Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl für Medientechnik

Multi-rate video encoding for adaptive HTTP streaming

Dipl.-Ing. Univ. Damien Philippe Schroeder

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Prof. Dr.-Ing. Klaus Diepold
Prüfer der Dissertation:	1.	Prof. Dr.-Ing. Eckehard Steinbach
	2.	Assoc. Prof. Dr. Christian Timmerer

Die Dissertation wurde am 21.11.2016 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 22.02.2017 angenommen.

Abstract

Nowadays, video streaming generates most of the traffic over the internet and efficient video compression techniques such as the recent High Efficiency Video Coding (HEVC) standard are needed to mitigate the stress on the networks. HEVC provides a bitrate reduction around 50% compared to the previous standard H.264, at the cost of an increased encoding complexity. Video streaming is predominantly implemented with the adaptive HTTP streaming paradigm, which requires a video to be encoded at multiple bitrates called representations.

The increased encoding complexity of HEVC combined with the need to encode multiple representations challenges the video providers who see their overall encoding complexity rise. Multi-rate encoding is a promising way to alleviate the overall encoding complexity. A multi-rate encoder directly encodes a video at different representations by using multiple single-layer encoders, and shares information between these encoders in order to reduce the redundancies of encoding the same video multiple times.

This thesis focuses on HEVC-based multi-rate encoding for adaptive HTTP streaming applications. In a first part, a multi-rate encoder that uses encoding information from a high-quality reference encoding to constrain the rate-distortion optimization of lower-quality encodings is considered. Various encoding decisions are used to decrease the overall encoding complexity without harming the compression efficiency. In a second part, the proposed multi-rate encoder is extended to the case where multiple spatial resolutions are required as output representations. Algorithms to extract encoding information from a high-resolution reference encoding are presented, and the information is used to lower the encoding complexity of lower-resolution encodings. The proposed multi-rate methods are combined to form a multi-rate encoder that outperforms a state-of-the-art encoder in terms of rate-distortion performance. The practical case of rate-control-based encoding is considered in the third part, and a method to share content-dependent information is shown to improve the overall rate-distortion performance of the system.

To sum up, the multi-rate methods proposed in this thesis can both reduce the overall video encoding complexity and improve the rate-distortion performance in the practical case of rate-control-based encoding.

Kurzfassung

Heutzutage besteht der größte Anteil des Internetverkehrs aus Video Streaming. Daher werden effiziente Videokompressionsverfahren wie der neue High Efficiency Video Coding (HEVC) Standard benötigt, um die Belastung der Netze zu mindern. HEVC bringt eine Datenratenreduzierung von etwa 50% gegenüber seinem Vorgänger H.264 zu Lasten einer erhöhten Codierungskomplexität. Video Streaming wird außerdem überwiegend als adaptives Streaming über HTTP implementiert, welches die Kompression eines Videos in verschiedenen Bitraten erfordert.

Die erhöhte Codierungskomplexität von HEVC sowie die Notwendigkeit, ein Video mehrmals zu codieren, sind eine Herausforderung für Videoanbieter, deren gesamte Codierungskomplexität steigt. *Multiraten-Codierung* ist eine vielversprechende Möglichkeit, die Codierungskomplexität zu senken. Ein Multiraten-Codierer codiert ein Video in verschiedenen Bitraten mithilfe mehrerer individuellen Encodierer und ermöglicht den Austausch von Codierungsinformationen zwischen den einzelnen Encodierern, so dass die Anzahl redundanter Rechenschritte verringert wird.

Diese Dissertation befasst sich mit HEVC-basierter Multiraten-Codierung für adaptives Streaming über HTTP. Im ersten Teil wird ein Multiraten-Codierer betrachtet, der Codierungsinformationen aus einer Referenzcodierung mit hoher Qualität verwendet, um den Suchraum der Raten-Verzerrungs-Optimierung von Codierungen niedrigerer Qualität einzuschränken. Verschiedene Codierungsentscheidungen werden verwendet, um die gesamte Codierungskomplexität zu verringern, ohne die Kompressionseffizienz zu beeinträchtigen. Im zweiten Teil wird der vorgeschlagene Multiraten-Codierer für den Fall erweitert, dass mehrere räumliche Auflösungen am Ausgang des Systems erforderlich sind. Algorithmen zum Extrahieren von Codierungsinformationen aus einer hochauflösenden Referenzcodierung werden vorgestellt und diese Informationen werden verwendet, um die Codierungskomplexität von Codierungen mit niedrigerer Auflösung zu senken. Die vorgeschlagenen Multiraten-Verfahren werden kombiniert, um einen Multiraten-Codierer zu bilden, der einen Codierer nach dem Stand der Technik in Bezug auf die Kompressionseffizienz übertrifft. Der praktische Fall der bitratensteuerungsbasierten Codierung wird im dritten Teil betrachtet und ein Verfahren zum Austausch inhaltsabhängiger Informationen wird vorgestellt, das die Raten-Verzerrungs-Performanz des Systems verbessert.

Zusammenfassend lässt sich sagen, dass die in dieser Dissertation vorgeschlagenen

Multiraten-Verfahren einerseits die gesamte Videocodierungskomplexität verringern und andererseits die Kompressionseffizienz im praktischen Fall der bitratensteuerungsbasierten Codierung verbessern.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Prof. Eckehard Steinbach for offering me the chance to do my PhD under his supervision at the Chair of Media Technology. I feel lucky to have benefited from his expert advice and vision and I am thankful for the trust he showed me.

I would also like to thank Prof. Christian Timmerer for accepting to serve as my second examiner and Prof. Klaus Diepold for chairing the thesis committee.

Furthermore, I would like to thank all my past and present colleagues at the Chair of Media Technology, including the foreign visitors, for the intelligent discussions, for the help provided, and for the amazing international and friendly environment. Special thanks to Dr. Ali El Essaili from whom I learned a lot and to Dr. Christian Lottermann for the interesting and successful projects together. I also thank Prof. Martin Reisslein from Arizona State University for his valuable advice and his commitment to our collaboration. In addition, I would like to thank the students who worked under my supervision for their contributions, especially Adithyan Ilangovan.

Last but not least, I would also like to thank my family and my friends for their encouragement throughout my PhD time. In particular, I would like to thank my parents for their constant support over the years and for giving me the opportunity to get great education. And many thanks to Julia for continuously supporting me!

Contents

Notation

Abbreviations

Abbreviation	Description	Definition
ANOVA	Analysis of Variance	page 32
AVC	Advanced Video Coding (H.264)	page 5
BD-PSNR	Bjøntegaard delta PSNR	page 13
BD-rate	Bjøntegaard delta rate	page 13
CABAC	Context-Adaptive Binary Arithmetic Coding	page 8
CDN	Content Delivery Network	page 15
CPU	Central Processing Unit	page 15
CTU	Coding Tree Unit	page 6
CU	Coding Unit	page 6
DASH	Dynamic Adaptive Streaming over HTTP	page 16
DCT	Discrete Cosine Transform	page 8
DST	Discrete Sine Transform	page 8
GOP	Group of Pictures	page 10
HD	High Definition	page 5
HEVC	High Efficiency Video Coding	page 5
HTTP	Hypertext Transfer Protocol	page 14
IEC	International Electrotechnical Commission	page 5
ISO	International Organization for Standardization	page 5
ITU	International Telecommunication Union	page 5
JCT-VC	Joint Collaborative Team on Video Coding	page 5
LTE	Long Term Evolution	page 16
MSE	Mean Squared Error	page 13
MV	Motion Vector	page 7
PSNR	Peak Signal-to-Noise Ratio	page 13
PU	Prediction Unit	page 6
QoE	Quality of Experience	page 16
QoS	Quality of Service	page 16
QP	Quantization Parameter	page 8
RD	Rate-Distortion	page 8
RDO	Rate-Distortion Optimization	page 8
RTP	Real-time Transport Protocol	page 14
SA	Spatial Activity	page 24
SNR	Signal-to-Noise Ratio	page 8
TA	Temporal Activity	page 24
TCP	Transmission Control Protocol	page 14

Abbreviation	Description	Definition
TU	Transform Unit	page 6
TV	Television	page 1
TZ	Test Zone	page 11
UDP	User Datagram Protocol	page 14
URL	Uniform Resource Locator	page 15

Symbols

b	number of bits
b_{coded}	number of bits already coded
b_{frame}	number of bits for a frame
$b_{\text{frame},coded}$	number of bits already coded in a frame
b_{GOP}	number of bits for a GOP
$b_{\text{GOP},coded}$	number of bits already coded in a GOP
b_{header}	number of bits in the header (estimated)
$b_{\text{target,frame}}$	target number of bits per frame
D	distortion
d_{avg}	average CU depth
D_{Had}	absolute sum of Hadamard transformed residual
dec	encoding decisions
dec_{opt}	optimal encoding decisions
J	rate-distortion cost
J_{rough}	rough rate-distortion cost approximation
q_{step}	quantization step
QP	quantization parameter
N	number of frames
N_{coded}	number of frames already coded
N_{GOP}	number of frames in a GOP
N_{SW}	number of frames in the sliding window
N_x, N_y	number of pixels horizontally and vertically, respectively
R	bitrate
R_{achieved}	achieved bitrate (when using rate control)
R_{target}	target bitrate (when using rate control)
α, β	video-content-dependent model parameters
δ_α, δ_β	update factors
ΔT	Encoding time difference
ϵ	model error
η	bitrate deviation
$\bar{\eta}$	mean bitrate deviation
θ	threshold parameter
λ	Lagrange multiplier
τ	threshold parameter
ψ	intra mode candidate list
ω	weight
ω_{CTU}	weight of a CTU
ω_{frame}	weight of a frame

Chapter 1

Introduction

The amount of video streaming over the internet has constantly been growing in the past years. This has been driven on one hand by the increased capacity of cellular networks and the advent of powerful smartphones and tablets with high-quality video displaying capabilities, and on the other hand by the continuous development of broadband internet connections in a large number of countries. According to [16], video and audio streaming accounts for more than 70% of North American downlink traffic in fixed access networks during peak evening hours, while this number was around 35% in 2010. In North American wireless networks, it is more than 40% of the peak downlink traffic that is due to video and audio streaming.

This rising volume of video puts pressure on the different networks. Therefore, efficient video codecs are required to compress the video streams. While H.264/AVC has been predominantly used over the past decade, its successor HEVC has been designed, at the cost of a higher encoding complexity, to provide a 50% bitrate reduction at the same perceptual quality [17]. It is therefore expected to gradually replace H.264.

From a streaming perspective, *adaptive HTTP streaming* is now the most widely used paradigm to watch video over the internet, with the largest video providers such as Netflix or Youtube implementing it [18]. Adaptive HTTP streaming provides streaming adaptivity by making the video content available at various bitrates called representations. Furthermore, it benefits from the HTTP protocol which allows to place the control of the streaming at the client side, and which provides reliable transmission of the video data as a result of the underlying TCP protocol [19].

Although adaptive HTTP streaming has mainly been developed for video-on-demand and live streaming over the internet as opposed to classical TV broadcast, these different services have recently started to converge [20]. As an example, adaptive HTTP streaming has been incorporated in the design of the recent ATSC 3.0 broadcast standard [21].

The widespread use of adaptive HTTP streaming means that there is a huge amount of video content that has to be encoded at multiple bitrates. With the multiple representations to encode and the more complex HEVC, the overall encoding complexity is drastically increasing. This is a major challenge for the video providers who see their encoding costs rise.

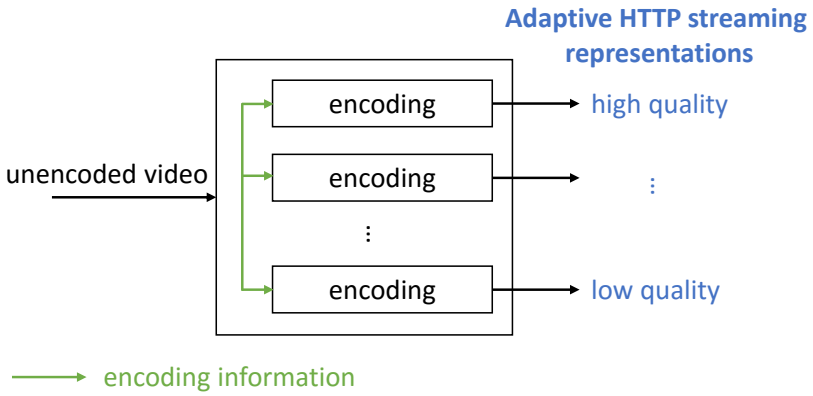


Figure 1.1: General schema of a multi-rate encoder. Encoding information is shared between different single-layer encoders within the multi-rate system. From a single input video, the multi-rate encoder outputs a set of representations at different bitrates and qualities.

One possibility of decreasing the overall encoding complexity is to use a *multi-rate encoder* [22]. A multi-rate encoder encodes a single unencoded video into different independently decodable representations by using multiple single-layer encoders that can share encoding information. The inherent redundancies of encoding the same video at different qualities can thus be reduced, which leads to an overall decrease in encoding complexity. Figure 1.1 shows a schema of a general multi-rate encoder.

Apart from reducing the overall encoding complexity, a multi-rate encoder can be beneficial in terms of rate-distortion performance, when the shared encoding information is used to ameliorate the encoding decisions in each single-layer encoder.

1.1 Main contributions

The topic of this thesis is multi-rate video encoding based on HEVC for adaptive HTTP streaming. The main contributions are as follows.

1. **Rate-distortion-optimization constrained multi-rate encoding:** The similarities of encoding a video at a single spatial resolution but different signal qualities with HEVC are examined and it is observed that the encoding decisions are always slightly differing between different representations. Therefore, it is shown that a multi-rate encoder cannot directly reuse encoding decisions from a reference encoding to a dependent encoding without harming the rate-distortion performance. On the contrary, methods to use the encoding decisions from a high-quality encoding to constrain the rate-distortion optimization of lower quality encodings are proposed. Four encoding decisions (block structure, prediction mode, intra mode, and motion vectors) are considered, and the encoding results show that the proposed multi-rate methods can decrease the encoding complexity without significantly decreasing the rate-distortion performance. The

different methods are combined to form a multi-rate encoder that exhibits a large encoding complexity reduction at the cost of a very low rate-distortion performance loss.

2. **Multi-rate encoding at different spatial resolutions:** The scenario of an adaptive HTTP streaming system with representations at multiple spatial resolutions is considered. The lack of correspondence between block structures of encoded videos at different resolutions when the downsampling factor is not a power of two is identified as the main challenge to apply multi-rate methods to the case of various spatial resolutions. Still, methods to extract information about the block structure, the prediction mode and the intra mode from a high-resolution reference encoding are proposed. This information is used to constrain the rate-distortion optimization of lower-resolution dependent encodings. The encoding results reveal that the encoding complexity can be reduced, again without significantly harming the rate-distortion performance. The combination of the proposed methods leads to a multi-rate encoder that can encode representations both at different spatial resolutions and different signal qualities. Finally, a comparison shows that the proposed multi-rate encoder outperforms a state-of-the-art method in terms of rate-distortion performance.

3. **Improved rate control for multi-rate encoding:** The scenario of a multi-rate encoder using rate control is considered. The existing HEVC rate control algorithm and the underlying rate-distortion model are analyzed. A method to share model parameters between different encodings in a multi-rate system is proposed. Results show that the rate-distortion performance can be improved on average. Furthermore, the proposed method can be combined with a method that constrains the rate-distortion optimization. This leads to a multi-rate encoder that both improves the rate-distortion performance and reduces the encoding complexity, compared to the reference HEVC encoder.

1.2 Organization

The thesis is organized as follows. Chapter 2 provides the background information on HEVC, adaptive HTTP streaming, and multi-rate encoding needed to understand the rest of the thesis. Additionally, the state-of-the-art is reviewed and previous work on multi-rate encoding is presented in details. The settings common to the entire thesis are introduced in Chapter 3. Chapter 4 presents multi-rate methods that constrain the rate-distortion optimization of low-quality encodings based on the information from a high-quality encoding. Specifically, methods to reuse the block structure, the prediction mode, the intra mode and the motion vectors are proposed. In a final step, the various methods are combined, which results in a multi-rate encoder exhibiting a high complexity reduction along with a small decrease of the rate-distortion performance. In Chapter 5, a multi-rate encoder scenario with representations at different spatial resolutions is considered. Methods to reuse information from a high-resolution encoding are proposed and encoding results show that the overall

encoding complexity can be decreased. The methods are combined and compared to a state-of-the-art multi-rate encoder which is outperformed in terms of rate-distortion performance. Chapter 6 considers the case where rate control is applied to the adaptive HTTP streaming representations. A method to pass rate-distortion model information is proposed, which leads to an improved rate-distortion performance. Finally, Chapter 7 concludes the thesis and possible directions for future work are suggested.

Parts of this thesis have been published in [1], [6] and [7].

Chapter 2

Background and related work

This chapter introduces the background information necessary to understand the thesis, and reviews the state-of-the-art work related to the topic of the present thesis. In Section 2.1, the recent video coding standard HEVC is presented along with the main coding steps of the standard. Important video coding concepts such as rate-distortion optimization and video coding metrics are introduced in the context of HEVC. Next, the encoding complexity of HEVC and its reference software HM is inspected and at last, the state-of-the-art HEVC encoding is explored. Section 2.2 examines the adaptive HTTP streaming technology, compares it to other streaming technologies, and reviews the state-of-the-art adaptive HTTP streaming. Finally, the concept of multi-rate encoding is introduced in Section 2.3 along with a detailed presentation of related work.

2.1 HEVC

High Efficiency Video Coding (HEVC) is the latest video coding standard by the International Telecommunication Union (ITU) Video Coding Experts Group and the International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) Moving Picture Experts Group working together as the Joint Collaborative Team on Video Coding (JCT-VC). The standard was approved early 2013 and published both by ITU and ISO/IEC [23], [24].

The standard was designed with the goal of providing 50% bitrate reduction compared with the predecessor H.264/Advanced Video Coding (AVC) at similar perceptual quality [17]. Furthermore, HEVC is especially targeting large spatial resolutions, in order to fit to the now widespread use of High Definition (HD) videos and to cope with the emergence of resolutions larger than HD such as 4K or 8K.

HEVC is a hybrid video codec such as its predecessors since H.261. The encoding is block-based and each block is first either *intra*-predicted (spatial prediction from blocks in the same frame) or *inter*-predicted (temporal prediction from blocks in other frames). After the prediction, the residual signal is transformed with a 2D transformation, the resulting coefficients are then scaled and quantized and finally entropy encoded. Additionally, HEVC

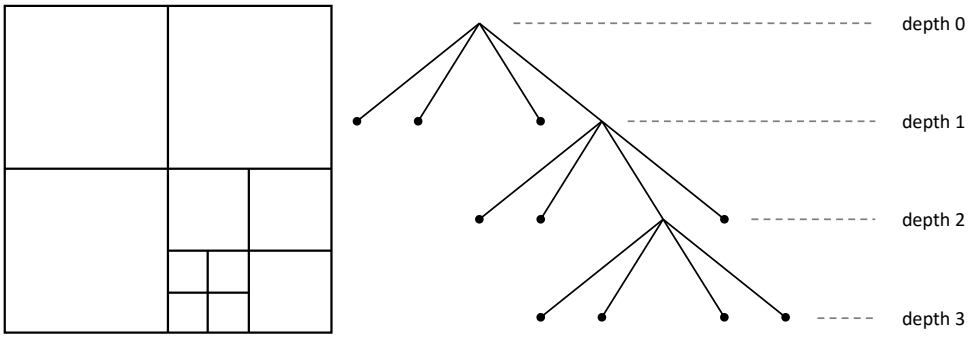


Figure 2.1: Example of a CTU partitioning with corresponding quadtree structure and depth.

allows two optional loop-filters (deblocking filter and sample-adaptive offset filter).

2.1.1 Encoding

This section presents the main components of HEVC. For a complete description of the standard, readers are referred to [25].

2.1.1.1 Block structure

The block structure in HEVC is one of the major novelties compared to the previous H.264/AVC [26]. A frame is first partitioned into basic blocks called Coding Tree Units (CTU). The size of the CTU can be chosen for a video sequence as 8×8, 16×16, 32×32, or 64×64 pixels. Compared to the macroblock size of 16×16 pixels in H.264/AVC, the maximum CTU size of 64×64 pixels allows to better capture spatial correlation in a single block for videos with large spatial resolutions.

The CTU is further partitioned using a quadtree representation into Coding Units (CU), which are the leave nodes of the quadtree. A CU size corresponds to a specific depth in the quadtree. Figure 2.1 shows an example CTU partitioned into multiple CUs. The decision of intra or inter-prediction is taken at CU level.

A CU contains one or more non-overlapping prediction units (PU), which define specific parameters for the prediction. In the case of intra-prediction, two possible PU partitions exist: $2N \times 2N$ and $N \times N$, with $2N$ being the length of a CU side. In the case of inter-prediction, eight possible PU partitions exist (two square partitions as in intra-prediction plus two rectangular partitions and four asymmetric partitions), as illustrated in Figure 2.2.

Finally, a CU contains one or more transform units (TU) as a nested quadtree. A TU is the basic unit for transformation and quantization and is independent of the PU structure in the inter-prediction case. In the intra-prediction case, PUs and TUs are coupled [26].

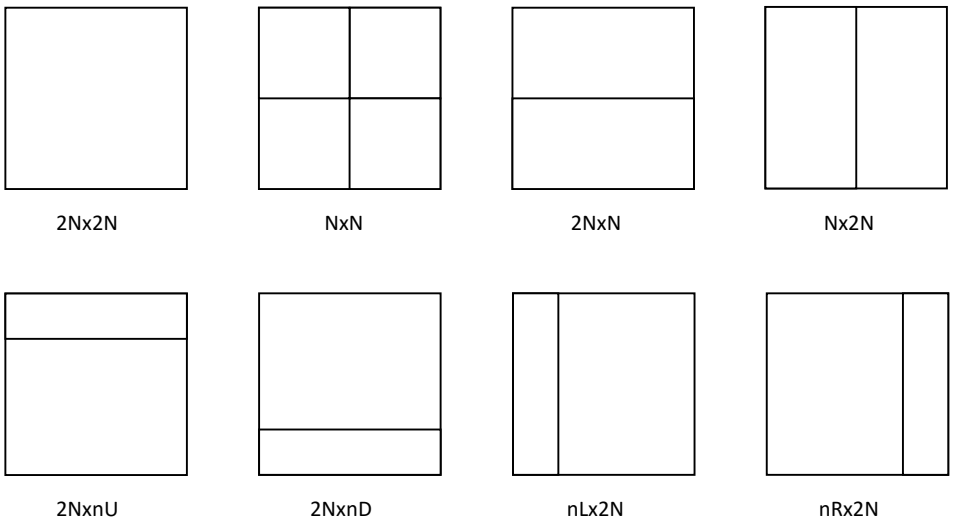


Figure 2.2: Eight PU partition types for HEVC [26].

2.1.1.2 Intra prediction

Intra prediction is the spatial prediction of PUs from already coded blocks in the same frame. HEVC defines 33 angular prediction modes, compared to 8 angular prediction modes in H.264/AVC [27]. Additionally, HEVC has a DC prediction mode and a planar prediction mode, which sums up to 35 intra prediction modes. The intra PU size can be between 4×4 and 64×64 pixels, and has to be the size of the parent CU, unless the PU size is 4×4.

Samples from the PU are predicted using the intra prediction mode and the reference samples from the reconstructed blocks left, above, above-right and optionally below-left. The reference samples are interpolated linearly with 1/32 pixel accuracy.

HEVC defines three *most probable modes* which are the intra prediction modes of the PUs left and above and a third mode assigned as planar, DC, or angular (vertical) in this order. These most probable modes can be coded with a 1 bit flag and a 2 bit index indicating the element of the most probable mode array. The remaining 32 intra prediction modes are coded with a fixed length coding of 5 bit.

Video frames entirely encoded with intra prediction are independently decodable as they do not rely on other frames and are called *I-frames*.

2.1.1.3 Inter prediction

Inter prediction is the temporal prediction of a PU from already coded blocks in other frames. The displacement to the predictor block is given by a motion vector (MV), which is associated to an index referring to the reference frame containing the predictor block. There can be either one prediction direction (simple prediction) or two predictions (bi-directional prediction), which allows for example to predict from multiple reference frames. Each prediction can use a temporally preceding or a temporally following reference frame. A frame using

simple prediction is called *P-frame* and a frame using bi-directional prediction is called *B-frame*.

Inter prediction is performed with a quarter-pixel accuracy in HEVC. An eight-tap filter is used to interpolate the half-sample positions and a seven-tap filter is then used to interpolate the quarter-sample positions in the reference frame [17].

2.1.1.4 Transform, quantization and entropy encoding

After the prediction, the prediction error residual in a TU is transformed using a 2D transform of size 4×4, 8×8, 16×16 or 32×32. HEVC defines an integer transform matrix of size 32×32 that approximates a 2D Discrete Cosine Transform (DCT). The transform matrix can be downsampled to accommodate for the other transform sizes. When the TU is of size 4×4, an alternative transform based on a Discrete Sine Transform (DST) is used.

The transform coefficients are quantized by division by a quantization step q_{step} [28]. The quantization step is determined based on the quantization parameter (QP) as follows:

$$q_{\text{step}} = \left(2^{1/6}\right)^{\text{QP}-4} \tag{2.1}$$

The QP is an integer that can take a value between 0 in 51 in HEVC, and can be used as input parameter to the encoder to determine the quality of the encoded output video. Quantization scaling matrices can be used to differentiate the amount of quantization of the different transform coefficients. This allows for example to adapt the quantization to the properties of the human visual system, e.g., by applying a stronger quantization to high-frequency coefficients, as the human visual system is less sensitive to high-frequency components. The quantization is a lossy process and the distortion introduced by the quantization is irreversible.

The quantized coefficients are finally entropy encoded using context-adaptive binary arithmetic coding (CABAC) [29].

