference between the layers, the performance results for the combined schemes are lower than the results when only the key picture concept is applied. When the results of the combined refinement approach compared to SHVC are analyzed, it can be seen that, depending on the configuration and the QP difference, the results range from a slight BD-rate loss of 0.43% to a BD-rate gain of 2.42%. The results furthermore vary with each sequence from the test set. While some sequences exhibit good results, other sequences seem to be less suited for the key picture concept and refinement approach.

As it was presented before in Chapter 3, the flexible inter layer prediction scheme of the key picture concept in combination with some minor restrictions can be used to construct a decoder that is able to reconstruct the higher layer of a scalable bitstream at a significantly reduced complexity. Since a decoder complexity reduction is one of the main objectives of the refinement scheme, this was further analyzed. It could be demonstrated that, using the combined approach, the number of inverse transform operations could on average be reduced by up to 8.95% where the application of the key picture concept only yields hardly any reduction. The average number of context coded bins can be reduced by a similar amount of up to 9.7% compared to conventional SHVC. Furthermore, in combination with the refinement scheme, there is also a considerable improvement of the worst case number of inverse transform operations compared to SHVC because reconstruction of the higher layer non-key pictures is always achievable with only one inverse transform operation.

In summary, the refinement coding can be well combined with the key picture concept in order to further reduce the decoder complexity when multiple SNR layers are coded. While the coding performance is slightly lower than for the key picture concept, the results in relation to SHVC, depending on the configuration, range from a small performance drop to a small performance increase. In any case, the additional complexity reduction of binary refinement coding is accompanied by a drop in coding performance compared to the plain key picture concept. Such a slight reduction of coding performance might be acceptable if thereby the decoder complexity can be significantly reduced.

While the evaluation of the combined coding approach is an important investigation, there is still room for improvement. For example, the coding of the refinement information could be further optimized. Furthermore, in the tested implementation, the CU size in both layers must match in order to perform refinement. This signaling constraint could be further relaxed to enable a more flexible signaling. As in the previous chapters, rate-distortion optimal quantization (RDOQ) and sign data hiding are disabled for the presented experiments. For a practical application, these techniques could be enabled using an encoder that is aware of the scalable layers and optimizes and applies these techniques jointly for all layers. A similar multi-layer encoder is necessary if it is a requirement to control the lower layer drift.

# 6 Summary and Conclusion

## 6.1 Summary

For the scalable extension to high efficiency video coding (HEVC), it was decided to adopt a multi-loop approach which refrains from changes to the lower level coding tools of HEVC. By adding the appropriate high-level syntax as well as an upsampling filter, scalability is enabled by copying the (possibly upsampled) lower layer reconstruction as an additional reference picture in the higher layer. The regular motion compensation functions can then be used to enable prediction from the reconstruction of the lower layer. On the one hand, this multi-loop approach offers a reasonable rate-distortion performance while only minimal modifications to the encoder and decoder in both layers are required. On the other hand, it requires full reconstruction of all pictures of all layers at the decoder side. This implies that the decoder complexity significantly increases with the number of layers. While the increase is still moderate in case of spatial scalability, it is severe for quality scalability. In Chapter 2, the average complexity increase as well as a worst case analysis is established. It is shown that for two quality layers, the average as well as the worst case decoder complexity is approximately doubled in case of SNR scalability. But also for spatial scalability, a severe surge in decoder complexity could be detected. While only a scenario with two layers was considered here, it can be expected that the decoder complexity further increases linearly with the number of additional layers. While a moderate increase in complexity as it is incurred for spatial scalability may be acceptable, the extensive increase for quality scalability disqualifies it for applications in which the decoder complexity is a limiting factor. This thesis focuses on scalable high efficiency coding (SHVC) and its complexity. It presents two approaches which aim at a reduction of the decoder complexity particularly in case of quality scalability.

The first technique, which is applied in Chapter 3, enables a more flexible prediction between the layers. In particular, it allows for a prediction in the lower layer from reference pictures in the higher layer. In combination with some minor restrictions at the lower layer, this results in a coding scheme with certain characteristics: Firstly, there is an increase of the overall coding performance. This gain is accompanied by a drift in the lower layer if the higher layer is not decoded. With periodic key pictures, which always allow for a drift free reconstruction, this drift can be controlled and restricted to a certain set of frames. In a visual test it was shown that while the drift lowers the reconstruction quality of the lower layer, it does not cause any particular visible artifacts. Secondly, and more importantly, the coding concept allows for a reconstruction of the higher layer at a significantly reduced complexity compared to conventional SHVC. While the decoder complexity for SHVC almost doubles for a scenario with two quality layers, the measured average results indicate that it is much closer to single layer coding with HEVC using the presented coding concept. This is supported by the measured average complexity values as well as the theoretical worst case complexity analysis. Lastly, the conventional SHVC implementation only allows for coarse grain scalability, i.e. from a scalable bitstream with two layers, only two distinct sub-streams can be extracted and decoded. The proposed concept, however, relaxes this constraint and enables a medium granular scalability in which multiple sub-streams at different bitrates can be extracted from the scalable bitstream. This is beneficial when a flexible and efficient adaption to varying channel conditions is required. Conceptually, the approach is similar to the key picture concept from scalable video coding (SVC), which employs a related flexible inter layer prediction scheme in order to accomplish low complexity decoding and medium grain scalability (MGS). However, at the same time, the modified prediction structure can be implemented without changes to the lower layer syntax or coding tools. Therefore, the concept can, like the SHVC coding scheme, be considered a high level syntax only approach. This is one of the most important differences compared to SVC in which several lower layer coding tools were altered or newly introduced.

The second technique from Chapter 4 focuses on the inter layer prediction process and how information is added in the higher layer. In SHVC, the encoder can choose to copy the lower layer reconstruction to the higher layer and add a supplemental residual signal in the spatial domain. Following the high level syntax only approach, this residual signal is coded using the unchanged HEVC coding scheme. In the proposed refinement technique, instead of coding an additional residual signal, the existing lower layer residual is refined by directly mapping from the lower layer quantizer reconstruction values to those of the higher layer quantizer with a smaller quantization step size. This is especially useful in combination with the flexible inter layer prediction technique, in which the prediction signal in the layers is identical for the non-key frames. With a plausible assumption on the coefficient distribution, the mapping between the two scalar quantizers is optimized using rate-distortion criteria. Since the refinement is directly performed in the transform domain and the reconstruction values in the lower layer as well as the mapping probabilities are known, also the coding of the mapping can be efficiently performed. Besides the conventional context based coding using CABAC, an alternative novel fixed probability coding mode is implemented and tested for the refinement information. A specific test compares the performance of the refinement technique to conventional coding of another residual signal in the spatial domain. In this test, the results show no significant difference in average coding performance. While some configurations lead to a slight drop, others result in a small gain. Specifically for the high rates (low QP values), the refinement scheme shows an even higher potential. Because the lower layer reconstruction is not directly utilized in the higher layer, the inverse quantization and transformation steps in the lower layer can be skipped when only the higher layer is reconstructed. Furthermore, the worst case number of context coded bins for the refinement information is significantly lower compared to coding of another residual signal. This indicates a potential gain in applications with many subsequent layers.

As it was previously mentioned, the refinement process is well suited to be combined with the key picture concept in order to further reduce the complexity of the multilayer decoder. Therefore, in Chapter 5, the flexible inter layer prediction concept from Chapter 3 and the refinement scheme from Chapter 4 are applied in combination. In addition to the complexity reduction which could be demonstrated in Chapter 3, particularly the average number of inverse transform operations and context coded bins can be further reduced. Moreover, the worst case complexity implications for the multilayer coding concept can be further decreased towards the complexity of single layer coding. For the coding performance, the combined approach is close to conventional coding using SHVC. Depending on the configuration, the results range from a slight loss to a small increase in coding performance. At the same time, there is a modest drop in coding performance when the combined results are compared to the results in which only the flexible inter layer prediction scheme is applied. It should be noted that the drift in the lower layer, which is induced by the flexible inter layer prediction scheme, is also present in this combined scheme.

In the past, scalable coding was never widely used in practical applications. For the scalable extension of HEVC, a pure multi loop approach was taken which has serious negative implications for the decoder complexity. This thesis focuses on the severe decoder complexity overhead of SHVC, particularly in the case of quality scalability. It is shown that the decoder complexity can be significantly reduced whereas the overall coding performance is comparable to or even slightly higher than that of conventional scalable coding using SHVC. On the downside, the proposed concept induces a drift when only the lower layer of the scalable bitstream is decoded. While it was shown that the visual impact of this drift is negligible, it still lowers the coding performance of the lower layer. All in all, there is a tradeoff between the reconstruction quality of the lower layer, the coding performance in the higher layer and the complexity of the multilayer decoder. Specifically for quality scalability or a scenario with more than two layers it is questionable if SHVC will be used for practical applications with the current implications for the decoder complexity. With the proposed coding scheme, a more reasonable approach to the tradeoff is given by which scalable coding might become a more viable option.

## 6.2 Future Work

This section is denoted to future aspects of the presented subject whose thorough discussion is beyond the scope of this thesis.

As it was described, the current implementation of the flexible inter layer prediction scheme uses a fixed key picture distance. However, for a real application this approach is impractical. Instead, a picture should be tagged as a key picture in one of the parameter sets. Similar to SVC a quality indicator may be used to enable a simple discarding of NAL units for the decoding of intermediate layers. This signaling should just have minimal impact on the overall coding performance. Furthermore, in this thesis, only a two layer scenario is considered as it is defined in the common testing conditions. While it can be expected that the complexity benefit of the presented flexible inter layer prediction scheme is even higher for more layers, it would be interesting to further investigate the inherent complexity reduction and coding performance in this case. With an additional downsampling filter it is furthermore possible to combine the flexible inter layer prediction concept with spatial scalability. However, it was shown that in this case the complexity overhead is smaller compared to quality scalability so that there is less room for improvement. It is also unknown how strong the influence of the drift is in this scenario. It should be noted that the additional downsampling filter will add further complexity to the coding scheme. Therefore, it is questionable if the extra downsampling filter can be justified by the coding performance and the complexity reduction. As it was explained, SHVC as well as the presented flexible inter layer prediction scheme comply to the high level syntax only approach in which no changes to the lower layer coding tools are applied. If this constraint would be relaxed, it would be possible to employ more advanced inter layer prediction tools that could further improve the overall coding performance and further lower the decoder complexity.

For the refinement coding there are also some additional aspects for further research. In this thesis it was shown how the lower layer residual signal can be refined using a binary mapping process and how this refinement information can be coded. Moreover, it is shown how coding can be performed using conventional context adaptive coding as well as entropy coding using a fixed probability. However, there is potentially still room for improvement for the entropy coding of this information. It is reasonable to assume that the probability distribution of the transform coefficients depends on further factors like the position of the coefficient within the transform. This could be used for an improved context selection or, in the case of fixed probability coding, for a better estimation of the probabilities which are used for coding. Furthermore, the optimization of the mapping process could benefit from a more precise estimation of the probability. If the probability distribution of the coefficients changes depending on the position of the coefficient within the transform, it could be advantageous to use a different mapping for each of these positions. Moreover, only a two layer scenario was covered in this thesis. While an extension of the presented mapping optimization approach is straightforward, it would be very interesting to analyze the achievable coding performance for this case. As explained, the two encoder optimization techniques rate-distortion optimal quantization (RDOQ) and sign data hiding are disabled for the presented residual refinement implementation. Both techniques deliberately alter the transform coefficients in order to improve the rate-distortion performance. While these techniques are not inherently incompatible with the presented refinement scheme, they would require a completely revised multilayer optimization at the encoder side. With the refinement approach, a modification of a transform coefficient in the lower layer has implications for the refinement mapping and the reconstruction in the higher layer. The encoder must therefore take all layers into account when RDOQ or sign data hiding is applied. This enhanced multilayer encoder could perform encoding for both layers with a focus on the higher layer reconstruction quality as opposed to the current implementation which encodes the lower layer without any knowledge of the higher layer. It is also conceivable to let the multilayer encoder choose the appropriate mapping between the layers and signal it within the bitstream. Furthermore, with regard to the flexible inter layer prediction the drift in the lower layer could be monitored or even controlled with this encoder approach. However, such inter layer optimization techniques at the encoder side are outside of the scope of this thesis. Finally, the presented fixed probability coding mode is not limited to refinement information. In other settings like conventional single layer coding this low complexity entropy coding mode might be advantageous as well.

# A Transform Coding Details

## A.1 Last significant position

Table A.1 presents the employed coding scheme. The x and y position are coded independently using the same coding scheme but separate contexts. Coding is performed using a unary coded prefix part and a suffix which uses fixed length coding. First, the variable number of prefix bits signal a specific value (for values 0 to 3) or a range of possible values (for values larger than 3). Each bit of the prefix part ($b_0$ to $b_8$) is coded using context adaption. Table A.2 demonstrates how the context is chosen depending on the color component, the transform size and the prefix bit index. 36 separate contexts are utilized for coding of the prefix (18 for the x, and 18 for the y position).

If the transform size is larger than 4x4, it is possible that the prefix signals a range of values. In this case, one to three additional suffix bits establish the particular value within this range. The number of suffix bits depends on the value range signaled by the prefix. The suffix is then a binary representation of the index of the actual value within the range. If present, the suffix bits are coded using bypass coding.

## A.2 Coefficient Level Coding

The coded sub-block flag is coded using each two contexts for luma and for chroma. One of them is selected if the coded sub-block flag is set for one or both of the adjacent lower or right sub-block. The other context is selected otherwise.

For the significance flag of each coefficient in the sub-block, one of the 42 contexts (27 for luma and 15 for chroma) is chosen according to these rules (see also Figure A.1):

- For the DC coefficient (coefficient 0 of sub-block 0), context 0 is selected.
- If the transform size is 4x4, one of 9 contexts is selected depending on the coefficient position within the block as depicted in Figure A.2.
- For larger transforms, depending on the transform size and the used scan as well as the sub-block index, one of three contexts is chosen (see Figure A.3): First it is determined if there are significant coefficients in the adjacent lower and right sub-block. Then, for each of the four possible cases, one of the three contexts is selected according to the position of the coefficient within the 4x4 sub-block.

**Table A.1** Coding of the last significant position. Each possible value (or value range) is coded using a variable number of context coded prefix bits ($b_0$ to $b_8$). The prefix bits in brackets are not coded if they can be inferred from the transform size. If the prefix signals a range of possible values, an additional bypass coded suffix of one to three bits is coded.

| transform | value | prefix bits | | | | | | | | | suffix bits | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| size | | $b_0$ | $b_1$ | $b_2$ | $b_3$ | $b_4$ | $b_5$ | $b_6$ | $b_7$ | $b_8$ | | | |
| all | 0 | 0 | | | | | | | | | | | |
| | 1 | 1 | 0 | | | | | | | | | | |
| | 2 | 1 | 1 | 0 | | | | | | | | | |
| | 3 | 1 | 1 | 1 | (0) | | | | | | | | |
| $\geq$ 8x8 | 4,5 | 1 | 1 | 1 | 1 | 0 | | | | | $x_0$ | | |
| | 6,7 | 1 | 1 | 1 | 1 | 1 | (0) | | | | $x_0$ | | |
| $\geq$ 16x16 | 8...11 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | $x_0$ | $x_1$ | |
| | 12...15 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | (0) | | $x_0$ | $x_1$ | |
| 32x32 | 16...23 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | $x_0$ | $x_1$ | $x_2$ |
| | 24...31 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | $x_0$ | $x_1$ | $x_2$ |

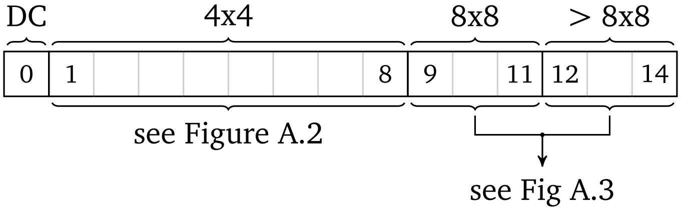


**Figure A.4** Context selection for the significance flag of the chroma component. For the DC component, context 0 is selected independent of transform size. 8 Contexts are used for 4x4 transformations. For larger transforms, depending on the size, a set of three possible contexts is selected.

For the chroma component, the context selection process works similar, but with less contexts. Analogous to the luma component, there is one context for the DC coefficient and a set of 9 coefficients for the transform size of 4x4 coefficients (see Figures A.4 and A.2). For larger transforms, one context is chosen from a set of three contexts according to the transform size. As for the luma component, one of these three contexts is chosen depending on the presence of significant coefficients in the lower and right sub-block and the position of the coefficient within the sub-block.

**Table A.2** The prefix bits $b_0$ to $b_8$ are coded using one of 18 contexts depending on the color component, the transform size and the prefix bit index.

| | transform | prefix bits context | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | size | $b_0$ | $b_1$ | $b_2$ | $b_3$ | $b_4$ | $b_5$ | $b_6$ | $b_7$ | $b_8$ |
| Luma | 4x4 | 0 | 1 | 2 | | | | | | |
| | 8x8 | 3 | 3 | 4 | 4 | 5 | | | | |
| | 16x16 | 6 | 6 | 7 | 7 | 8 | 8 | 9 | | |
| | 32x32 | 10 | 10 | 11 | 11 | 12 | 12 | 13 | 13 | 14 |
| Chroma | 4x4 | 15 | 16 | 17 | | | | | | |
| | 8x8 | 15 | 15 | 16 | 16 | 17 | | | | |
| | 16x16 | 15 | 15 | 15 | 15 | 16 | 16 | 16 | | |

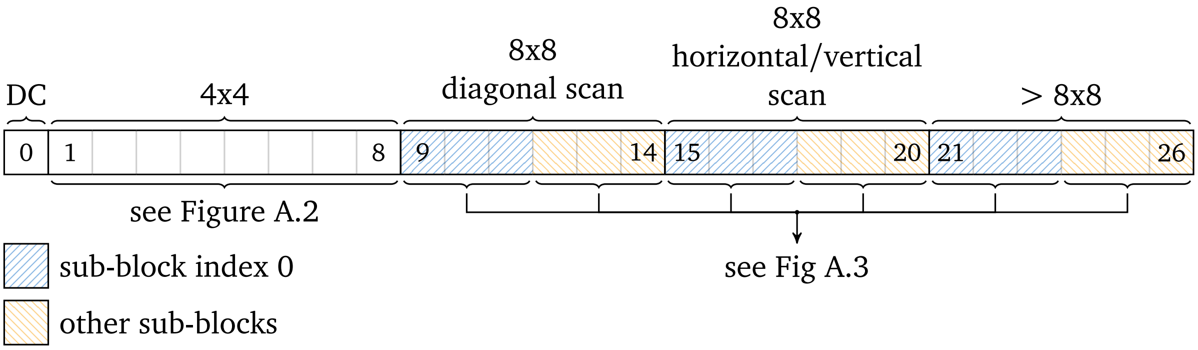


**Figure A.1** Context selection for the significance coefficient flag of the luma component. For the DC component, context 0 is selected independent of the transform size. 8 Contexts are used for 4x4 transformations. For larger transforms, depending on the size, the scan direction and the index of the sub-block, a set of three possible contexts is selected.

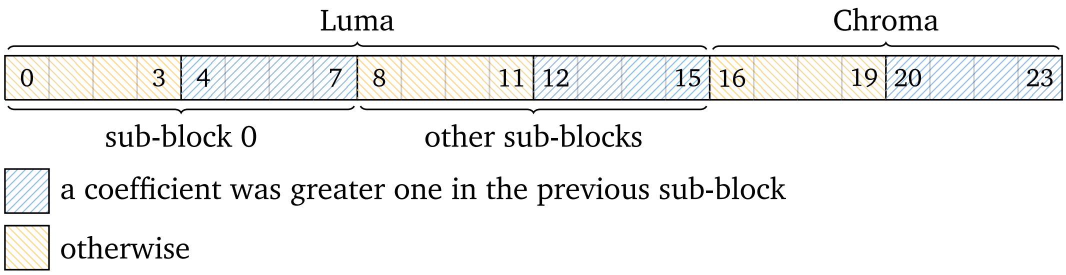
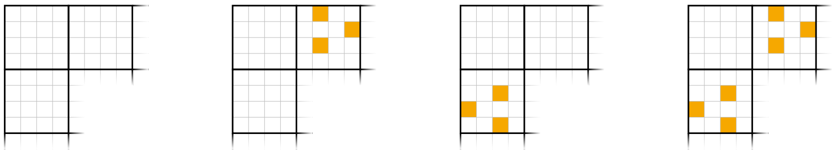


**Figure A.5** Context selection for the coefficient greater one flag. Depending on the component (luma or chroma), the sub-block index (sub-block 0 or not) and if there was a coefficient greater than one in the previous sub-block, a set of 4 contexts is chosen. One of these four contexts is then selected according to Figure A.6

| | | | |
|---|---|---|---|
| 0 | 1 | 4 | 5 |
| 2 | 3 | 4 | 5 |
| 6 | 6 | 8 | 8 |
| 7 | 7 | 8 | |

**Figure A.2** Significance flag context selection for 4x4 transformations depending on the coefficient position. For the bottom right coefficient, the significance flag can be inferred from the last significant position.

1. Check right and lower sub-block for significant coefficients.



2. Select context from coefficient position in sub-block.

| | | | |
|---|---|---|---|
| 2 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| 2 | 2 | 2 | 2 |
| 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| 2 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 |

| | | | |
|---|---|---|---|
| 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 |

**Figure A.3** From each set of three contexts, one is selected. In the first step, it is checked if the right and/or bottom sub-block contain significant coefficients. In the second step, depending on this, a context is selected using the position of the current coefficient within the sub-block.

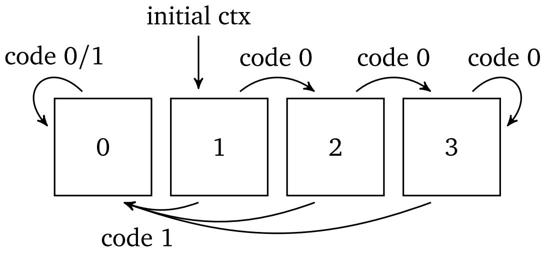


**Figure A.6** State machine to select one of four contexts for the coefficient greater one flag. For every sub-block, context one is initially selected. The state machine then jumps to the other contexts depending on the coded bits.

For the level greater one flags, the 24 contexts are arranged in sets of four contexts (see Figure A.5). A set of four contexts is selected depending on the condition if at least one coefficient in the previously coded sub-block was greater than one. For luma, the context set selection also depends on the sub-block position; There are separate contexts for sub-block 0. One context from the set of four contexts is chosen using a tiny state machine (see Figure A.6). Initially context 1 is selected. Now, depending on the coded greater one flags, the state machine traverses to the other contexts.