2.1.2 Rate-distortion optimization

A lossy compression introduces distortion in the encoded signal. The original signal is said to be altered in the Signal-to-Noise Ratio (SNR) domain. There is a fundamental tradeoff between the distortion D and the bitrate R resulting of the compression. Classic optimization problems can be to minimize the distortion subject to a constrained bitrate, or inversely minimizing the bitrate subject to a constrained distortion. These optimization problems can be reformulated as an unconstrained optimization problem called *rate-distortion optimization* (RDO) as follows:

$$\min J = D + \lambda R \tag{2.2}$$

where λ is a Lagrange multiplier and J is called the rate-distortion (RD) cost. Figure 2.3 shows a typical rate-distortion curve $D(R)$ along with constant J lines, which have a slope of $-\lambda$. To achieve an optimal RD performance, the value of λ should be small at a high bitrate/low distortion and should be high for a low bitrate/high distortion. Generally, the

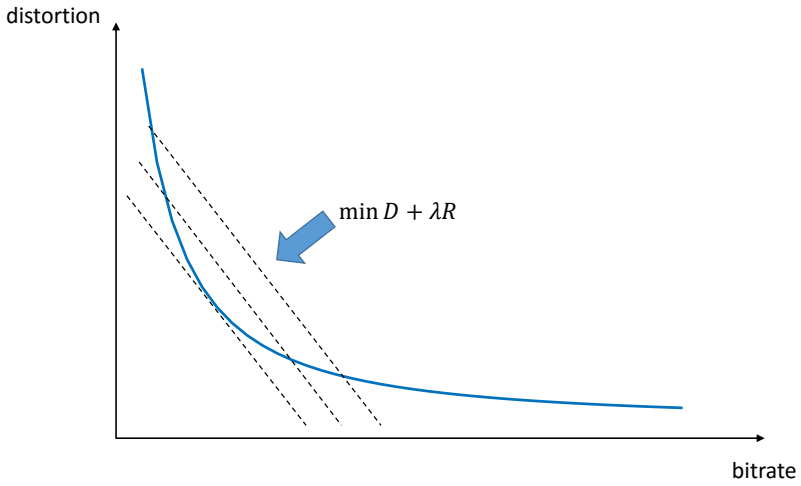


Figure 2.3: Typical rate-distortion curve (blue). The dashed lines are constant J curves with a slope of $-\lambda$. Adapted from [30]

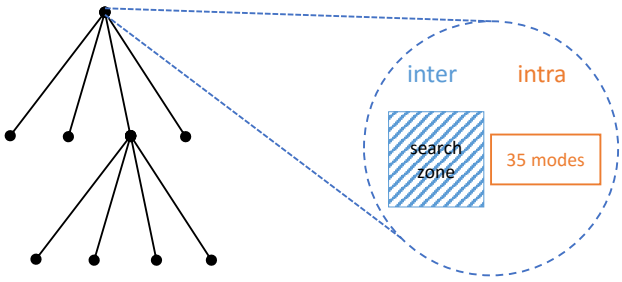


Figure 2.4: Schema of the RDO in HEVC: traversal of the CTU quadtree to analyze each CU.

value of λ is heuristically chosen based on the value of the QP. As the QP increases, the value of λ increases as well.

In the case of HEVC, the achieved bitrate and distortion for a given QP and λ pair are dependent of the coding decisions (block structure, intra or inter prediction, intra prediction mode, MVs, etc.). Thus, the RDO is equivalent to finding the encoding decisions that minimize the RD cost J. Mathematically, Eq. (2.2) can be reformulated as finding the optimal encoding decisions $\mathrm{dec}_{\mathrm{opt}}$ from the set of all possible decisions $\{\mathrm{dec}\}$ that minimize J:

$$\mathrm{dec}_{\mathrm{opt}} = \underset{\{\mathrm{dec}\}}{\arg\min}\, J \tag{2.3}$$

Practically, due to the quadtree structure of the CTU, the RDO process during encoding consists of a tree traversal, where each node of the CTU quadtree has to be analyzed. The encoder starts analyzing the largest CU (i.e., depth 0 of the quadtree, where the CU size is equal to the CTU size) and performs inter and intra prediction for this CU. The RD cost J of each possible prediction is calculated and the prediction leading to the minimum RD cost $J_{\mathrm{min},0}$ is stored as the best candidate for depth 0. The CU at depth 0 is then split into four CUs at depth 1, and each CU at depth 1 is then analyzed. The prediction leading to the

minimum RD cost for each CU is stored again. This process is repeated recursively so that the entire quadtree is traversed. Finally, the combination of block structure and prediction which leads to the overall minimum RD cost $J_{\min}$ is chosen to encode the CTU. The RDO process is illustrated in Figure 2.4.

2.1.3 Rate control

Certain applications require the bitrate of a compressed video stream to be as close as possible to a given target bitrate, especially when the applications involve transmission over a channel with a specific throughput. *Rate control* is an operating mode of a video encoder which aims at achieving an output bitrate equal to the target bitrate.

Rate control algorithms typically consist of two steps. In the first step, the bit budget (derived from the target bitrate) is allocated at different levels of encoding, and in the second step, the encoder spends the allocated bits as exactly as possible [31].

In the allocation step, the bits are generally allocated at group-of-pictures (GOP) level, frame level and basic block level (i.e., CTU level in HEVC). For some applications such as low-delay video streaming, the bits are not allocated at GOP level, but equally to all frames.

To achieve a specific number of bits in the second step, rate control algorithms rely on modeling the bitrate behavior as a function of a specific encoding parameter that can be tuned. Depending on the chosen parameter, the rate control is said to be performed in the *Q-domain*, if the tuning parameter is the QP, in the *λ-domain*, if the parameter is the λ from the RDO, or in the *ρ-domain*, if the bitrate is modeled based on the percentage ρ of zeros among quantized coefficients [32].

2.1.4 Encoding complexity

The complexity of an HEVC encoder is mainly due to the RDO. Experiments using the HM reference software [33] show that the RDO accounts for more than 75% of the encoding time, while less than 15% of the encoding time is due to the entropy encoding [34]. Within the RDO, the complexity stems from the high number of possible CTU encoding decisions, which all have to be tested and compared if optimal decisions in the RD sense are to be made for a CTU.

2.1.4.1 Intra prediction

In the case of intra prediction, the 35 different intra prediction modes are the main factor of complexity. A full search among all 35 modes consists of performing the intra prediction with each prediction mode, then transforming the residuals, quantizing the coefficients and finally do the entropy encoding. The resulting encoding distortion D and bitrate R for each mode can be used to calculate the RD costs and choose the best intra mode. This full search is, however, too complex to be practically implemented.

The HEVC reference software HM [33] therefore implements a suboptimal *fast encoding algorithm* [27] that includes two stages. In the first stage, a rough mode decision is performed to choose a set of candidates: 3 or 8 modes depending on the PU size. In that sense, a modified RD cost function is evaluated for all 35 modes:

$$J_{\text{rough}} = D_{\text{Had}} + \lambda R_{\text{mode}} \tag{2.4}$$

where D_{Had} is the absolute sum of the Hadamard transformed residual signal and R_{mode} is the number of bits needed to signal the prediction mode. The modes leading to the lowest J_{rough} are chosen as candidates. In the second stage, the candidates are compared using the full RD cost J, and the one mode leading to the lowest RD cost is selected.

2.1.4.2 Inter prediction

The process of finding the MV for a PU is called *motion estimation*. Motion estimation is generally performed by comparing the PU to possible predictor blocks in the reference frames, calculating a difference metric and finally picking the predictor block that leads to the lowest difference. In order to find the optimum MV, a full search has to be performed, that is, each position in each reference frame has to be tested. However, such a full search is prohibitively complex. Fast and efficient motion estimation algorithms have to be implemented in order to reduce the computational complexity of the motion estimation to an acceptable level. In practice, a *search range* is defined, which restricts the motion estimation to an area smaller than the entire frame. Additionally, search patterns are used to avoid testing all possibilities within the search range.

The HM reference software uses a test zone (TZ) search algorithm [35] that works in two stages: first an integer pixel accuracy search and then a sub-pixel refinement. The integer pixel accuracy search itself consists of four steps:

1. A starting motion vector is selected: either the motion vectors from the neighboring blocks (left, above, above-left), or a median of these vectors, or the zero vector.

2. A diamond search around the point given by the starting vector is performed. That is, a diamond pattern with 8 points is tested, where the distance to the starting point is iteratively multiplied by 2 (cf. Figure 2.5). The point leading to the lowest cost is selected as candidate.

3. A raster scan search is performed over the entire search range in order to avoid being stuck in a local optimum (cf. Figure 2.6). The best candidate is updated if a better point is found.

4. A second diamond search is performed around the current best candidate for refinement.

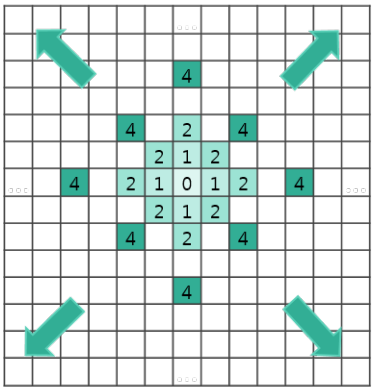


Figure 2.5: Diamond search with iterative testing of an 8 points diamond pattern. (Source: [36])

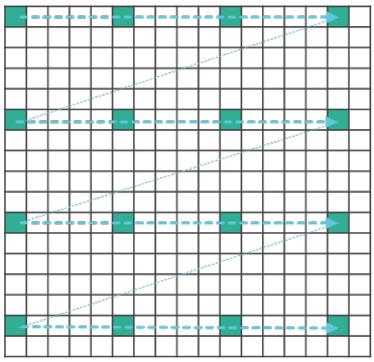


Figure 2.6: Raster search over a given search range. (Source: [36])

2.1.5 State-of-the-art HEVC encoding

Due to the large number of possible encoding decisions, a primitive HEVC encoder that would perform a full-search RDO would not be practical for any real-world application. Thus, in the last few years, the research on HEVC has mainly focused on reducing the encoding complexity by proposing methods that make fast suboptimal RDO decisions, with the often contradicting targets to be as fast as possible and to have an RD performance as good as possible. Thus, the proposed methods are generally evaluated by their complexity reduction compared to a reference HEVC encoder and by the RD performance difference compared to the same reference encoder.

The determination of the block structure, especially the CU structure, is a major part of the RDO. The depth of a CU can be predicted using neighboring blocks and colocated CUs in previous frames [37]. The authors in [38] propose an early CU splitting and early CU pruning in the intra case based on an online statistical model of the RD cost. In a similar way, the authors in [39] propose a fast CU partitioning method based on a two-class classification problem using a minimum risk Bayesian decision rule and on the RD cost function.

In the case of intra mode decision, [40] makes use of the intra modes of the neighboring blocks to construct a set of probable modes to be checked during RDO. The authors in [41] calculate gradients of the pixel values to determine potential intra modes. [42] proposes a

two-step method where the first step is a fast rough mode decision.

In the case of inter prediction, research has focused on reducing the complexity of the motion estimation, which is one of the most computationally complex parts of the RDO. For example, [43] performs a fast motion estimation with the help of statistical inference. In another direction, the authors in [44] propose a method to reduce the complexity of the motion compensation. An early merge mode decision has been proposed in [45] to further reduce the inter prediction complexity.

Finally, researchers have focused on the parallelization of encoding processes to further speed up HEVC encoding, e.g., for parallel motion estimation [46], parallel intra prediction [47], or for an improved wavefront parallel processing [48].

2.1.6 Video coding metrics

Video encoders are generally compared with respect to their RD performance. Therefore, the distortion is commonly measured using the Mean Squared Error (MSE), which is calculated by performing a pixel-to-pixel comparison between the original unencoded picture X and the encoded picture $\hat{X}$ as follows:

$$\mathrm{MSE} = \frac{1}{N_x \cdot N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} \left(X(x,y) - \hat{X}(x,y) \right)^2 \tag{2.5}$$

where N_x and N_y are the number of pixel horizontally and vertically, respectively.

The Peak-Signal-to-Noise-Ratio (PSNR) is a video quality metric expressed in decibel (dB), based on the MSE, and which can be calculated as follows:

$$\mathrm{PSNR} = 10 \cdot \log_{10} \frac{(2^b - 1)^2}{\mathrm{MSE}} \, dB \tag{2.6}$$

where b is the number of bits used to represent a pixel in a channel. The PSNR can be calculated for the different channels (one luminance channel and two chrominance channels), but PSNR generally refers to the luminance PSNR, when not stated otherwise. For a video sequence, the PSNR is calculated for each frame and an arithmetic mean of the values is calculated as the sequence PSNR. In this thesis, only video sequences with $b = 8$ are considered and the luminance PSNR is denoted as PSNR.

The bitrate of an encoded video sequence is calculated as the number of bits of the encoded video stream divided by the duration of the video sequence in seconds, and can thus be typically expressed in kb/s or Mb/s.

An RD-curve is generally plotted as the PSNR as a function of the bitrate. The RD performance of two encoders is compared using the Bjøntegaard delta rate (BD-rate), which expresses the average bitrate difference in % over a specific PSNR interval, and the Bjøntegaard delta PSNR (BD-PSNR), which expresses the average PSNR difference over a specific bitrate interval [49], [50]. Four data points (PSNR/bitrate) per curve are needed to interpolate the RD-curves. The average PSNR difference in decibel (dB) and the average bitrate difference in % are calculated as the difference of the integrals divided by the integration interval.

In the last 15 years, new metrics targeted at measuring the perceptual quality of an encoded video have been developed, because metrics such as PSNR fail to accurately determine the quality perceived by human viewers under certain circumstances [51], especially when different types of distortions (e.g., blurring or gaussian noise) are compared. These metrics are out of the scope of this thesis, because here only one video encoder is considered, and thus, it is assumed that the type of distortion introduced by the lossy compression is always the same. In that case, the PSNR is monotonically related to the perceptual quality, and the PSNR can be used to assess the performance of the encoder.

2.2 Adaptive HTTP streaming

2.2.1 Video streaming protocols

In order to ensure a reliable streaming session and thus an acceptable video quality to the users, streaming protocols have to be employed for video streaming over the internet. Streaming protocols can be divided into either *push-based* protocols or *pull-based* protocols [52].

2.2.1.1 Push-based protocols

Push-based protocols like the Real-time Transport Protocol (RTP) require a session to be established between the server and the client. The session control is at the server, which decides at what rate the video is sent to the client, typically using the User Datagram Protocol (UDP). The client can send feedback to the server, e.g., about buffer-level, throughput, or round-trip time.

The major drawbacks of push-based protocols are on one hand the need for specialized servers that can handle stateful sessions. On the other hand, the unreliability of UDP can lead to a large number of packet losses and UDP is sometimes blocked by firewalls in certain cellular network configurations.

2.2.1.2 Pull-based protocols

In pull-based protocols, the streaming control lies at the client side. The client can typically request the video from the server using Hypertext Transfer Protocol (HTTP) over Transmission Control Protocol (TCP). The main advantages compared to push-based streaming are that no specialized servers are required and that HTTP servers are already widely deployed. Furthermore, as the control lies at the client, the computational requirement at the server is less and the streaming can be more easily scaled to a large number of users.

Progressive download over HTTP is a simple pull-based streaming approach. A client requests a video stream at a specific bitrate and receives the video stream progressively at a pace determined by the selected bitrate and the current throughput. The client can start

watching the video as soon as the buffer reaches a predefined level. The limitations of progressive download are that it does not enable live streaming and that it provides no bitrate adaptivity during a session. Furthermore, network resources might be wasted due to the continuous download of the video, because the user can stop watching the video at any time [19].

2.2.2 Adaptive HTTP streaming

Adaptive HTTP streaming is a streaming technology designed to overcome the drawbacks of the streaming protocols previously introduced. It builds on top of progressive download over HTTP as it uses the pull-based paradigm and the HTTP protocol. Thus, it benefits from the advantages of HTTP [19], namely:

1. HTTP is widely deployed and the existing architecture, including content delivery networks (CDNs), can be reused.
2. Firewall issues are avoided using HTTP.
3. Reliable transmission is provided by the underlying TCP protocol.
4. HTTP allows the control of the streaming to be placed at the client.

In addition, the video content is encoded at different bitrates (and thus different qualities) called *representations*, and the encoded representations are segmented in the time domain into time-aligned segments (with a typical duration of two to ten seconds). These two features enable bitrate adaptivity, as the client can switch between representations (and thus bitrates) within a streaming session at each segment boundary. Furthermore, the segments have to be requested one by one, which reduces the bandwidth wastage when a user stops watching a video. Finally, live streaming is made possible, as segments can be published on the server as soon as they are encoded, which reduces the end-to-end latency compared to a scenario where a whole video has to be encoded before being published.

The available representations and segments along with their characteristics (bitrate, time, Uniform Resource Locator (URL)) are summarized in a manifest document that is made available to the client at the beginning of the streaming session. A rate-adaptation algorithm at the client is responsible for making the adaptation decisions. Rate-adaptation algorithm are generally based on the measured throughput, the video buffer level, and the decoder state (e.g., central processing unit (CPU) usage). Figure 2.7 shows an example adaptive HTTP streaming session.

2.2.3 State-of-the-art adaptive HTTP streaming

Adaptive HTTP streaming was first implemented in commercial solutions (e.g., Apple HTTP live streaming [53] in 2009, Microsoft Smooth Streaming [54] in 2010). An international stan-

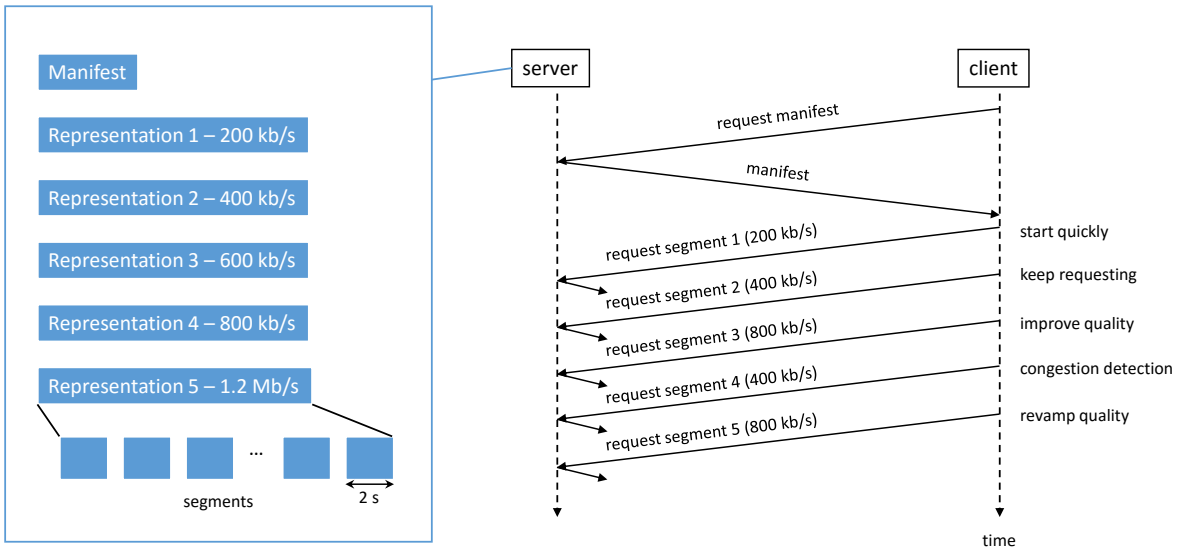


Figure 2.7: Example of an adaptive HTTP streaming session. The client first requests the manifest and can then request segments at different bitrates depending on the rate-adaptation algorithm. Adapted from [52]

dard called Dynamic Adaptive Streaming over HTTP (DASH) was then ratified in 2011 and published in 2012 [55].

Research has first focused on the rate-adaptation algorithms at the client side. Liu *et al.* relate the segment duration to the segment fetch time to decide if the requested representation has to be switched up or switched down [56]. Miller *et al.* target a streaming without interruption by avoiding buffer underrun, and additionally try to minimize the number of switches and maximize the average video quality as secondary objectives [57]. The FESTIVE algorithm considers the interaction of multiple clients and uses for example a randomized scheduler to avoid a synchronization of the requests of multiple clients [58]. Similarly, the authors in [59] propose a new network probing method that better estimates the share of the resources their PANDA algorithm can use. These and further adaptation algorithms have been tested in [60], which comes to the conclusion that there is no perfect rate-adaption algorithm with respect to multiple criteria, and that relatively simple rate-adaptation approaches perform well under different circumstances.

To investigate the subjective user experience of a video streaming session, the evaluation of the Quality-of-Experience (QoE) has been an important topic of research. The study [61] relates the network Quality-of-Service (QoS) metrics to application QoS and finally to user QoE, and finds that rebuffering events have the strongest impact on the users' QoE. The authors in [62] conduct an evaluation of adaptive HTTP streaming over a simulated Long Term Evolution (LTE) network by measuring the rebuffering events. [63] evaluates the QoE with a random neural network by taking into account both the rebuffering events and the video encoding quality.

From the network perspective, it has been observed that multiple adaptive HTTP stream-

ing users that compete for resources in a bottleneck link can lead to instability, unfairness and underutilization of the resources. Akhshabi *et al.* show in [64] and [65] that the application layer is responsible for this behavior, especially because of the requesting pattern of the rate-adaptation algorithm in the steady state. Based on these observations, [66] proposes to shape the bandwidth of competing users in a residential gateway, which improves the stability and convergence time of rate adaptation, which in turn affects the QoE of the competing users positively. In [4], the wireless resources are allocated and the HTTP requests are optionally rewritten at a proxy, so that the sum of QoE is maximized over multiple users sharing the wireless resources in a cellular network. Other network studies examine the impact of a cache engine between the origin server and the client [67], or investigate the effect of the new HTTP 2.0 protocol on adaptive HTTP streaming [68].

Meanwhile, there have been comparatively few studies on the server side (preparation of the video content) of adaptive HTTP streaming systems. In [8], an uplink video streaming in an automotive case is considered and the number of representations needed to provide a good QoE is studied. A method to select a subset of representations to be encoded based on the feedback from the network is proposed, in order to reduce the computational load of encoding the entire set of possible representations. [69] proposes to use scene-cuts as segment boundaries, which allows to place I-frames only at scene-cuts and thus improve the encoding efficiency. Finally, Toni *et al.* formulate an optimization problem taking into account video content, network capacity, and type of users to determine an optimal set of representations that maximizes the user satisfaction [70].

2.3 Multi-rate video encoding

2.3.1 Video encoding methods

In order to avoid buffer underflow in video streaming sessions, the bitrate of the compressed video stream has to be adapted to the communication channel. The bitrate of a video compressed with a specific encoder is influenced by the spatial resolution, the temporal resolution (i.e., the frame-rate) and the signal fidelity (i.e., level of distortion introduced by lossy compression). A video can be compressed at a target bitrate using rate control. Different rate control methods have recently been proposed for HEVC, e.g., [31], [71]. Another possibility is to transcode (or transrate) an already encoded video to another bitrate [72]–[74]. The drawback of transcoding is that the RD performance is decreased due to requantization. Both rate control and transcoding are designed to target one specific representation.

On the other hand, a video can be encoded to provide inherent scalability. Scalable video coding [75] encodes a video into a base layer and several enhancement layers. Decoding an enhancement layer requires the availability of the base layer at the decoder. Major drawbacks of scalable video coding are the increased decoding complexity compared to a single-layer encoded representation with no decoding dependencies, and the decreased RD performance.

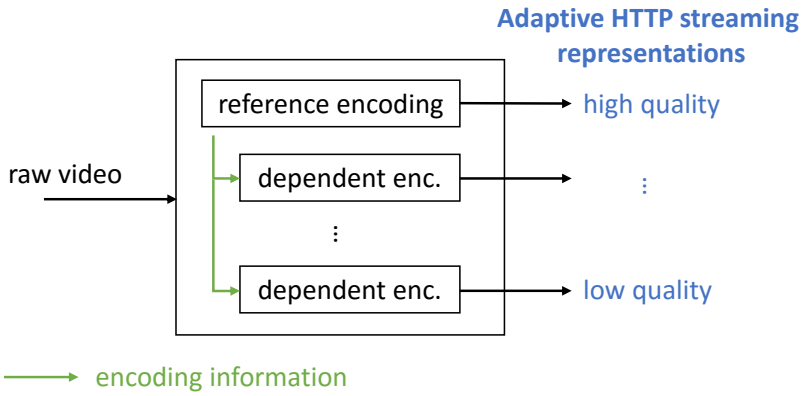


Figure 2.8: Schema of a multi-rate encoder. Encoding information is passed from a reference encoding to dependent encodings within the multi-rate system.

For example, the scalable extension of HEVC increases the bitrate by at least 14.3% [76] compared to single-layer HEVC. Furthermore, in an HTTP streaming scenario, a client needs to send a request per layer, which leads to inefficient multiple requests to obtain a high-quality representation. Due to these drawbacks, scalable coding is not expected to be widely deployed for adaptive HTTP streaming, contrarily to single-layer HEVC.

2.3.2 Multi-rate encoding

In *multi-rate encoding*, a video is directly encoded at multiple bitrates, each one being independently decodable [22]. This is particularly suited for adaptive HTTP streaming applications, where a video needs to be encoded in different representations. The redundancy of encoding the same video at different bitrates is exploited, either in order to reduce the overall encoding complexity, or in order to improve the overall RD performance. Figure 2.8 shows a schema of a multi-rate encoder, which contains multiple instances of single-layer encoders. One single-layer encoding is used as *reference*, and encoding information from the reference is passed to *dependent* single-layer encoders [77]. The system takes one unencoded raw video as input and outputs a set of predefined representations at different qualities and different bitrates.

2.3.3 State-of-the-art multi-rate encoding

2.3.3.1 Multi-rate encoder in the DCT domain

The first work on multi-rate encoding was published in 2002 by Zaccarin *et al.* [77]. Although there was no adaptive HTTP streaming at that time, the authors target video streaming offered at different bitrates. The authors argue that both scalable encoding and transcoding present lower encoding efficiency in the RD sense than single-layer encoding, and thus introduce the idea of multi-rate encoding with a reference encoder and dependent encoders.

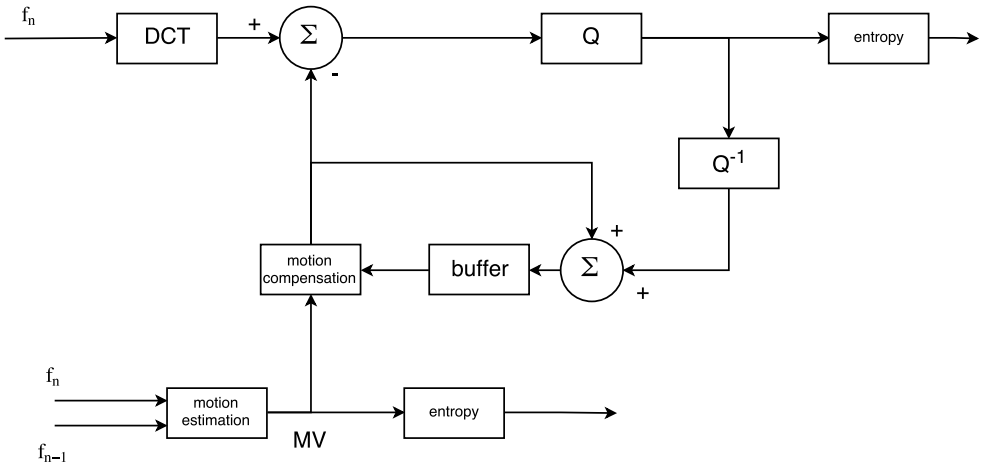


Figure 2.9: Block diagram of the reference encoder by Zaccarin *et al.* The DCT and the motion estimation are performed outside the prediction loop. Adapted from [77]

The presented multi-rate encoder applies a DCT to the frames before prediction, which allows to perform the DCT only once for all bitrates. Similarly, the motion estimation is performed only once in the temporal domain, and the motion vectors are used to perform motion compensation in the DCT domain [78]. Furthermore, the authors propose to approximate the motion compensation for the dependent encodings by using the DCT coefficients from the reference encoding, which can further reduce the computational complexity, but contributes to a drift error. The drift error is kept small by resetting with the appropriate DCT coefficients. Figure 2.9 shows the block diagram of the proposed reference encoder. The prediction loop is the same for the dependent encoders, but the motion compensation can be modified.