A total of six contexts are used for coding of the level greater two flag. For chroma, the context selection only depends on the condition if there was at least one coefficient greater than one when coding the previous sub-block. For luma, the selection furthermore depends on the sub-block position. Two more separate contest are used for sub-block 0.

# B Additional Complexity Results

## B.1 Worst Case Complexity Considerations

**Table B.1** Worst case number of read bytes per pixel at the higher layer resolution. All PUs are of size 8x8 and use bi-directional inter prediction with fractional motion vectors. For the lower layer, the relative number of read bits is lower since the spatial resolution of the lower layer is reduced. For the upsampling it is assumed that the entire lower layer reconstruction is read once. The aggregated values are then put into relation with the worst case result for single layer coding at the enhancement layer resolution.

| | SNR | 1.5x | 2x |
|---|---|---|---|
| Enhancement Layer | 10.1 | 10.1 | 10.1 |
| Base Layer | 10.1 | 4.49 | 2.52 |
| Upsampling | 1.5 | 0.667 | 0.375 |
| Sum | 21.7 | 15.25 | 12.99 |
| Relative to Single Layer Coding | 215% | 151% | 129 % |

**Table B.2** Worst case number of arithmetic operations (addition and multiplication) per pixel at the higher layer resolution. All PUs are of size 8x8 and use bi-directional inter prediction with fractional motion vectors. For the lower layer, the relative number of arithmetic operations is lower since the spatial resolution of the lower layer is reduced. For the upsampling it is assumed that the entire lower layer reconstruction is always upsampled. In case of SNR scalability, the lower layer reconstruction is directly copied and no arithmetic operations are required for the upsampling. The aggregated values are then put into relation with the worst case result for single layer coding at the enhancement layer resolution.

| | SNR | 1.5x | 2x |
|---|---|---|---|
| Enhancement Layer | 57 | 57 | 57 |
| Base Layer | 57 | 25.3 | 14.25 |
| Upsampling | 0 | 13.3 | 9 |
| Sum | 114 | 95.67 | 80.25 |
| Relative to Single Layer Coding | 200% | 168% | 141% |

## B.2 Complexity Results Compared to Simulcast

**Table B.3** Average decoding time difference of the SHVC reference decoder compared to simulcast coding.

| | 2x | 1.5x | SNR |
|---|---|---|---|
| All intra | 160% | 191% | |
| Low delay | 107% | 128% | 100% |
| Random access | 113% | 134% | 104% |

**Table B.4** Average memory access operations as well as arithmetic (multiplication and addition) operations used for filtering relative to the simulcast reference.

| | | Block upsampling | | | Picture upsampling | | |
|---|---|---|---|---|---|---|---|
| | | 2x | 1.5x | SNR | 2x | 1.5x | SNR |
| Memory Access | Low delay | 87% | 75% | 96% | 103% | 93% | 96% |
| | Random access | 94% | 79% | 99% | 109% | 95% | 99% |
| Arithmetic | Low delay | 91% | 127% | 89% | 160% | 187% | 89% |
| | Random access | 96% | 130% | 93% | 162% | 190% | 93% |

**Table B.5** Average difference in number of pixels that an inverse transform is applied for, that are considered by the deblocking process and that are considered by the sample adaptive offset (SAO) process. All values are relative to the simulcast reference.

| | | 2x | 1.5x | SNR |
|---|---|---|---|---|
| Inverse Transform | All intra | 90% | 60% | |
| | Low delay | 84% | 67% | 99% |
| | Random access | 83% | 65% | 100% |
| Deblocking | All intra | 66% | 47% | |
| | Low delay | 90% | 66% | 96% |
| | Random access | 85% | 63% | 94% |
| SAO | All intra | 118% | 86% | |
| | Low delay | 80% | 68% | 99% |
| | Random access | 87% | 73% | 107% |

## B.3 Complexity Results for Different Memory Architectures

**Table B.6** Average memory access operations of SHVC relative to the single layer- and simulcast references for the DDR2 memory architecture (Read operations are aligned to 8 bytes. The size of each read operation is a multiple of 32 bytes.).

| | | Block upsampling | | | Picture upsampling | | |
|---|---|---|---|---|---|---|---|
| | | 2x | 1.5x | SNR | 2x | 1.5x | SNR |
| single layer | Low delay | 109% | 102% | 176% | 129% | 123% | 176% |
| | Random access | 118% | 107% | 190% | 138% | 127% | 190% |
| simulcast | Low delay | 87% | 73% | 97% | 103% | 88% | 97% |
| | Random access | 93% | 77% | 101% | 109% | 92% | 101% |

**Table B.7** Average memory access operations of SHVC relative to the single layer- and simulcast references for the DDR3 memory architecture (Read operations are aligned to 8 bytes. The size of each read operation is a multiple of 64 bytes.).

| | | Block upsampling | | | Picture upsampling | | |
|---|---|---|---|---|---|---|---|
| | | 2x | 1.5x | SNR | 2x | 1.5x | SNR |
| single layer | Low delay | 109% | 97% | 170% | 122% | 111% | 170% |
| | Random access | 117% | 104% | 186% | 131% | 118% | 186% |
| simulcast | Low delay | 86% | 70% | 96% | 96% | 80% | 96% |
| | Random access | 92% | 75% | 101% | 102% | 85% | 101% |

# C Decoding of Intermediate Layers for the Low Delay Configuration

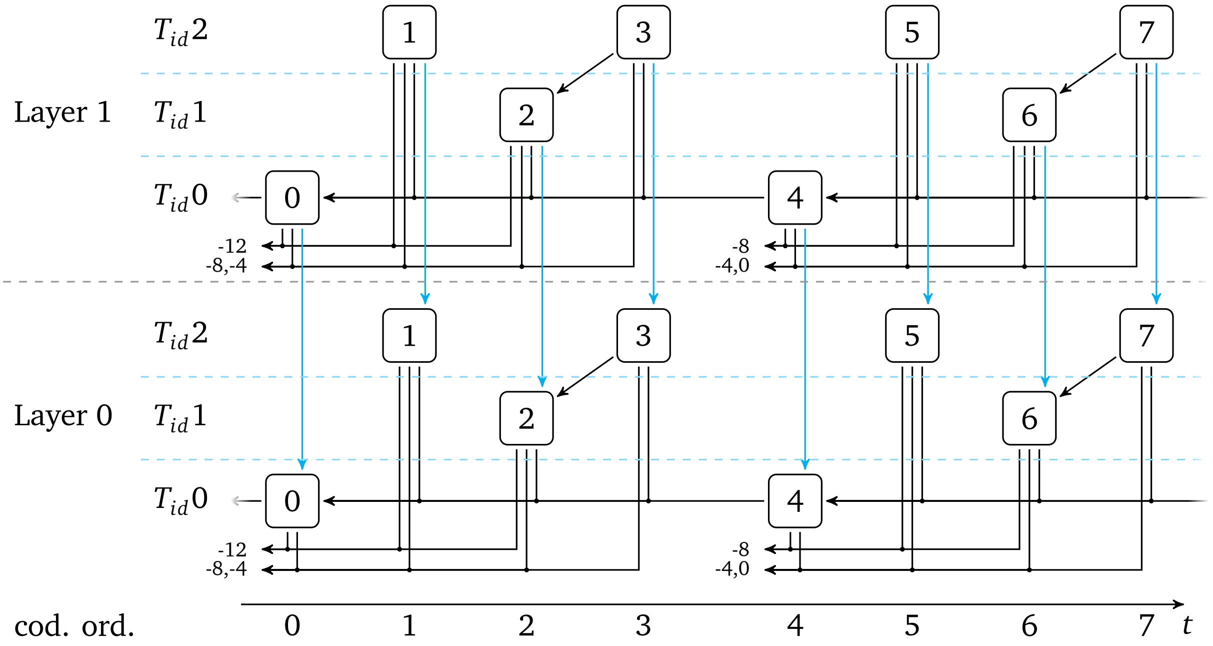


**Figure C.1** One GOP of the low delay configuration with a GOP size of 4 pictures and two layers. Each layer uses temporal scalability with 4 temporal sub-layers ($T_{id}0$ to $T_{id}2$).

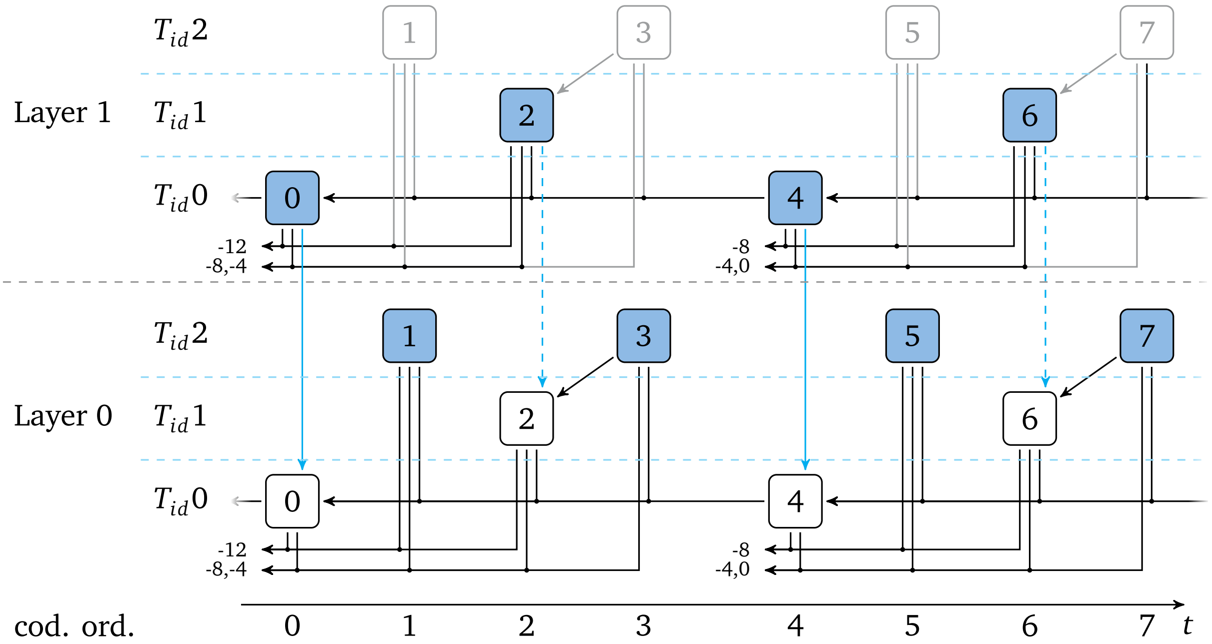


**Figure C.2** One GOP of the low delay configuration with a GOP size of 4 pictures, two layers and three temporal layers. While all temporal layers of layer 0 are decoded, only the temporal layers $T_{id}0$ and $T_{id}1$ of layer 1 are decoded. The pictures that are output are marked in blue.

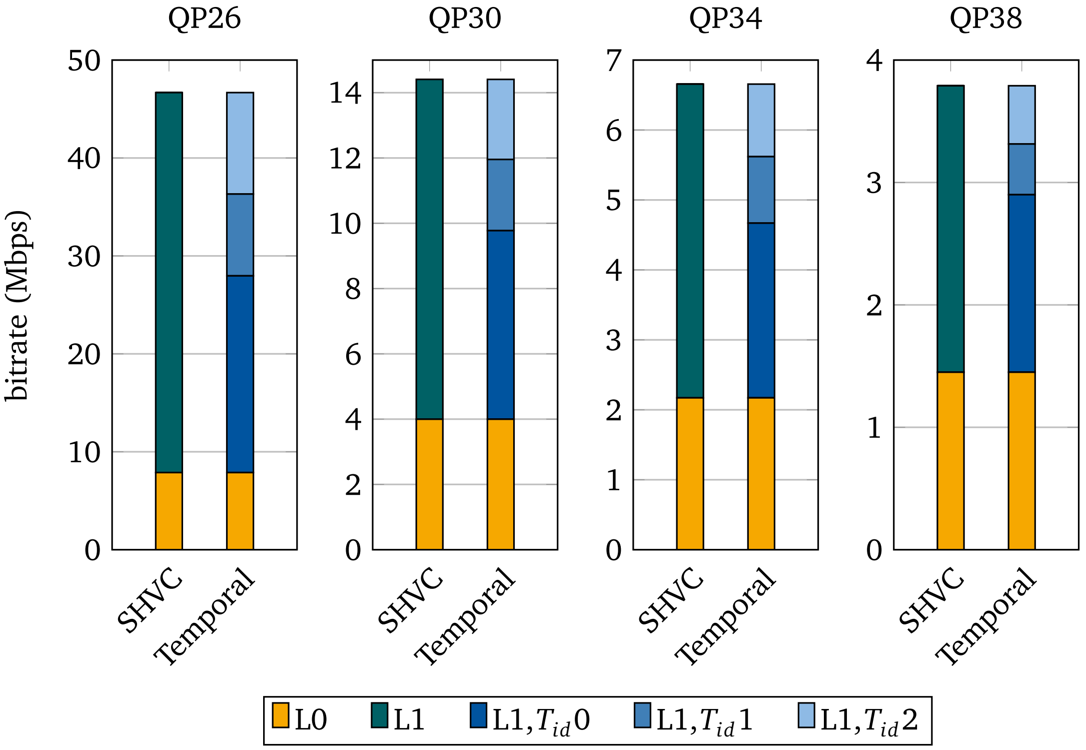


**Figure C.3** Bitrates of the base layer (L0) and the enhancement layer (L1) of SHVC for the sequence Cactus and a $\Delta QP$ of 6 using the low delay configuration and temporal scalability with three layers. In the respective right graph the bitrate of the enhancement layer is split into the four temporal sub-layers ($T_{id}0$ to $T_{id}3$).

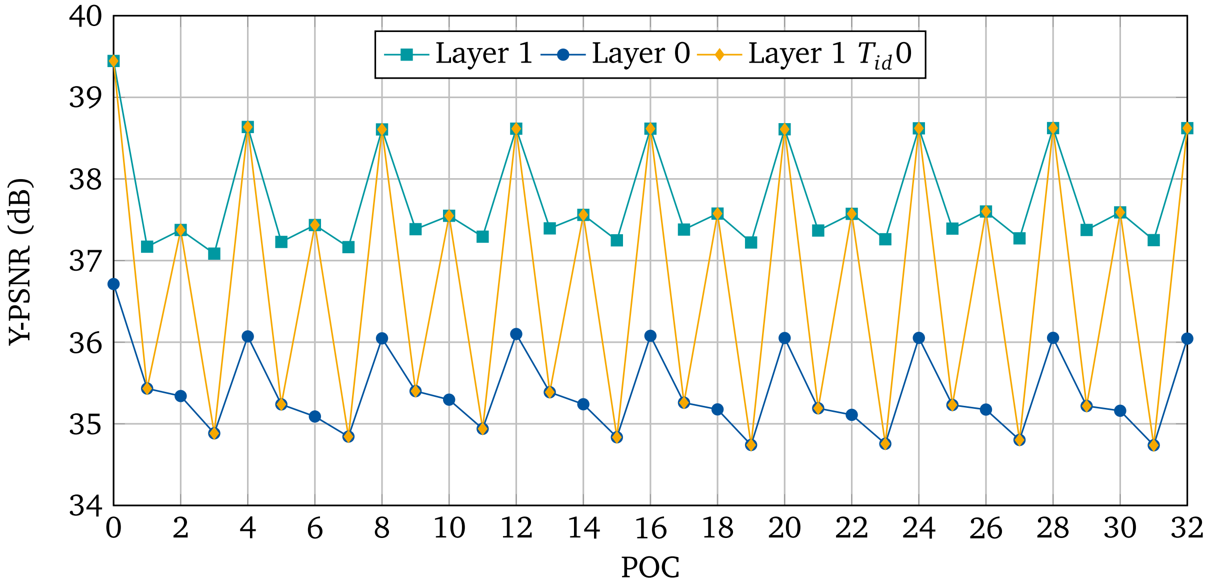


**Figure C.4** The Y-PSNR of the first 33 frames of the sequence Cactus with a base layer QP of 30, an enhancement layer QP of 24, the low delay configuration and 3 temporal sub layers. For the orange curve (—◆—) only the lowest two temporal layers of the enhancement layer ($T_{id}0$ and $T_{id}1$) are decoded. It can be seen that over time, the reconstruction quality fluctuates heavily. This fluctuation has a severe visual impact on the perceived quality.

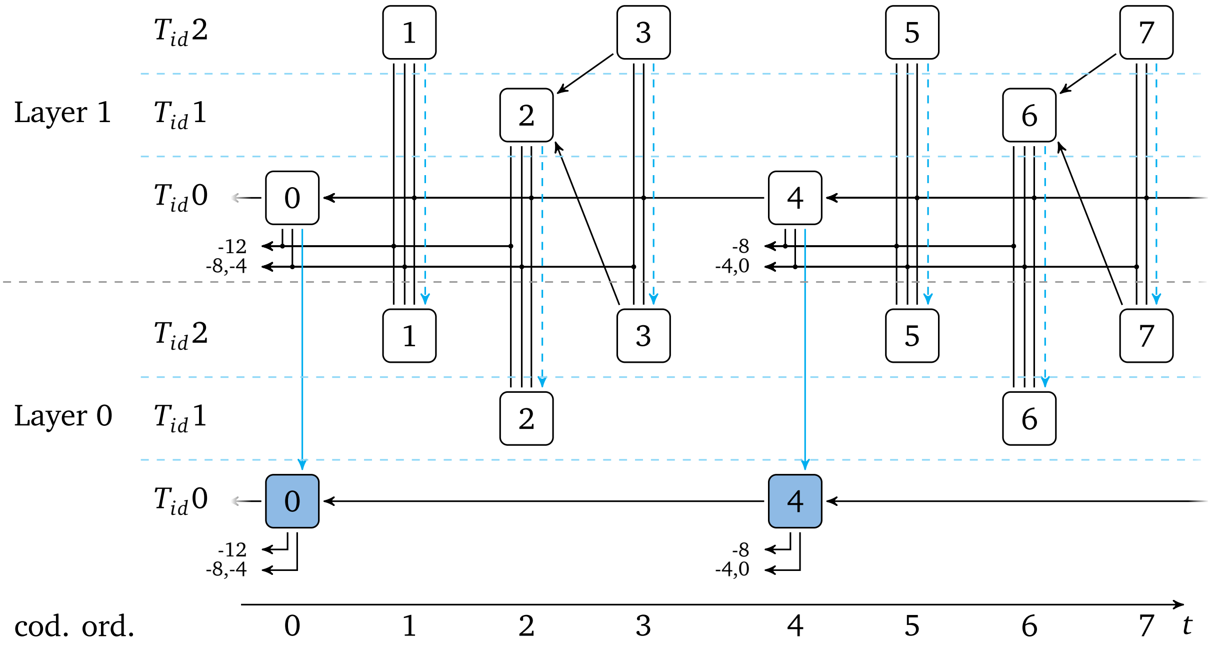


**Figure C.5** One GOP of the low delay configuration with a GOP size of 4 pictures and two layers using the key picture concept. Each layer uses temporal scalability with 3 temporal sub-layers ($T_{id}0$ to $T_{id}3$). The key pictures in the lower layer are marked in blue.

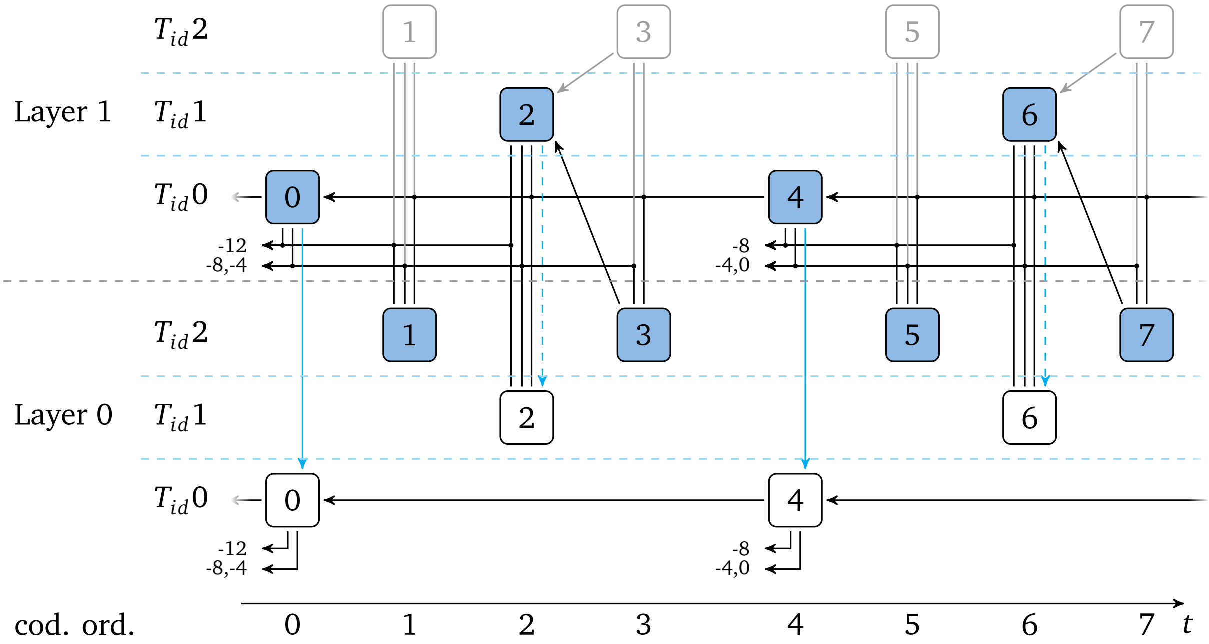


**Figure C.6** One GOP of the low delay configuration with a GOP size of 4 pictures and two layers. Each layer uses temporal scalability with 3 temporal sub-layers ($T_{id}0$ to $T_{id}2$). The pictures that are output are marked in blue. With the key picture prediction structure, the non-key pictures in the lower layer can use the enhanced reconstructed pictures from layer 1 as references.