Experimental results for one video sequence show a RD performance degradation of less than 0.3 dB PSNR. However, the gains in computational complexity are not presented. A major drawback of the proposed multi-rate encoder is that it is not standard compatible due to the DCT applied before the prediction.

2.3.3.2 VP8 multi-rate encoder

With the emergence of adaptive HTTP streaming, Finstad *et al.* pick up the idea of multi-rate encoding again in 2011 [22]. The authors consider an implementation with the open-source VP8 encoder [79]. Profiling of the VP8 encoding shows that more than 80% of the encoding time is spent for the analysis part, that is, for the RDO. Thus, the authors propose to directly reuse analysis decisions from a reference encoder in the dependent encoders. The reused analysis information contains the macroblock mode decision, the intra-prediction, and the inter-prediction [22].

Experimental results show that the encoding time can be sped up by up to 2.5 times, but the PSNR decrease is between 1 dB and 1.5 dB in the case where a small range of bitrates

is used for the output (1.4 Mb/s to 2.8 Mb/s for an HD video). If a broader range of bitrates is chosen, the RD performance decrease can reach 6 dB. The bad RD-performance of this scheme can be explained by the direct reuse of the analysis decisions from the reference encoder, while the optimal decisions for the dependent encodings would always be different.

2.3.3.3 Machine learning based multi-rate encoder

De Praeter *et al.* apply the idea of multi-rate encoding to HEVC by predicting the CU structure of dependent encodings using machine learning [80].

The authors propose to use the *random forest* algorithm [81]. The random forest has to be trained on the first M frames of the video, by using features from the reference encoding to build a set of decision trees for each dependent encoding. Each tree uses a random set of features, which increases the robustness of the algorithm compared to a single decision tree. The possible features consist of the mean, the variance, the maximum, and the minimum of the CU, PU, and TU block sizes, as well as the variance of the transform coefficients, and the motion vector variance.

In the next frames, the random forest model determines if a CU has to be split or not based on the decision trees. The authors set the threshold for a CU not to be split if a split would result in a tree node containing less than 1% of the total number of samples used in the tree. In the case of multiple resolutions, where a block to be determined does not exactly correspond to a block in the reference encoding, the information is weighted according to the percentage of area that is corresponding.

In their experiments, the authors try out different reference encodings (from high quality to low quality). They come to the conclusion that for a given spatial resolution, using a higher quality reference leads to a higher prediction accuracy (percentage of correct CU split decisions) than using a lower quality reference. Furthermore, a higher quality reference leads to a lower BD-rate increase (thus, to a better RD-performance) than with a lower quality reference. However, the higher quality reference does not necessarily lead to the highest complexity reduction. In a scenario with output streams at different resolutions, the proposed methods achieves complexity reduction between 51.9% and 71.8% for an average BD-rate increase between 4.7% and 10.2%. As storage and transmission costs increase with an increasing BD-rate for a given quality, a BD-rate increase of 4.7% could already be prohibitive for a video provider.

2.3.3.4 Simultaneous H.264/AVC and HEVC encoding

Cebrián-Márquez *et al.* propose a multi-rate encoder which uses an H.264/AVC reference encoding to speed up HEVC dependent encodings [82]. The arguments to use an H.264/AVC reference are on one side to provide backwards compatibility for devices which do not support HEVC, and on the other side the relative lower complexity of H.264/AVC.

The authors propose a motion vector reuse algorithm, which uses the MVs from the H.264/AVC reference to initialize the motion estimation of the HEVC encoder. As the H.264 macroblock has a size of 16×16 pixels, while a PU can be larger, the median of the corresponding MVs can be used to initialize the motion estimation of a large PU. As the H.264/AVC and HEVC motion vectors show a significant similarity, the search range of the HEVC motion estimation is reduced to 4 pixels in order to reduce the encoding time. Instead of the default diamond search pattern in the reference HEVC encoder, the authors propose to use a hexagonal pattern.

Experimental results show that the complexity of the motion estimation of the dependent HEVC encodings can be reduced by 43% with the proposed method, which leads to an overall encoding time reduction of almost 9%. This encoding time reduction, however, comes at the price of a slightly degraded RD performance, with an average BD-rate increase of 1.2%.

2.3.3.5 Remarks on the related work

The drawbacks observed in the related work are used as a basis for the motivation of certain aspects in the rest of the thesis:

- The proposed multi-rate encoder should be standard compatible, so that the methods can be widely applicable and integrated in existing systems. Specifically, the recent HEVC standard is considered in this thesis.
- The encoding information from the reference encoding should be reused in an intelligent way, that is, no direct reuse is implemented, as this would lead to a low RD performance.
- In order to provide a good acceptability for the proposed multi-rate encoder, the objective is to reach an RD performance as close as possible to the original single-layer encoder, because a decreased RD performance leads to increased costs for transmission and storage of the video content.

Chapter 3

Settings

This chapter presents the settings common to the entire thesis. Section 3.1 introduces the HEVC encoder and its configuration. The hardware is described in Section 3.2. Section 3.3 presents the data set consisting of various video sequences at different spatial resolutions.

3.1 HEVC encoder

The reference HEVC encoder HM 16.5 [33] compiled with *gcc* 4.8.4 is used as software encoder throughout the thesis. The unmodified encoder is used to gather observations and as a baseline for comparison with the proposed methods. The proposed methods are implemented based on HM 16.5 in order to allow for fair comparisons.

Furthermore, the JCT-VC common test conditions and software reference configurations from [83] are followed. For adaptive HTTP streaming, the video representations must be segmented in the time domain into individually decodable segments, that is, the segments need to start with an I-frame. As the focus is on adaptive HTTP streaming, unless stated otherwise, the *random access, main* profile defined in [83] is used, which provides periodic I-frames in the encoding structure.

3.2 Hardware

All encodings and measurements in this thesis are performed on an Ubuntu 14.04 server with an Intel Q9550 @ 2.83 GHz processor and 8 GB RAM.

3.3 Video sequences

A main set of ten different sequences with an original spatial resolution of 1920×1080 pixels (1080p) and different frame-rates expressed in frames per second (fps) is used. Two sequences *Kimono* and *ParkScene* are from [83] and the eight other sequences *BlueSky*, *CrowdRun*, *DucksTakeOff*, *ParkJoy* , *PedestrianArea*, *Riverbed*, *RushHour*, and *Sunflower* are from [84]. The video sequences present a variety of contents, as can be seen from the thumbnails in

Table 3.1: Main set of ten video sequences

Sequence	SA	TA	fps
BlueSky	79.71	33.70	25
CrowdRun	89.77	21.43	50
DucksTakeOff	77.69	15.25	50
Kimono	22.80	13.60	24
ParkJoy	100.46	33.36	50
ParkScene	49.15	10.71	24
PedestrianArea	32.56	15.46	25
Riverbed	35.95	26.57	25
RushHour	23.61	9.13	25
Sunflower	31.29	13.96	25

Table 3.2: Alternative set of six video sequences at different spatial resolutions

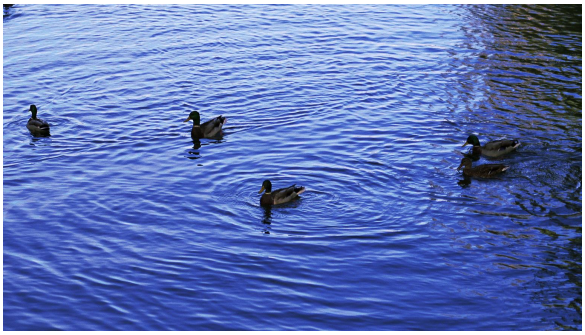
Sequence	resolution	SA	TA	fps
BasketballPass	416×240	70.52	10.51	50
BlowingBubbles	416×240	72.59	16.42	50
BQMall	832×480	89.21	17.21	60
PartyScene	832×480	103.95	13.81	50
PeopleOnStreet	2560×1600	80.90	21.90	30
Traffic	2560×1600	61.77	11.58	30

Figure 3.1. Thus, the sequences have different temporal and spatial characteristics, that can be measured with the spatial activity (SA) and temporal activity (TA) metrics [85], which are summarized in Table 3.1. In the case of multiple spatial resolutions, the original 1080p uncompressed sequence is downsampled to 1280×720 (720p) and 640×360 (360p) pixels.

Additionally, a set of six video sequences from [83] with alternative original video resolutions between 416×240 and 2560×1600 pixels is used, see Table 3.2. These video sequences, represented in Figure 3.2, are used to demonstrate that the proposed methods are not specific to the single resolution from the main set. Furthermore, the alternative set is used as validation set, when the proposed methods are based on observations made on the main set.

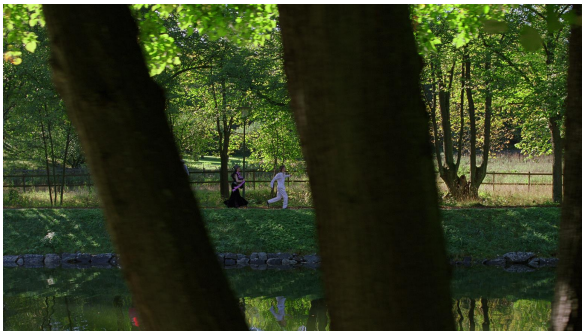
(a) BlueSky

(b) CrowdRun



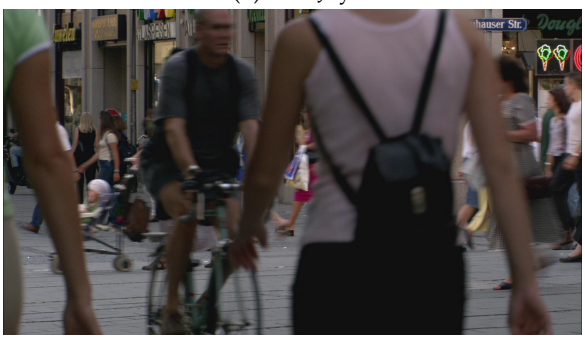
(c) DucksTakeOff

(d) Kimono



(e) ParkJoy

(f) ParkScene



(g) PedestrianArea

(h) RiverBed

(i) RushHour

(j) Sunflower

Figure 3.1: Thumbnails of the main set of ten 1080p video sequences used in this thesis.

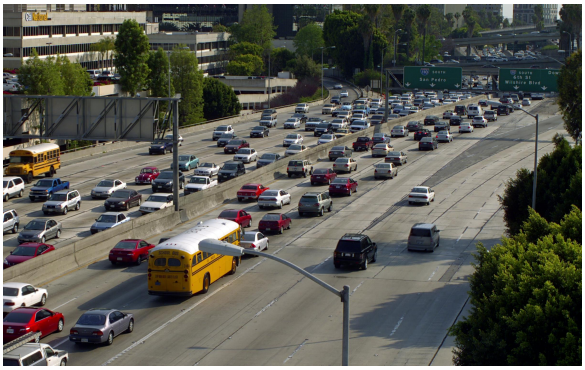
(a) BasketballPass

(b) BlowingBubbles

(c) BQMall

(d) PartyScene

(e) PeopleOnStreet



(f) traffic

Figure 3.2: Scaled thumbnails of the alternative set of six video sequences at different spatial resolutions.

Chapter 4

RDO-constrained multi-rate encoding

4.1 Introduction

The computational complexity of encoding a video at multiple representations for adaptive HTTP streaming is very high, given that a typical adaptive HTTP streaming requires around 10 to 15 representations [70]. The intrinsic redundancy of encoding the same video multiple times is an argument to consider a multi-rate encoding system instead of multiple independent single-layer encoders. Especially, the overall computational complexity is expected to be decreased if the complexity of the redundant parts can be reduced. For that purpose, the video encoding processes have to be understood and the redundant parts have to be identified. In the case of HEVC, the large number of encoding decisions that can be made for each CTU leads to an RDO which accounts for the largest part of the encoding complexity [34].

In this chapter, the goal is to reduce the complexity of encoding multiple representations with HEVC. Importantly, the RD performance of the proposed multi-rate encoder should remain as close as possible to the reference single-layer encoder in order to push the acceptability of the method. Indeed, a decreased RD performance leads to increased video storage and transmission costs, and thus would hinder the widespread use of multi-rate encoding.

First, it is observed that the optimal encoding decisions between representations at different SNR qualities are always slightly different. Thus, unlike previous work on multi-rate encoding [22], encoding decisions from a reference encoding are not directly reused to speed up dependent encodings, because this harms the RD performance. On the contrary, the information from the reference encoding is used to constrain the RDO in the dependent encodings, i.e., the number of possible encoding decisions to be tested during RDO is reduced. The overall encoding time can be decreased for different types of encoding information from the reference encoding, each time without significantly decreasing the RD performance. This chapter is limited to representations at a single spatial resolution and single frame-rate, that is, representations with varying quality in the SNR domain are considered.

The rest of the chapter is organized as follows. A preliminary study to compare the

Table 4.1: Average analysis time for different CU depths (in ms)

depth	CU	intra prediction	inter prediction	intra mode	motion vectors
0	17.35	3.03	10.75	1.85	2.25
1	5.76	0.90	3.70	0.49	0.61
2	1.94	0.28	1.23	0.13	0.20
3	0.54	0.36	0.29	0.05	0.06

relative complexity of different RDO steps is presented in Section 4.2. The reuse methods for an HEVC multi-rate system including observations, proposed methods, and results for the different RDO parts are presented in Section 4.3 for the CU structure, in Section 4.4 for the prediction mode, in Section 4.5 for the intra prediction mode, and in Section 4.6 for the motion vectors. The effects of combining the different proposed methods is examined in Section 4.7. Section 4.8 summarizes the chapter.

4.2 Preliminary study

The potential encoding time reductions differ depending on the information which is reused in the multi-rate system. As an example, reusing the CU structure information for a dependent encoding is equivalent to skipping the analysis of certain nodes in the quadtree. Thus, the prediction mode decision at CU level and underlying intra direction or motion vector search are skipped as well in these nodes. In this example, this means that the CU structure reuse is expected to lead to larger encoding time reductions than the prediction mode reuse.

This is illustrated by time measurements of individual RDO steps. While complexity assessment is a topic in its own right (cf. [34]), a time measurement is a good measure of the underlying complexity of a software encoder. Although the exact value of the time measurements is not relevant because the encoding time depends on the computer configuration, the relative times give insight into the relative complexities.

Table 4.1 shows the average analysis time (in ms) of different steps of the RDO at different CU depths. The average is taken over 12,240 CTUs from different sequences of the main set. As expected, the time to analyze an entire CU is the largest, as it includes analysis of intra and inter prediction. The results also indicate that the inter prediction part of the RDO takes longer than the intra prediction part, mainly due to the various possible PU partitions of inter prediction (cf. Section 2.1.1.1). The sum of intra and inter time does not have to be smaller than the CU time (e.g., at depth 3), because the HM encoder implements early decision algorithms and does not always analyze all possibilities. Finally, the times to determine the intra mode for intra prediction and the motion vectors for inter prediction are the shortest times.

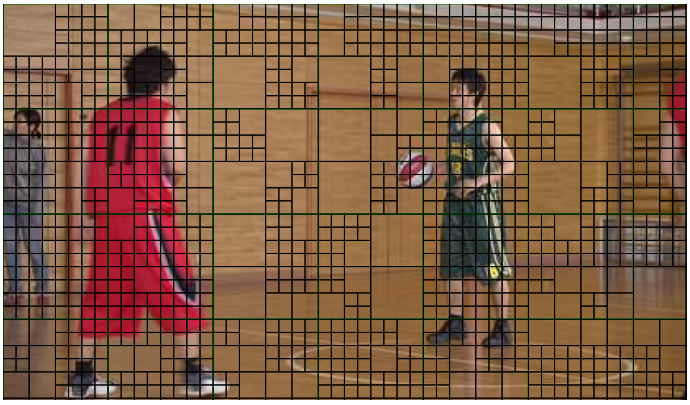


Figure 4.1: First frame of *BasketballPass* encoded at QP 22 and resulting CU structure.

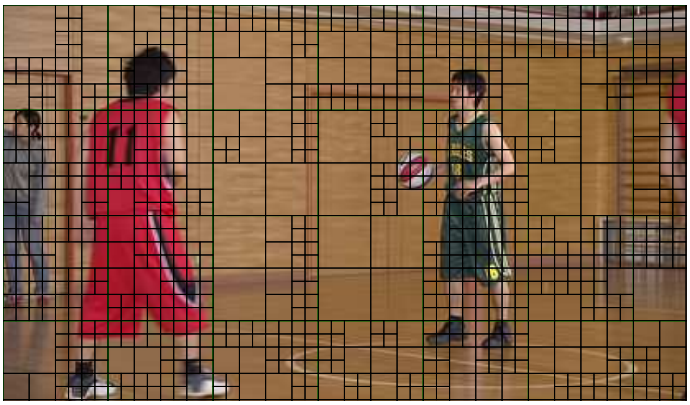


Figure 4.2: First frame of *BasketballPass* encoded at QP 26 and resulting CU structure.

4.3 CU structure reuse

4.3.1 Observations

Strong similarities in the CU structure are observed across multiple encodings of a single video sequence at different SNR qualities. As an example, Figure 4.1 shows the first frame of the *BasketballPass* sequence encoded with the reference software HM [33] at QP 22 and the resulting CU structure. Figure 4.2 shows the same frame encoded at QP 26 with the resulting CU structure. Similarities in the block sizes can be observed between the two figures. For example, the top-left block is encoded at a large CU size and is surrounded by small blocks. On the other hand, differences in the central area of the frame can also be observed, where the frame encoded at QP 26 has more larger blocks.

In order to quantify the similarity between two encodings with different QPs, the percentage of the area of the frames where the CU size is the same is calculated, that is, the CU has the same depth. If the CU depth is not identical, it can either have a greater depth (i.e., a smaller CU size) or a lower depth (i.e., a larger CU size).

Figure 4.3 shows the percentage of the area of the frames where the block depth is greater, identical, or lower than the reference depth given by the encoding at QP 22, as a mean over

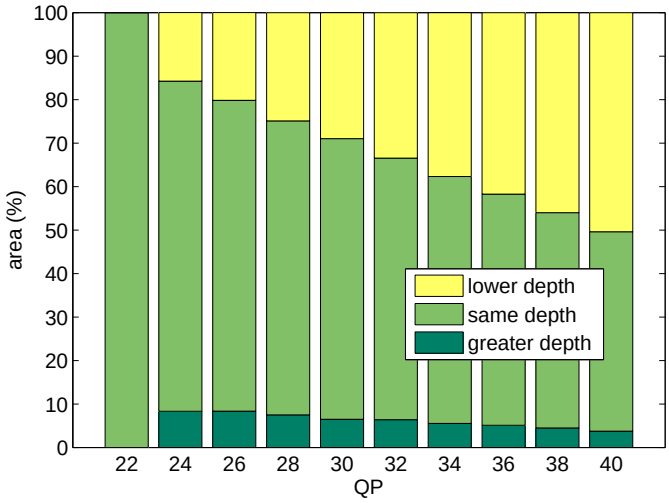


Figure 4.3: Average percentage of the area of the 10 sequences of the main set with block depth greater, identical or lower than the reference encoding at QP 22.

one second of 10 different video sequences of the main set. The encoding at QP 22 shows 100% similarity with the reference encoding QP 22, as expected. The percentage of CUs with the same depth as in the reference encoding decreases as the QP increases (down to approximately 45% at QP 40). In a multi-rate encoding scenario, this means that the CU structure cannot be directly reused for lower quality dependent encodings, as numerous suboptimal CU size decisions would be made.

Interestingly, with growing QP, the videos tend to have more CUs with a lower depth, that is, with a larger CU size. This confirms the observation made on the frames of the *BasketballPass* sequence in Figures 4.1 and 4.2 On the other hand, less CUs will have a greater depth. This behavior can be intuitively explained by the fact that at a greater QP, the strong quantization leads to less details in the image and thus encoding can be more easily performed at a larger block size.

4.3.2 Information reuse

Given the similarities in the CU structure across multiple encodings of a single video at different qualities, it is proposed to reuse the information of the CU structure of a high-quality reference encoding to shorten the RDO process of lower-quality dependent encodings, and thus reduce the encoding time.

As explained in Section 2.1.2, the RDO process is implemented starting from the largest CU size, that is, depth 0. After analysis of the CU at depth 0, the CU is split into 4 CUs at depth 1, and the RDO is recursively applied to these CUs, until the maximum CU depth is reached.

Knowing that, on one hand, the CUs of an encoding will mostly have a lower or equal depth compared to a high quality reference encoding, and that the amount of CUs having a greater depth than in the reference encoding is relatively small (cf. Figure 4.3), and on the

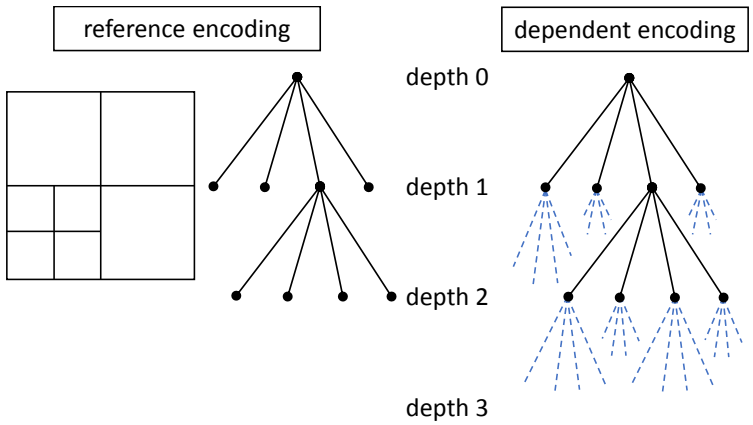


Figure 4.4: Example CTU block structure and quadtree for the reference encoding on the left, and quadtree checked during the RDO process for the dependent encoding on the right.

other hand, the RDO process of the HEVC encoder is implemented recursively starting with depth 0, it is proposed to stop the RDO process of the dependent encodings at the depth given by the high-quality reference encoding for each CU in the video sequence. This is illustrated in Figure 4.4, where an example CU structure of the reference encoding and the corresponding quadtree is shown on the left and on the right the quadtree that is checked during RDO of a dependent encoding. The depths depicted in dashed lines are not checked during the dependent RDO process, which leads to significant encoding time savings.

In the case of CUs that should have a greater depth as in the reference encoding, a suboptimal CU size in the RD sense will be chosen. The overall RD loss should be small, however, as this concerns only a relatively small percentage of all CUs (cf. Figure 4.3). On the other hand, the relatively large number of CUs which have a lower depth in a dependent encoding will still have the optimal CU size in the RD sense as the RDO process will pass through these low depths during the dependent encoding.

The fact that the highest quality encoding tends to have the most small CUs combined with the recursive RDO process implementation is an argument to choose the highest quality encoding as the reference encoding, if best RD performance has to be achieved.

4.3.3 Results

Table 4.2 shows the encoding results of the implementation of the proposed CU structure reuse method based on HM compared with the unmodified encoder[1]. As introduced in Section 3.1, the *random access, main* profile is used, and four representations encoded with QP 22, 27, 32, and 37. On average, the encoding time is decreased by 33.61% for four representations, while the average BD-rate is increased by 0.53%.

Table 4.3 shows the encoding results for the alternative set with different resolutions. The average encoding time is decreased by 27.94% and the average BD-rate is increased by

[1] The source code of the CU structure reuse method presented in [7] is available at https://github.com/damjeux/multi-rate-HEVC.

Table 4.2: Comparison of encoding with CU structure reuse vs. conventional encoding for the main set at 1080p.

Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	0.37%	−0.014 dB	−39.23%
CrowdRun	0.51%	−0.022 dB	−19.23%
DucksTakeOff	0.29%	−0.008 dB	−25.53%
Kimono	0.74%	−0.025 dB	−39.71%
ParkJoy	0.33%	−0.014 dB	−26.13%
ParkScene	0.63%	−0.021 dB	−35.39%
PedestrianArea	1.00%	−0.031 dB	−35.94%
Riverbed	0.37%	−0.015 dB	−40.89%
RushHour	0.14%	−0.002 dB	−35.02%
Sunflower	0.91%	−0.017 dB	−39.04%
Average	**0.53%**	**−0.017 dB**	**−33.61%**

Table 4.3: Comparison of encoding with CU structure reuse vs. conventional encoding for the alternative set.

Resolution	Sequence	BD-rate	BD-PSNR	ΔT
416 × 240	*BasketballPass*	0.50%	−0.024 dB	−35.19%
	BlowingBubbles	0.32%	−0.013 dB	−20.95%
832 × 480	*BQMall*	0.61%	−0.027 dB	−29.13%
	PartyScene	0.29%	−0.013 dB	−22.25%
2560 × 1600	*PeopleOnStreet*	0.85%	−0.039 dB	−19.87%
	Traffic	0.52%	−0.019 dB	−40.23%
	Average	**0.52%**	**−0.023 dB**	**−27.94%**

Table 4.4: Two-way ANOVA with resolution and average depth of the reference.

Source	Sum Sq.	df	Mean Sq.	F-value	p-value
resolution	16.13	3	5.377	0.37	0.773
average depth	618.92	1	618.92	43.13	< 0.0001
Error	157.84	11	14.349		

0.52%. The average encoding time reduction is slightly less than for the main set where all videos are at 1080p. However, the proposed method is shown not to depend on the spatial resolution.

For that, a two-way analysis of variance (ANOVA) of the encoding time reduction [86] is performed. The spatial resolution is used as a categorical explaining variable with four different resolutions possible. The second explaining variable is the average CU depth of the reference encoding, and is a continuous variable. The outcome of the two-way ANOVA

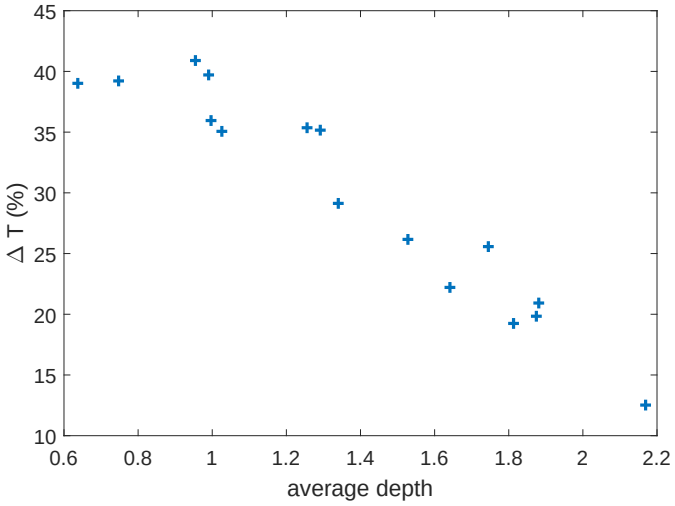


Figure 4.5: Encoding time reduction ΔT as a function of the average depth of the reference encoding for 16 videos.

is summarized in Table 4.4. For the spatial resolution, the *p-value* of 0.773 indicates that the mean encoding time reduction for the different resolutions is not significantly different. On the other hand, the *p-value* for the average depth is less than 0.05 and is thus small enough to conclude that the encoding time reduction is significantly different for different average depth values.