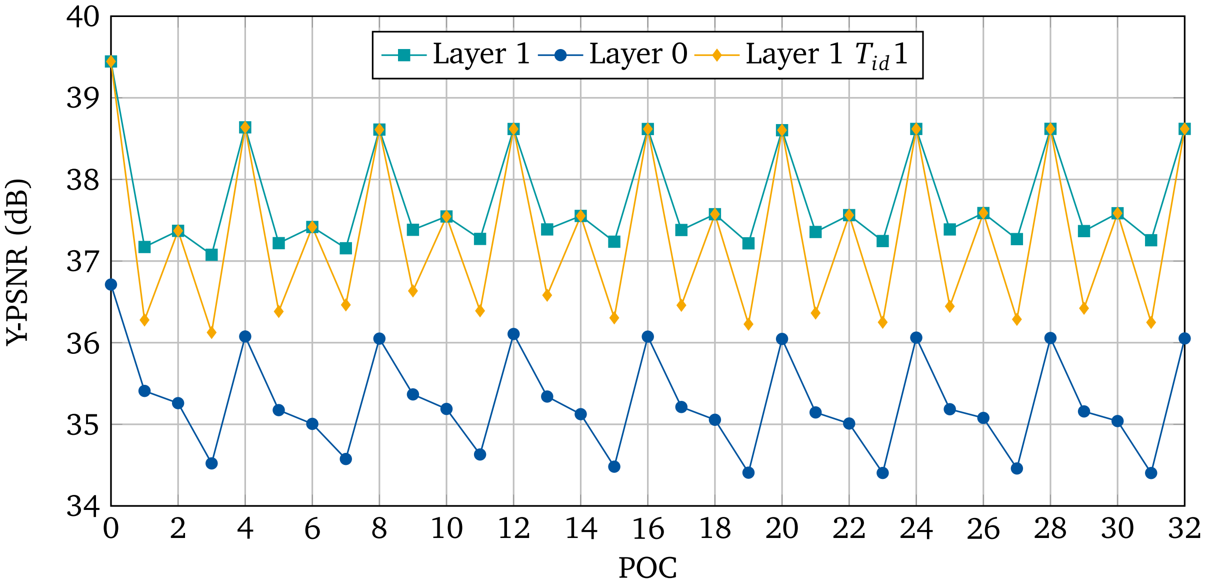


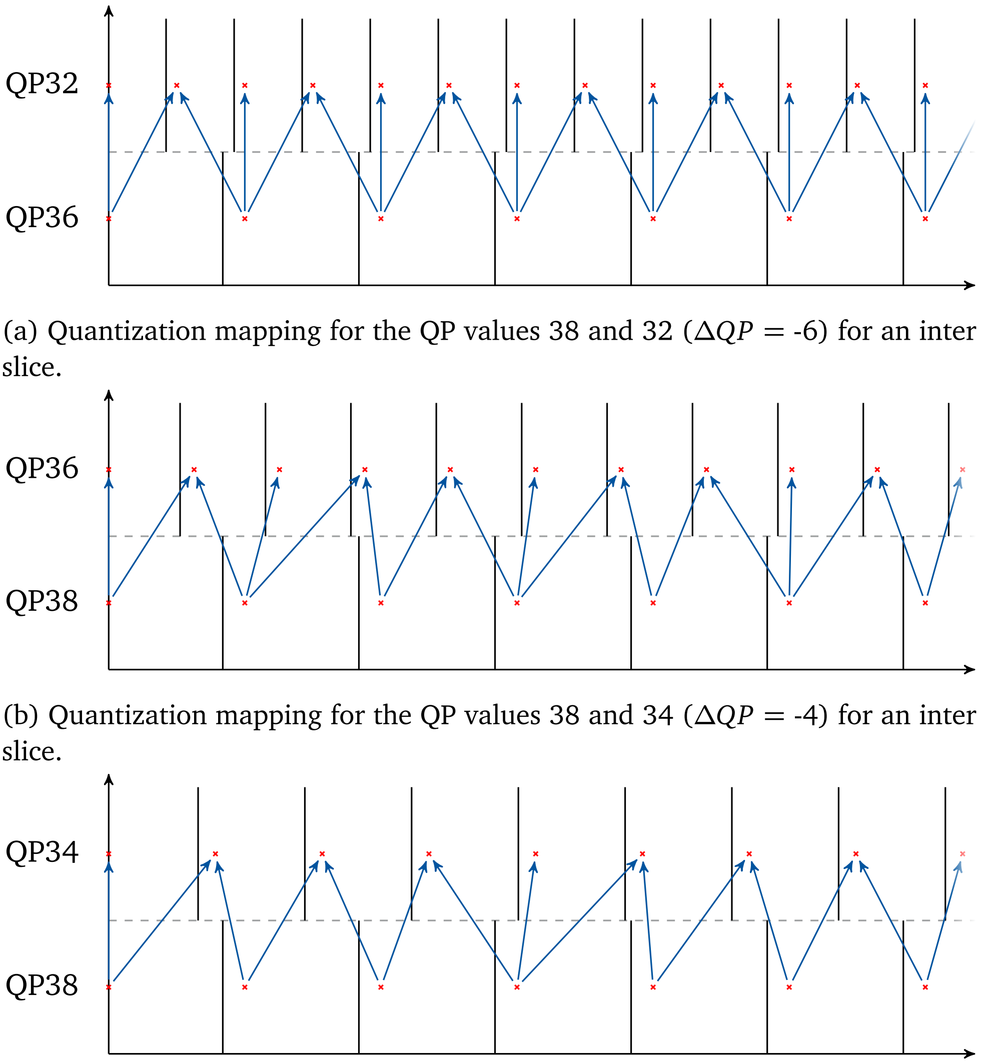
**Figure C.7** The Y-PSNR of the first 33 frames of the sequence Cactus with a base layer QP of 30, an enhancement layer QP of 24, the low delay configuration and 3 temporal sub layers. For the orange curve (—◆—) only the lower two temporal layers of the enhancement layer ($T_{id}0$ and $T_{id}1$) are decoded. It can be seen that while only the enhancement of every picture with an even POC is added, also the reconstruction of the pictures with an uneven POC benefit from the additional information.

**Table C.1** Coding performance of decoding intermediate layers using the higher layer temporal layers. For each temporal layer that is discarded in layer 1, we compare the performance using the key picture concept to conventional SHVC using temporal layers. The low delay configuration is used and a key picture distances (KPD) of 4 pictures for the key picture concept. When only the base layer is decoded (L0), there is a small performance reduction due to the drift which is introduced by the key picture concept. At the same time there is a performance increase compared to SHVC if all temporal sub-layers from layer 1 are decoded (L1,$T_{id}$2). When $T_{id}$2 or $T_{id}$1 and $T_{id}$2 are discarded (decoding of intermediate layers), a significant gain in reconstruction quality becomes evident.

| | | BD-rate (%) | | | BD-PSNR (dB) | | |
|---|---|---|---|---|---|---|---|
| | $\Delta QP$ | Y | U | V | Y | U | V |
| L1, $T_{id}$2 | 4 | -2.11% | -6.28% | -6.78% | 0.0611 | 0.1087 | 0.1206 |
| | 6 | -1.36% | -5.37% | -6.05% | 0.0388 | 0.0966 | 0.1102 |
| | Avg. | -1.74% | -5.82% | -6.42% | 0.0500 | 0.1027 | 0.1154 |
| L1, $T_{id}$1 | 4 | -17.37% | -23.14% | -22.82% | 0.5278 | 0.3981 | 0.4121 |
| | 6 | -24.31% | -32.19% | -32.78% | 0.7581 | 0.6235 | 0.6742 |
| | Avg. | -20.84% | -27.66% | -27.80% | 0.6429 | 0.5108 | 0.5431 |
| L1, $T_{id}$0 | 4 | -18.83% | -27.99% | -27.38% | 0.5868 | 0.4954 | 0.5068 |
| | 6 | -25.94% | -39.21% | -39.52% | 0.8384 | 0.7915 | 0.8558 |
| | Avg. | -22.39% | -33.60% | -33.45% | 0.7126 | 0.6434 | 0.6813 |
| L0 | 4 | 3.13% | 1.10% | 1.39% | -0.0984 | -0.0162 | -0.0252 |
| | 6 | 4.09% | 1.59% | 2.15% | -0.1267 | -0.0230 | -0.0351 |
| | Avg. | 3.61% | 1.34% | 1.77% | -0.1126 | -0.0196 | -0.0301 |

# D Coefficient Refinement

## D.1 Binary Refinement Mapping for Inter Prediction

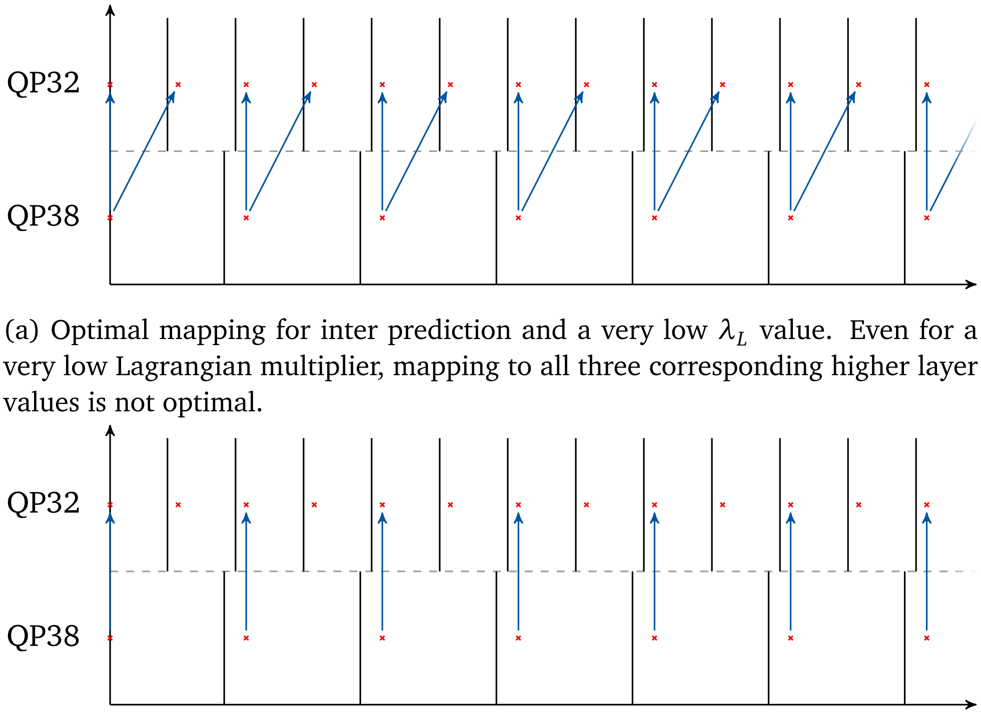


(a) Quantization mapping for the QP values 38 and 32 ($\Delta QP$ = -6) for an inter slice.

(b) Quantization mapping for the QP values 38 and 34 ($\Delta QP$ = -4) for an inter slice.

(c) Quantization mapping for the QP values 38 and 36 ($\Delta QP$ = -2) for an inter slice.

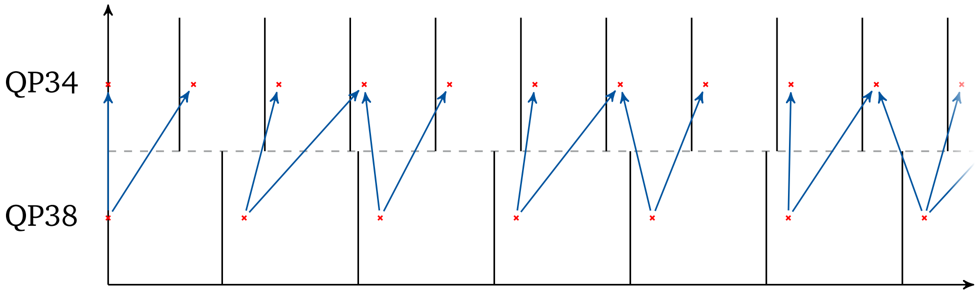
**Figure D.1** Quantization mapping for a $\Delta QP$ of -6, -4 and -2 for an inter slice. Depending on the lower layer reconstruction value, there are two or three reconstruction values that the lower layer value can be mapped to in the higher layer quantizer.



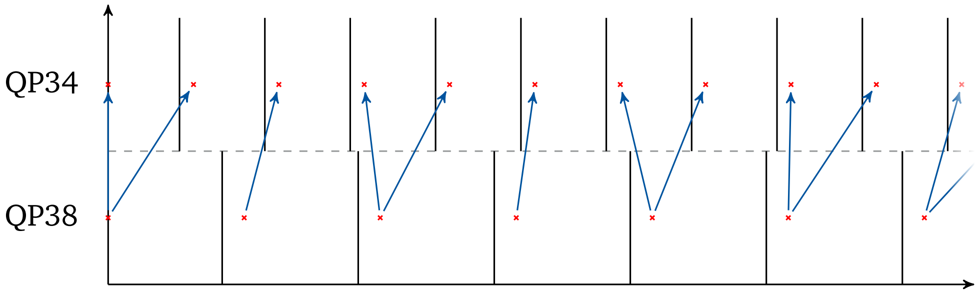
(a) Optimal mapping for inter prediction and a very low $\lambda_L$ value. Even for a very low Lagrangian multiplier, mapping to all three corresponding higher layer values is not optimal.

(b) When the $\lambda_L$ value is increased in the case of inter prediction, at some point the optimal mapping only allows mapping to the same value in the higher layer.

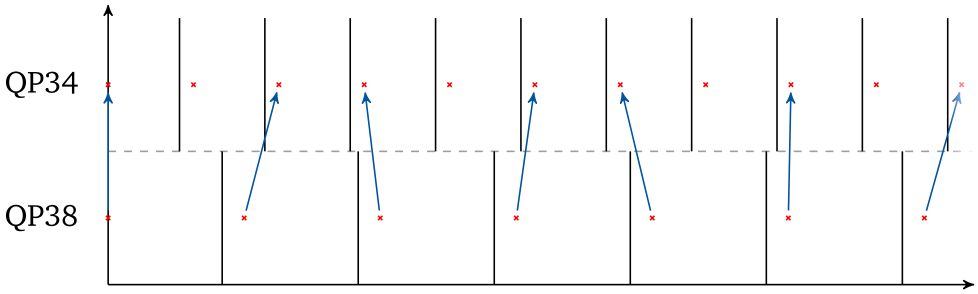
**Figure D.2** Optimal mappings for different values of $\lambda_L$, a QP delta of -6 and inter prediction.



(a) Optimal mapping for a very low $\lambda_L$ value. Even for this very low penalty on the bitrate, some feasible mappings are not optimal (E.g. from the first significant coefficient in the lower layer to the first significant coefficient in the higher layer.)

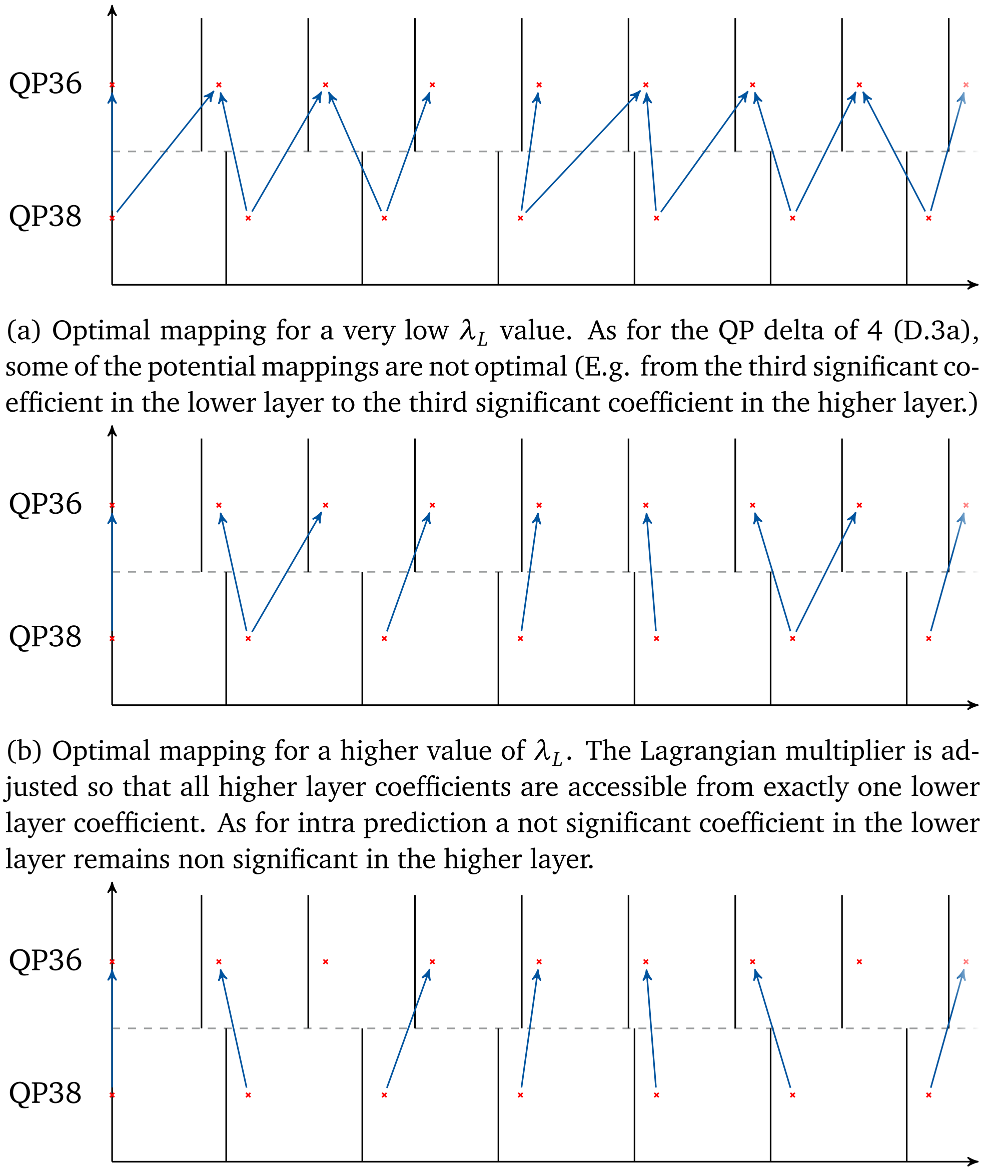


(b) Optimal mapping for a higher value of $\lambda_L$. The Lagrangian multiplier is adjusted so that all higher layer coefficients are accessible from exactly one lower layer coefficient.



(c) Optimal mapping for a high value of $\lambda_L$. The mapping converges to a pure one to one mapping in which a lot of higher layer coefficients are not reachable. The reconstruction values are modified but no additional information is added in the higher layer.

**Figure D.3** Optimal mappings for different values of $\lambda_L$, a QP delta of -4 and inter prediction.



(a) Optimal mapping for a very low $\lambda_L$ value. As for the QP delta of 4 (D.3a), some of the potential mappings are not optimal (E.g. from the third significant coefficient in the lower layer to the third significant coefficient in the higher layer.)

(b) Optimal mapping for a higher value of $\lambda_L$. The Lagrangian multiplier is adjusted so that all higher layer coefficients are accessible from exactly one lower layer coefficient. As for intra prediction a not significant coefficient in the lower layer remains non significant in the higher layer.

(c) Optimal mapping for a high value of $\lambda_L$. Also here, the mapping reverts to a pure one to one mapping in which a lot of higher layer coefficients are not reachable. The reconstruction values are modified but no additional information is added in the higher layer.

**Figure D.4** Optimal mappings for different values of $\lambda_L$, a QP delta of -2 and inter prediction.

## D.2 Binary Refinement Mapping for 3 Layers

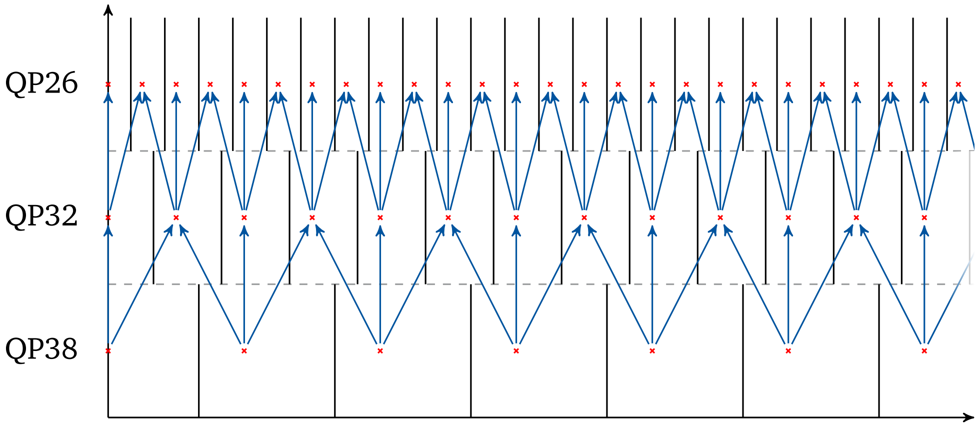


**Figure D.5** Per layer optimized mapping for a very low $\lambda_L$ value and a delta QP value of -6 between the layers. For each layer, all possible higher layer coefficients are reachable from the lower layer reconstruction values. Mapping is possible from all reconstruction values in the lowest layer to all reconstruction values in the highest layer.

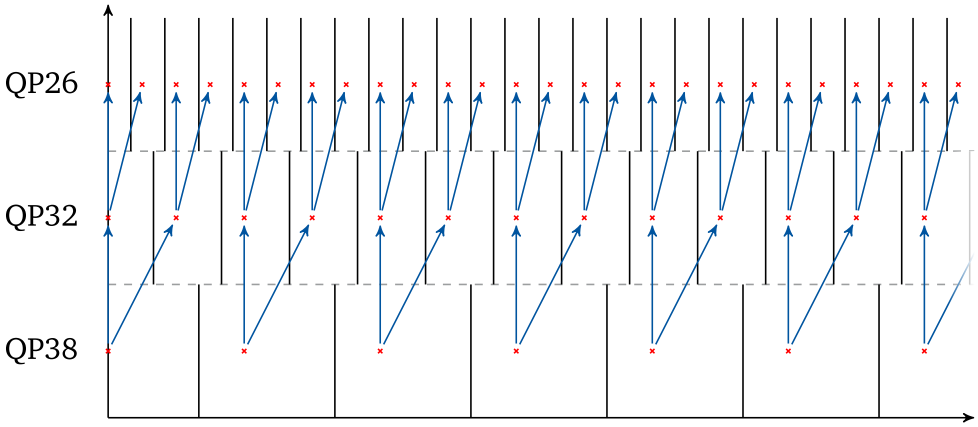


**Figure D.6** Per layer optimized mapping for a higher $\lambda_L$ value and a delta QP value of -6 between the layers. For each layer, mapping is possible from each reconstruction value to two reconstruction values in the higher layer. Each coefficient in the highest layer is reachable from exactly one reconstruction value in the lowest layer.

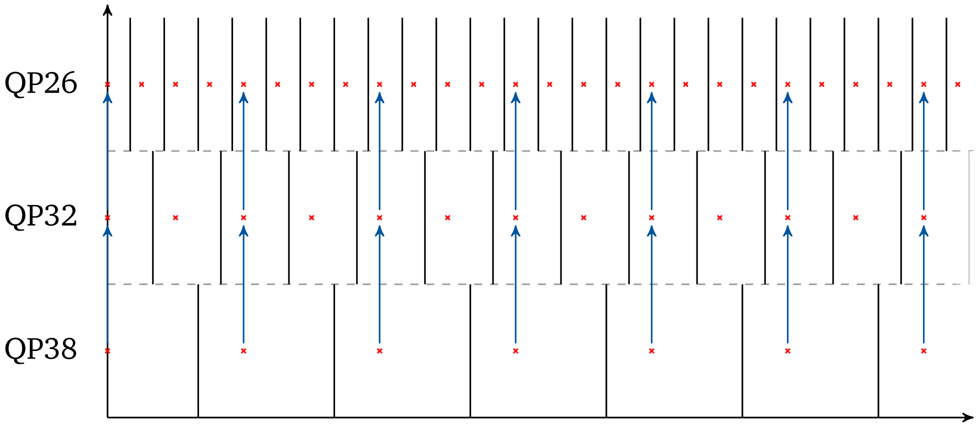


**Figure D.7** Per layer optimized mapping for a high $\lambda_L$ value and a delta QP value of -6 between the layers. No refinement is performed and the reconstruction value for each reconstruction value in the lowest layer remains unchanged and many of the higher layer coefficients can not be reached.

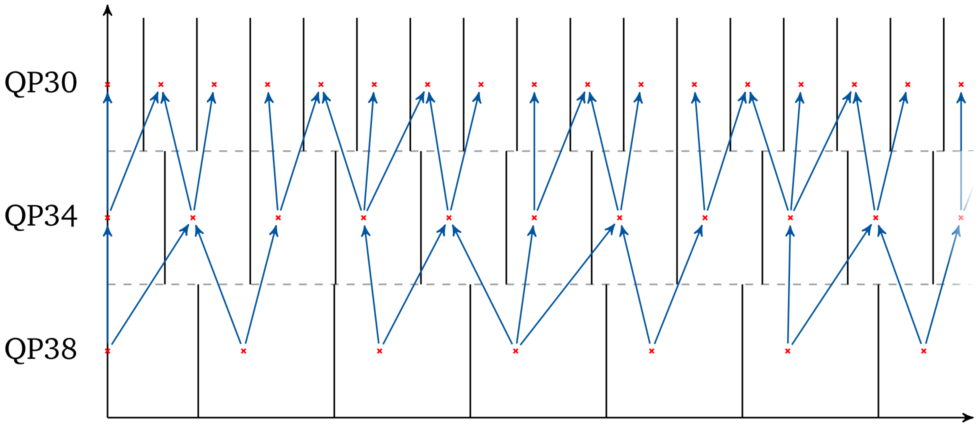


**Figure D.8** Per layer optimized mapping for a very low $\lambda_L$ value and a delta QP value of -4 between the layers. For each layer, all possible higher layer coefficients are reachable from the lower layer reconstruction values. Mapping is possible from all reconstruction values in the lowest layer to all reconstruction values in the highest layer.

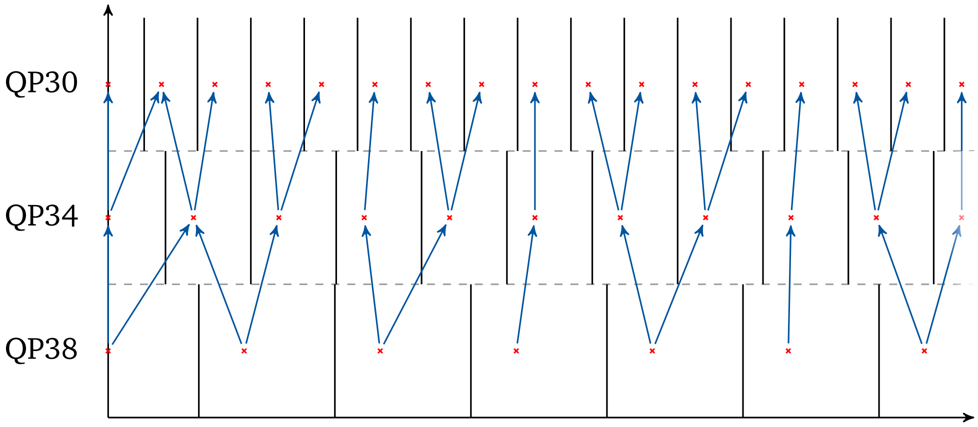


**Figure D.9** Per layer optimized mapping for a higher $\lambda_L$ value and a delta QP value of -4 between the layers. For each layer, mapping is possible from each reconstruction value to one or two reconstruction values in the higher layer. Each coefficient in the highest layer is reachable from exactly one reconstruction value in the lowest layer.

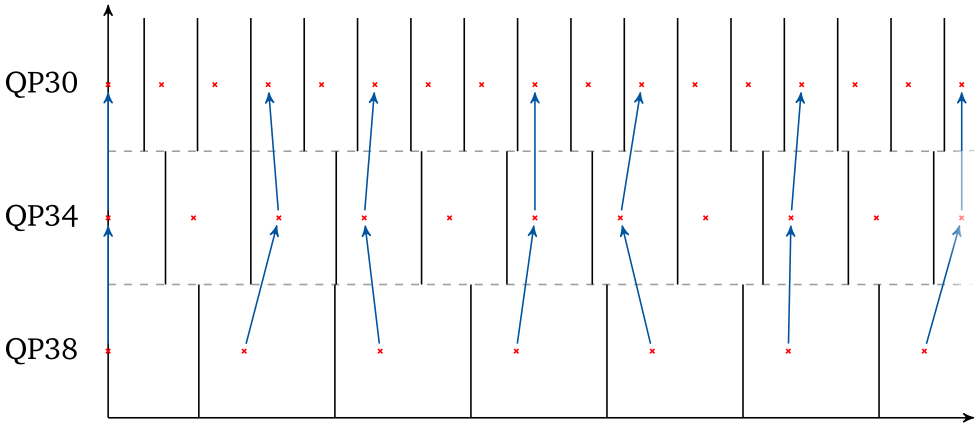


**Figure D.10** Per layer optimized mapping for a high $\lambda_L$ value and a delta QP value of -4 between the layers. No refinement is performed. The reconstruction values are just shifted for every layer and many of the higher layer coefficients can not be reached.

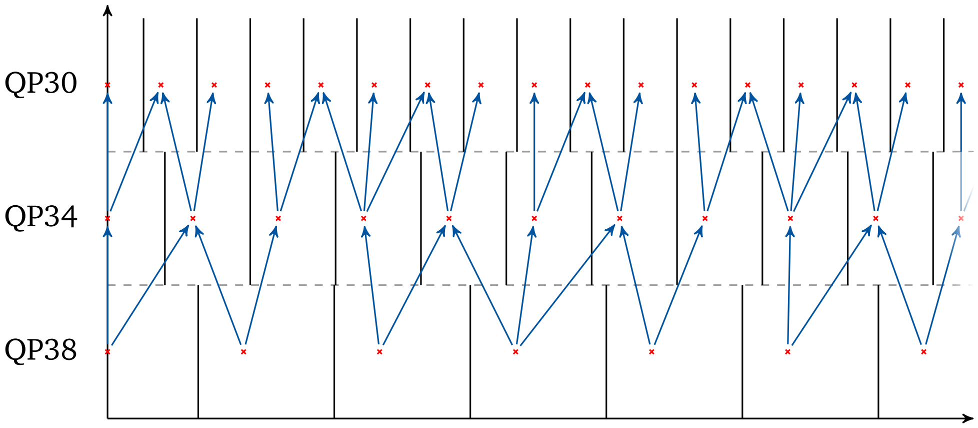


**Figure D.11** Per layer optimized mapping for a very low $\lambda_L$ value and a delta QP value of -2 between the layers. For each layer, all possible higher layer coefficients are reachable from the lower layer reconstruction values. Mapping is possible from all reconstruction values in the lowest layer to all reconstruction values in the highest layer.

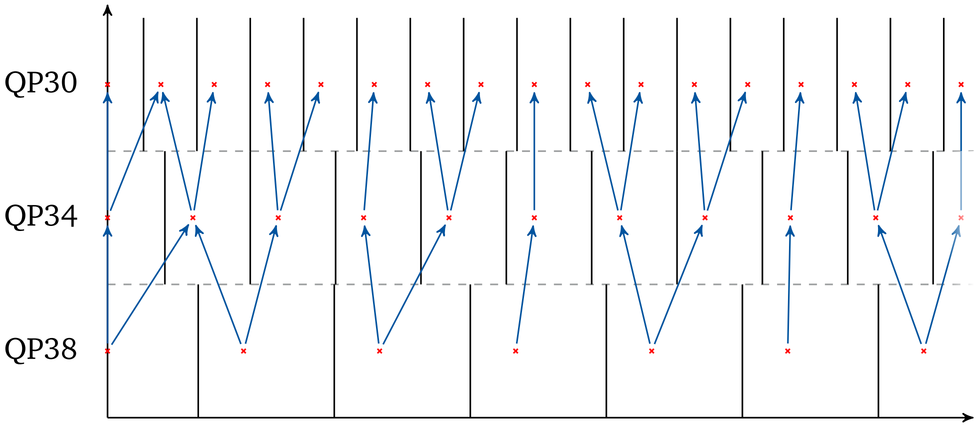


**Figure D.12** Per layer optimized mapping for a higher $\lambda_L$ value and a delta QP value of -2 between the layers. For each layer, mapping is possible from each reconstruction value to one or two reconstruction values in the higher layer. Each coefficient in the highest layer is reachable from exactly one reconstruction value in the lowest layer.

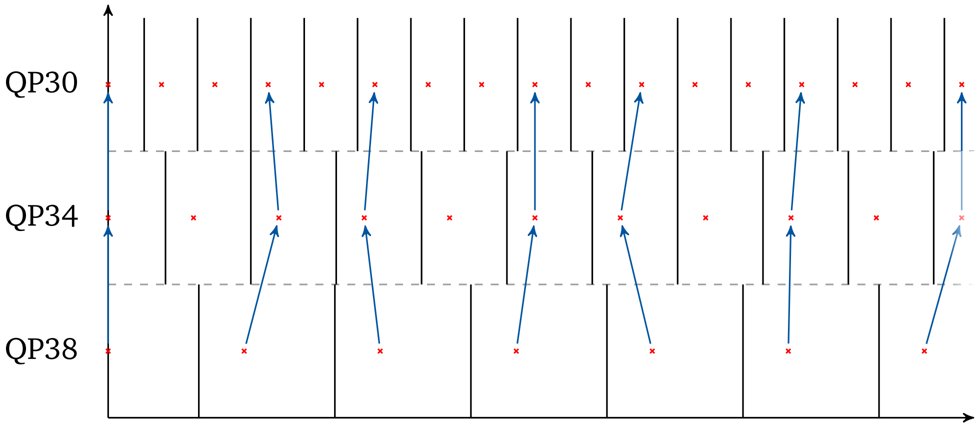


**Figure D.13** Per layer optimized mapping for a high $\lambda_L$ value and a delta QP value of -2 between the layers. No refinement is performed. The reconstruction values are just shifted for every layer and many of the higher layer coefficients can not be reached.

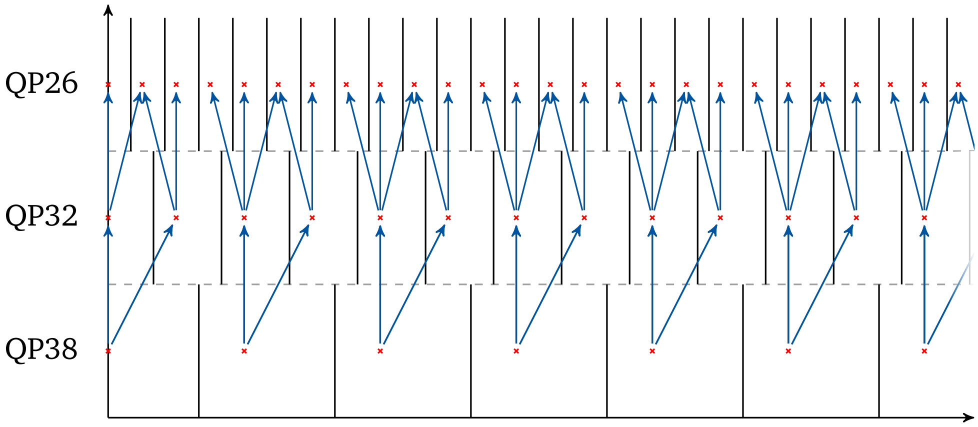


**Figure D.14** Overall optimized mapping for a very low $\lambda_L$ value and a delta QP value of -6 between the layers. Mapping is possible from all reconstruction values in the lowest layer to all reconstruction values in the highest layer. Some mappings through the middle layer are not possible.

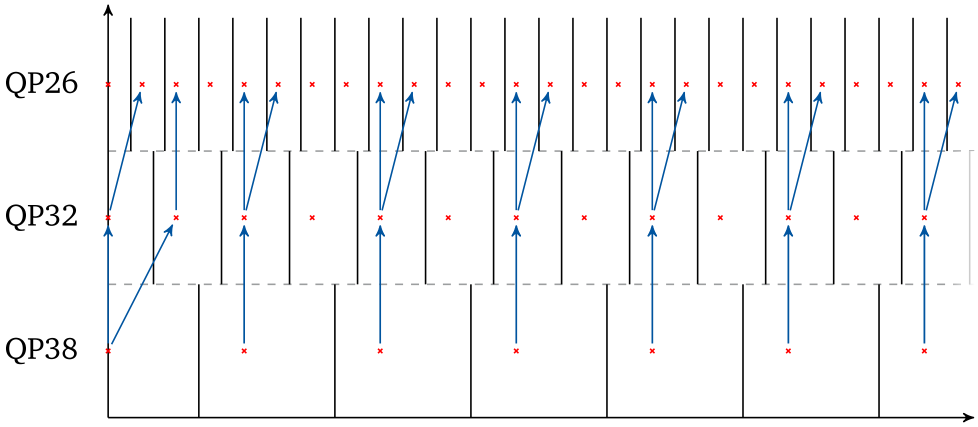


**Figure D.15** Overall optimized mapping for a higher $\lambda_L$ value and a delta QP value of -6 between the layers. Mapping is possible from each reconstruction values in the lowest layer to two reconstruction values in the highest layer. Mapping is always performed through one value in the middle layer.

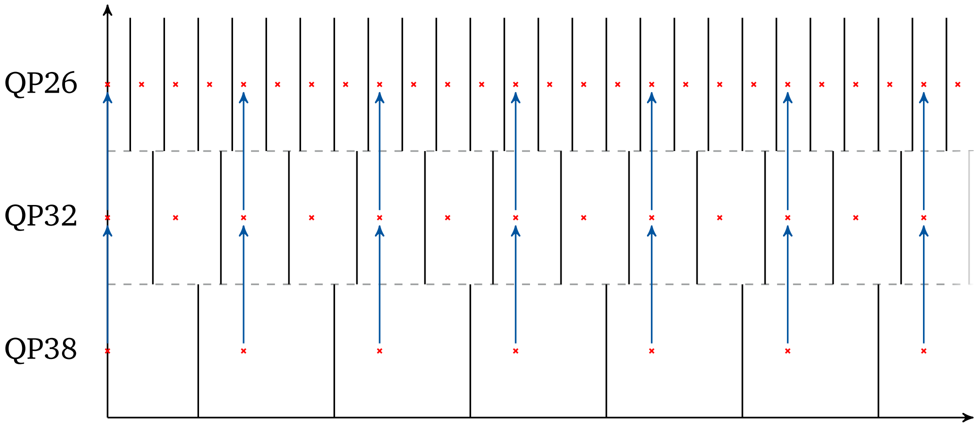


**Figure D.16** Overall optimized mapping for a high $\lambda_L$ value and a delta QP value of -6 between the layers. No refinement is performed and all reconstruction values are identical in all layers. Many of the higher layer reconstruction values can not be reached.

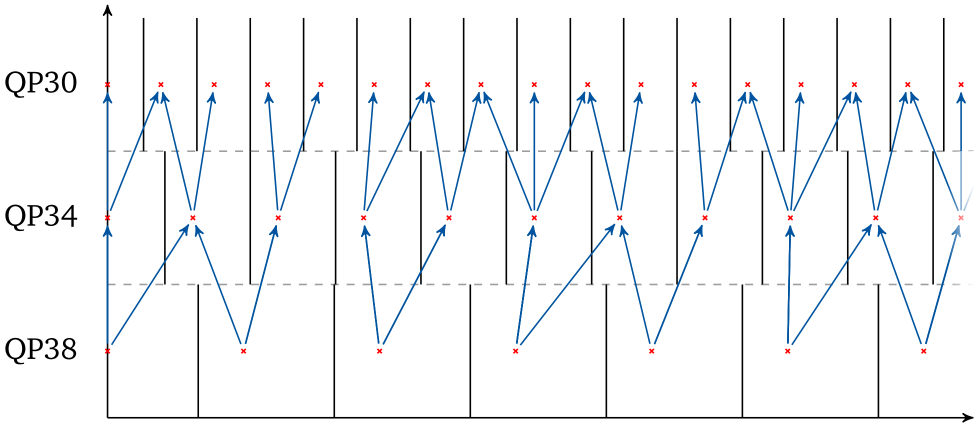


**Figure D.17** Overall optimized mapping for a very low $\lambda_L$ value and a delta QP value of -4 between the layers. Mapping is possible from all reconstruction values in the lowest layer to all reconstruction values in the highest layer. Some mappings through the middle layer are not possible.

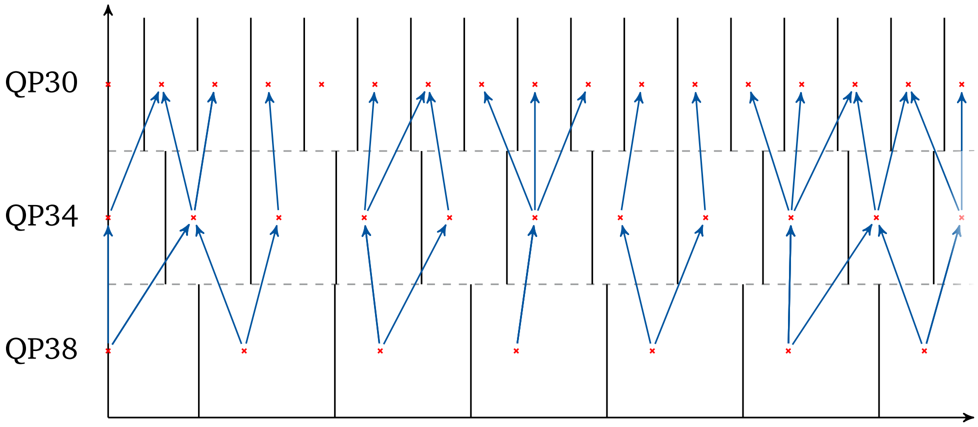


**Figure D.18** Overall optimized mapping for a higher $\lambda_L$ value and a delta QP value of -4 between the layers. Mapping is possible from each reconstruction values in the lowest layer to two or three reconstruction values in the highest layer. For some coefficients mapping is possible through multiple paths in the middle layer.

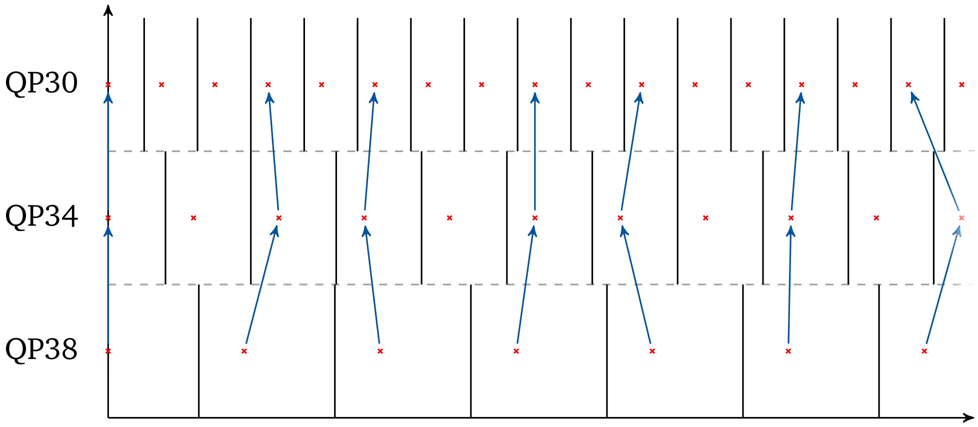


**Figure D.19** Overall optimized mapping for a high $\lambda_L$ value and a delta QP value of -4 between the layers. No refinement is performed. The reconstruction values are just shifted for every layer and many of the higher layer coefficients can not be reached.

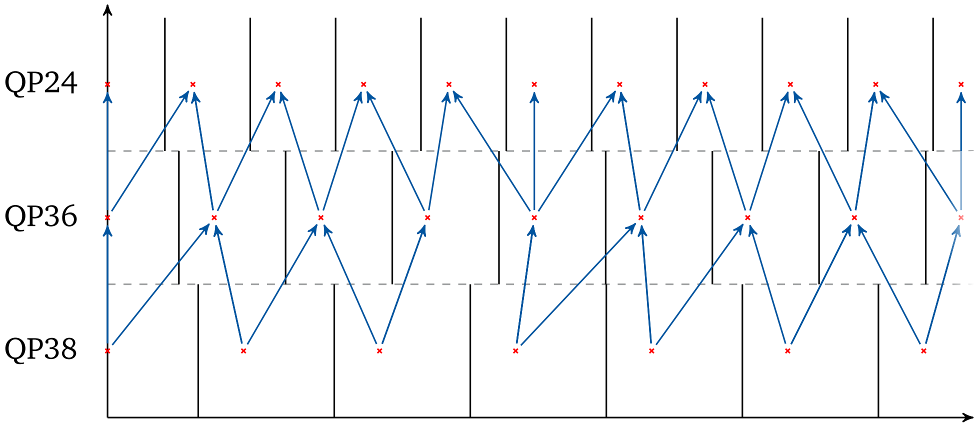


**Figure D.20** Overall optimized mapping for a very low $\lambda_L$ value and a delta QP value of -2 between the layers. Mapping is possible from all reconstruction values in the lowest layer to all reconstruction values in the highest layer. Some mappings through the middle layer are not possible.