Figure 4.5 shows the encoding time reduction ΔT as a function of the average depth of the reference encoding for the 16 videos. There is a clear trend of decreasing ΔT for an increasing average depth of the reference. This can be explained by the proposed method, which stops the RDO process at the depth given by the reference encoding. The lower the average depth, the less CUs have to be analyzed and thus the highest the encoding time reduction. The average depth of the reference is mainly influenced by the content of the video.

4.4 Prediction mode reuse

4.4.1 Observations

In order to identify the similarities in prediction mode across different qualities, the ten videos from the main set are encoded at different QPs ranging from 22 to 40. The intra/inter decision is gathered at every node of the quadtree during the RDO, that is, for every possible CU depth. The percentage of inter predicted CUs in inter predicted frames (i.e., I-frames are omitted) is first examined in Figure 4.6. On average, the percentage of inter CUs is observed to increase with increasing QP, independently of the CU's depth.

It is next investigated if the decision for a node in a low-quality encoding (QP from 24 to 40) is the same as the decision in the reference encoding (QP 22). Figure 4.7a shows what percentage of intra encoded CUs in the reference encoding is still intra encoded in the lower quality encodings, as a function of the QP. At QP 22, 100% of intra CUs are in common, as

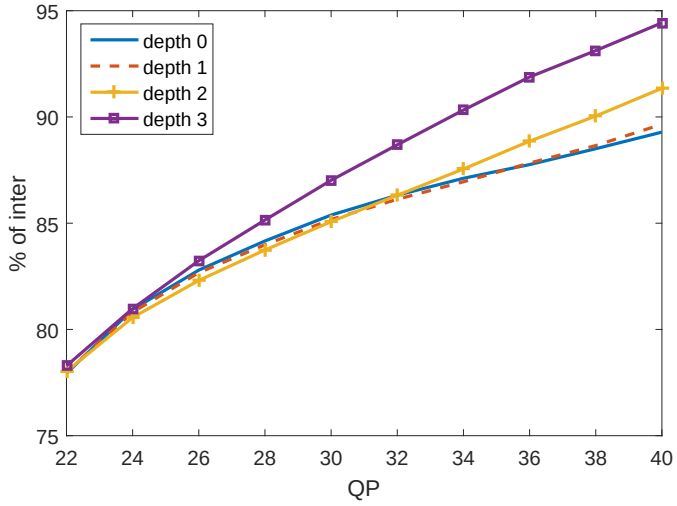
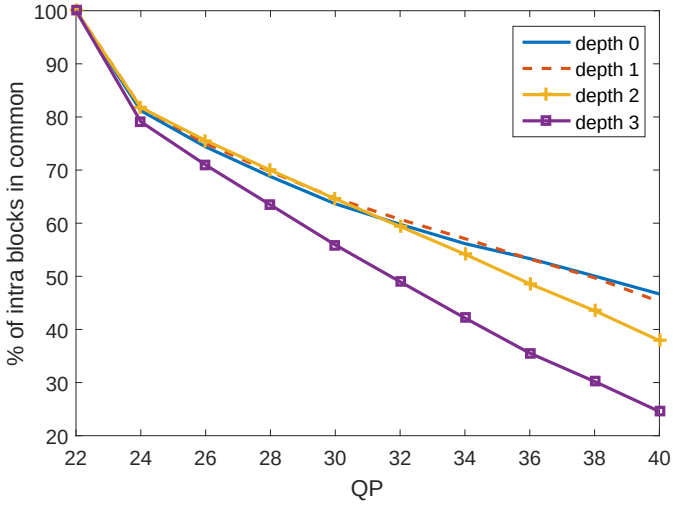


Figure 4.6: Percentage of inter blocks in inter predicted frames as a function of the QP.

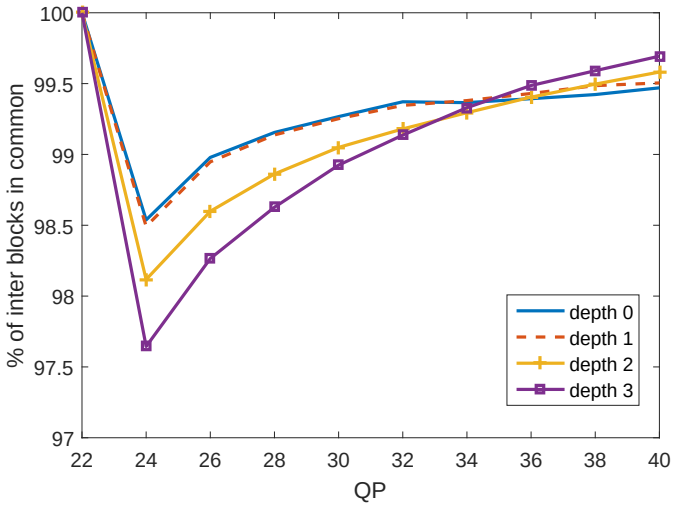
expected, as an encoding is compared with itself. As the QP increases to 24, only 80% of the CUs intra encoded at QP 22 are still intra encoded. This means that if the intra encoding information from the reference at QP 22 would be directly reused, a suboptimal decision would be made for 20% of the CUs. The percentage of intra CUs in common decreases further as the QP increases. It is concluded from these results that the intra encoding information cannot be reused from a high quality reference encoding to skip the inter analysis part, as this would lead to a large number of suboptimal decisions and thus to a decreased RD performance. As the prediction mode is a binary decision, the percentage of decisions in common should be very high so that only a small number of suboptimal decisions are made.

Figure 4.7b shows the percentage of inter encoded CUs in the reference encoding (QP 22) that is still inter encoded in the lower quality encodings, as a function of the QP. Unlike the intra case, a very high percentage of inter CUs in common can be observed across the range of QPs, with a minimum around 97.6% at QP 24 and depth 3. These results indicate that a CU that is encoded in inter mode in the reference encoding will be inter encoded in a lower quality representation with a very high probability.

A low quality reference (QP 40) is also tested to check if it could alternatively be used to speed up higher quality dependent encodings (QP 22 to 38). Figures 4.8a and 4.8b show that in that case, the percentage of inter and intra CUs in common with the reference can be as low as 83% and 80%, respectively. This indicates that a substantial number of suboptimal decisions would be made if a low quality reference was to be used as a reference. This means that a low-quality reference is a worse choice than a high-quality reference in terms of overall RD performance. As a high quality reference is already used in the CU structure reuse method, a high quality reference is kept for the prediction mode reuse method and for the rest of the thesis.



(a) intra CUs



(b) inter CUs

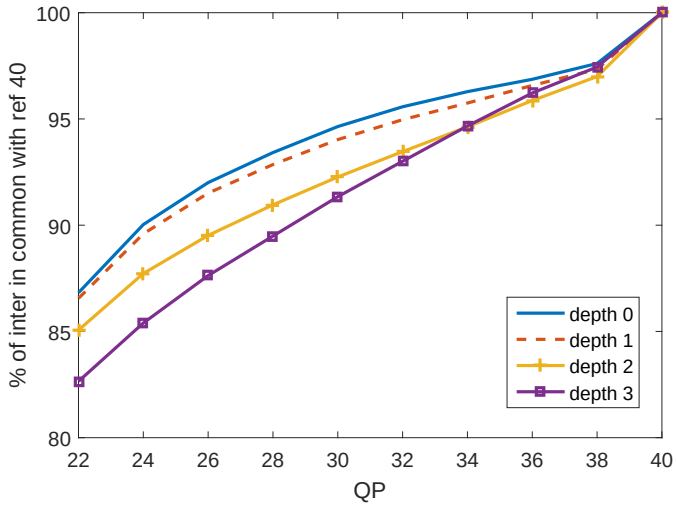
Figure 4.7: Percentage of intra or inter CUs in common with the reference at QP 22 at different depths.

4.4.2 Information reuse

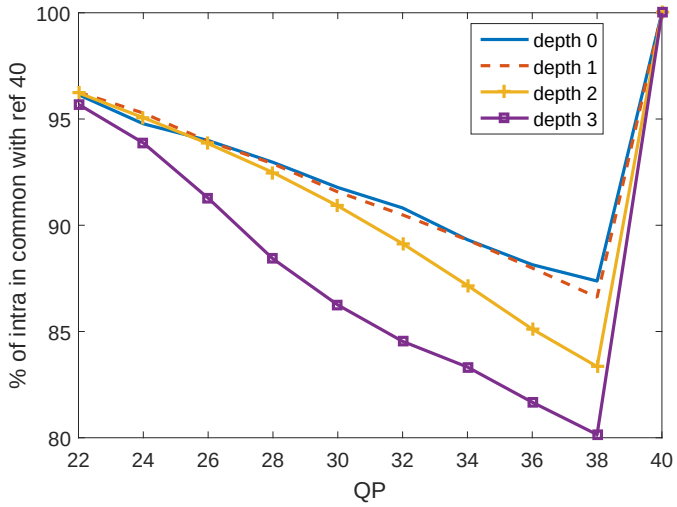
Given the preceding observations, information about the prediction mode from a high-quality reference encoding is proposed to be reused in order to speed up the RDO of lower-quality dependent encodings. Specifically, the prediction mode decision at every CU of the quadtree during the RDO of the reference encoding is gathered. That is, 1 decision is stored at depth 0, 4 decisions at depth 1, 16 decisions at depth 2, and 64 decisions at depth 3.

If the decision from the reference encoding is inter mode for a specific CU, intra prediction for that same CU is not checked in the dependent encodings, because the decision for the CU will be inter with a very high probability. This information reuse scheme leads to a suboptimal decision with a small probability. The few CUs with suboptimal decision will contribute to a small decrease in RD performance.

On the other hand, the inter analysis part cannot be skipped if there is an intra encoded



(a) inter CUs



(b) intra CUs

Figure 4.8: Percentage of intra or inter CUs in common with the reference at QP 40 at different depths.

CU in the reference encoding, because this would lead to numerous suboptimal decisions and thus substantially harm the overall RD performance.

4.4.3 Results

The proposed method is implemented to assess the impact of reusing only the prediction mode decision on a multi-rate system. The HM based implementation is compared with the original HM encoder, and the results for the main set are listed in Table 4.5. On average, the proposed method shows a BD-rate increase of approximately only 0.15%, while the overall encoding time over 4 representations is reduced by 1.60%. The average time gain is relatively small, which is due to the fact that the inter analysis part cannot be skipped, which would have resulted in higher time gains (cf. Table 4.1). Additionally, the videos tend to have more inter encoded CUs than intra encoded CUs in the *random access, main* profile (cf. Figure 4.6).

Table 4.5: Comparison of encoding with prediction mode reuse vs. conventional encoding for the main set at 1080p.

Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	0.12%	−0.005 dB	−0.58%
CrowdRun	0.13%	−0.005 dB	−3.04%
DucksTakeOff	0.02%	−0.0001 dB	−1.96%
Kimono	0.09%	−0.003 dB	−2.17%
ParkJoy	0.07%	−0.003 dB	−2.92%
ParkScene	0.08%	−0.003 dB	−1.21%
PedestrianArea	0.64%	−0.020 dB	−1.28%
RiverBed	0.07%	−0.003 dB	−0.49%
RushHour	0.30%	−0.007 dB	−1.56%
Sunflower	0.02%	−0.002 dB	−0.78%
Average	**0.15%**	**−0.005 dB**	**−1.60%**

Table 4.6: Comparison of encoding with prediction mode reuse vs. conventional encoding for the alternative set.

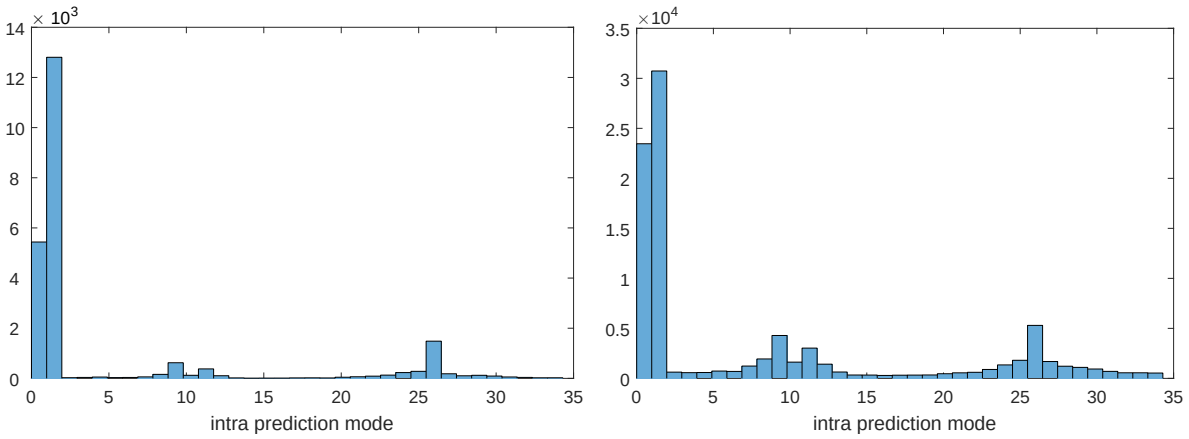
Resolution	Sequence	BD-rate	BD-PSNR	ΔT
416 × 240	*BasketballPass*	0.28%	−0.014 dB	−0.39%
	BlowingBubbles	0.10%	−0.004 dB	−2.71%
832 × 480	*BQMall*	0.18%	−0.008 dB	−1.00%
	PartyScene	0.10%	−0.004 dB	−2.39%
2560 × 1600	*PeopleOnStreet*	0.36%	−0.017 dB	−3.37%
	Traffic	0.09%	−0.003 dB	−2.47%
	Average	**0.19%**	**−0.008 dB**	**−2.06%**

Further results for the alternative set of videos at different spatial resolutions are listed in Table 4.6. On average, the encoding time is reduced by 2.06% while the average BD-rate is increased by 0.19%. These results are comparable to the results from the main set.

4.5 Intra mode reuse

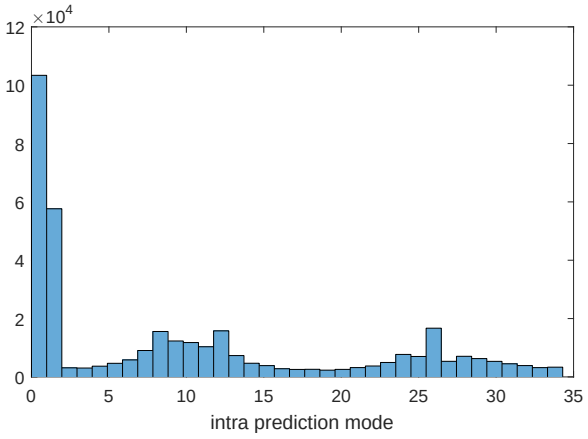
4.5.1 Observations

An intra PU is characterized by its intra prediction mode, which can be planar prediction (mode 0), DC prediction (mode 1) or one of 33 angular predictions (modes 2 to 34), which sums up to 35 different possibles intra prediction modes in HEVC [27] (cf. Section 2.1.1.2). An intra CU at depth between 0 and 2 always contains only one PU, whereas a CU at depth 3 can contain one PU or four square PUs [26]. From a PU perspective, this last partitioning is equivalent to a depth 4.

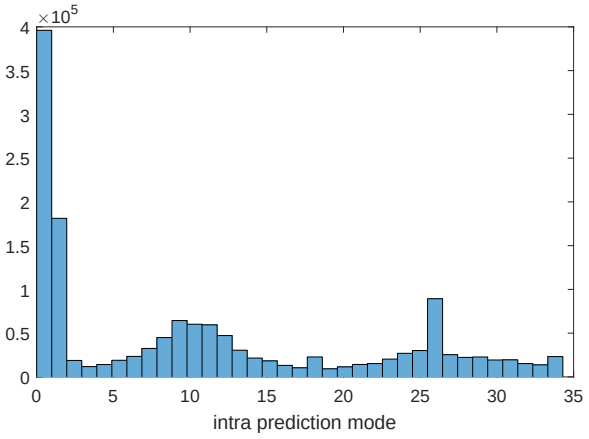


(a) depth 0

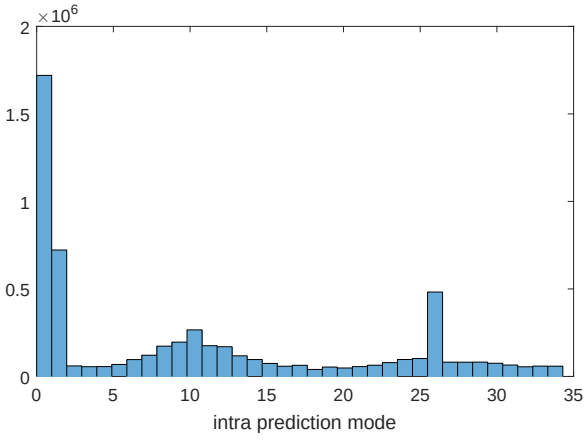
(b) depth 1



(c) depth 2

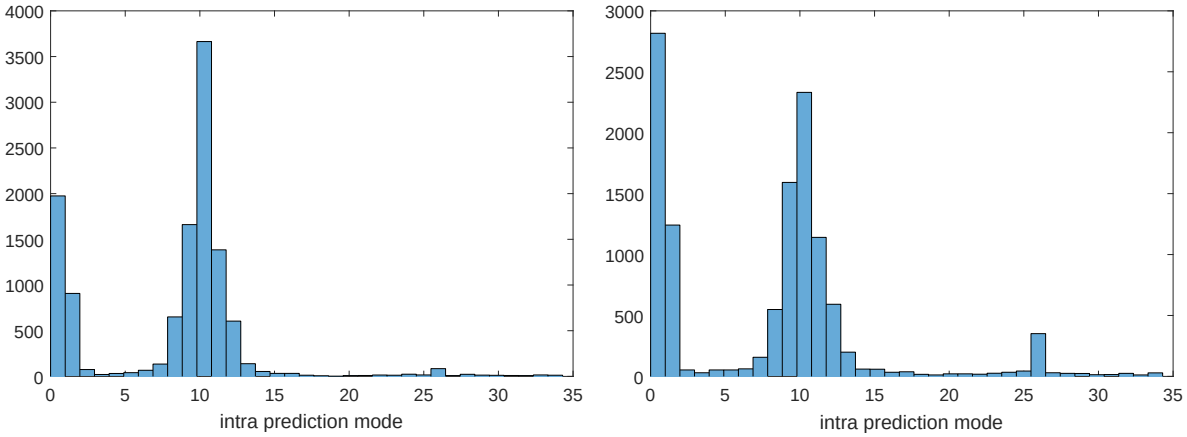


(d) depth 3



(e) depth 4

Figure 4.9: Histograms of the luma intra prediction mode for 10 videos of the main set at QP 22 and different PU depths.



(a) QP 30 (b) QP 40

Figure 4.10: Histograms of the luma intra prediction mode at depth 2 for 10 videos of the main set at different QPs, for PUs which were intra mode 10 at QP 22.

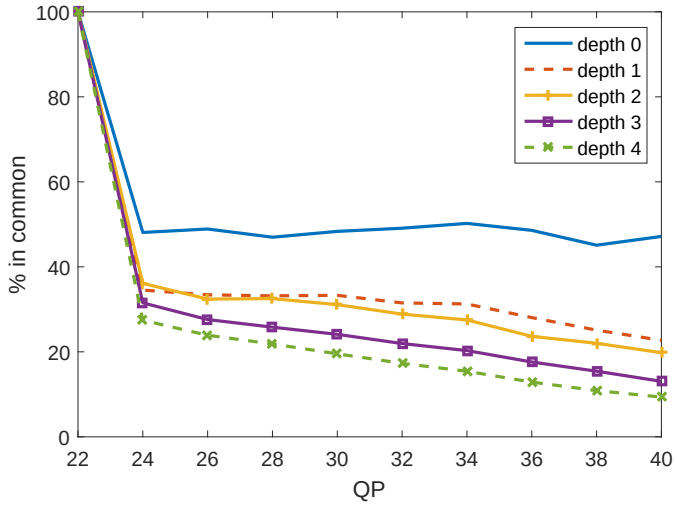


Figure 4.11: Percentage of PUs with intra mode 10 at QP 22, which are still intra mode 10 at lower quality QPs at different depths.

The 10 first frames of the 10 videos from the main set are encoded with the *intra, main* profile [83] (that is, only I-frames) with QP ranging from 22 to 40, in order to assess the similarities in intra mode from the luma component across videos with different qualities. The distribution of the intra prediction modes at different depths for the QP 22 videos are shown as histograms in Figure 4.9. The easiest way to reuse the intra mode information in a multi-rate system would be to directly reuse the intra mode of a PU from the reference for the same PU in a low-quality encoding.

Figure 4.10 shows the intra mode of PUs at depth 2 for encodings at QP 30 and 40, respectively, which were intra mode 10 (horizontal prediction) in the reference encoding at QP 22. At QP 30 (Figure 4.10a), intra mode 10 is still the intra mode with the most elements, however, it accounts for only 32% of all PUs. This means that directly reusing the intra mode 10 from the reference at QP 22 for a low quality encoding at QP 30 would lead to 68% of

suboptimal decisions at depth 2 for these PUs. At QP 40 (Figure 4.10a), intra mode 10 is not the intra mode with the most elements, but it still accounts for 20% of all PUs. Figure 4.11 shows what percentage of PUs which are intra mode 10 at QP 22 are still intra mode 10 at QPs between 22 and 40 and different depths. The values range between 10% and 50%. Similar trends were observed with other intra modes. It is inferred from these observations that the intra mode cannot be reused directly. However, the intra mode from the reference can still be considered a “good candidate", with a probability between 10% and 50%.

4.5.2 Information reuse

Calculating the full RD costs for all 35 intra modes is too complex to be practical. Thus, HM implements a suboptimal fast intra algorithm which first evaluates an approximated cost for all 35 modes and then makes a candidate list with the best 3 or 8 candidates (depending on the PU size) which are in turn fully analyzed [27] (cf. Section 2.1.4.1). Based on the observation that the intra mode from the reference is a “good candidate", it is proposed to reduce the candidate list to 3 for all PU sizes and then check if the reference intra mode is in this list. If it is not, then the reference intra mode is added to this short list, which then contains 4 candidates to be fully analyzed. The choice not to reduce the list down to less than 3 candidates comes from the fact that the approximated cost is sensitive to the 3 *most probable intra modes* defined in HEVC [27] (cf. Section 2.1.1.2).

4.5.3 Results

The proposed method is implemented to assess the impact of reusing only the intra mode information on a multi-rate system. First results with the *random access, main* profile lead to an average encoding time reduction of 0.88% and to a BD-rate increase of 0.004% for the main set of videos. The low encoding time reduction comes from the high number of inter encoded frames, where the intra mode reuse does not have a big impact.

The videos are now encoded with the *intra, main* profile in order to focus on I-frames only, where the intra mode reuse method will have the highest impact. The results for the main set are presented in Table 4.7. The average time gain of almost 14% comes at the expense of a very small BD-rate increase of 0.03%. A further observation is that the RD performance is actually improved compared to the original HM encoder for the *Kimono, PedestrianArea, Riverbed, RushHour* and *Sunflower* sequences. This confirms that the intra mode information from a high quality can be considered a “good candidate". Although the proposed method is specific to the HM encoder, it is believed that similar reuse schemes can be explored for various HEVC encoders.

The results for the alternative set are presented in Table 4.8. The average encoding time reduction is 14.36% whereas the average BD-rate increase is 0.23%.

Table 4.7: Comparison of encoding with intra mode reuse vs. conventional encoding for the main set at 1080p.

Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	0.08%	−0.005 dB	−14.16%
CrowdRun	0.28%	−0.02 dB	−13.60%
DucksTakeOff	0.05%	−0.0001 dB	−14.18%
Kimono	−0.10%	0.004 dB	−14.33%
ParkJoy	0.17%	−0.01 dB	−13.97%
ParkScene	0.12%	−0.005 dB	−13.41%
PedestrianArea	−0.02%	0.001 dB	−13.53%
Riverbed	−0.06%	0.002 dB	−13.67%
RushHour	−0.15%	0.004 dB	−13.49%
Sunflower	−0.05%	0.003 dB	−13.49%
Average	**0.03%**	**−0.003 dB**	**−13.78%**

Table 4.8: Comparison of encoding with intra mode reuse vs. conventional encoding for the alternative set.

Resolution	Sequence	BD-rate	BD-PSNR	ΔT
416 × 240	*BasketballPass*	0.14%	−0.008 dB	−14.76%
	BlowingBubbles	0.32%	−0.018 dB	−14.02%
832 × 480	*BQMall*	0.27%	−0.017 dB	−14.93%
	PartyScene	0.35%	−0.026 dB	−14.49%
2560 × 1600	*PeopleOnStreet*	0.15%	−0.008 dB	−14.36%
	Traffic	0.14%	−0.007 dB	−13.58%
	Average	**0.23%**	**−0.014 dB**	**−14.36%**

4.6 Motion vector reuse

4.6.1 Observations

Inter predicted frames rely on a motion-compensated prediction based on previously encoded frames. An inter CU contains either one PU (called 2N×2N), or two, or four PUs [26] (cf. Section 2.1.1.1). Each PU is characterized by one or two two-dimensional motion vectors (MV) that point to the predictor block in a specified reference frame. The *random access, main* profile has an encoding structure with B-frames, that is, frames can be predicted from two reference frames. The reference frames are listed in two lists L0 and L1.

The MVs of a video at different qualities are examined. Therefore, the MVs of the 2N×2N PU at each CU depth found during the inter analysis process are compared for 10 videos encoded with a QP ranging from 22 to 40. As an example, Figures 4.12 and 4.13 show the MVs at depth 0 and list L0 from the second frame of the *BlueSky* sequence encoded at QP 22 and 24, respectively. Blocks with no displayed MVs are intra predicted at depth 0. The MVs are

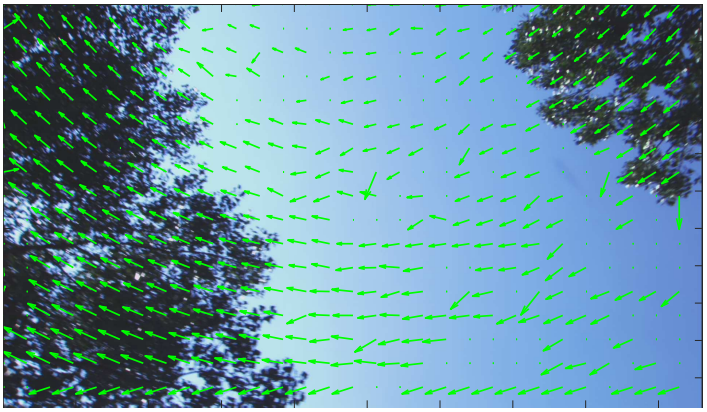


Figure 4.12: MVs at depth 0 and list L0 for the second frame of *BlueSky* at QP 22.

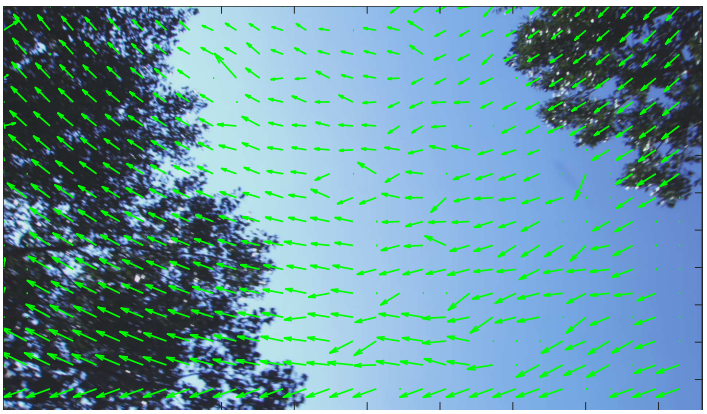


Figure 4.13: MVs at depth 0 and list L0 for the second frame of *BlueSky* at QP 24.

scaled uniformly for better visualization. Strong similarities can be observed, and comparable similarities have been observed at different qualities and for other videos. However, there is always a small MV difference. Thus, the MVs from the reference encoding cannot be directly reused. To quantify the similarity, it is determined if the difference vector of an MV with the corresponding MV in the reference at QP 22 has a norm smaller than 4 pixels. Figure 4.14 shows the percentage of PUs that have an MV difference with the corresponding reference MV smaller than 4 pixels in the case of the L0 list. This percentage is around 95% at depth 0, whereas it can go down to 85% at depths 2 or 3. Results for the L1 list are very similar.

4.6.2 Information reuse

Based on the insight that the MVs at lower quality encodings are very similar to the MVs from the high-quality reference encoding, it is proposed to restrict the motion estimation to the vicinity of the reference MV in the dependent encodings. For this, the MV of the 2N×2N PU in the reference encoding for each possible CU (i.e., at each node in the quadtree) are first collected for both the L0 and L1 lists. The index of the reference frame in the lists are not collected.

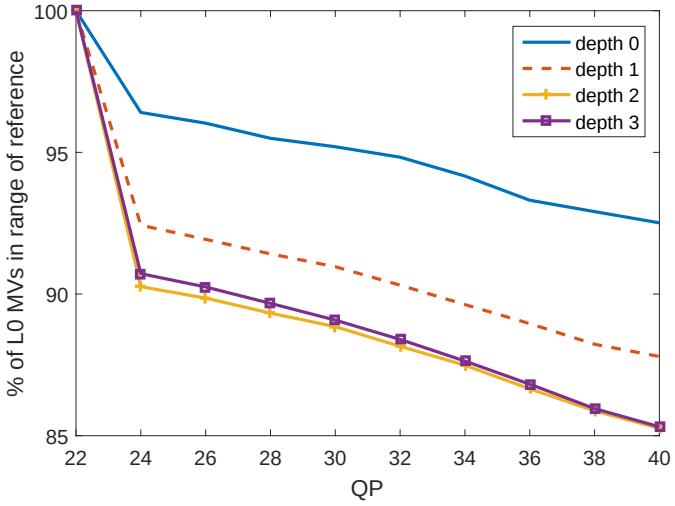


Figure 4.14: Percentage of PUs that have a MV difference with the reference MV smaller than 4 pixels for the L0 list.

The HM encoder implements a test zone (TZ) search algorithm [35], [87] (combination of diamond search and raster search) with a default search range of 64 pixels in the *random access*, *main* profile and up to 2 reference frames in both lists L0 and L1. The TZ search algorithm is initialized with either the 0 vector or with a vector predicted from neighboring blocks (cf. Section 2.1.4.2).

In the proposed method, the TZ search algorithm is initialized with the MV from the reference encoding for the corresponding list. The search range is restricted to 4 pixels and the raster search part of the TZ search is deactivated. As the reference frame information is not available, the motion estimation is still run for all possible reference frames in each list. The dependent encoder is not restricted to the 2N×2N PU partitioning, and uses the reference MV from the 2N×2N PU to initialize the motion estimation of the different possible PU partitions in the same CU.

4.6.3 Results

Table 4.9 shows the comparison results of the implementation of the proposed MV information reuse method with the original HM encoder for the main set at 1080p. On average, the proposed method can reduce the encoding time over 4 representations by 6.21%. Interestingly, the proposed reuse method improves the RD performance, as the average BD-rate is decreased by 0.12% and the average BD-PSNR is increased by 0.004 dB. On one hand, this is due to the fact that the original HM encoder does not perform a full-search motion estimation, and thus, does not always find the optimal MV in the RD sense. On the other hand, it shows that initializing the motion estimation with a good guess is beneficial in terms of RD performance, even if only one reference MV per CU is used although there can be multiple PUs, and even if the search range is drastically reduced, as in the proposed method. Even though the proposed method is examined in the context of the HM encoder, it is believed that

Table 4.9: Comparison of encoding with MV reuse vs. conventional encoding for the main set at 1080p.

Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	0.07%	−0.003 dB	−4.17%
CrowdRun	−0.08%	0.003 dB	−4.89%
DucksTakeOff	−0.09%	0.002 dB	−5.60%
Kimono	−0.06%	0.002 dB	−6.25%
ParkJoy	−0.20%	0.008 dB	−5.71%
ParkScene	−0.05%	0.002 dB	−2.73%
PedestrianArea	−0.19%	0.006 dB	−8.75%
Riverbed	−0.05%	0.002 dB	−11.58%
RushHour	−0.09%	0.002 dB	−6.16%
Sunflower	−0.52%	0.016 dB	−6.25%
Average	**−0.12%**	**0.004 dB**	**−6.21%**

Table 4.10: Comparison of encoding with MV reuse vs. conventional encoding for the alternative set.

Resolution	Sequence	BD-rate	BD-PSNR	ΔT
416 × 240	*BasketballPass*	0.22%	−0.010 dB	−1.87%
	BlowingBubbles	−0.15%	0.007 dB	−3.86%
832 × 480	*BQMall*	−0.06%	0.003 dB	−4.17%
	PartyScene	−0.03%	0.001 dB	−4.44%
2560 × 1600	*PeopleOnStreet*	−0.13%	0.006 dB	−7.27%
	Traffic	−0.05%	0.002 dB	−4.89%
	Average	**−0.03%**	**0.002 dB**	**−4.42%**

the reuse of MV information is also beneficial for both the RD performance and the encoding time of other encoders that do not rely on a full-search motion estimation.

Table 4.10 shows the comparison results for the alternative set at different spatial resolutions. The average encoding time is reduced by 4.42% and the RD performance is increased, as the BD-rate is decreased by 0.03%, similarly to the case of the main set in Table 4.9.

4.6.4 Comparison with the state-of-the-art

The hybrid motion vector reuse method by Cebrián-Márquez *et al.* uses the MVs from an H.264/AVC encoding to speed up the motion estimation in an HEVC encoder (cf. Section 2.3.3.4). The results presented in their paper [82] are based on HM 16.6 and use the same configuration as in this thesis (*random access*, *main* profile and QPs 22, 27, 32, and 37). Their results are compared with the results of the proposed MV reuse method in Table 4.11. On average, Cebrián-Márquez *et al.* achieve a time reduction of 8.41% while the proposed method achieves a lower time reduction of 4.44%. However, the proposed method

Table 4.11: Comparison of the proposed MV reuse method with a state-of-the-art method.

	Proposed		Cebrián-Márquez *et al.* [82]	
Sequence	**BD-rate**	**ΔT**	**BD-rate**	**ΔT**
BasketballPass	0.22%	−1.87%	0.9%	−10.46%
BlowingBubbles	−0.15%	−3.86%	0.8%	−6.46%
BQMall	−0.06%	−4.17%	0.7%	−8.40%
PartyScene	−0.03%	−4.44%	0.8%	−6.63%
Kimono	−0.06%	−6.25%	0.7%	−10.15%
ParkScene	−0.05%	−2.73%	0.5%	−8.14%
PeopleOnStreet	−0.13%	−7.27%	1.8%	−10.20%
Traffic	−0.05%	−4.89%	1.0%	−6.80%
Average	**−0.04%**	**−4.44%**	**0.9%**	**−8.41%**

improves the average RD performance, with a BD-rate decrease of 0.04%, while the method by Cebrián-Márquez *et al.* degrades the RD performance with an average BD-rate increase of 0.9%. Their larger time reduction can be explained by the fact that their reference encoding is the H.264/AVC encoding, which is not taken into account in the time reduction calculation. Thus, all four HEVC representations are dependent encodings. On the other hand, as the proposed method only encodes with HEVC, the representation at best quality (QP 22) is a reference encoding and is thus not accelerated in the proposed method.

4.7 Combination of methods

So far, different encoding decisions (CU structure, prediction mode, intra prediction mode and motion vectors) that can be reused from a high quality reference encoding to constrain the RDO of lower quality dependent encodings have been identified. Methods to reuse this information from the reference encoding have been proposed and the results show that each method can reduce the overall encoding complexity, while the RD performance is only very slightly degraded or even improved.

From the results, it can seen that the CU structure reuse leads to the highest encoding time reduction among the proposed methods. This was expected from the preliminary study (cf. Table 4.1), as the analysis of a CU encloses the prediction mode decision as well as the underlying intra and inter analysis. The proposed prediction mode reuse does not offer a large encoding time reduction, mostly because the observations show that the decision of intra encoding cannot be reused for lower quality dependent encodings if the RD performance of the multi-rate system has to be kept high. For the intra prediction mode reuse, the results show that the proposed method primarily makes sense in an all-intra encoding case, where all frames are affected by the proposed method. Finally, the motion vector reuse method shows that an average improvement in RD performance can be achieved simultaneously to

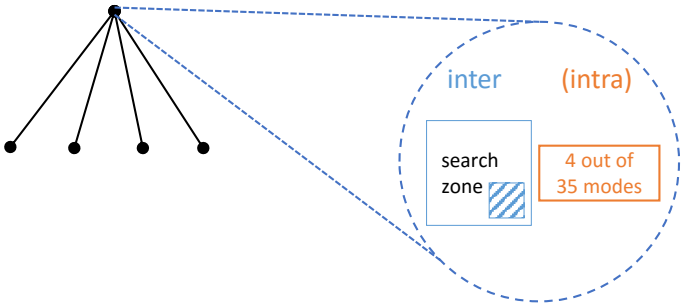


Figure 4.15: Conceptual schema of the constrained RDO in the proposed multi-rate encoder: Compared to the original RDO, see Figure 2.4, the quadtree traversal is shortened, the intra analysis is potentially skipped, also fewer intra modes and a smaller inter-prediction motion vector search zone are considered.

Table 4.12: Encoding results for a combination of CU structure and prediction mode reuse.

Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	0.42%	−0.016 dB	−40.39%
CrowdRun	0.60%	−0.025 dB	−22.13%
DucksTakeOff	0.27%	−0.007 dB	−26.88%
Kimono	0.75%	−0.025 dB	−40.36%
ParkJoy	0.40%	−0.017 dB	−29.01%
ParkScene	0.63%	−0.021 dB	−35.39%
PedestrianArea	1.02%	−0.032 dB	−37.44%
Riverbed	0.44%	−0.019 dB	−41.59%
RushHour	0.29%	−0.010 dB	−36.76%
Sunflower	0.99%	−0.023 dB	−39.84%
Average	**0.58%**	**−0.020 dB**	**−34.98%**

an encoding time reduction.

As the proposed methods have considered different steps of the RDO, they can potentially be combined. In this section, the effect of the combination of the proposed methods is examined. Figure 4.15 conceptually shows how the RDO in the proposed multi-rate encoder is constrained. The quadtree to be analyzed is shortened with the CU structure reuse method. The intra analysis part can be skipped with the prediction mode reuse method. In the intra mode reuse method, the number of intra modes to be checked is reduced. Finally, in the motion vector reuse method, the size of the search zone in the motion estimation is reduced.

4.7.1 CU structure and prediction mode

The CU structure reuse method, which achieves the largest encoding time reduction, is first combined with the prediction mode reuse method. The results of the combination are listed in Table 4.12. Compared to the CU structure reuse method alone, the average encoding

Table 4.13: Encoding results for a combination of CU structure, prediction mode, and intra prediction mode reuse.

Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	2.23%	−0.086 dB	−41.28%
CrowdRun	1.43%	−0.058 dB	−23.44%
DucksTakeOff	0.57%	−0.014 dB	−28.03%
Kimono	1.26%	−0.042 dB	−40.57%
ParkJoy	0.91%	−0.037 dB	−29.80%
ParkScene	1.59%	−0.051 dB	−36.41%
PedestrianArea	2.33%	−0.071 dB	−37.83%
Riverbed	0.32%	−0.018 dB	−41.89%
RushHour	1.78%	−0.061 dB	−37.27%
Sunflower	1.24%	−0.032 dB	−40.29%
Average	**1.37%**	**−0.047 dB**	**−35.68%**

time reduction is increased from 33.61% to 34.98%. The difference (1.37 percentage points) is slightly less than the encoding time reduction achieved with the prediction mode reuse method alone (1.60%, Table 4.5). This can be explained by the fact that the CU structure reuse methods skips the analysis of some CUs, and thus skips the underlying prediction mode decision as well. In the case of the RD performance, the combination increases slightly the BD-rate from 0.53% to 0.58%. The difference (0.05 percentage points) is also slightly less than the BD-rate increase of the prediction mode reuse method alone (0.15%, Table 4.5).

4.7.2 CU structure, prediction mode, and intra prediction mode

The CU structure is next combined with the prediction mode and the intra prediction mode reuse methods and the results for the main set are presented in Table 4.13. Compared to CU structure and prediction mode (Table 4.12), the average encoding time reduction is slightly increased from 34.98% to 35.68%. However, the RD performance is largely reduced, as the average BD-rate increases from 0.58% to 1.37%. This may be explained by the fact that the intra prediction mode reuse affects the I-frames, which are used as reference for the following P and B-frames.

4.7.3 CU structure, prediction mode, and motion vectors

As the addition of the intra mode reuse in a *random access, main* profile does not bring a significant advantage in encoding time reduction, but decreases the RD performance, it is not further considered in the combination of methods. Thus, the combination of CU structure, prediction mode and motion vectors reuse methods is now tested.

Encoding results are presented in Table 4.14. Compared to the CU structure and prediction mode combination, the encoding time reduction is increased by 2.36 percentage points,

Table 4.14: Encoding results for a combination of CU structure, prediction mode, and motion vectors reuse.

Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	0.47%	−0.017 dB	−41.57%
CrowdRun	0.44%	−0.018 dB	−24.20%
DucksTakeOff	−1.07%	0.027 dB	−29.83%
Kimono	0.80%	−0.027 dB	−42.80%
ParkJoy	0.30%	−0.012 dB	−29.83%
ParkScene	0.65%	−0.021 dB	−37.86%
PedestrianArea	1.17%	−0.036 dB	−41.06%
Riverbed	0.47%	−0.020 dB	−45.49%
RushHour	0.53%	−0.010 dB	−38.65%
Sunflower	0.87%	−0.016 dB	−42.10%
Average	**0.46%**	**−0.015 dB**	**−37.34%**

which is less than the encoding time reduction achieved by the motion vector reuse method alone (6.21%, Table 4.9), which can be explained, again, by the CU structure reuse methods, which skips the analysis of entire CUs, and thus, also the underlying motion estimation. The RD performance is improved from 0.58% BD-rate increase to only 0.46% BD-rate increase. This confirms that the reuse of motion vectors from a high quality reference encoding is beneficial for the RD performance.

4.8 Summary

In this chapter, an HEVC multi-rate encoding system that encodes representations at a single spatial resolution and different SNR qualities has been examined. Based on observations of similarities between different representations, methods that reuse encoding information from a high-quality reference encoding to speed up lower quality dependent encodings have been proposed. The goal of the methods is to reduce the overall encoding complexity while keeping the RD performance as close as possible to the RD performance of a system with independent single-layer encoders. Therefore, the reuse methods have been designed to constrain the RDO of the dependent encodings. Four encoding decisions from the reference encoding that can constrain the RDO have been identified: the CU structure, the prediction mode, the intra prediction mode and the motion vectors. Encoding results of the proposed methods compared to conventional encoding show that the encoding time of multiple representations can be reduced while the RD performance is almost not degraded or even improved in the case of the motion vectors reuse method. The different proposed methods can also be combined, which leads to a larger overall encoding time reduction at a very low RD performance decrease.

Chapter 5

Multi-rate encoding with multiple spatial resolutions

5.1 Introduction

The different representations of an adaptive HTTP streaming system generally span multiple spatial resolutions. The main reason is to accommodate for different streaming devices, as a mobile device such as a smartphone probably requires a lower resolution than a static device such as a television. Furthermore, varying the spatial resolution is an effective way to vary the bitrate of the encoded video, and thus, a wide range of different bitrates can be achieved by encoding at different resolutions.

Unlike the preceding chapter, the encoding of a single video sequence is now considered at different resolutions in this chapter. The goal is again to reduce the overall computational complexity. The video at highest resolution is chosen to be the reference encoding. Intuitively, the video at highest resolution contains the most information, because the downsampling process to achieve a lower resolution is a lossy process.

Depending on the downsampling ratio, there might not be a direct correspondence (from a covered frame area point of view) between blocks of the representations at different resolutions. This is identified as the main challenge for multi-rate encoding at different resolutions, because the encoding decisions in HEVC are taken at block level, and thus, it is not possible to directly map a decision from a high-resolution reference to a lower resolution dependent encoding if there is no correspondence in the block structure.

Therefore, in this chapter, methods to extract information from a high-resolution reference to then constrain the RDO of dependent encodings in a multiple resolutions perspective are evaluated. In Section 5.2, the reuse of the CU structure is considered. The prediction mode reuse is presented in Section 5.3 and the reuse of the intra prediction mode information is examined in Section 5.4. The proposed methods are combined in Section 5.5 to form an efficient multi-rate encoder, which is compared to a multi-rate encoder from related work. Finally, Section 5.6 summarizes the chapter.

5.2 CU structure reuse

5.2.1 CU structure similarities

To determine the similarities in the CU structure of a video encoded at different resolutions, a test video is encoded at three different resolutions (original at 1920×1080 pixels and two downsampled versions at 1280×720 and 640×360 pixels). Figure 5.1 shows the CU structure for the 20th frame of the *ParkScene* sequence at QP 22 and with a CTU size chosen as 64×64 pixels at all three resolutions. Although the 64×64 CTUs do not cover the same image area at different resolutions, certain areas of the frame will be encoded similarly at different resolutions, in the sense that homogeneous regions such as the tree on the left tend to be coded with large CUs, whereas frame regions with a high detail level tend to be coded with small CUs. A similar behavior is observed in other videos as well.

5.2.2 CU matching across resolutions

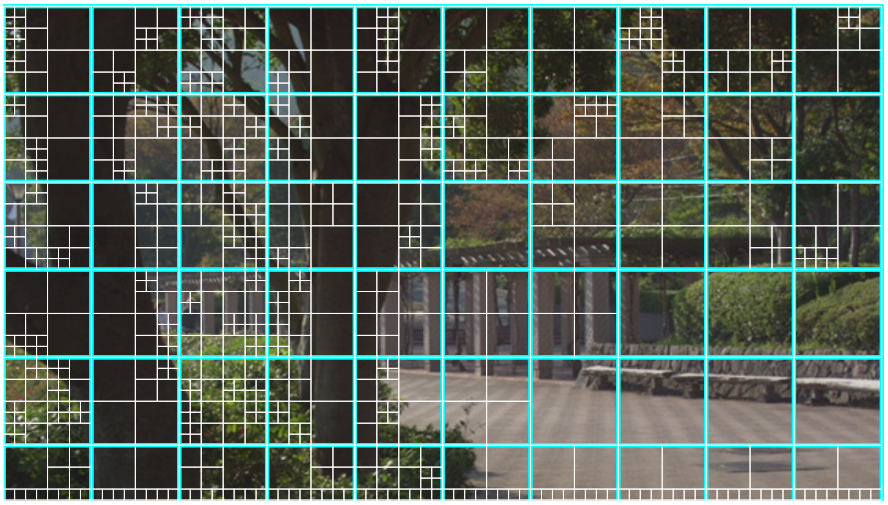
To reuse the CU structure information from a high-resolution encoding of the video to speed up lower-resolution encodings, the CU structure at a high resolution needs to be matched to the CU structure at a low resolution. In the case where the downsampling ratio is a power of 2, the CU structure can be easily matched across resolutions due to the CTU quadtree structure. E.g., Figure 5.2 shows that a CU at depth 1 at 360p corresponds to a CU at depth 0 at 720p from the perspective of the frame area covered. However, there is no direct correspondence between the CUs at different resolutions if the downsampling ratio is different than a power of 2. As an example, Figure 5.2 shows that a CTU at 720p covers a frame area which is larger than one CTU but smaller than four CTUs at 1080p.

5.2.3 CU structure extraction algorithm

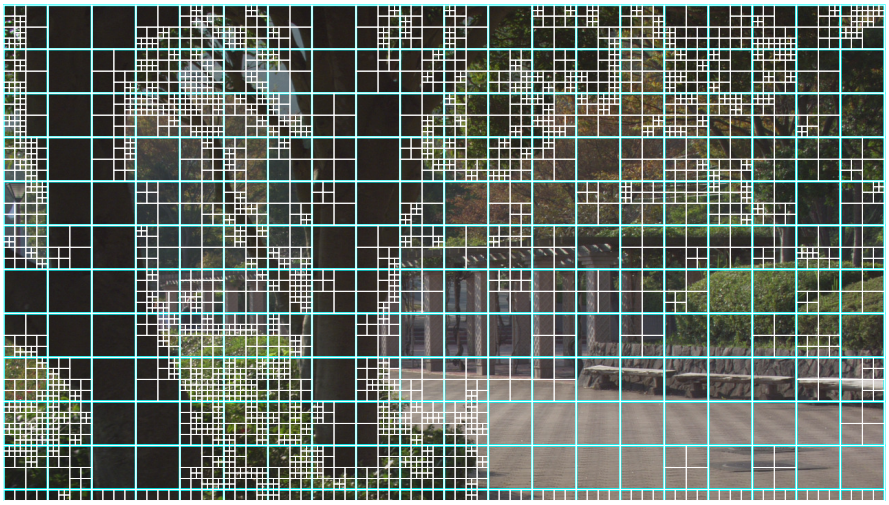
To be able to reuse CU structure information from a high-resolution reference encoding for lower-resolution dependent encodings with an arbitrary downsampling ratio, an algorithm which extracts CU structure information from the high-resolution video is proposed. The output of the algorithm is a virtual CU structure at a low resolution, which is called *extracted CU structure*. On the other hand, an *original encoding* is an independent encoding with an unmodified HEVC encoder.

The proposed extraction of the CU structure information is done at the CTU level, i.e., the extracted CU structure is computed for the low resolution representation CTU by CTU.

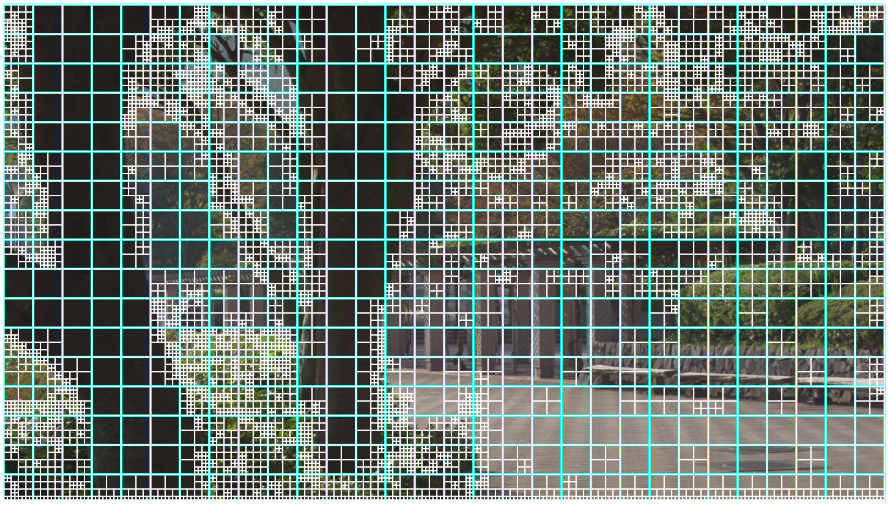
The algorithm follows a quadtree depth-first traversal, that is, first, the CU at depth 0 is selected in the low-resolution video. Then, the area A in the reference encoding that corresponds to the current CU is selected. The percentage p_0 of A encoded at depth (less than or equal to) 0, i.e., highest possible CU size, is then determined. In general, the percentage p_i with $i \in \{0, 1, 2\}$ is defined as the percentage of the corresponding area in the reference encoding with depth less or equal to i.



(a) 640 × 360



(b) 1280 × 720



(c) 1920 × 1080

Figure 5.1: CTU (blue) and CU (white) structure of the 20th frame of the *ParkScene* sequence encoded at QP 22.

(a) 360p (b) 720p (c) 1080p

Figure 5.2: Correspondence between CTUs (blue) of size 64×64 pixels and CUs (white) at different resolutions for a specific frame area.