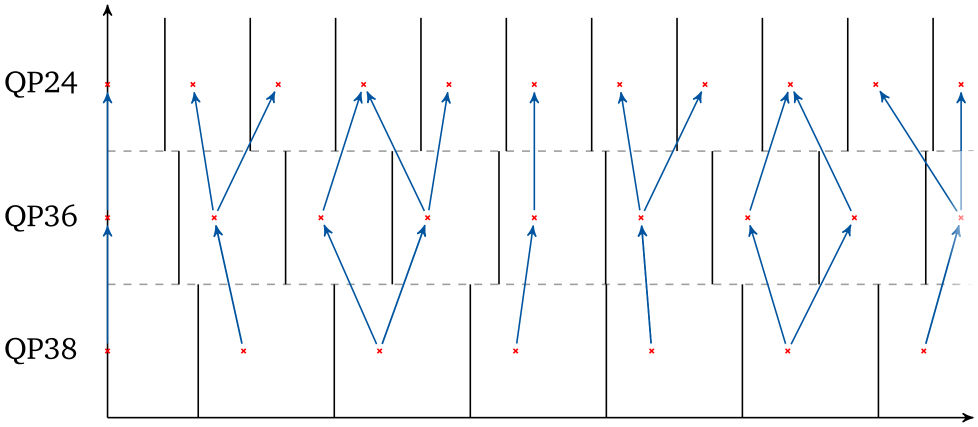


**Figure D.21** Overall optimized mapping for a higher $\lambda_L$ value and a delta QP value of -2 between the layers. Mapping is possible from each reconstruction values in the lowest layer to two or three reconstruction values in the highest layer. For some coefficients mapping is possible through multiple paths in the middle layer.

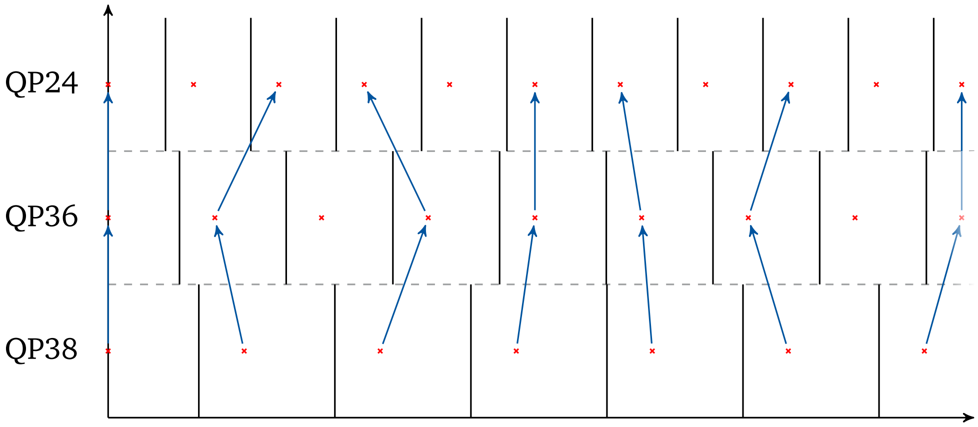
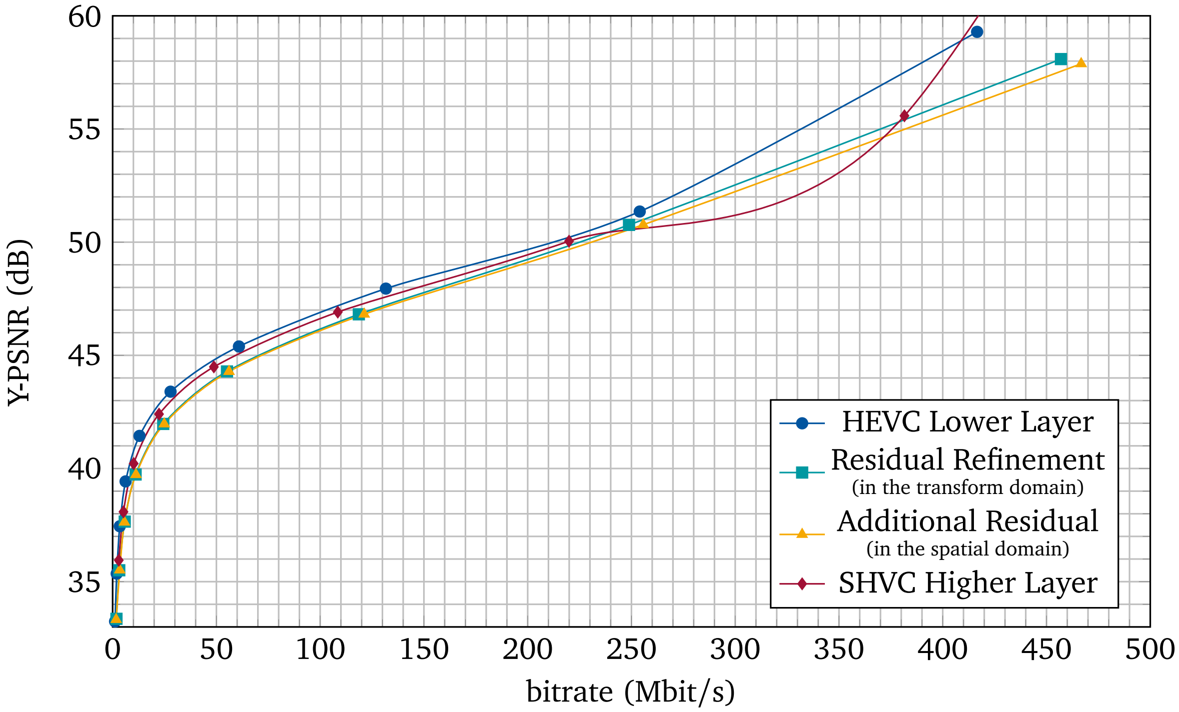
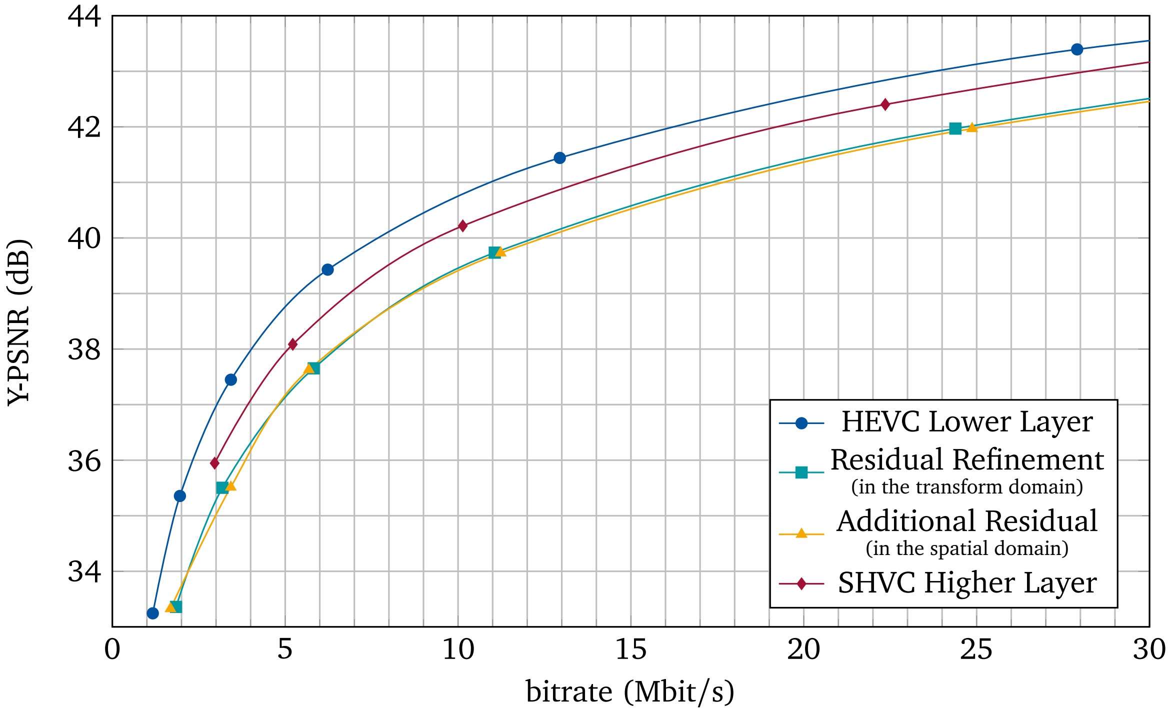


**Figure D.22** Overall optimized mapping for a high $\lambda_L$ value and a delta QP value of -2 between the layers. No refinement is performed. The reconstruction values are just shifted for every layer and many of the higher layer coefficients can not be reached.

## D.3 Additional Coding Performance Results

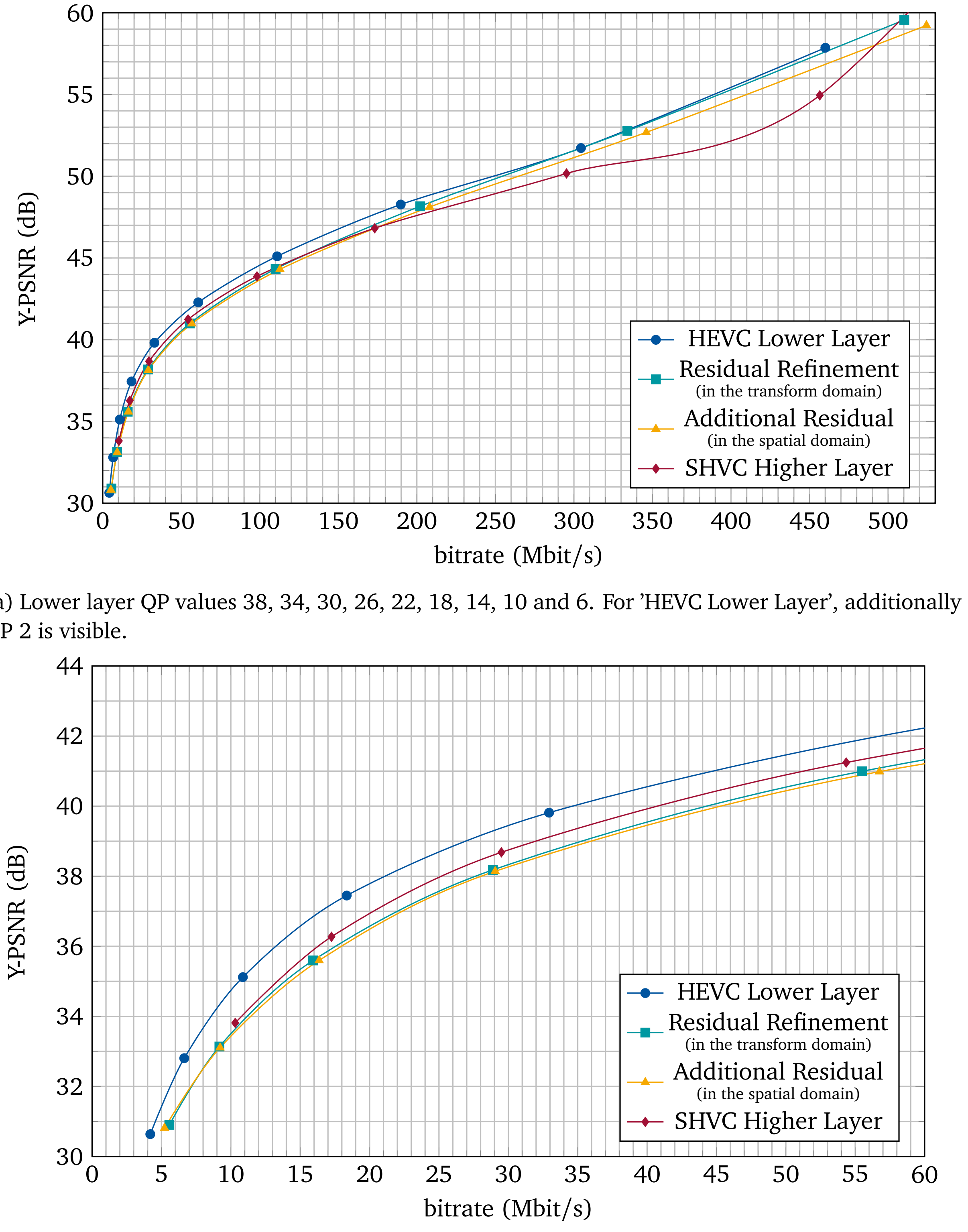


(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For ’HEVC Lower Layer’, additionally QP 2 is visible.



(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22. For ’HEVC Single Layer’ also QP 18 is visible.

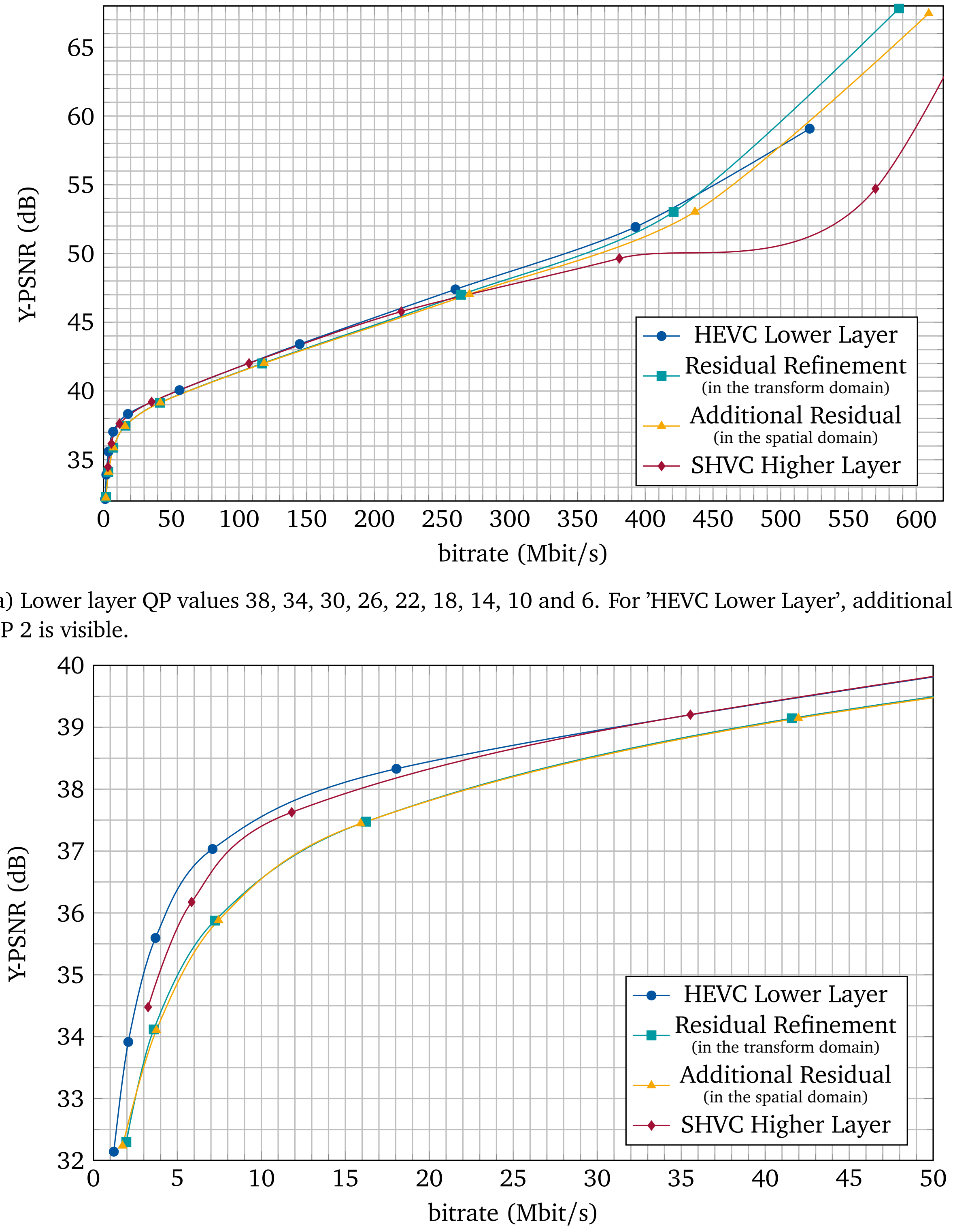
**Figure D.23** Coding performance results of the lower layer for the sequence Traffic using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -6.



(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.

(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

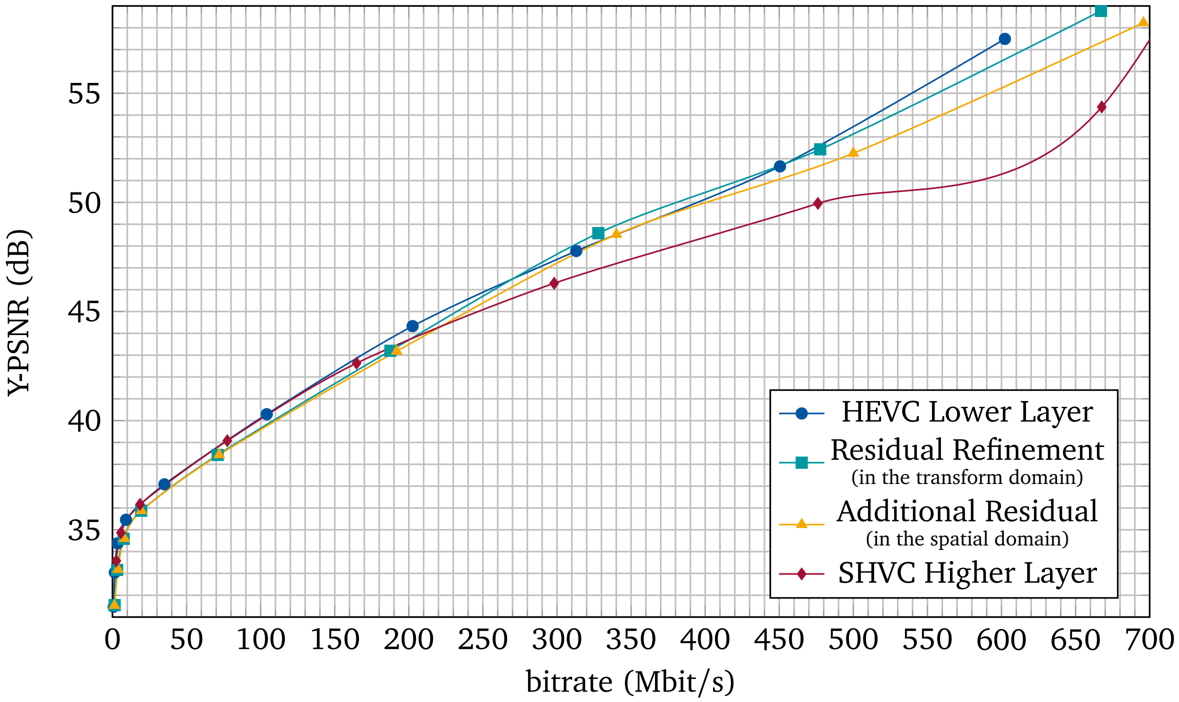
**Figure D.24** Coding performance results of the lower layer for the sequence PeopleOnStreet using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -6.



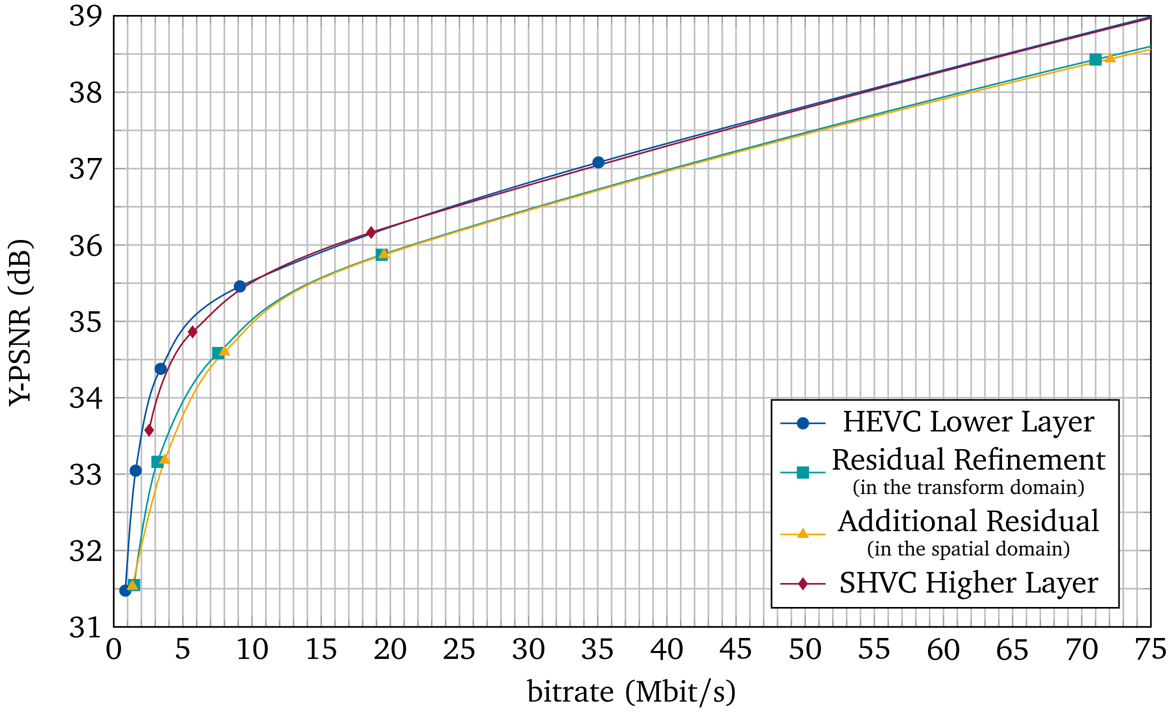
(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.

(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

**Figure D.25** Coding performance results of the lower layer for the sequence Cactus using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -6.