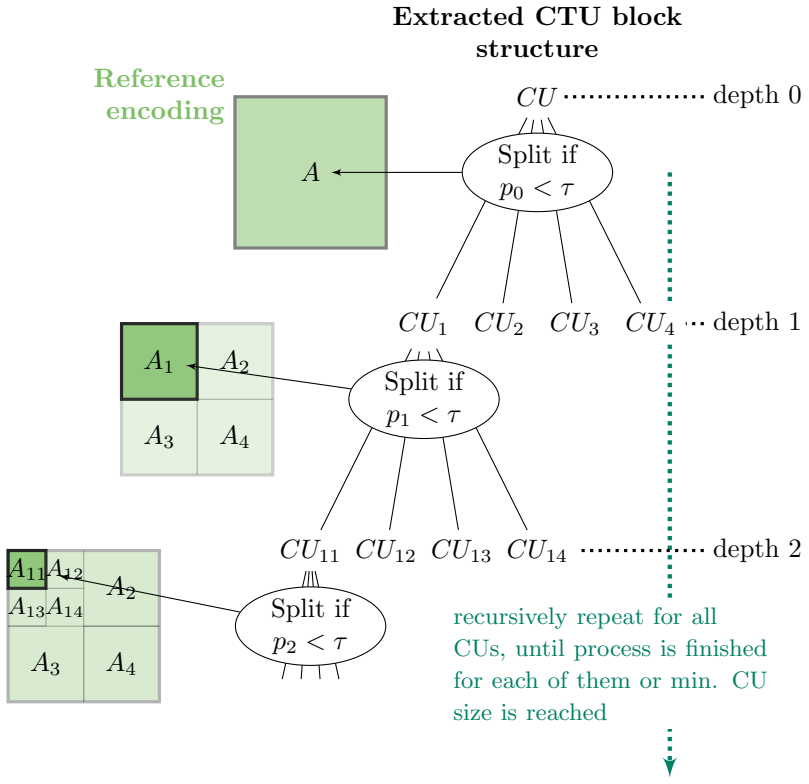


Figure 5.3: Algorithm to extract the CU structure from a high resolution reference encoding, for a threshold τ.

If p_0 is greater than or equal to a threshold τ, then the current CU is not split and the process moves on to the next CTU. On the contrary, if p_0 is less than τ, then the current CU is split into four smaller CUs at depth 1 (CU_1, CU_2, CU_3 and CU_4). This process is recursively repeated for all the CUs in order to traverse the quadtree, until the process is finished for each CU or the minimum CU size is reached. As an example, the decision whether to split/not split each of the depth 1 CUs is examined. Starting with CU_1, the area A_1 in the reference

encoding that corresponds to CU_1 is selected. Now, the percentage p_1 of A_1 at depth less than or equal to 1 is measured. If p_1 is greater than or equal to τ, then CU_1 is not split and the next CU is considered. However if p_1 is less than τ, CU_1 is further split into four smaller CUs at depth 2 (CU_{11}, CU_{12}, CU_{13} and CU_{14}). The algorithm is represented graphically in Figure 5.3.

The threshold τ determines how conservative the algorithm is. For instance, for $\tau = 100$, a CU will be extracted at highest possible CU size only if 100% of A is at depth 0. In comparison, for $\tau = 60$, a CU will be extracted at highest possible CU size only if at least 60% of A is at depth 0, which means that a part of the area A can have a higher depth.

Figure 5.4 shows the extracted CU structure for the 20th frame of the *Parkscene* sequence at 720p (extracted from a 1080p reference encoding), for two different thresholds. It can be seen that for $\tau = 60$, more CUs tend to be extracted at larger CU size compared to $\tau = 80$. The original CU structure obtained from an independent encoding with an unmodified encoder is also shown for comparison in Figure 5.4c.

5.2.4 Similarity quantification

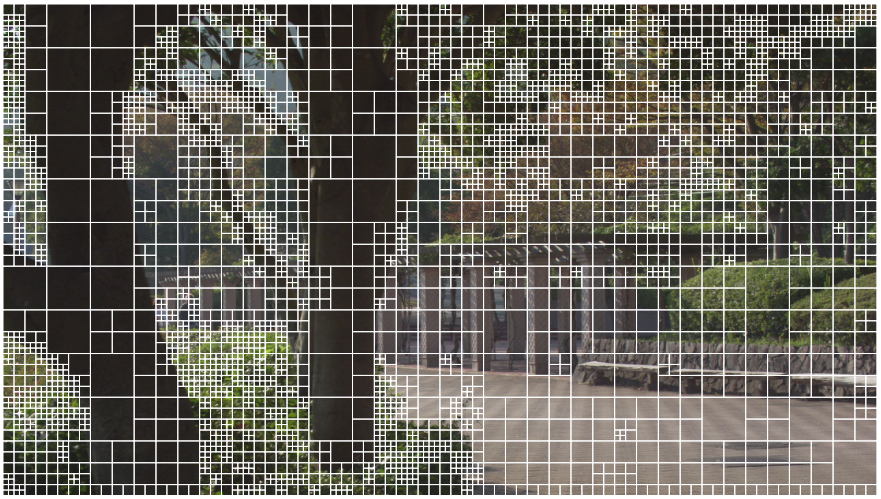
In order to quantify the similarity between the extracted CU structure and the CU structure of the original encoding, the percentage of the area of the frames where the CUs have the same depth is calculated. If the CU depth is not identical, it can either have a greater depth (i.e., a smaller CU size) or lower depth (i.e., a larger CU size). Figure 5.5 shows the comparison between Figures 5.4b and 5.4c. The yellow region is where both have same CU depths, the dark green indicates that the original encoding has CUs at a greater depth and the light green indicates the original encoding has CUs at a lower depth. To generalize the results, the methodology is repeated for the 10 sequences of the main set and the percentages of these regions for two different values of τ are shown in Figure 5.6.

When τ decreases, the number of CUs of the extracted structure at lower depth increases. Thus, the percentage of area having greater depth in the original structure increases, which can be seen by the increase in the dark green region in Figure 5.6, when τ goes from 80 to 60. The percentage of the area where the CU depth is not identical (sum of dark and light green regions) is not negligible. Therefore, the extracted CU structure for the dependent encodings cannot be directly reused. Still, a majority of CUs of the original structure will have lower or same depth as the extracted CU structure. E.g., for $\tau = 80$, roughly 95% of the frame area will have lower or same depth.

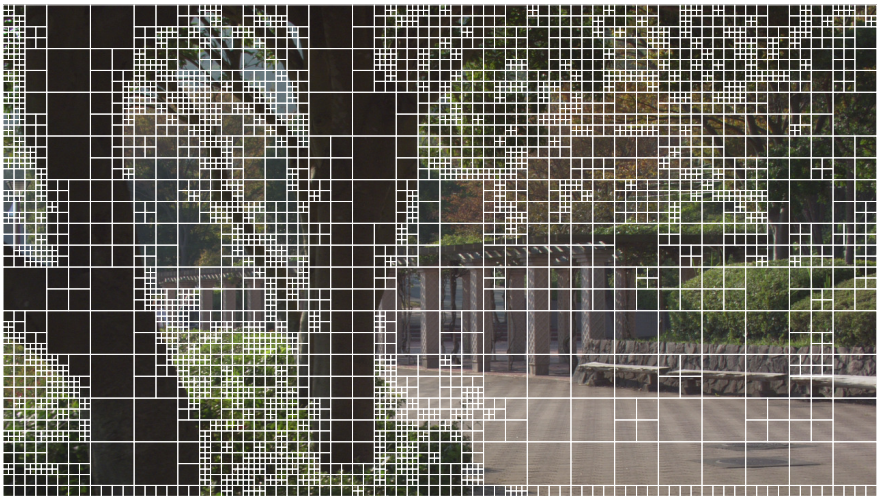
5.2.5 Extracted CU structure reuse

As seen in the preceding section, most of the area in the original encoding either has lower or same depth as the extracted CU structure. Combining this observation and the fact that the RDO process of the HEVC encoder is implemented recursively starting from the lowest depth, similar to Section 4.3, the RDO process of the dependent low resolution encoding is

(a) Extracted CU structure $\tau = 80$



(b) Extracted CU structure $\tau = 60$



(c) CU structure of the original encoding

Figure 5.4: Extracted CU structure ($\tau = 60$ and $\tau = 80$) for a frame of the *ParkScene* (720p) sequence from a reference encoding at 1080p and original encoding.

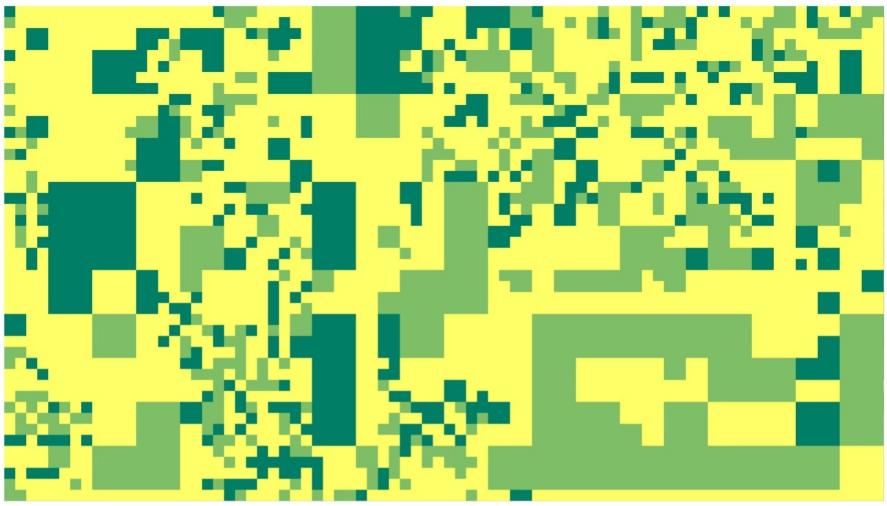
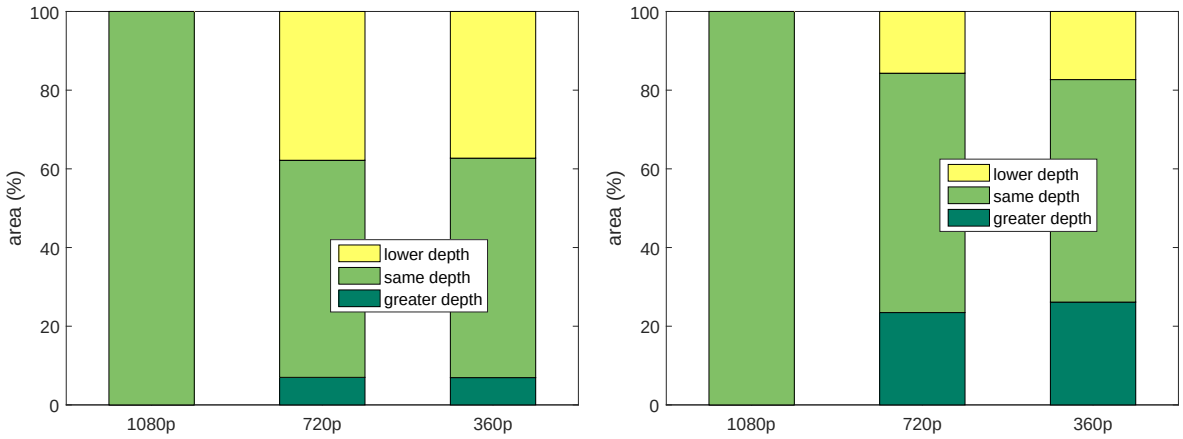


Figure 5.5: Areas in the original CU structure of Figure 5.4c with greater (dark green), same (yellow) or lower (light green) depth when compared with the extracted CU structure ($\tau = 60$) shown in Figure 5.4b.



(a) $\tau = 80$ (b) $\tau = 60$

Figure 5.6: Average percentage of areas in the original encoding with depths lower, identical or greater than the extracted CU structure for 10 sequences.

stopped at the depth given by the extracted CU structure. As shown in Figure 5.6, there is a small percentage of the area where the depth is greater in the original encoding than in the extracted CU structure. As the RDO process is stopped at the depth of the extracted CU structure, a suboptimal CU size in the RD sense will be chosen for the dependent encoding. However, the overall RD loss due to such cases should be small because this concerns only a small percentage of the frame area, e.g., roughly 5% (dark green region) in Figure 5.6a for $\tau = 80$.

In general, when τ is decreased, the percentage of the area at greater depth in the original encoding (dark green region) increases (cf. Figure 5.6). Consequently, there will be a higher RD loss. In the case of a lower τ value, on average, the RDO process is stopped earlier than for a higher τ value and so there will be higher encoding time savings. Thus, there is a trade-

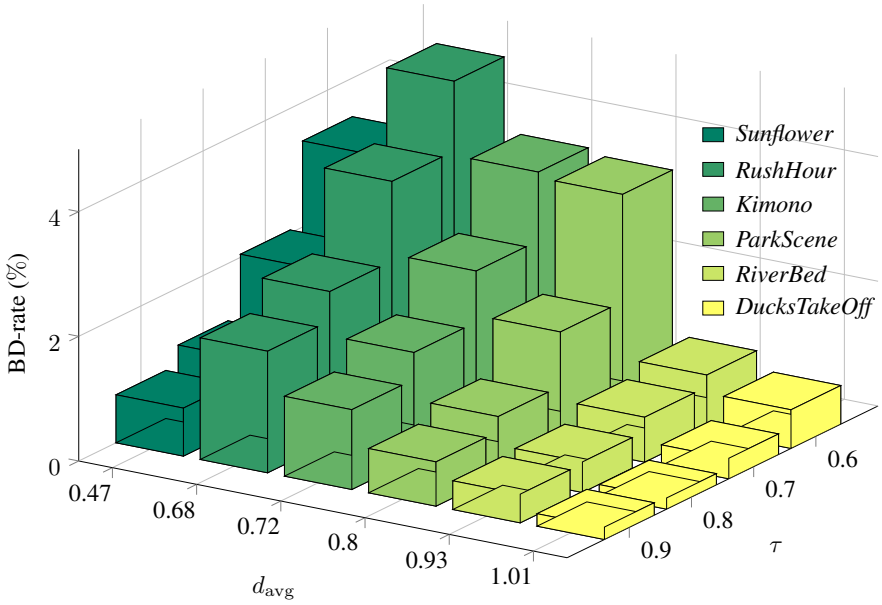


Figure 5.7: BD-rate for different thresholds for the 720p sequence and average depths of the reference encoding.

off between the encoding time savings and the RD performance loss, which can be balanced by τ.

5.2.6 Threshold determination

5.2.6.1 Observations

How to choose the threshold τ for the proposed method is discussed next. To gain first insights, the proposed method is evaluated for different values of τ. The proposed implementation is compared with the unmodified HM encoder.

The 1080p reference video sequence is encoded at four different QPs (22, 27, 32, and 37). The CU structure is then extracted based on τ for each of the four references. The dependent encodings are encoded at the same QPs and use the extracted CU structure from the reference at the same QP.

Six video sequences are used for this initial comparison: *Sunflower*, *RushHour*, *Kimono*, *ParkScene*, *RiverBed* and *DucksTakeOff*. The proposed method is evaluated for four values of τ (60, 70, 80 and 90). The resulting BD-rate is shown in Figure 5.7. The average depth of the CUs d_{avg} of the 1080p reference encoding is also calculated. This is a weighted average where the weight is the percentage of the frame area that the CU represents. Two observations can be made. First, for all sequences, a larger value of τ leads to a lower BD-rate than for a smaller value of τ, as explained in Section 5.2.5. Second, for a fixed τ, sequences with lower d_{avg} tend to have a higher BD-rate than the ones with higher d_{avg}. This can be intuitively explained by the impact of stopping the RDO before actually reaching the optimal depth. The probability of not reaching the optimal depth increases if the reference depth, and thus

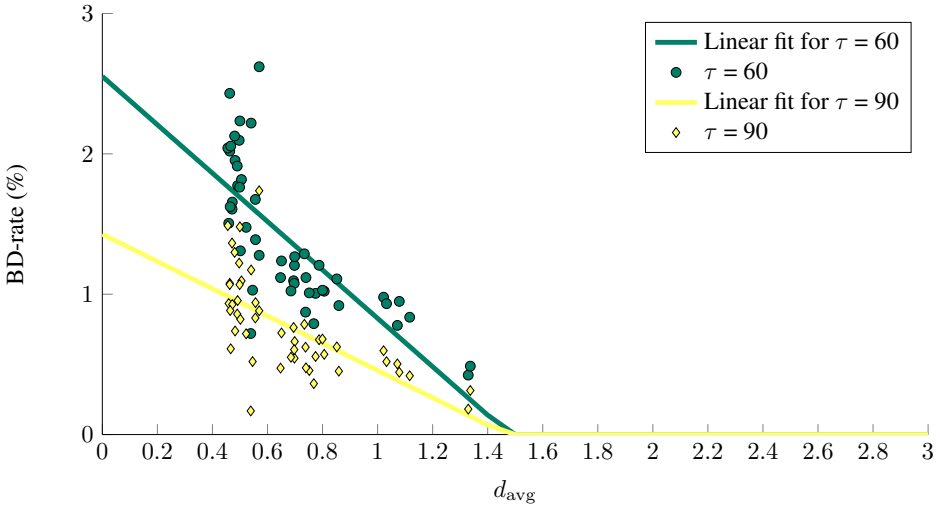


Figure 5.8: Scatter plot of the BD-rate as a function of the average depth d_{avg} for each frame of the *RiverBed* sequence and linear fit of the BD-rate for different values of τ. d_{avg} is calculated per frame from the reference 1080p sequence.

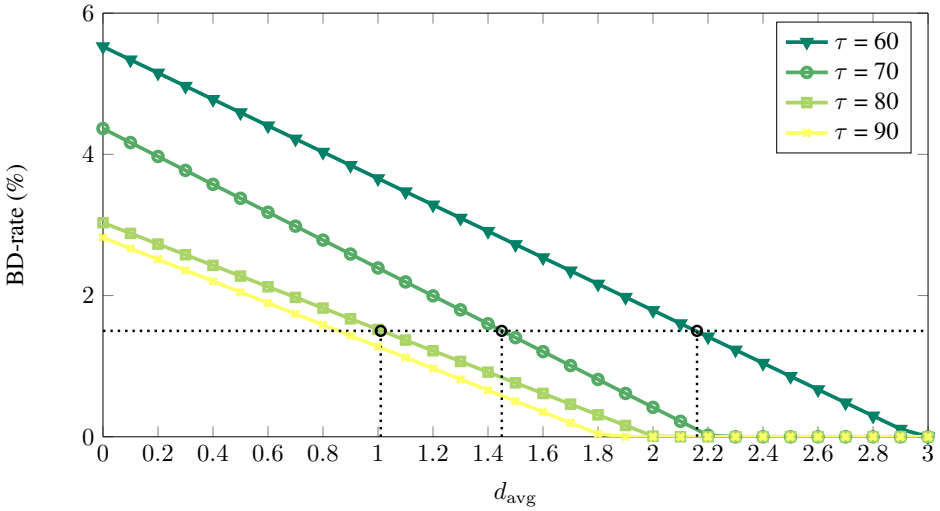


Figure 5.9: Linear fit of the BD-rate as a function of the average depth for different values of τ for the 10 sequences of the main set.

the derived depth, are low. In order to balance this effect, a large value of τ should be used for sequences with a low d_{avg}. Similarly, a comparatively small value of τ can be used for a sequence with a high d_{avg}.

In a practical scenario, the parameter τ should be available before the entire sequence is encoded at the reference high resolution in order to encode the dependent low resolution representations in parallel. Determining τ on a frame by frame basis enables to encode the dependent frame directly after the reference frame has been encoded. Figure 5.8 shows the average depth d_{avg} from each 1080p reference frame, and the resulting BD-rate at 720p for 50 frames of the *Riverbed* video for two different τ values. A linear curve is fitted to the point cloud as a first order approximation in order to get a coarse estimation of the BD-rate de-

Table 5.1: Mapping between d_{avg} and τ for 720p.

d_{avg}	[0, 1.01)	[1.01, 1.45)	[1.45, 2.16)	[2.16, 3]
τ	90	80	70	60

Table 5.2: Mapping between d_{avg} and τ for 360p.

d_{avg}	[0, 1.5)	[1.5, 1.65)	[1.65, 1.79)	[1.79, 3]
τ	90	80	70	60

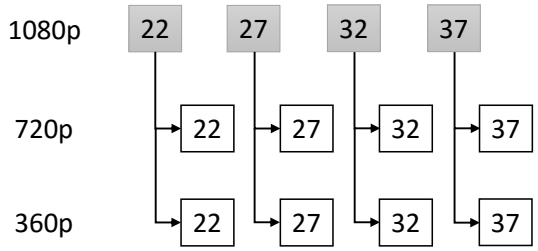


Figure 5.10: Reference encodings (gray) and dependent encodings (white) with QPs in the multiple references case.

pending on the average depth. In Figure 5.9, the process is repeated and averaged over the 10 sequences of the main set.

5.2.6.2 Proposed threshold choosing method

In order to keep the BD-rate at a low level while reducing the encoding time, the value of τ is adapted for every frame based on the value d_{avg} of the current frame of the reference encoding. An arbitrary low BD-rate value of 1.5% is chosen and based on Figure 5.9, a simple mapping described in Table 5.1 is proposed for 720p dependent encodings, where the average depth d_{avg} of the current reference frame is mapped to a threshold τ for the corresponding dependent encoding frame. A threshold of 100 is not considered as this results in a negligible encoding time gain. The same methodology is used to find a mapping for the 360p dependent encodings as shown in Table 5.2.

5.2.7 Results

The proposed CU structure reuse method across resolutions uses a high resolution reference to extract the CU structure which is used to speed up the dependent encodings at lower resolution. This means that there is no dependencies across the SNR dimension (that is, if the QP is varied). However, in order to calculate the BD-rate, four representations at a specific spatial resolution need to be encoded. The first possibility is to use one reference encoding per value of QP, as shown in Figure 5.10. Tables 5.3 and 5.4 show the performance of the proposed encoding method compared to the original HM reference for 720p and 360p, respectively, when multiple references are used. The encoding time for the 4 low-resolution

Table 5.3: Comparison of encoding results for 720p with multiple 1080p references.

Sequence	**BD-rate**	**BD-PSNR**	**ΔT**
BlueSky	0.95%	-0.05 dB	59.40%
CrowdRun	1.88%	-0.07 dB	62.25%
DucksTakeOff	0.10%	0.00 dB	44.37%
Kimono	1.42%	-0.06 dB	50.68%
ParkJoy	0.52%	-0.02 dB	31.76%
ParkScene	0.86%	-0.03 dB	45.54%
PedestrianArea	2.39%	-0.10 dB	47.30%
RiverBed	0.49%	-0.02 dB	49.11%
RushHour	2.17%	-0.07 dB	51.78%
Sunflower	1.24%	-0.05 dB	65.53%
Average	**1.20%**	**-0.05 dB**	**50.77%**

Table 5.4: Comparison of encoding results for 360p with multiple 1080p references.

Sequence	**BD-rate**	**BD-PSNR**	**ΔT**
BlueSky	1.57%	-0.10 dB	55.36%
CrowdRun	4.39%	-0.22 dB	58.82%
DucksTakeOff	0.10%	0.00 dB	43.61%
Kimono	3.04%	-0.13 dB	46.13%
ParkJoy	0.66%	-0.03 dB	31.19%
ParkScene	0.82%	-0.04 dB	42.71%
PedestrianArea	3.87%	-0.21 dB	41.32%
RiverBed	0.66%	-0.03 dB	47.32%
RushHour	5.08%	-0.2 dB1	48.77%
Sunflower	1.44%	-0.08 dB	61.68%
Average	**2.16%**	**-0.11 dB**	**47.69%**

representations is reduced on average by 50.77% and 47.69% for 720p and 360p, respectively. In terms of RD performance, the dependent 720p encodings show an average BD-rate increase of 1.20%. The dependent 360p encodings have an average BD-rate increase of 2.16%.

The encoding time reduction is calculated for the lower resolution encodings only, which is illustrated as an example for the *ParkScene* sequence in Figure 5.11. However, the encoding time increases when the spatial resolution increases. Thus, the encoding times for the reference encodings at 1080p are the largest times. In a practical scenario, it thus doesn't make sense to have multiple high-resolution references, as the overall encoding time reduction (including the reference encoding) would be relatively low. In the following, only one high-resolution and high quality (QP 22) reference is used, as illustrated in Figure 5.12.

The results for the CU structure reuse in the case of a single reference encoding at 1080p

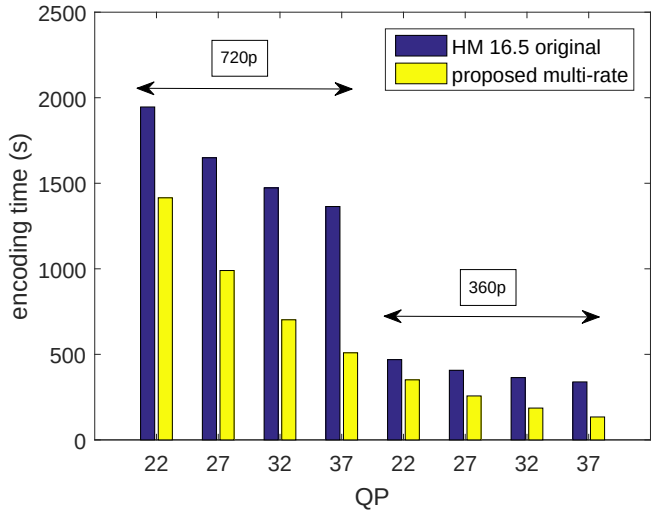


Figure 5.11: Encoding time for the *ParkScene* sequence in the case of multiple references.

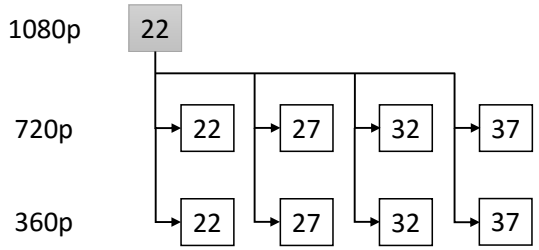


Figure 5.12: Reference encoding (gray) and dependent encodings (white) with QPs in the single reference case.

and QP 22 are presented in Tables 5.5 and 5.6. The average time reduction of 38.62% and 31.81% for 720p and 360p, respectively, is lower than in the case with multiple references (when it is calculated for the low-resolution representations only). However, the RD performance is better with an average BD-rate increase of 0.76% and 0.93%, respectively. This can be explained by the fact that only the highest quality reference is used, which has the largest CU depth. This puts the weakest constraint on the RDO and thus leads to the best RD performance.

Table 5.5: Comparison of encoding results for 720p based on a single 1080p reference, when the CU structure is reused.

Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	0.60%	−0.03 dB	−50.86%
CrowdRun	0.57%	−0.03 dB	−19.48%
DucksTakeOff	0.13%	0.00 dB	−22.91%
Kimono	1.02%	−0.04 dB	−44.10%
ParkJoy	0.41%	−0.02 dB	−25.57%
ParkScene	0.82%	−0.03 dB	−34.47%
PedestrianArea	1.39%	−0.06 dB	−40.01%
Riverbed	0.36%	−0.02 dB	−46.15%
RushHour	1.27%	−0.04 dB	−43.04%
Sunflower	1.00%	−0.04 dB	−59.64%
Average	**0.76%**	**−0.03 dB**	**−38.62%**

Table 5.6: Comparison of encoding results for 360p based on a single 1080p reference, when the CU structure is reused.

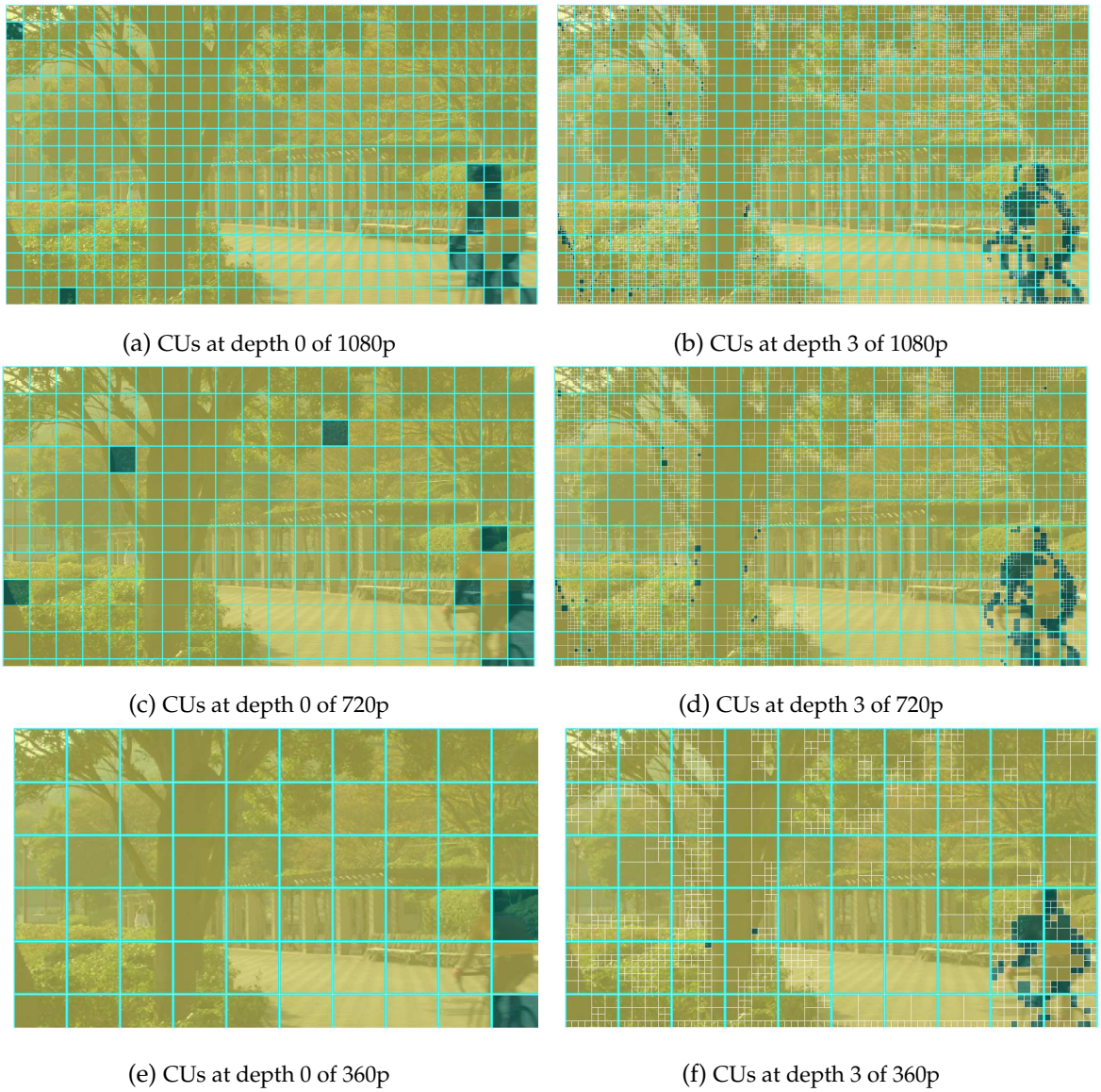
Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	0.59%	−0.04 dB	−44.43%
CrowdRun	0.45%	−0.02 dB	−16.61%
DucksTakeOff	0.17%	−0.01 dB	−22.80%
Kimono	1.70%	−0.07 dB	−33.57%
ParkJoy	0.36%	−0.02 dB	−20.78%
ParkScene	0.55%	−0.02 dB	−27.73%
PedestrianArea	2.11%	−0.11 dB	−30.85%
RiverBed	0.46%	−0.02 dB	−35.25%
RushHour	2.38%	−0.10 dB	−34.03%
Sunflower	0.55%	−0.03 dB	−52.03%
Average	**0.93%**	**−0.04 dB**	**−31.81%**

5.3 Prediction mode reuse

5.3.1 Observations

The idea from Section 4.4 of reusing the prediction mode decision is applied to the case where multiple spatial resolutions of the video have to be encoded. First, the similarities in prediction modes at different resolutions and different CU depths are analyzed.

As an example, Figure 5.13 shows the prediction mode decision of the 55th frame of the *ParkScene* sequence at different resolutions, where a cyclist first enters the frame from the right, both for depth 0 CUs and depth 3 CUs. Similarities can be seen across resolutions such as the background being mostly inter predicted, because the background information



(a) CUs at depth 0 of 1080p

(b) CUs at depth 3 of 1080p

(c) CUs at depth 0 of 720p

(d) CUs at depth 3 of 720p

(e) CUs at depth 0 of 360p

(f) CUs at depth 3 of 360p

Figure 5.13: Inter (yellow) and intra (green) mode decision for the 55th frame of the *ParkScene* sequence encoded at QP 22.

is already present in other frames. On the contrary, the cyclist entering the frame at the right bottom corner is partly intra predicted. This behavior can be observed at different resolutions and at different depths. Similarities were also observed at depths 1 and 2.

5.3.2 Prediction mode extraction algorithm

As explained in the preceding section, the challenge of reusing information from a high-resolution reference encoding comes from the fact that there is no direct correspondence (i.e., overlap) between blocks at different resolutions if the downsampling factor is not a power of 2.

Therefore, an algorithm to determine the prediction mode of a CU at each depth for the

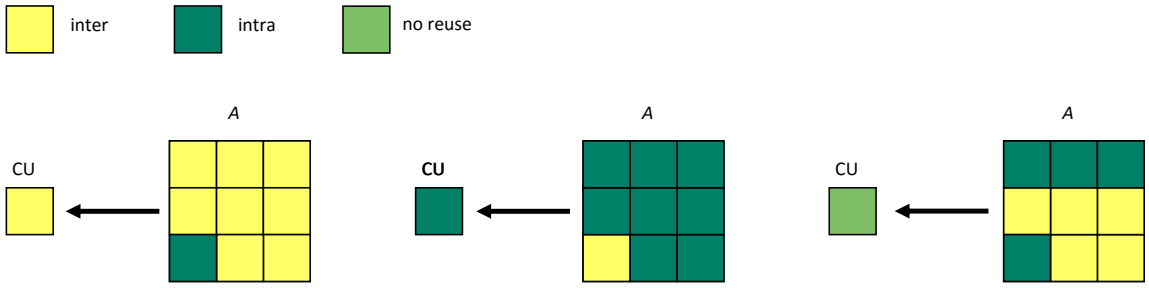


Figure 5.14: Extraction of the prediction mode for a low-resolution CU from its corresponding area A in the reference encoding for $\theta = 80$.

dependent low-resolution encodings is proposed: for a CU of the low-resolution video at depth i, the corresponding area A of the reference at the same depth i is selected. The percentage p of A which is encoded with inter prediction is measured. The prediction mode of the current CU is then determined depending on the value of p according to Table 5.7. As a parameter, the threshold θ can take a value between 50 and 100.

Table 5.7: Prediction mode decision for multiple resolutions according to inter-prediction percentage p.

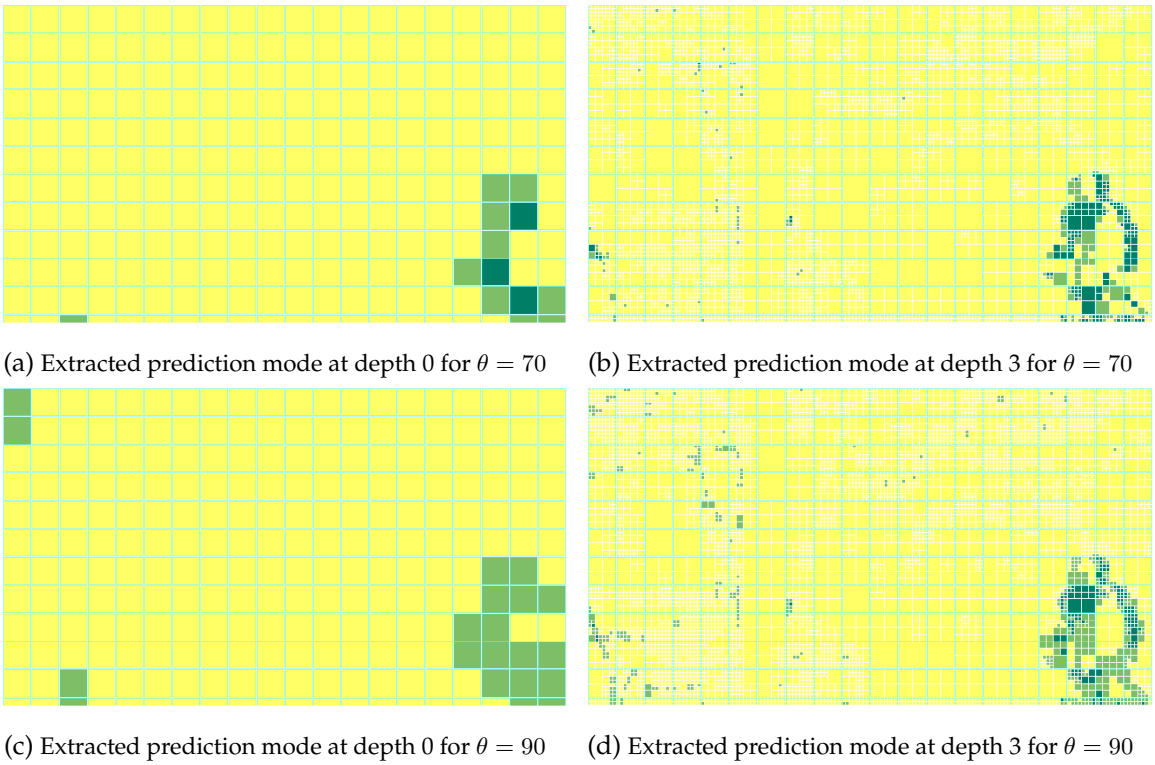
Condition	Prediction mode
$p \geq \theta$	*inter*
$p < 100 - \theta$	*intra*
else	*no reuse*

The algorithm is illustrated with an example in Figure 5.14, where the threshold θ is set to 80. If more than 80% of the high-resolution area A corresponding to the current CU is inter predicted, then the current CU is set to inter prediction as well. If less than 20% of the corresponding area is inter predicted, then the current CU is set to intra prediction. Finally, if the percentage of the area that is inter predicted is between 20% and 80%, the current CU is set to "no reuse" mode.

5.3.3 Prediction mode reuse

If a CU is set to inter mode, the intra analysis part is skipped during the RDO of the dependent encoding. Similarly, if a CU is set to intra, the inter analysis part is skipped during the RDO of the dependent encoding. Finally, for the "no reuse" case, neither analysis part is skipped.

Figure 5.15 shows the extracted prediction mode for the CUs of the 55th frame of the *ParkScene* sequence at depth 0 and 3 and for θ values 70 and 90. More CUs are set to the "no reuse" mode for the larger value of θ. Indeed, if the value of θ increases, the condition for a CU to be set to a specific prediction mode gets harsher. On the contrary, if the value of θ increases, the probability for a CU to be set to the "no reuse" mode increases. On one hand,



(a) Extracted prediction mode at depth 0 for $\theta = 70$

(b) Extracted prediction mode at depth 3 for $\theta = 70$

(c) Extracted prediction mode at depth 0 for $\theta = 90$

(d) Extracted prediction mode at depth 3 for $\theta = 90$

Figure 5.15: Extracted prediction modes for the 55th frame of the *ParkScene* (720p) sequence (extracted from the reference encoding at 1080p) for different depths and different values of θ. Yellow, dark green, and light green correspond to *inter* mode, *intra* mode, and *no reuse*, respectively.

having more "no reuse" CUs means that the RD performance will be close to the original performance, as no analysis part is skipped during the dependent encoding. On the other hand, the encoding time gains decrease with more "no reuse" CUs.

For a preliminary evaluation, the effect of the proposed method is evaluated on four different video sequences at 720p using different values of θ. Table 5.8 shows the RD performance and encoding time decrease of the proposed method compared to the reference encoder. For the following, a value of $\theta = 80$ is selected, as it presents a good compromise between encoding time reduction and BD-rate increase.

5.3.4 Results

Comparison results for the main set at 720p and 360p using one 1080p reference at QP 22 and a threshold value $\theta = 80$ are presented in Tables 5.9 and 5.10. The average BD-rate increase is kept low at 0.58% and 0.84% for 720p and 360p, respectively. The average time reduction is higher than in the single resolution case (Section 4.4). This is mainly due to the fact that both inter and intra analysis parts can be skipped in the multi-resolution case, whereas only the intra analysis part can be skipped in the method proposed for the single-resolution case.

Table 5.8: Comparison of encoding results for 720p sequences, with different values of θ.

Sequence	θ	**BD-rate**	$\mathbf{\Delta T}$
DucksTakeOff	60	2.10%	−20.05%
	70	1.98%	−19.97%
	80	1.77%	−19.65%
	90	1.62%	−18.86%
	100	1.48%	−14.32%
Kimono	60	1.01%	−8.97%
	70	0.91%	−8.68%
	80	0.74%	−8.48%
	90	0.70%	−7.89%
	100	0.69%	−7.84%
ParkJoy	60	1.02%	−10.12%
	70	0.79%	−9.80%
	80	0.47%	−9.29%
	90	0.42%	−8.53%
	100	0.36%	−7.06%
ParkScene	60	0.63%	−3.51%
	70	0.41%	−3.48%
	80	0.24%	−3.22%
	90	0.22%	−2.93%
	100	0.19%	−2.25%
Average	**60**	**1.19%**	**−10.66%**
	70	**1.02%**	**−10.48%**
	80	**0.81%**	**−10.16%**
	90	**0.74%**	**−9.55%**
	100	**0.68%**	**−7.87%**

Table 5.9: Comparison of encoding results for 720p based on a 1080p reference. The prediction mode is set using a threshold of $\theta = 80$.

Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	0.13%	−0.01 dB	−2.68%
CrowdRun	0.56%	−0.03 dB	−7.82%
DucksTakeOff	1.77%	−0.06 dB	−19.65%
Kimono	0.74%	−0.03 dB	−8.48%
ParkJoy	0.47%	−0.02 dB	−9.29%
ParkScene	0.24%	−0.01 dB	−3.22%
PedestrianArea	0.61%	−0.03 dB	−13.75%
Riverbed	0.22%	−0.01 dB	−52.11%
RushHour	0.49%	−0.02 dB	−7.72%
Sunflower	0.53%	−0.02 dB	−2.85%
Average	**0.58%**	**−0.02 dB**	**−12.76%**

Table 5.10: Comparison of encoding results for 360p based on a 1080p reference. The prediction mode is set using a threshold of $\theta = 80$.

Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	0.06%	0.00 dB	−3.14%
CrowdRun	0.56%	−0.03 dB	−5.41%
DucksTakeOff	3.45%	−0.17 dB	−18.72%
Kimono	0.87%	−0.04 dB	−7.20%
ParkJoy	0.54%	−0.03 dB	−7.61%
ParkScene	0.23%	−0.01 dB	−2.67%
PedestrianArea	0.87%	−0.05 dB	−11.52%
Riverbed	0.97%	−0.04 dB	−51.66%
RushHour	0.37%	−0.02 dB	−5.90%
Sunflower	0.50%	−0.03 dB	−2.63%
Average	**0.84%**	**−0.04 dB**	**−11.65%**

5.4 Intra mode reuse

5.4.1 Intra mode reuse

As explained in Section 2.1.4.1, the HM reference software uses a two-step approach to determine the intra mode. First a rough mode decision determines a set of intra mode candidates and second the candidates are evaluated using a full mode decision. Similar to Section 4.2, the processor time of the two steps is measured. The results are listed in Table 5.11 and indicate that the full mode decision, even with fewer intra modes to check, takes longer than the rough intra mode decision on 35 modes. The time taken for PU depth 4 is not measured because the HM encoder does not always check the PUs at that depth.

In the single resolution case, the proposed multi-rate method aims at reducing the full mode decision complexity by using the reference intra mode as candidate and reducing the total number of candidates to be checked (see Section 4.5). However, for a PU at a low resolution, there is not necessarily a single corresponding PU at a reference high resolution, and thus there may be multiple different intra modes in the reference area of the low resolution PU. There is no indication about which one would make the most sense. Adding possibly multiple intra modes to the candidates does not make sense as this would increase the computation time. Therefore, the goal is to reduce the complexity of the first step of the intra mode decision, that is, the rough mode decision.

The candidate lists ψ_k from the high-resolution PUs k which overlap the area of the considered low-resolution PU are merged into a multiset $\psi_{\text{merge}} = \uplus_k \psi_k$. To obtain the final extracted candidate list ψ for the low resolution PU, the 3 elements with the highest multiplicity (that is, the elements which occur most often in the multiset) are picked. Ties are resolved randomly.

A numerical example is given and illustrated by Figure 5.16. The area of the reference encoding corresponding to the current PU overlaps four PUs with candidate lists $\psi_1 = \{0, 1, 21\}$ $\psi_2 = \{0, 11, 25\}$ $\psi_3 = \{0, 1, 26\}$ $\psi_4 = \{1, 19, 21\}$. The multiset is then $\psi_{\text{merge}} = \{0, 0, 0, 1, 1, 1, 11, 19, 21, 21, 25, 26\}$ and the final candidate list $\psi = \{0, 1, 21\}$.

5.4.2 Method assessment

The accuracy of the merging and clipping is assessed by checking if the intra mode found in the original encoding is in the extracted candidate list. Table 5.12 shows the percentage of extracted candidate lists containing the best intra mode. This percentage decreases with increasing depth at both 720p and 360p. For depths 0 and 1, the percentage of extracted candidates lists which contain the intra mode of the original encoding is above 75%. However, from depth 2 on, the percentage is below 65%. In order to avoid making too many suboptimal decisions, the proposed method is only applied until depth 1 in the following.

Table 5.11: Average time for the intra mode decision at different PU depths (in ms)

depth	rough mode decision	full mode decision	total intra mode decision
0	0.66	1.19	1.85
1	0.19	0.30	0.49
2	0.05	0.08	0.13
3	0.01	0.04	0.05

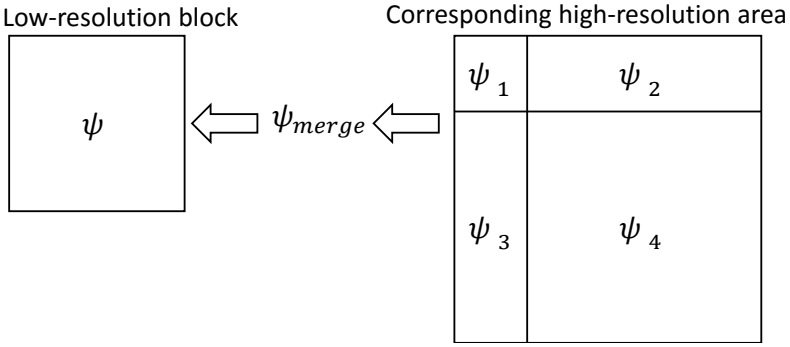


Figure 5.16: Example of merging and clipping of candidate lists from a high-resolution reference.

Table 5.12: Percentage of candidate lists containing the optimal intra mode, as given by the original encoder.

Resolution	**PUs at depth**				
	0	**1**	**2**	**3**	**4**
720p	93.7%	79.7%	61.0%	64.3%	57.8%
360p	96.9%	82.4%	64.4%	64.2%	58.7%

5.4.3 Results

In the case of the *random access, main* profile, the average encoding time reduction is 0.77% and 0.76%, while the BD-rate increase is 0.58% and 0.30% for 720p and 360p, respectively. Similar to the single resolution case, these results can be explained by the relative low number of intra encoded blocks in this profile, where mostly P and B-frames are used.

Results for the *intra, main* profile are shown in Tables 5.13 and 5.14. The average encoding time decrease is 8.37% and 7.54% for 720p and 360p, respectively, which is lower than the results achieved in the single-resolution case. This can be explained on one hand by the fact that here the time gain comes from skipping the evaluation of the approximated costs, while in the single resolution case, the full RD cost calculation is shortened (see Table 5.11). On the other hand, only the approximated costs at depths 0 and 1 are skipped here, whereas all depths are affected in the single resolution case.

Table 5.13: Comparison of encoding results for 720p based on a 1080p reference. The extracted intra mode candidate list is reused until depth 1.

Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	0.19%	−0.01 dB	−7.95%
CrowdRun	0.07%	−0.01 dB	−7.32%
DucksTakeOff	0.41%	−0.02 dB	−6.34%
Kimono	0.65%	−0.03 dB	−8.97%
ParkJoy	0.09%	−0.01 dB	−7.56%
ParkScene	0.25%	−0.01 dB	−7.79%
PedestrianArea	1.50%	−0.07 dB	−9.68%
Riverbed	0.41%	−0.02 dB	−8.40%
RushHour	1.97%	−0.09 dB	−9.87%
Sunflower	1.25%	−0.07 dB	−9.86%
Average	**0.68%**	**−0.03 dB**	**−8.37%**

Table 5.14: Comparison of encoding results for 360p based on a 1080p reference. The extracted intra mode candidate list is reused until depth 1.

Sequence	BD-rate	BD-PSNR	ΔT
BlueSky	0.07%	−0.01 dB	−8.88%
CrowdRun	0.06%	0.00 dB	−5.88%
DucksTakeOff	0.19%	−0.01 dB	−5.94%
Kimono	0.30%	−0.02 dB	−8.74%
ParkJoy	0.16%	−0.01 dB	−7.24%
ParkScene	0.23%	−0.01 dB	−7.65%
PedestrianArea	0.92%	−0.06 dB	−7.34%
RiverBed	0.18%	−0.01 dB	−7.35%
RushHour	0.69%	−0.05 dB	−8.42%
Sunflower	0.50%	−0.04 dB	−8.02%
Average	**0.33%**	**−0.02 dB**	**−7.54%**

5.5 Multi-resolution multi-rate encoder

5.5.1 Combined proposed methods

In this chapter, similar to the single-resolution case in Chapter 4, different possible information reuse methods have been investigated separately and their individual effect on RD performance and encoding time have been examined. As in the single-resolution case, the CU structure reuse leads to the highest encoding time reduction, as expected from Table 4.1. The prediction mode reuse achieves larger encoding time reductions as in the single-resolution case, with an average encoding time reduction of 12.76% and 11.65% for 720p and 360p, respectively, while the RD performance loss is small (BD-rate increase of 0.58% and 0.84%). The

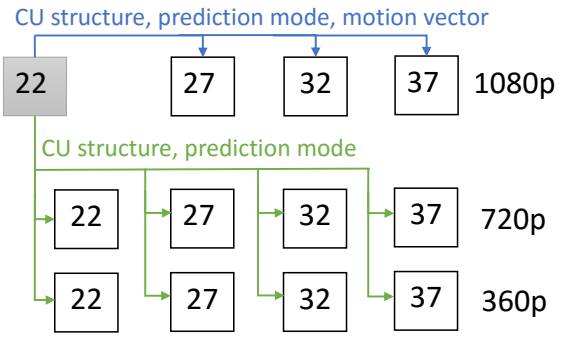


Figure 5.17: Schema of the multi-rate encoding system with the reference encoding (gray) and dependent encodings (white) and corresponding QPs.

intra mode reuse methods leads to encoding time reduction of 8.37% and 7.54% for 720p and 360p, respectively, if all frames are encoded as I-frames. However, the method is relatively inefficient in a configuration with mostly inter-predicted frames.

Similar to Section 4.7, the proposed methods are combined to leverage all possible encoding time reductions. So far, a single-resolution system and a multiple-resolutions system have been studied separately. A system with 12 representations is now considered, spanning both multiple resolutions and different SNR qualities, as shown in Figure 5.17. The encoding at 1080p and QP 22 is used as reference encoding (largest resolution and highest signal quality). For the three 1080p dependent encodings, the proposed CU structure reuse, prediction mode reuse, and motion vector reuse are implemented. Section 4.7.2 showed that incorporating the proposed intra mode reuse method leads to a lower RD performance, without achieving significant time reduction. Similarly, for the low-resolution dependent encodings, the proposed CU structure reuse and prediction mode reuse are implemented. Results from Section 5.4.3 indicate that the intra candidate list reuse method is not beneficial in a *random access* profile with inter-predicted frames.

5.5.2 Results

The encoding results compared with the original HM encoder are presented in Table 5.15. The BD-rate and BD-PSNR are measured separately for the different resolutions. The overall encoding time difference ΔT_{12} is now measured over the 12 representations, which includes the reference encoding. The RD performance loss is smallest for the 1080p representations with an average BD-rate increase of 0.55%. Averaged over the three resolutions, the BD-rate is increased by 1.11% and the BD-PSNR is reduced by 0.047 dB. The average encoding time reduction is 41.79%. These results indicate that a complete multi-rate encoder for a practical adaptive HTTP streaming scenario which requires 10 to 15 representations spanning different spatial resolutions and different signal qualities can be achieved with the proposed methods.

Table 5.15: Comparison of encoding results for a fixed QP representations set.

Sequence	**BD-rate (%)**			**BD-PSNR (dB)**			**ΔT (%)**			**ΔT_{12} (%)**
	1080p	720p	360p	1080p	720p	360p	1080p	720p	360p	
BlueSky	0.68	0.62	0.63	−0.027	−0.032	−0.039	−47.72	−53.25	−47.01	−49.35
CrowdRun	0.44	1.16	1.11	−0.018	−0.055	−0.058	−23.91	−27.04	−22.83	−24.76
DucksTakeOff	−0.004	1.83	2.43	0.0001	−0.063	−0.115	−25.27	−39.53	−40.02	−30.19
Kimono	0.70	1.65	1.71	−0.023	−0.066	−0.063	−43.95	−48.78	−38.63	−45.00
ParkJoy	0.16	0.86	1.04	−0.007	−0.039	−0.050	−29.21	−33.90	−28.24	−30.54
ParkScene	0.56	0.99	0.77	−0.018	−0.037	−0.033	−36.33	−37.52	−30.47	−36.27
PedestrianArea	1.09	2.13	2.06	−0.036	−0.099	−0.112	−43.40	−50.29	−40.80	−45.30
Riverbed	0.49	0.80	1.54	−0.021	−0.037	−0.073	−45.45	−78.05	−72.71	−57.27
RushHour	0.89	1.71	2.39	−0.023	−0.058	−0.102	−42.15	−48.01	−38.81	−43.67
Sunflower	0.50	1.27	1.01	−0.018	−0.051	−0.053	−53.02	−61.26	−54.07	−55.58
Average	**0.55**	**1.30**	**1.47**	**−0.019**	**−0.053**	**−0.070**	**−39.04**	**−47.76**	**−41.36**	**−41.79**
	1.11			**−0.047**			**−42.72**			

Table 5.16: Target bitrates for encoding with rate control (in kb/s).