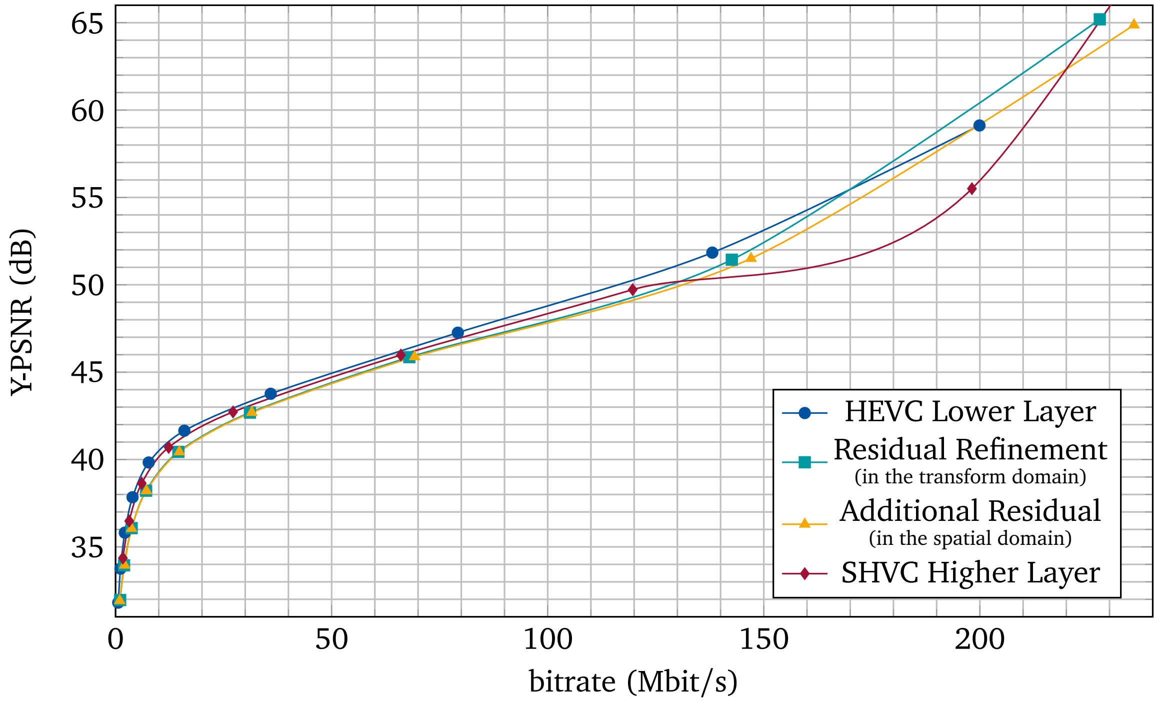


(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.

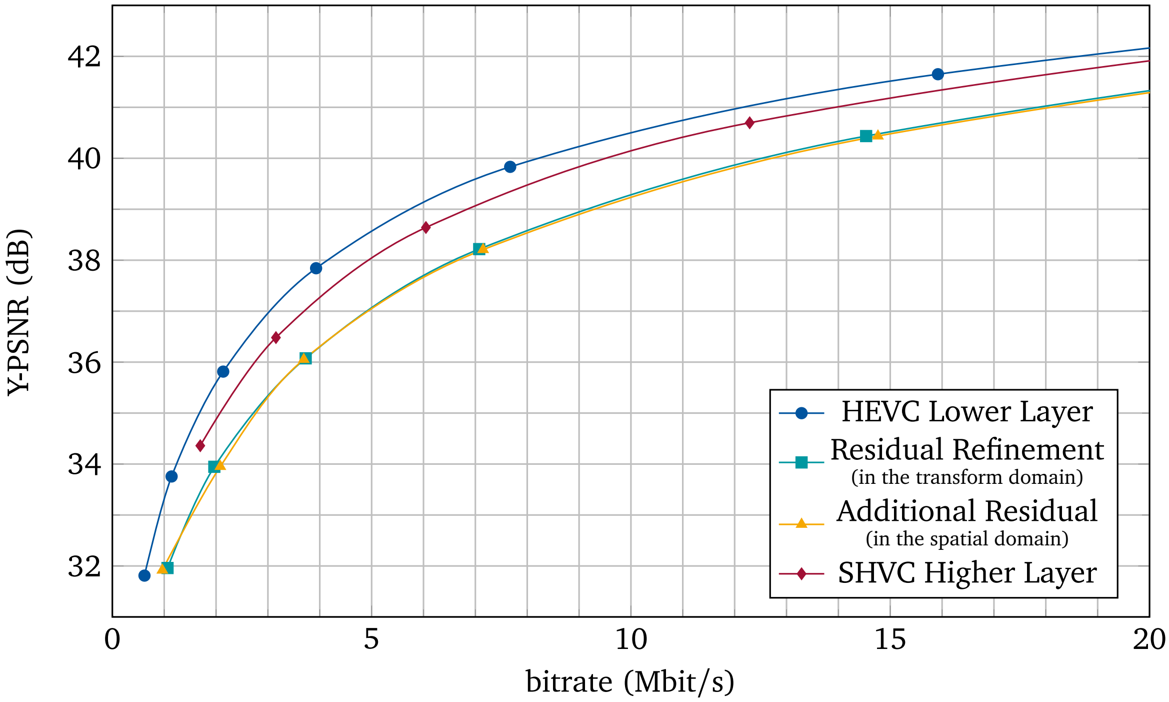


(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

**Figure D.26** Coding performance results of the lower layer for the sequence BQTerrace using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -6.

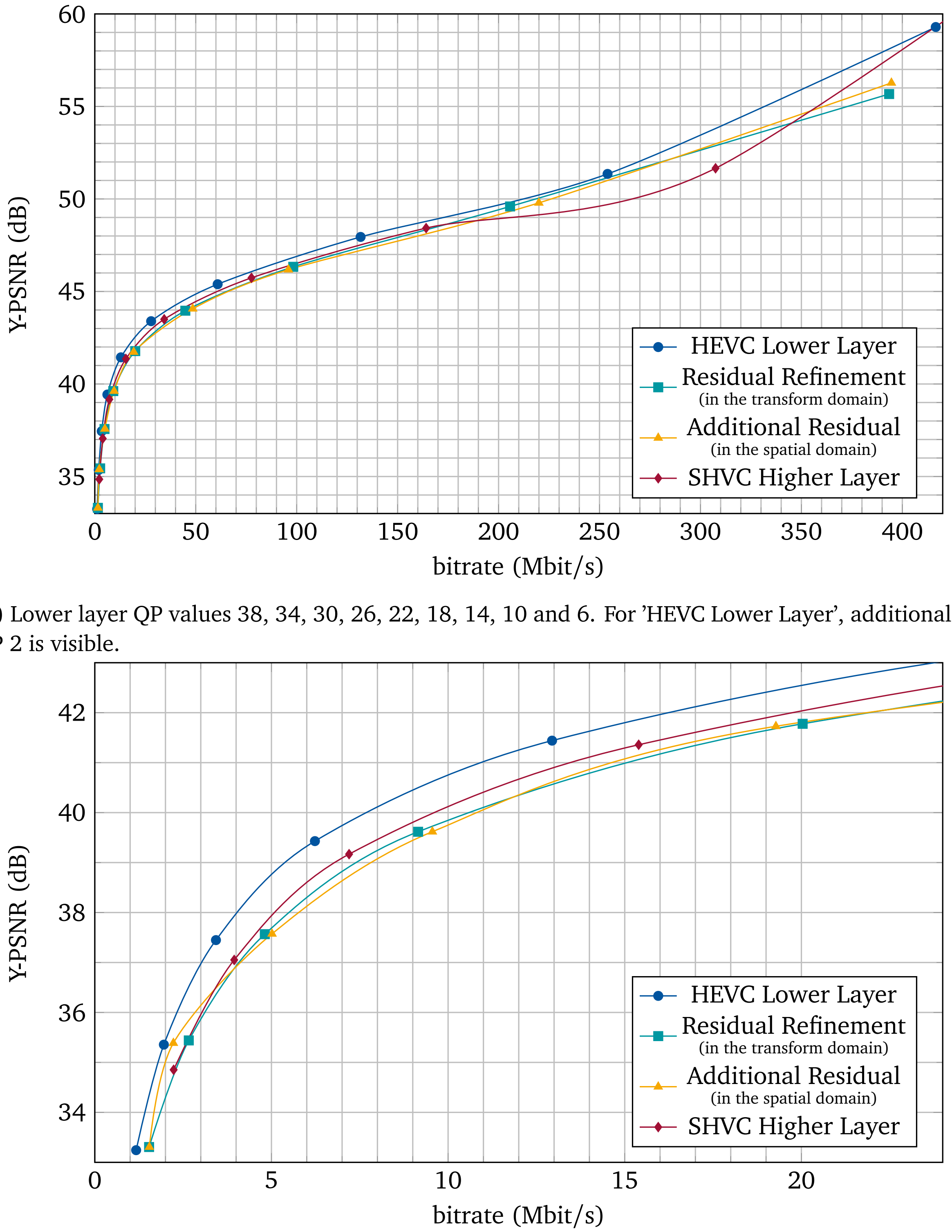


(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.



(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22. For 'HEVC Single Layer' also QP 18 is visible.

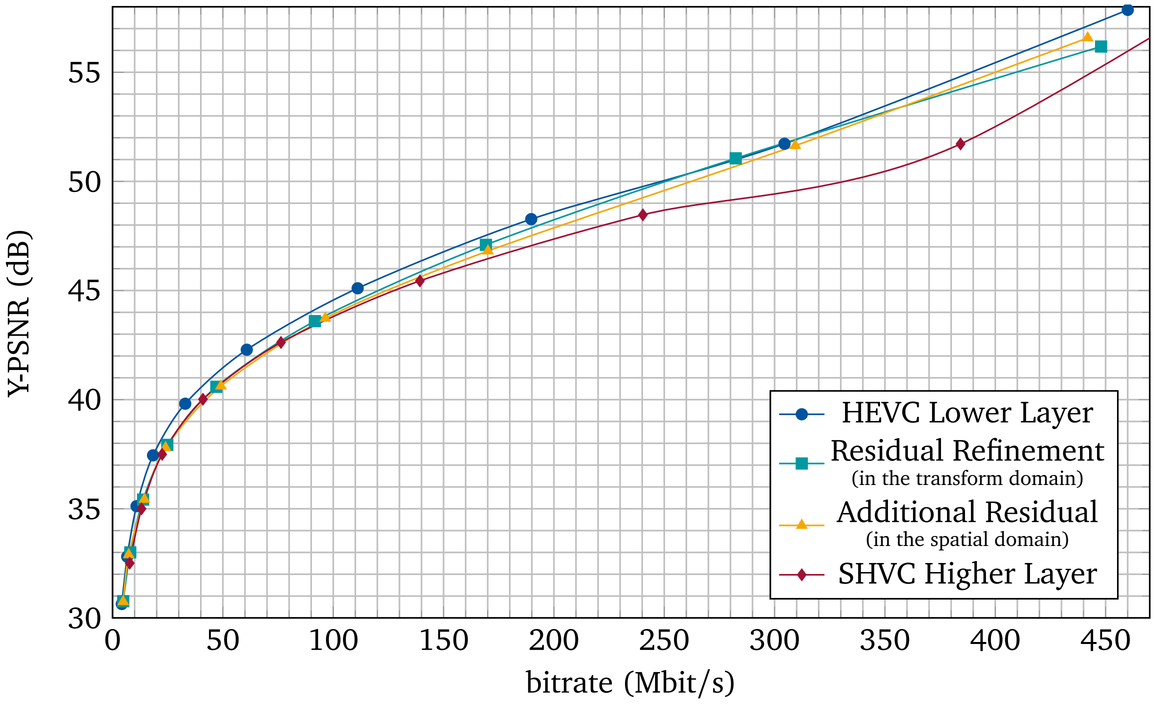
**Figure D.27** Coding performance results of the lower layer for the sequence ParkScene using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -6.



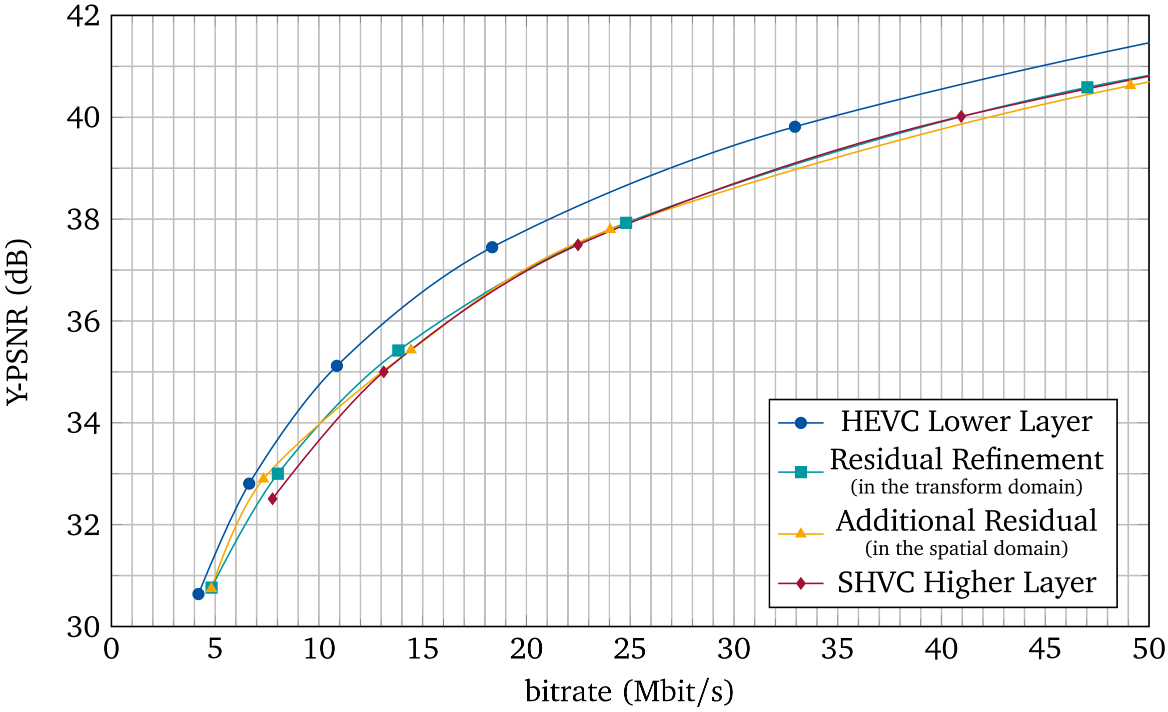
(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.

(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

**Figure D.28** Coding performance results of the lower layer for the sequence Traffic using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -4.

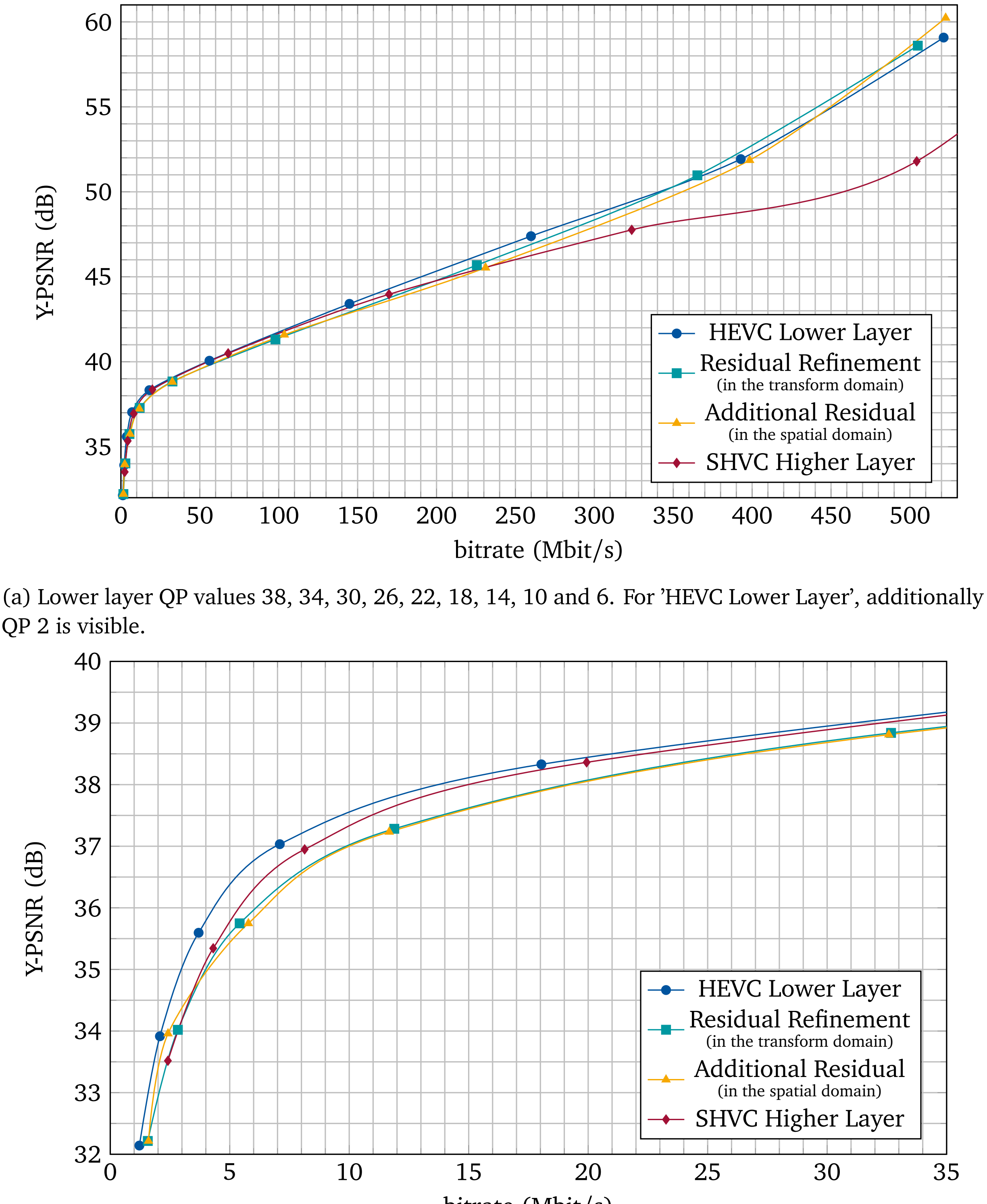


(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.



(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

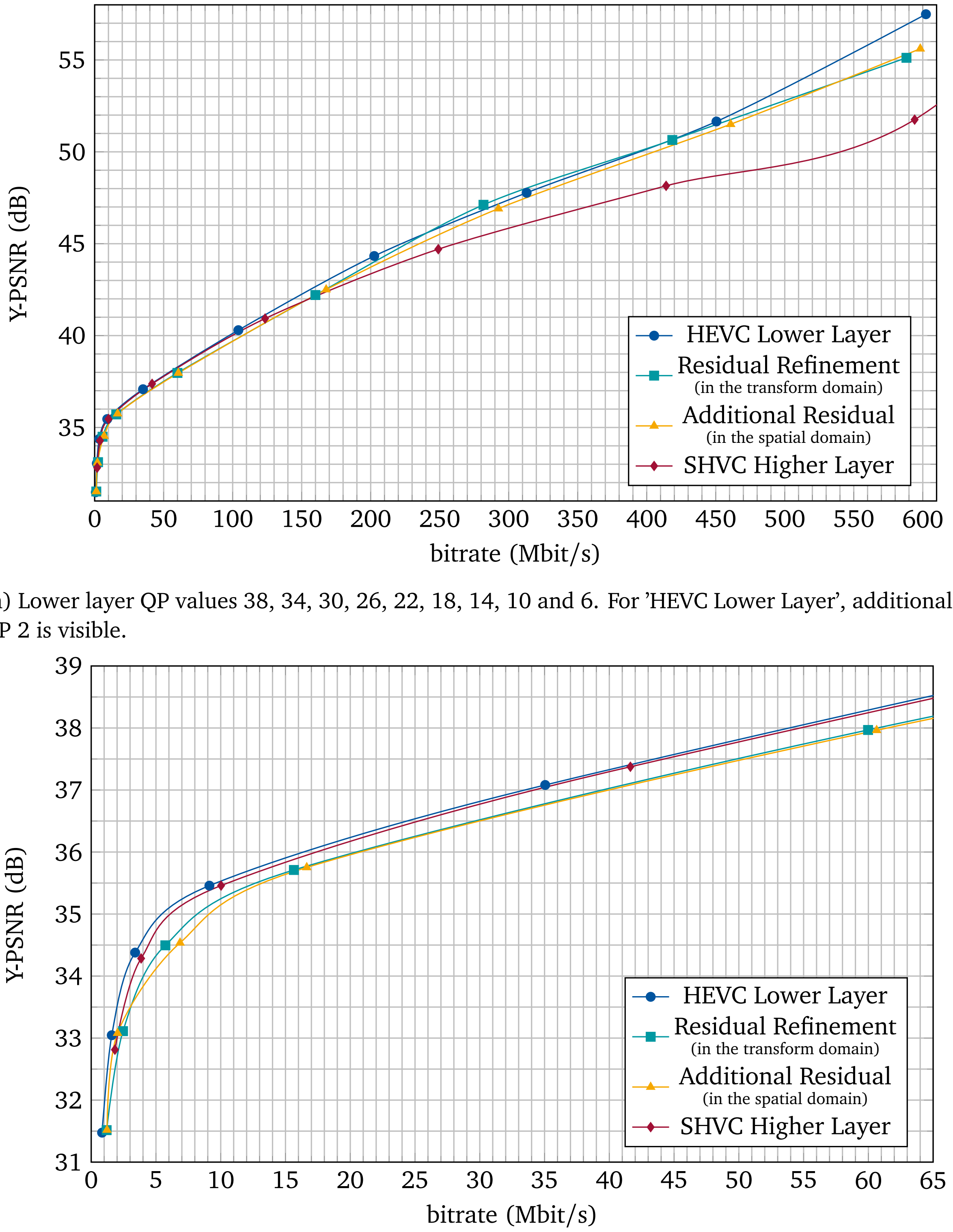
**Figure D.29** Coding performance results of the lower layer for the sequence PeopleOnStreet using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -4.



(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.

(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

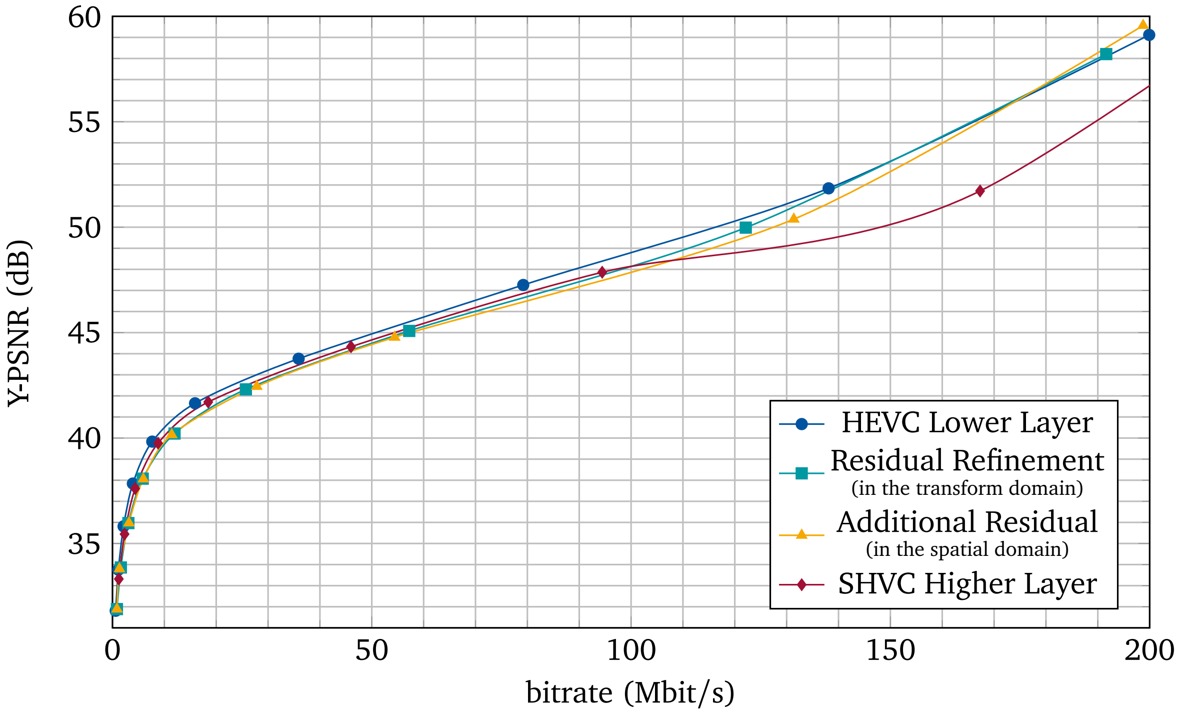
**Figure D.30** Coding performance results of the lower layer for the sequence Cactus using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -4.



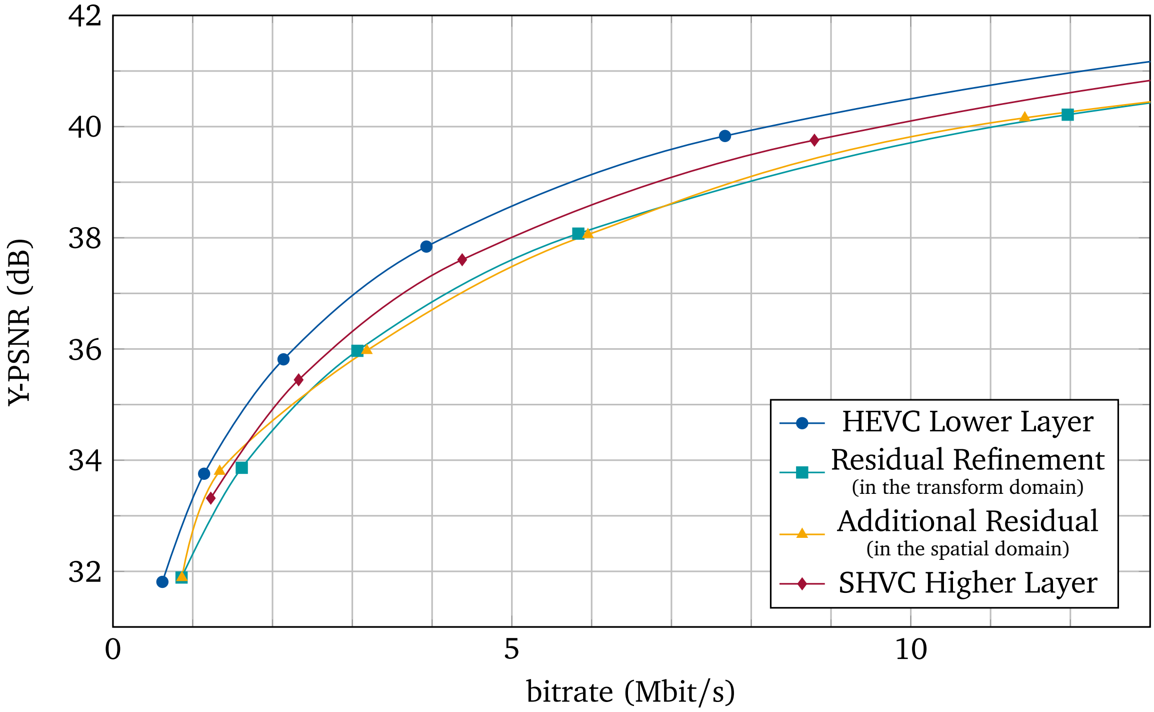
(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.

(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

**Figure D.31** Coding performance results of the lower layer for the sequence BQTerrace using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -4.

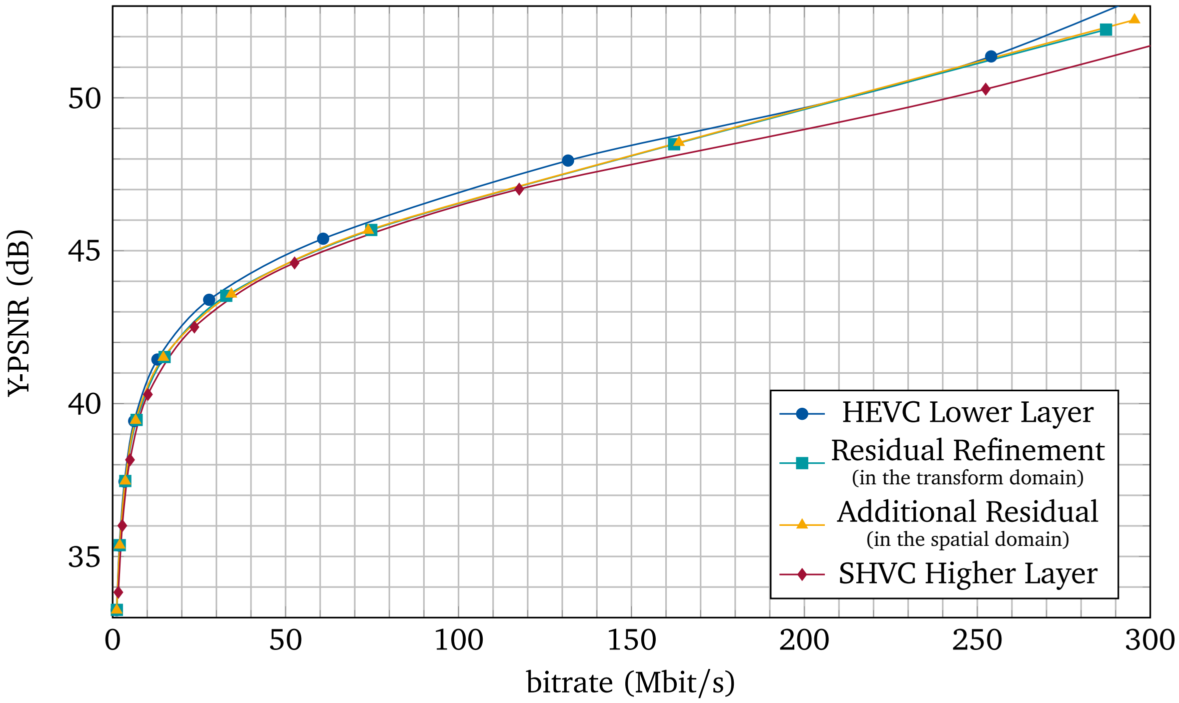


(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.

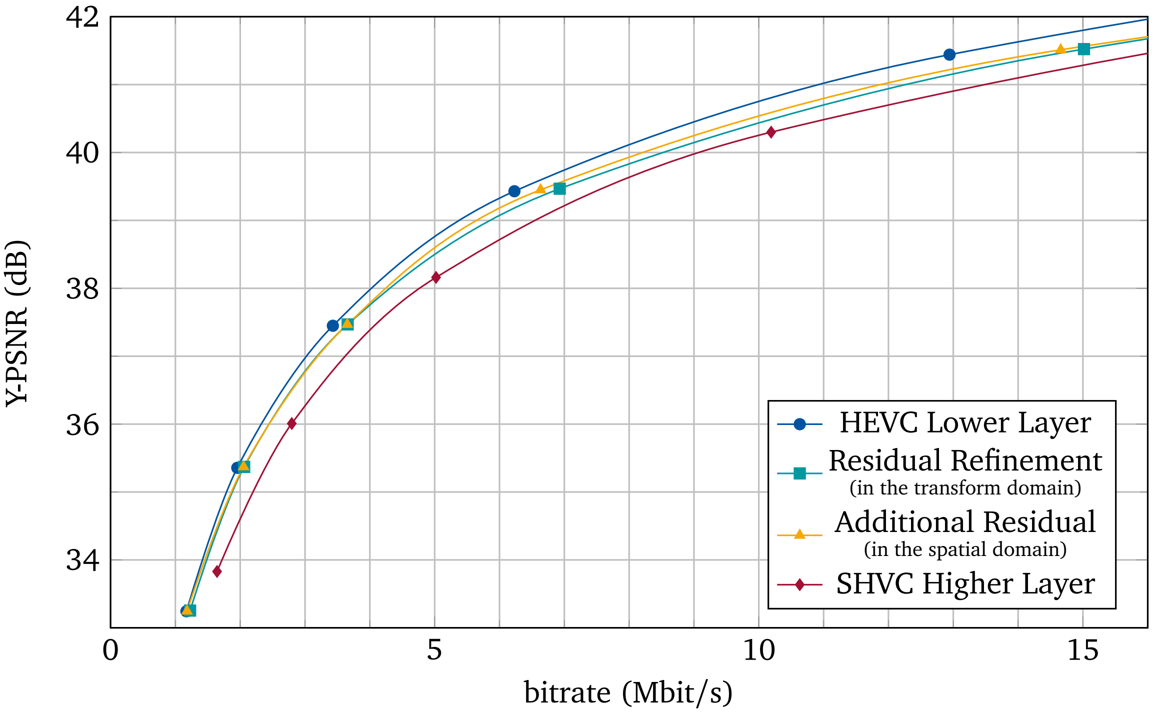


(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

**Figure D.32** Coding performance results of the lower layer for the sequence ParkScene using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -4.

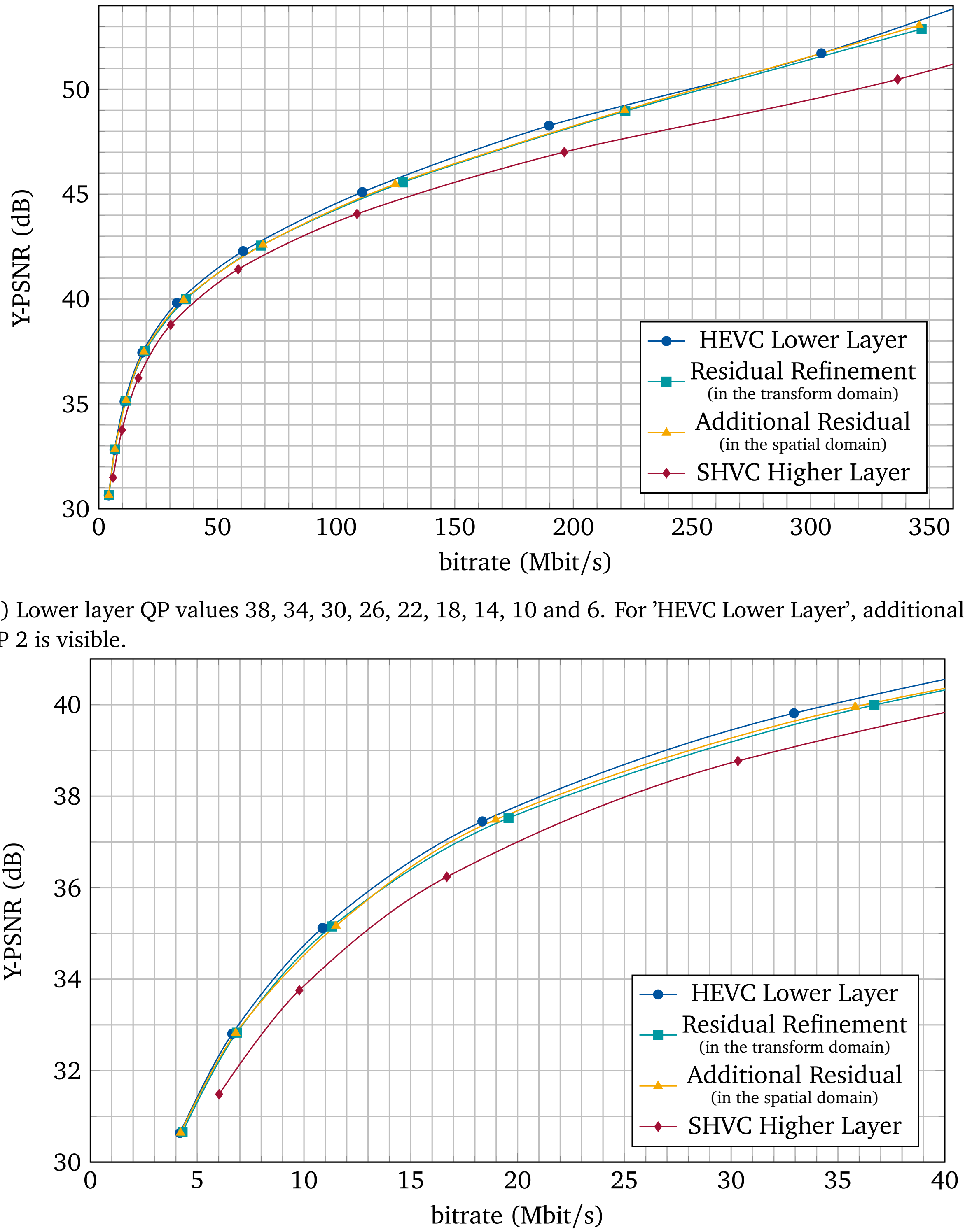


(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.



(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

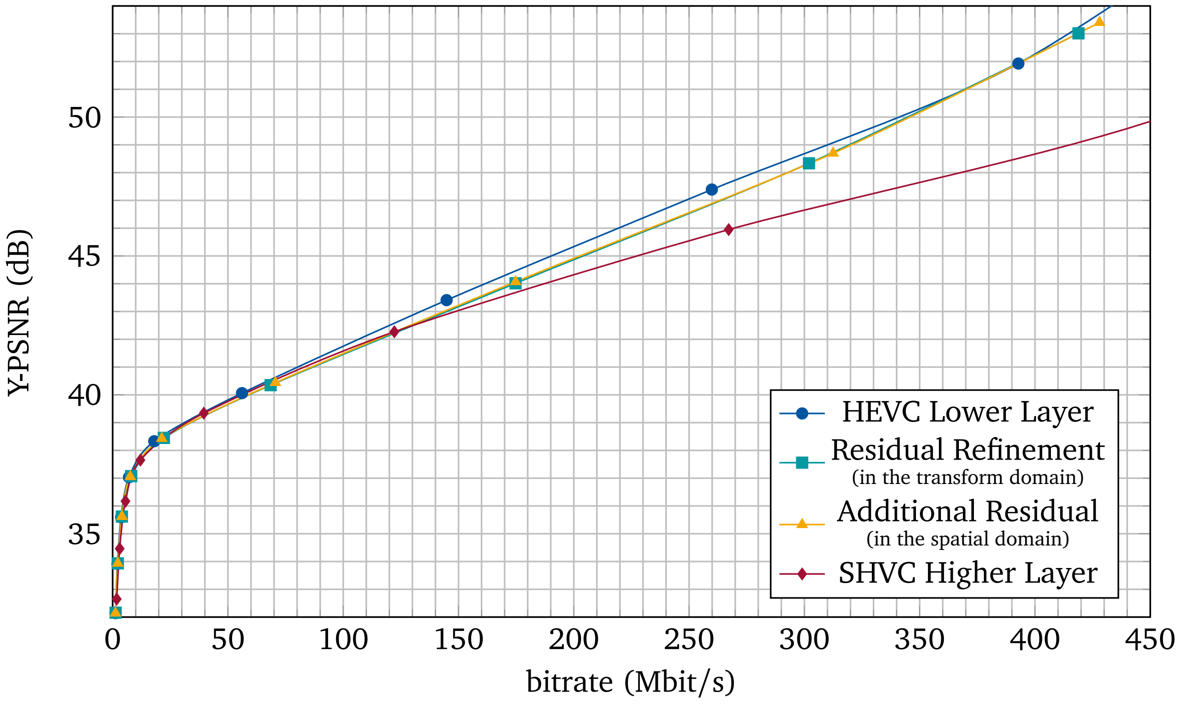
**Figure D.33** Coding performance results of the lower layer for the sequence Traffic using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -2.



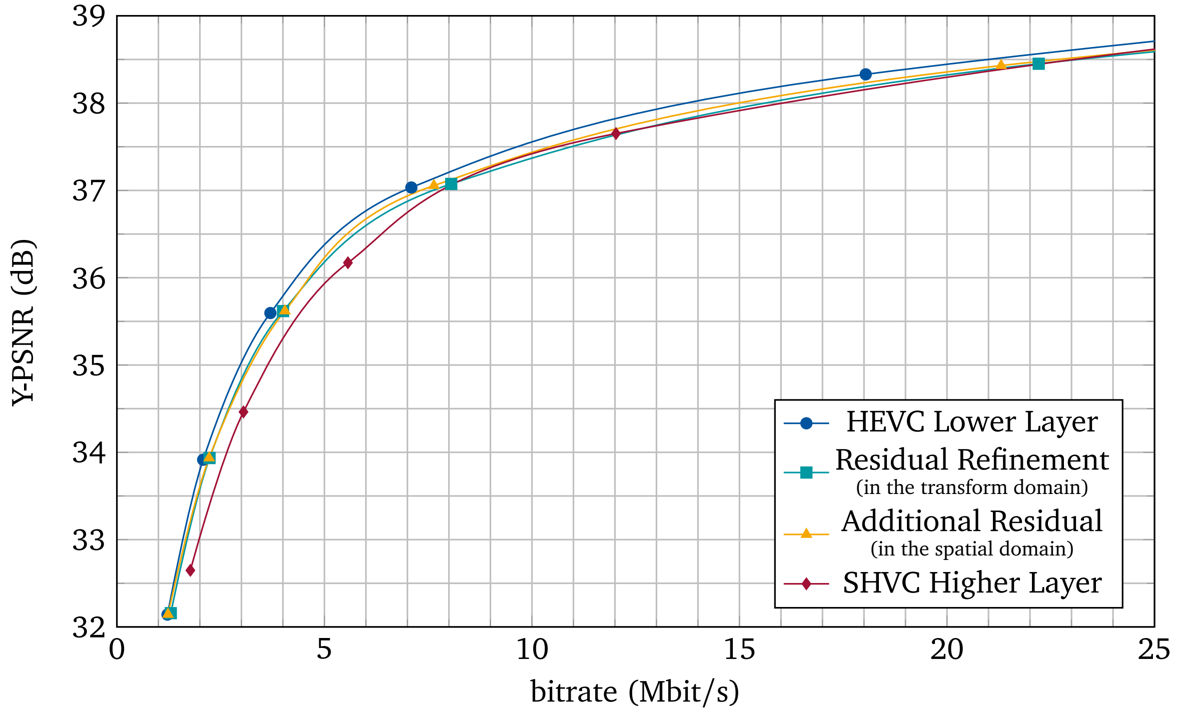
(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.

(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

**Figure D.34** Coding performance results of the lower layer for the sequence PeopleOnStreet using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -2.

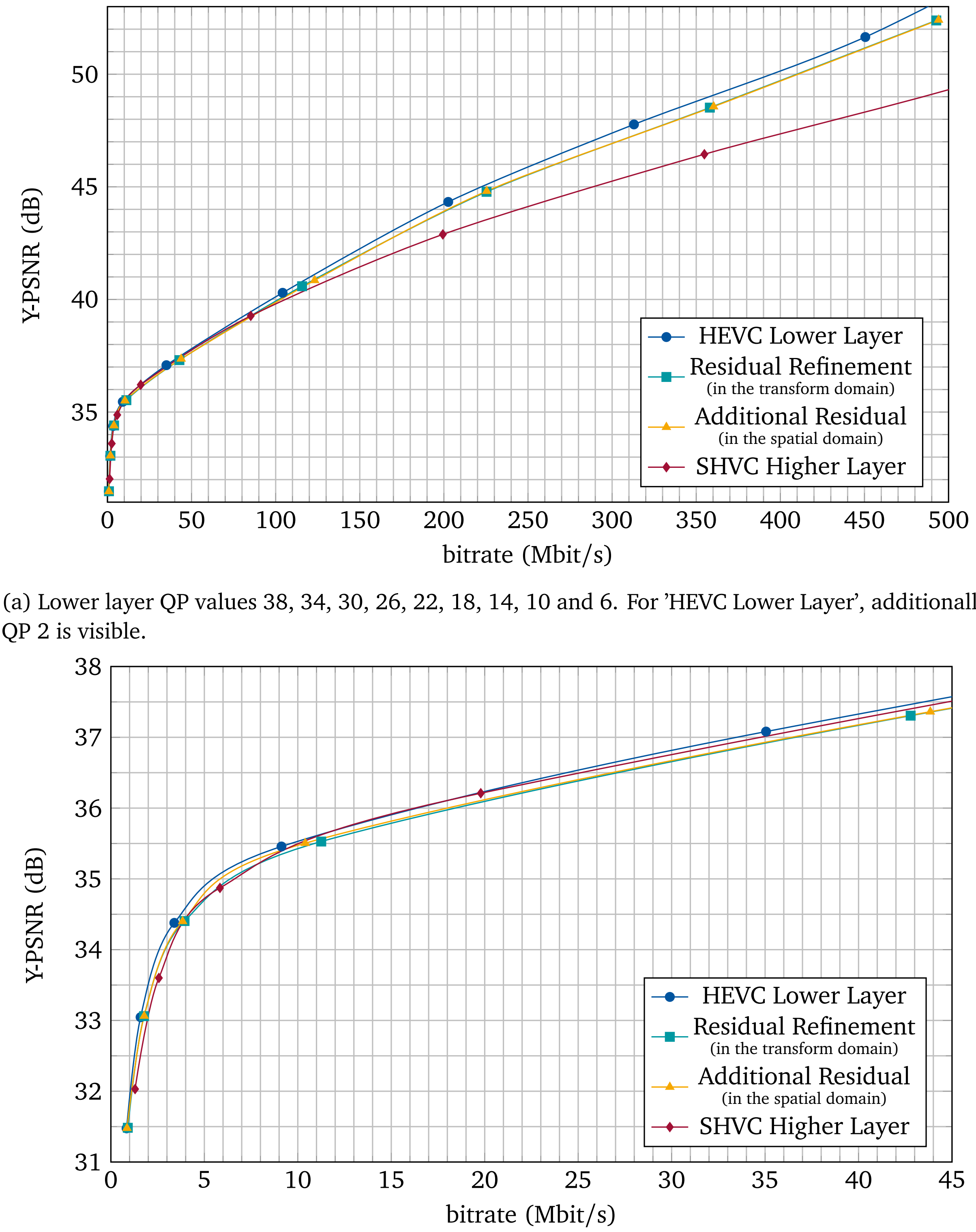


(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.



(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

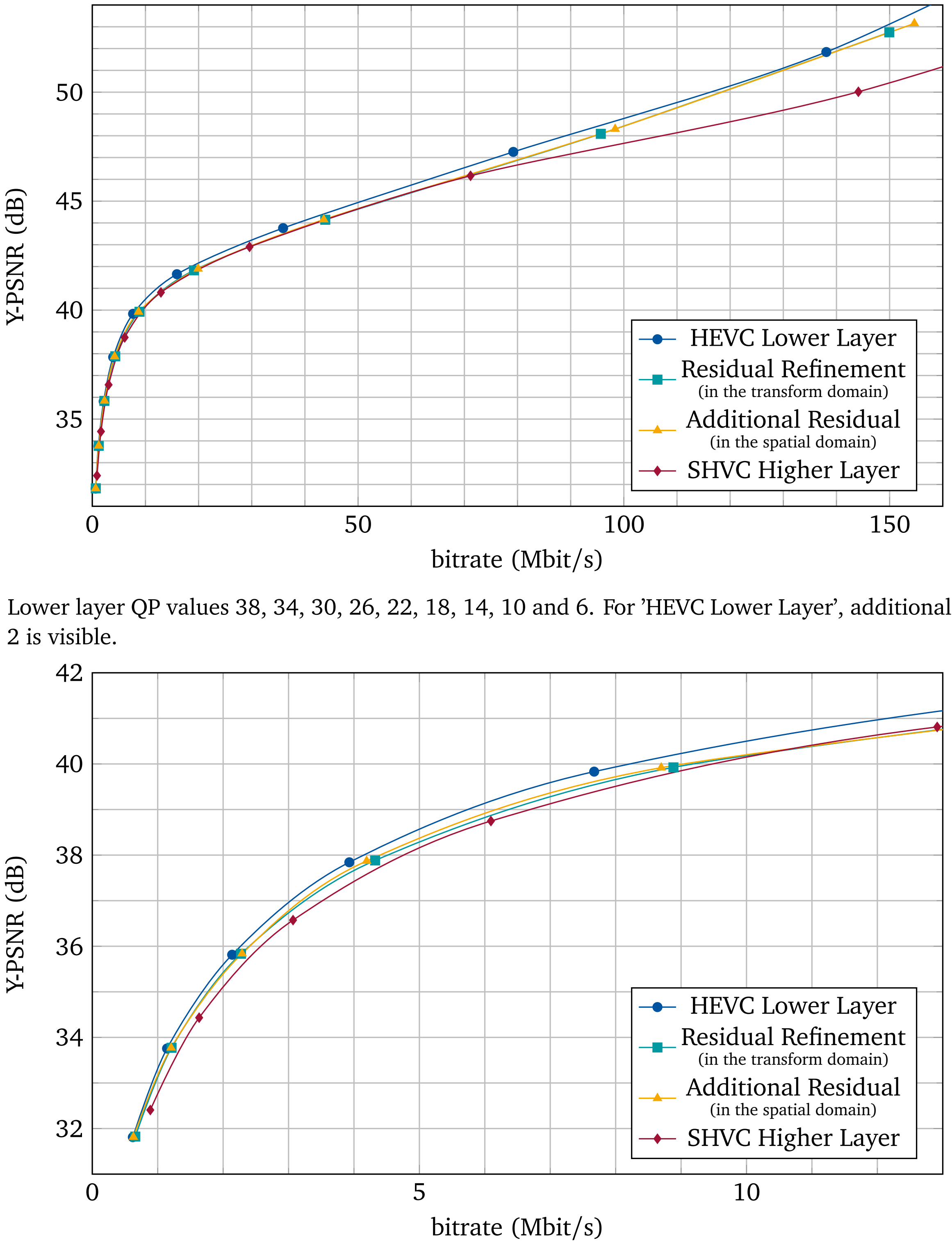
**Figure D.35** Coding performance results of the lower layer for the sequence Cactus using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -2.



(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.

(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

**Figure D.36** Coding performance results of the lower layer for the sequence BQTerrace using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -2.



(a) Lower layer QP values 38, 34, 30, 26, 22, 18, 14, 10 and 6. For 'HEVC Lower Layer', additionally QP 2 is visible.

(b) Detailed view for the lower layer QP values 38, 34, 30, 26 and 22.

**Figure D.37** Coding performance results of the lower layer for the sequence ParkScene using single layer coding and the two coding modes for the higher layer (coding an additional residual signal and refining the existing lower layer residual) for a delta QP of -2.

# E Additional Coding Performance Results for the Combined approach

**Table E.1** BD-rate results for the key picture concept and refinement coding compared to conventional SHVC for the random access configuration and a QP difference of -4.

| | key picture concept | | | combination | | |
|---|---|---|---|---|---|---|
| | Y | U | V | Y | U | V |
| Traffic | -3.13% | -4.74% | -5.45% | -1.67% | -3.80% | -4.64% |
| PeopleOnStreet | -1.23% | -15.10% | -13.94% | 1.12% | -14.37% | -13.23% |
| Kimono | -2.69% | -8.39% | -10.88% | -1.87% | -7.45% | -10.54% |
| ParkScene | -2.26% | -4.86% | -5.71% | -1.11% | -4.37% | -5.57% |
| Cactus | -3.23% | -5.84% | -7.30% | -1.65% | -4.99% | -6.42% |
| BasketballDrive | -0.21% | -7.25% | -6.12% | 1.71% | -6.46% | -4.50% |
| BQTerrace | -1.18% | -7.66% | -7.79% | 0.38% | -7.67% | -7.62% |
| Average | -1.99% | -7.69% | -8.17% | -0.44% | -7.02% | -7.50% |

**Table E.2** BD-rate results for the key picture concept and refinement coding compared to conventional SHVC for the low delay configuration and a QP difference of -4.

| | key picture concept | | | combination | | |
|---|---|---|---|---|---|---|
| | Y | U | V | Y | U | V |
| Traffic | -4.71% | -6.38% | -6.95% | -3.11% | -5.25% | -5.85% |
| PeopleOnStreet | -3.05% | -13.71% | -12.50% | -1.52% | -13.03% | -11.86% |
| Kimono | -4.49% | -8.65% | -10.62% | -3.93% | -7.30% | -9.85% |
| ParkScene | -3.58% | -5.55% | -6.35% | -2.62% | -5.08% | -6.19% |
| Cactus | -5.10% | -7.47% | -8.65% | -3.86% | -6.50% | -7.76% |
| BasketballDrive | -2.15% | -7.75% | -6.88% | -1.05% | -6.89% | -5.17% |
| BQTerrace | -2.09% | -8.17% | -8.22% | -0.82% | -7.99% | -7.73% |
| Average | -3.60% | -8.24% | -8.60% | -2.42% | -7.43% | -7.77% |

# Bibliography

[AK04] Y. Altunbasak and N. Kamaci. “An analysis of the DCT coefficient distribution with the H.264 video coder.” In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Volume 3. May 2004, iii-177-80 vol.3. DOI: 10.1109/ICASSP.2004.1326510 (cited on page 106).

[BAZ13] Philippe Bordes, Pierre Andrivon, and Roshanak Zakizadeh. *AHG14: Color Gamut Scalable Video Coding using 3D LUT*. Doc. JCTVC-M0197. Incheon, KR, 13th meeting: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Apr. 2013 (cited on page 37).

[Bjo01] Gisle Bjontegaard. *Calculation of average PSNR differences between RD-curves*. Doc. VCEG-M33. Austin, Texas: Video Coding Experts Group (VCEG) Meeting (ITU-T SG16 Q.6), Apr. 2001 (cited on page 13).

[Bos+12] F. Bossen, B. Bross, K. Suhring, and D. Flynn. “HEVC Complexity and Implementation Analysis.” In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.12 (Dec. 2012), pages 1685–1696. ISSN: 1051-8215 (cited on pages 23, 28).

[Boy+16] J.M. Boyce, Yan Ye, Jianle Chen, and A.K. Ramasubramonian. “Overview of SHVC: Scalable Extensions of the High Efficiency Video Coding Standard.” In: *Circuits and Systems for Video Technology, IEEE Transactions on* 26.1 (Jan. 2016), pages 20–34. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2015.2461951 (cited on page 37).

[Bud+13] M. Budagavi, A. Fuldseth, G. Bjøntegaard, V. Sze, and M. Sadafale. “Core Transform Design in the High Efficiency Video Coding (HEVC) Standard.” In: *IEEE Journal of Selected Topics in Signal Processing* 7.6 (Dec. 2013), pages 1029–1041. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2013.2270429 (cited on page 21).

[CHJ11] Gordon Clare, Felix Henry, and Joel Jung. *Sign Data Hiding*. Doc. JCTVC-G271. Geneva, CH, 7th meeting: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Nov. 2011 (cited on page 112).

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. 1991 (cited on page 5).

[EF04] A. Eshet and M. Feder. “Multistage quantization via conditional hierarchical mapping.” In: *Proceedings of the 2004 11th IEEE International Conference on Electronics, Circuits and Systems, 2004. ICECS 2004.* Dec. 2004, pages 302–305. DOI: 10.1109/ICECS.2004.1399678 (cited on pages 96, 122).

[Fly+16] D. Flynn, D. Marpe, M. Naccari, T. Nguyen, C. Rosewarne, K. Sharman, J. Sole, and J. Xu. "Overview of the Range Extensions for the HEVC Standard: Tools, Profiles, and Performance." In: *IEEE Transactions on Circuits and Systems for Video Technology* 26.1 (Jan. 2016), pages 4–19. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2015.2478707 (cited on page 15).

[Fra+13] Edouard Francois, Polin Lai, Do-Kyoung Kwon, and Ankur Saxena. *TE3: Summary Report of Tool Experiment on Combined Prediction in SHVC*. Doc. JCTVC-L0023. Geneva, CH, 12th meeting: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Jan. 2013 (cited on pages 44, 54).

[FTA13] Edouard Francois, Ali Tabatabai, and Elena Alshina. *BoG report: Methodoly for evaluating complexity of combined and residual prediction methods in SHVC*. Doc. JCTVC-L0440. Geneva, CH, 12th meeting: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Jan. 2013 (cited on pages 44, 45, 49).

[Fu+12] C. M. Fu, E. Alshina, A. Alshin, Y. W. Huang, C. Y. Chen, C. Y. Tsai, C. W. Hsu, S. M. Lei, J. H. Park, and W. J. Han. "Sample Adaptive Offset in the HEVC Standard." In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.12 (Dec. 2012), pages 1755–1764. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2012.2221529 (cited on page 27).

[h.261] ITU-T Rec. H.261. *Video codec for audiovisual services at p × 64 kbit/s*. URL: https://www.itu.int/rec/T-REC-H.261/en (cited on page 9).

[h.262] ISO/IEC 13818-2 / ITU-T Rec. H.262 (MPEG2). *Information technology – Generic coding of moving pictures and associated audio information: Video*. URL: http://www.itu.int/rec/T-REC-H.262/en (cited on page 9).

[h.263] ITU-T Rec. H.263. *Video coding for low bit rate communication*. URL: https://www.itu.int/rec/T-REC-H.263/en (cited on page 9).

[h.264/AVC] ITU-T Rec. H.264. *Advanced video coding for generic audiovisual services*. URL: https://www.itu.int/rec/T-REC-H.264/en (cited on pages 9, 21, 29, 82).

[h.265/HEVC] ITU-T Rec. H.265. *High Efficiency video coding*. Apr. 2015 (cited on pages 15, 21, 33, 34, 50).

[Hab+15] P. Habermann, C. C. Chi, M. Alvarez-Mesa, and B. Juurlink. "Optimizing HEVC CABAC Decoding with a Context Model Cache and Application-Specific Prefetching." In: *2015 IEEE International Symposium on Multimedia (ISM)*. Dec. 2015, pages 429–434. DOI: 10.1109/ISM.2015.97 (cited on page 111).

[Hab74] A. Habibi. "Hybrid Coding of Pictorial Data." In: *IEEE Transactions on Communications* 22.5 (May 1974), pages 614–624. ISSN: 0090-6778. DOI: 10.1109/TCOM.1974.1092258 (cited on page 9).

[HM-14.0] *JCT-VC HEVC Test Model Software Version 14.0*. Apr. 3, 2014. URL: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-14.0 (cited on page 112).

[HM-16.7] *JCT-VC HEVC Test Model Software Version 16.7*. Oct. 14, 2015. URL: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.7 (cited on pages 43, 92).

[HM-Software] *JCT-VC HEVC Test Model Software Repository*. URL: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware (cited on pages 15, 18, 21).

[ITU-R BT.500] ITU-R BT.500-13. *Methodology for the subjective assessment of the quality of television pictures*. Jan. 2012 (cited on page 72).

[ITU-R BT.710] ITU-R BT.710-4. *Subjective assessment methods for image quality in high-definition television*. Nov. 1998 (cited on page 72).

[ITU-T B.910] ITU-T B.910. *Subjective video quality assessment methods for multimedia applications*. Apr. 2008 (cited on page 72).

[JCTVC-Docs] *JCT-VC Document management system*. URL: http://phenix.int-evry.fr/jct/ (cited on page 54).

[Kir+07] Heiner Kirchhoffer, Detlev Marpe, Heiko Schwarz, and Thomas Wiegand. “A low-complexity approach for increasing the granularity of packet-based fidelity scalability in scalable video coding.” In: *2007 Picture Coding Symposium*. Lisboa, Protugal, Nov. 2007 (cited on page 85).

[KYC08] Marta Karczewicz, Yan Ye, and Insuk Chong. *Rate Distortion Optimized Quantization*. Doc. VCEG-AH21. Antalya, TR, 34th meeting: Video Coding Experts Group (VCEG) of ITU-T, Jan. 2008 (cited on pages 21, 112).

[Lai+12] J. Lainema, F. Bossen, W. J. Han, J. Min, and K. Ugur. “Intra Coding of the HEVC Standard.” In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.12 (Dec. 2012), pages 1792–1801. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2012.2221525 (cited on page 17).

[LG00] E. Y. Lam and J. W. Goodman. “A mathematical analysis of the DCT coefficient distributions for images.” In: *IEEE Transactions on Image Processing* 9.10 (Oct. 2000), pages 1661–1666. ISSN: 1057-7149. DOI: 10.1109/83.869177 (cited on pages 19, 96, 106).

[Mar+02] Michael W. Marcellin, Margaret A. Lepley, Ali Bilgin, Thomas J. Flohr, Troy T. Chinen, and James H. Kasner. “An overview of quantization in JPEG 2000.” In: *Signal Processing: Image Communication* 17.1 (2002). {JPEG} 2000, pages 73–84. ISSN: 0923-5965. DOI: http://dx.doi.org/10.1016/S0923-5965(01)00027-3. URL: http://www.sciencedirect.com/science/article/pii/S0923596501000273 (cited on page 91).

[Mis+13] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou. “An Overview of Tiles in HEVC.” In: *IEEE Journal of Selected Topics in Signal Processing* 7.6 (Dec. 2013), pages 969–977. ISSN: 1932-4553 (cited on page 16).

[MSW03] D. Marpe, H. Schwarz, and T. Wiegand. “Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard.” In: *IEEE Transactions on Circuits and Systems for Video Technology* 13.7 (July 2003), pages 620–636. ISSN: 1051-8215 (cited on page 22).

[MW03] D. Marpe and T. Wiegand. “A highly efficient multiplication-free binary arithmetic coder and its application in video coding.” In: *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*. Volume 2. Sept. 2003, II-263-6 vol.3 (cited on page 22).

[Nor+12] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. Van der Auwera. “HEVC Deblocking Filter.” In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.12 (Dec. 2012), pages 1746–1754. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2012.2223053 (cited on page 27).

[Ohm15] Jens-Rainer Ohm. *Multimedia Signal Coding and Transmission*. 2015 (cited on pages 5, 8, 22, 89, 100).

[Poy12] Charles Poynton. *Digital Video and HD*. Second Edition. The Morgan Kaufmann Series in Computer Graphics. Boston: Morgan Kaufmann, 2012, pages i–iii. ISBN: 978-0-12-391926-7. DOI: 10.1016/B978-0-12-391926-7.50058-8 (cited on page 15).

[RG83] R. Reininger and J. Gibson. “Distributions of the Two-Dimensional DCT Coefficients for Images.” In: *IEEE Transactions on Communications* 31.6 (June 1983), pages 835–839. ISSN: 0090-6778. DOI: 10.1109/TCOM.1983.1095893 (cited on pages 19, 96, 106).

[SB12] V. Sze and M. Budagavi. “High Throughput CABAC Entropy Coding in HEVC.” In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.12 (Dec. 2012), pages 1778–1791. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2012.2221526 (cited on pages 23, 27, 111).

[SBS14] Vivienne Sze, Madhukar Budagavi, and Gary J. Sullivan. *High Efficiency Video Coding (HEVC)*. Springer International Publishing, 2014. ISBN: 978-3-319-06894-7. DOI: 10.1007/978-3-319-06895-4 (cited on page 15).

[Ser+13] Vadim Seregin, Patrice Onno, Shan Liu, Tammy Lee, Chulkeun Kim, Haitao Yang, and Haricharan Laksman. *TE3: Summary Report of Tool Experiment on Combined Prediction in SHVC*. Doc. JCTVC-L0025. Geneva, CH, 12th meeting: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Jan. 2013 (cited on pages 44, 54).

[SF11] Ankur Saxena and Fernandes Felix. *CE7: Mode-dependent DCT/DST without 4*4 full matrix multiplication for intra prediction*. Doc. JCTVC-E125. Geneva, CH, 5th meeting: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Mar. 2011 (cited on pages 18, 46).

[SH14] Vadim Seregin and Yong He. *Common SHM test conditions and software reference configurations*. Doc. JCTVC-Q1009. Valencia, ES, 17th meeting: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Mar. 2014 (cited on pages 37, 41, 45).

[SHM-11] *JCT-VC SHVC Test Model Software Version 11.0*. Jan. 21, 2016. URL: `https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/tags/SHM-11.0/` (cited on pages 41, 43, 45).

[SMW07] H. Schwarz, D. Marpe, and T. Wiegand. "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard." In: *IEEE Transactions on Circuits and Systems for Video Technology* 17.9 (Sept. 2007), pages 1103–1120. ISSN: 1051-8215. DOI: `10.1109/TCSVT.2007.905532` (cited on pages 29, 33, 82).

[Sol+12] J. Sole, R. Joshi, N. Nguyen, T. Ji, M. Karczewicz, G. Clare, F. Henry, and A. Duenas. "Transform Coefficient Coding in HEVC." In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.12 (Dec. 2012), pages 1765–1777. ISSN: 1051-8215. DOI: `10.1109/TCSVT.2012.2223055` (cited on page 27).

[Sul+12] G. J. Sullivan, J. R. Ohm, Woo-Jin Han, and T. Wiegand. "Overview of the High Efficiency Video Coding (HEVC) Standard." In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.12 (Dec. 2012), pages 1649–1668. ISSN: 1051-8215. DOI: `10.1109/TCSVT.2012.2221191` (cited on pages 15, 16, 50).

[Sul96] G. J. Sullivan. "Efficient scalar quantization of exponential and Laplacian random variables." In: *IEEE Transactions on Information Theory* 42.5 (Sept. 1996), pages 1365–1374. ISSN: 0018-9448. DOI: `10.1109/18.532878` (cited on page 8).

[SW08] H. Schwartz and M. Wien. "The Scalable Video Coding Extension of the H.264/AVC Standard [Standards in a Nutshell]." In: *IEEE Signal Processing Magazine* 25.2 (Mar. 2008), pages 135–141. ISSN: 1053-5888. DOI: `10.1109/MSP.2007.914712` (cited on pages 29, 82).

[TM02] David S. Taubman and Michael W. Marcellin. *JPEG2000 Image Compression Fundamentals, Standards and Practice*. 2002 (cited on page 91).

[Ugu+13] K. Ugur, A. Alshin, E. Alshina, F. Bossen, W. J. Han, J. H. Park, and J. Lainema. "Motion Compensated Prediction and Interpolation Filter Design in H.265/HEVC." In: *IEEE Journal of Selected Topics in Signal Processing* 7.6 (Dec. 2013), pages 946–956. ISSN: 1932-4553 (cited on page 17).

[Wie15] Mathias Wien. *High Efficiency Video Coding*. Springer-Verlag Berlin Heidelberg, 2015. ISBN: 978-3-662-44275-3. DOI: `10.1007/978-3-662-44276-0` (cited on pages 15, 16, 27, 50, 112).

[YH05] Wei Yu and Yun He. "A high performance CABAC decoding architecture." In: *IEEE Transactions on Consumer Electronics* 51.4 (Nov. 2005), pages 1352–1359. ISSN: 0098-3063. DOI: `10.1109/TCE.2005.1561867` (cited on page 111).

[Zha14] Bin Zhang. *Robust Video Streaming with H.264/AVC Scalable Video Coding for Wireless Unicast and Multicast*. Aachen Series on Multimedia and Communications Engineering 14. Aachen: Shaker, 2014. ISBN: 978-3-8440-3091-4 (cited on pages 31, 85).

[ZSS08] Jie Zhao, Yeping Su, and Andrew Segall. *On the calculation of PSNR and bit-rate differences for the SVT test data*. Doc. COM16-C404-E. Geneva, CH: International telecommunication union, Study Group 16, Question 6 (ITU-T SH16 Q.6), Apr. 2008 (cited on page 14).