24 fps and 25 fps			50 fps		
1080p	**720p**	**360p**	**1080p**	**720p**	**360p**
7,500	5,200	2,300	55,000	45,000	20,000
3,500	2,500	1,100	25,000	20,000	9,000
1,500	1,200	540	10,000	9,000	4,500
800	600	260	5,000	4,500	2,000

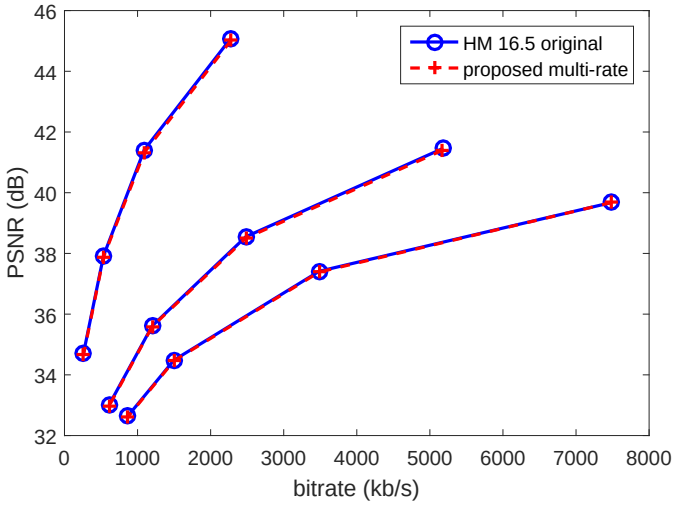


Figure 5.18: RD curves for the *ParkScene* sequence at 1080p, 720p, and 360p.

5.5.3 Rate-control-based encoding

In practical deployments, rate control is generally used instead of fixed QP encoding. To show the effect of the proposed methods for rate-control deployments, a set of 12 representations based on spatial resolution and target bitrate is now determined, instead of spatial resolution and fixed QP. To determine the target bitrates for the 12 representations, an average of the bitrates of the videos of the main set encoded at QPs 22, 27, 32, and 37 is used. As there are videos at 24, 25, and 50 fps, two bitrate sets are determined: one for the videos at 24 fps and 25 fps, and one for the videos at 50 fps. The target bitrates are listed in Table 5.16.

The multi-rate system with 12 representations is run where the 1080p representation at highest bitrate is the reference encoding. The default HM rate control is used. Table 5.17 shows the encoding results compared with the original HM encoder. The results are comparable with the results from the fixed QP representations set. Again, the 1080p representations show the least RD performance loss with a BD-rate increase of 0.46%. The overall average encoding time reduction is 37.92%, which is slightly less as in the fixed QP case. However, the overall RD performance is also slightly better than in the fixed QP case with an overall average BD-rate increase of 0.96%, instead of 1.11%. These results show on one side that the proposed methods are not only applicable to fixed QP encoding, but also to rate-control-based

Table 5.17: Comparison of encoding results for the set based on rate control.

Sequence	BD-rate (%)			BD-PSNR (dB)			ΔT (%)			ΔT_{12} (%)
	1080p	720p	360p	1080p	720p	360p	1080p	720p	360p	
BlueSky	0.47	1.12	0.98	−0.017	−0.051	−0.057	−41.57	−46.53	−39.63	−42.98
CrowdRun	0.44	0.72	1.09	−0.018	−0.035	−0.074	−24.20	−25.80	−24.93	−24.78
DucksTakeOff	−1.07	0.24	0.73	0.027	−0.010	−0.053	−29.83	−33.98	−32.07	−31.39
Kimono	0.80	0.91	2.31	−0.027	−0.039	−0.103	−42.80	−46.45	−36.84	−43.47
ParkJoy	0.30	0.62	1.86	−0.012	−0.031	−0.144	−29.83	−31.34	−28.38	−30.18
ParkScene	0.65	0.96	1.39	−0.021	−0.037	−0.065	−37.86	−38.72	−32.62	−37.68
PedestrianArea	1.17	1.73	2.00	−0.036	−0.070	−0.108	−41.06	−38.67	−30.85	−39.45
Riverbed	0.47	0.43	0.01	−0.020	−0.018	−0.007	−45.49	−51.39	−45.87	−47.38
RushHour	0.53	1.51	2.71	−0.010	−0.045	−0.116	−38.65	−41.38	−33.07	−39.06
Sunflower	0.87	1.01	1.76	−0.016	−0.028	−0.073	−42.10	−45.26	−38.47	−42.79
Average	**0.46**	**0.93**	**1.48**	**−0.015**	**−0.036**	**−0.080**	**−37.34**	**−39.85**	**−34.27**	**−37.92**
	0.96			**−0.044**			**−37.15**			

Table 5.18: Comparison of encoding results for videos with alternative spatial resolutions.

Sequence	**BD-rate (%)**			**BD-PSNR (dB)**			**ΔT (%)**			**ΔT_{12} (%)**
	1600p	1080p	720p	1600p	1080p	720p	1600p	1080p	720p	
PeopleOnStreet	1.00	0.89	0.75	−0.046	−0.045	−0.043	−26.05	−20.29	−17.91	−23.40
Traffic	0.56	0.51	0.70	−0.020	−0.022	−0.037	−39.96	−37.67	−36.17	−38.88
	480p	360p	240p	480p	360p	240p	480p	360p	240p	
BQMall	0.68	0.71	0.27	−0.030	−0.038	−0.015	−31.41	−31.59	−28.14	−31.01
PartyScene	0.42	0.73	0.44	−0.019	−0.038	−0.022	−26.19	−28.74	−26.44	−26.97
Average	**0.67**	**0.71**	**0.54**	**−0.029**	**−0.035**	**−0.029**	**−30.90**	**−29.57**	**−27.17**	**−30.07**
	0.64			**−0.031**			**−29.21**			

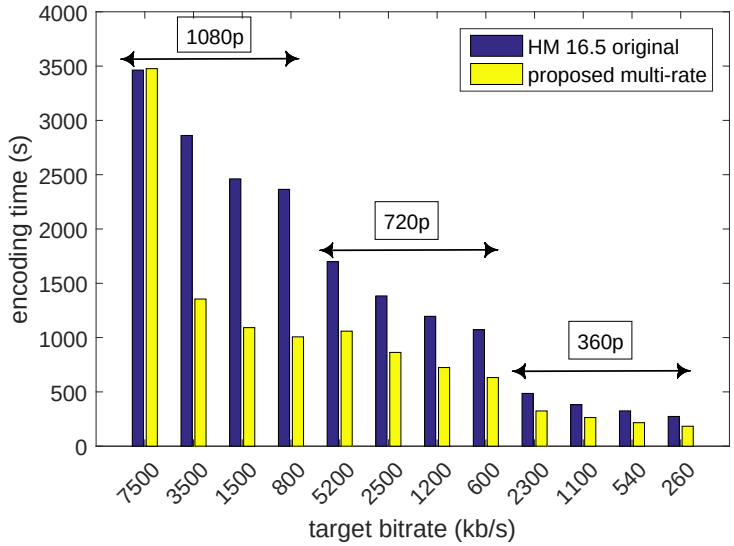


Figure 5.19: Encoding time of the 12 representations of the *ParkScene* sequence.

encoding. On the other side, the results show that a practical deployment of the proposed methods in a rate-control-based environment is beneficial in terms of complexity reduction, with only a minor RD-performance loss.

To visualize the achieved results in the rate-control case, Figure 5.18 shows an example of RD curves for the *ParkScene* sequence, both for the original HM encoder and for the proposed multi-rate encoder. Figure 5.19 shows the corresponding encoding time with the original encoder and the proposed multi-rate encoder.

5.5.4 Alternative spatial resolutions

In addition to the main set of 1080p videos, the impact of the proposed multi-rate system is assessed on video sequences with different reference resolutions from the alternative set. For larger resolutions, the sequences *PeopleOnStreet* and *Traffic* with an original resolution of 2500×1600 pixels are used, and two dependent resolutions of 1728×1080 and 1124×720 pixels are defined. For lower resolutions, the sequences *BQMall* and *Partyscene* with an original resolution of 832×480 pixels are used, and two dependent resolutions of 624×360 and 416×240 pixels are defined.

Table 5.18 shows the encoding results compared with the original HM encoder. The videos are encoded with a fixed QP (22, 27, 32, and 37) at each resolution. On average, the encoding time for 12 representations can be decreased by 30.07% while the BD-rate is increased by 0.64%. The results are comparable to the results for the 1080p set, although the encoding time reduction is lower, but the BD-rate increase is lower as well.

5.5.5 Comparison with related work

De Praeter et al. [80] proposed a multi-rate encoding system for HEVC, where the CU structure of the dependent encodings is predicted with machine learning based on the CU structure of the reference encoding (see Section 2.3.3). The reference encoding does not have to be the encoding with the highest quality. In their results, they encode five videos *BasketballDrive*, *BQTerrace*, *Cactus*, *Kimono*, and *ParkScene* [83] at three different resolutions (1920×1080, 1280×720, and 960×536 pixels) and different fixed QPs. Results are provided in [80] for different possible references. In order to be consistent in the comparison, their results are shown here based on the reference encoding that leads to the lowest BD-rate increase, as this is also the aim of the proposed methods. For the comparison, the same video sequences are encoded at the same three resolutions using the same fixed QPs with the proposed method.

Table 5.19: Comparison with related work.

Sequence	proposed		De Praeter et al. [80]	
	BD-rate	ΔT	BD-rate	ΔT
BasketballDrive	0.78%	−41.94%	6.4%	−59.6%
BQTerrace	0.36%	−31.91%	5.6%	−70.7%
Cactus	0.79%	−35.74%	6.5%	−59.4%
Kimono	1.12%	−46.02%	4.7%	−57.8%
ParkScene	0.77%	−36.51%	4.8%	−57.4%
Average	**0.76%**	**−38.42%**	**5.6%**	**−61.0%**

Table 5.19 shows the average BD-rate and the overall time reduction over all representations for the method by De Praeter et al. [80] and for the proposed method. Although the proposed method achieves a lower overall time reduction, the average BD-rate increase of the proposed method is less than 0.8%, which is very close to the original performance of the HM encoder. In contrast, the average BD-rate increase with the method by De Praeter et al. is 5.6% on average, which may be prohibitive for video streaming providers, as storage and transmission costs increase.

5.6 Summary

In this chapter, the focus was on multi-rate encoding methods where the reference and dependent encodings are at different spatial resolutions. The main identified challenge is that there is no direct correspondence between blocks at different resolutions if the downsampling factor is different from a power of 2. In fact, encoding decisions in HEVC are taken at block level, and thus, there is no direct way to map decisions from a high-resolution reference encoding to blocks in a lower-resolution dependent encoding.

Therefore, the multi-rate methods proposed in Chapter 4 are not applicable to the case of multiple resolutions. In this chapter, the behaviors of the CU structure, the prediction mode

and the intra prediction mode across encodings at different resolutions have been studied. Methods to extract information from a high-resolution to lower-resolution dependent encodings for any given downsampling ratio have been proposed. The extracted information is then used to constrain the RDO of the dependent encodings and thus, reduce their computational complexity.

In a final step, the different proposed methods have been combined along with the methods from Chapter 4, which leads to a multi-rate encoder capable of encoding representations at different resolutions and different levels of signal quality. Encoding results show that, for a specific set of 12 representations, the overall encoding time can be reduced by 38% at the cost of less than 1% of average bitrate increase. The proposed multi-rate encoder has also been shown not to be limited to fixed QP encoding, but is also efficient in a more practical scenario where rate control is used.

Chapter 6

Improved rate control for HEVC multi-rate encoding

6.1 Introduction

Previous chapters in this thesis have concentrated on reducing the encoding complexity of a multi-rate HEVC encoder. If the literature on multi-rate encoding is analyzed, one can notice that all proposed multi-rate methods so far have targeted complexity reduction [22], [77], [80], [82], because using similarities in encodings at different bitrates to eliminate redundancies is the first obvious benefit of a multi-rate system. However, the complexity reduction generally comes at the price of a small decrease in RD performance. In this chapter, the possibility of improving the overall RD performance of a multi-rate system is examined, compared to a classical system of multiple independent encoders, by sharing information about the video characteristics between the different representations. Specifically, the case of rate-control-based encoding is considered.

Rate control is often used in the context of adaptive HTTP streaming. First, a video provider who uses an adaptive HTTP system to provide video content is generally targeting a panel of various users. The different users can typically access the streaming service through different channels (cellular networks, wireless or wired connections, etc.), which can have specific constraints on the data rate. Second, adaptive HTTP streaming relies on one hand on a set of segmented video representations, and on the other hand on a manifest file which the users need to know the bitrates of the available representations. For that, the segments should comply with the advertised bitrate, which can be ensured by using rate control.

Rate control generally uses a video-content-dependent model that combines the bitrate with an encoding parameter, so that the bitrate can be adjusted for any specific video content. In this chapter, the content-dependent model information is passed between the reference and dependent encodings in order to improve the quality of the model, and thus improve the RD performance of the rate-control-based encoding. The idea can be compared to a two-pass (or multipass) encoding (e.g., [88]), where the encoder first passes through the video to

be encoded to learn its characteristics, before actually encoding the video in the second pass. However, a two-pass rate control outputs one specific bitrate, and it increases the encoding complexity due to the two passes. On the other hand, in this chapter a multi-rate scenario with multiple output rates is targeted, and the encoding complexity should not increase.

Section 6.2 reviews the state-of-the-art HEVC rate control implemented in the reference software HM. The performance of the rate control encoding is assessed and the limits of the rate-control algorithm are investigated in Section 6.3. Section 6.4 presents the proposed multi-rate method for improved rate control along with the results. In Section 6.5, the proposed method is combined with a method from Chapter 4 to decrease the encoding complexity. Finally, Section 6.6 summarizes the chapter.

6.2 Rate control for HEVC

Rate control for video compression can generally be divided into two steps. The first one is bit allocation at different levels of encoding and the second step is achieving the target bits during the encoding. To achieve the target bits during the encoding, rate control relies on estimating the rate-distortion function of the video to be encoded. A rate control method implemented in early versions of the HM reference software was using a rate-quantization model, that is, the bitrate was controlled by varying the quantization (i.e., the QP) [89]. Li *et al.* [31] argued that the rate-quantization model is only precise enough when all coding decisions are fixed. Thus, this model is not well suited for HEVC which allows a large number of encoding modes. Li *et al.* then proposed a new rate-control method in the so-called λ-domain, which was accepted at the 11th JCT-VC meeting [90] and implemented into the HM reference software.

6.2.1 Bit allocation

The rate-control method by Li *et al.* [31] allocates the bits at three levels: group-of-pictures (GOP) level, frame level and CTU level.

Given a target bitrate R_{target}, the average number of bits per frame should be:

$$b_{\text{target,frame}} = R_{\text{target}}/f \tag{6.1}$$

where f is the frame-rate in frames/s.

- At the **GOP level**, the number of bits per GOP should be $b_{\text{target,frame}} \cdot N_{\text{GOP}}$ where N_{GOP} is the number of frames in a GOP. However, achieving the exact number of bits for each GOP in a single pass is difficult, and thus Li *et al.* propose to adapt the target number of bits depending on the previous encoded GOPs as follows:

$$b_{\text{GOP}} = \frac{b_{\text{target,frame}} \cdot (N_{\text{coded}} + N_{SW}) - b_{\text{coded}}}{N_{SW}} \cdot N_{\text{GOP}} \tag{6.2}$$

where N_{SW} is the size of a sliding window and is set to 40, N_{coded} is the number of already encoded frames and b_{coded} is the total number of bits of all already encoded frames.

- At the **frame level**, the target number of bits for each frame i is given by:

$$b_{\text{frame},i} = \frac{b_{\text{GOP}} - b_{\text{GOP},coded}}{\sum_{\{AllNotCodedFrames\}} \omega_{\text{frame}}} \cdot \omega_{\text{frame},i} \tag{6.3}$$

where $b_{\text{GOP},coded}$ is the number of bits already coded in the GOP, and $\omega_{\text{frame},i}$ is the weight of frame i. That is, the remaining bits in the budget of the GOP are assigned according to the weights ω of each frame. The frame weights can be equal in certain applications (for example for low-delay applications). The *random access* profile [83], however, uses hierarchical bit allocation, where the weights are different based on the level in the encoding hierarchy (i.e., the position of the frame in the GOP structure).

- Finally, at the **CTU level**, the target number of bits for each CTU j is allocated such that the leftover bits in the frame budget are allocated based on the weight $\omega_{\text{CTU},j}$ of each CTU.

$$b_{\text{CTU},j} = \frac{b_{\text{frame}} - b_{\text{header}} - b_{\text{frame},coded}}{\sum_{\{AllNotCodedCTUs\}} \omega_{\text{CTU}}} \cdot \omega_{\text{CTU},j} \tag{6.4}$$

where b_{header} is the number of estimated header bits, and $b_{\text{frame},coded}$ is the number of bits already coded in the frame. The weights for the CTU are calculated based on the mean average difference (MAD) of the colocated CTU in the preceding frame at the same hierarchical level.

6.2.2 Rate control in the λ domain

After the bit allocation, the second step of rate control is to achieve the allocated bits. As seen in Section 2.1.2, the RDO can be expressed as an unconstrained optimization problem as follows:

$$\text{dec}_{\text{opt}} = \underset{\{\text{dec}\}}{\arg\min} \, (D + \lambda R) \tag{6.5}$$

where D is the distortion, R is the bitrate, and λ is the Lagrange multiplier. In the case of HEVC, the set of encoding decisions $\{\text{dec}\}$ contains for example the block structure, the motion vectors, the intra prediction modes, etc. The optimization problem has to be solved in the vicinity of the target bitrate R_{target} in the case of rate control. Li *et al.* [31] use a hyperbolic model for the RD relationship:

$$D(R) = CR^{-K} \tag{6.6}$$

where D is expressed in terms of MSE of the luma component, R is expressed in terms of *bit per pixel* (bpp), and C and K are two model parameters that depend on the video content. Putting the derivative of $D + \lambda R$ in (6.5) to zero to find the minimum, λ can be expressed as:

$$\lambda = -\frac{\partial D}{\partial R} = CK \cdot R^{-K-1} \equiv \alpha R^{\beta} \tag{6.7}$$

where α and β are two other parameters that describe the video content. Finally, (6.7) can be rewritten to:

$$R = \left(\frac{\lambda}{\alpha}\right)^{\frac{1}{\beta}} \tag{6.8}$$

which means that on an optimal point on the RD curve, the bitrate R is a function of λ. Li *et al.* thus propose to solve the rate-control problem in the λ domain: for a target bitrate R_{target} and known model parameters α and β, the Lagrange multiplier λ for the RDO is determined with (6.7).

Finally, the QP corresponding to the λ value is computed as follows [31]:

$$QP = \text{round}\left(4.2005 \cdot \ln \lambda + 13.7122\right) \tag{6.9}$$

The rounding operation is introduced as the QP can only be an integer.

6.2.3 Challenge

Because in general all frames are different and all CTUs are different, the content dependent parameters α and β need to be known for each frame and for each CTU to solve the rate-control problem. This is, however, not the case at the beginning of the encoding process, where no information about the video sequence is available. Thus, Li *et al.* [31] use predetermined α and β values for the first frames. For the first I-frame, α is set to 6.7542 and β is set to 1.7860. For P and B-frames, α is set to 3.2003 and β to -1.367.

After each encoded frame, the information gathered from the encoding of the frame is used to update the α and β values. The update process is as follows.

$$\lambda_{\text{meas}} = \alpha_{\text{used}} {R_{\text{meas}}}^{\beta_{\text{used}}} \tag{6.10}$$

$$\alpha_{\text{new}} = \alpha_{\text{used}} + \delta_\alpha \cdot \left(\ln \lambda_{\text{used}} - \ln \lambda_{\text{meas}}\right) \cdot \alpha_{\text{used}} \tag{6.11}$$

$$\beta_{\text{new}} = \beta_{\text{used}} + \delta_\beta \cdot \left(\ln \lambda_{\text{used}} - \ln \lambda_{\text{meas}}\right) \cdot \ln R_{\text{meas}} \tag{6.12}$$

First, the λ_{meas} value is calculated based on the measured bitrate R_{meas} and the values α_{used} and β_{used} that have been used. The new values α_{new} and β_{new} are calculated based on the difference in the logarithmic domain of the measured λ_{meas} and the λ_{used} that has been used, with update factors $\delta_\alpha = 0.1$ and $\delta_\beta = 0.05$, respectively [31]. The new values are used for the next frame at the same hierarchical level.

The determination of the “true" α and β values for each hierarchical frame level can thus be seen as an iterative process throughout the encoding. The underlying assumption is that a frame is very similar to its preceding frame at the same hierarchical level. Both this assumption and the simple update method lead to suboptimal determination of the model parameters α and β, and thus lead to suboptimal determination of λ for the rate control and the RDO. This is especially true for a scene change, where the model parameters substantially change.

Table 6.1: Target bitrates (kb/s) for the main set.

Sequence	QP 22	QP 27	QP 32	QP 37
BlueSky	4625	2254	1223	707
CrowdRun	53797	24336	11656	5786
DucksTakeOff	68417	21275	8669	4090
Kimono	5690	2711	1338	687
ParkJoy	68089	30581	13642	6037
ParkScene	8342	3614	1654	767
PedestrianArea	4303	1952	1004	558
Riverbed	21278	10433	4970	2451
RushHour	3830	1652	782	404
Sunflower	2365	1140	601	348

6.2.4 Model mismatch

The difference in the logarithmic domain of λ_{used} and λ_{meas} can be interpreted as the mismatch of the used model. In fact, if this difference is 0, then the values of α and β are not updated in Equations (6.11) and (6.12), as the model by Li *et al.* is considered to be matching. On the other hand, the larger the difference, the larger is the update of the parameter values, as the model is considered to poorly match the current frames. In this thesis, the *model error* ϵ is defined as the absolute value of this difference:

$$\epsilon = |\ln \lambda_{\text{used}} - \ln \lambda_{\text{meas}}| \tag{6.13}$$

The model error is used to determine how far the current values are from the "true" values.

6.2.5 Data set

In this chapter, the HEVC encoding is based on rate control instead of using a fixed QP. To determine target bitrates that are comparable to previous results in this thesis, 100 frames of each video are encoded at four QPs (22, 27, 32, and 37) [83] and the achieved bitrate is used as target for rate-control-based encoding, as listed in Table 6.1. The representation at highest quality (largest bitrate) is called representation 1. Lower quality representations are numbered representations 2, 3, and 4, with representation 4 being the representation at the lowest quality (lowest bitrate).

6.3 Performance of the original rate control

6.3.1 Frame-level rate control

The performance of the rate control is first evaluated at the frame level. That is, the model parameters α and β are only used at the frame level and not at the CTU level. Similarly,

Table 6.2: Comparison of encoding results between the fixed QP encoding and the encoding based on rate control at the frame level for the main set, and accuracy of the rate control.

Sequence	BD-rate	BD-PSNR	Mean deviation
BlueSky	4.43%	−0.17 dB	5.12%
CrowdRun	7.57%	−0.30 dB	1.35%
DucksTakeOff	16.25%	−0.36 dB	0.92%
Kimono	6.24%	−0.19 dB	0.70%
ParkJoy	3.66%	−0.15 dB	1.97%
ParkScene	2.80%	−0.09 dB	2.40%
PedestrianArea	12.71%	−0.37 dB	2.14%
Riverbed	3.18%	−0.14 dB	0.05%
RushHour	11.19%	−0.25 dB	1.48%
Sunflower	15.02%	−0.45 dB	7.05%
Average	**8.31%**	**−0.25 dB**	**2.32%**

the bit allocation is determined at the GOP level and at the frame level. On the contrary, no target bits are determined for the different CTUs within a frame. The frame-level rate control is activated in HM by putting the configuration parameter *LCULevelRateControl* to 0.

The rate control at frame level is compared to the encoding with fixed QP in terms of RD performance by using the target bitrates in Table 6.1. Table 6.2 shows the RD performance of the encoding with rate control at frame level compared to the encoding at fixed QP. With an average BD-rate increase of 8.31%, the rate-control-based encoding achieves a much lower RD performance than the fixed QP encoding. This RD performance loss is due to suboptimal decisions during the RDO. Based on the equation for the RDO (6.5), a suboptimal decision can be taken if the choice of λ is not optimal. The choice of λ is suboptimal if λ does not correspond to the slope of the tangent to the RD curve at the working point (cf. Figure 2.3). This can be the case during rate control if the RD model does not fit the actual video content.

Additionally, the bitrate accuracy of the rate control is examined by defining the bitrate deviation η as follows:

$$\eta = \frac{|R_{\text{target}} - R_{\text{achieved}}|}{R_{\text{target}}} \tag{6.14}$$

Table 6.2 presents the mean bitrate deviation over the four representations 1 to 4, calculated as:

$$\bar{\eta} = \frac{1}{4}\sum_{i=1}^{4} \eta_i \tag{6.15}$$

The mean bitrate deviation of the rate control averaged over the 10 videos of the main set is 2.32%.

Table 6.3: Comparison of encoding results between the fixed QP encoding and the encoding based on rate control at the CTU level for the main set, and accuracy of the rate control.

Sequence	BD-rate	BD-PSNR	Mean deviation
BlueSky	4.77%	−0.20 dB	6.16%
CrowdRun	4.75%	−0.19 dB	0.40%
DucksTakeOff	8.43%	−0.20 dB	0.25%
Kimono	12.19%	−0.37 dB	0.46%
ParkJoy	3.13%	−0.13 dB	1.37%
ParkScene	2.26%	−0.07 dB	2.97%
PedestrianArea	11.27%	−0.33 dB	2.54%
Riverbed	6.21%	−0.26 dB	0.04%
RushHour	16.35%	−0.37 dB	0.50%
Sunflower	11.74%	−0.36 dB	6.08%
Average	**8.11%**	**−0.25 dB**	**2.08%**

6.3.2 CTU-level rate control

The rate control is now performed at the CTU level. That is, the bit allocation is performed at the GOP level, frame level, and CTU level. Additionally to the frame model parameters α and β, each CTU now also has an own parameter pair.

The RD performance of the rate control at CTU level is compared to the RD performance of the encoding with fixed QP. Table 6.3 shows the compared RD performance. With a BD-rate increase of 8.11%, the RD performance of the rate control at CTU level is severely degraded compared to the fixed QP encoding case. However, the RD performance loss is slightly less than in the case of rate control at frame level (8.31%, cf. Table 6.2).

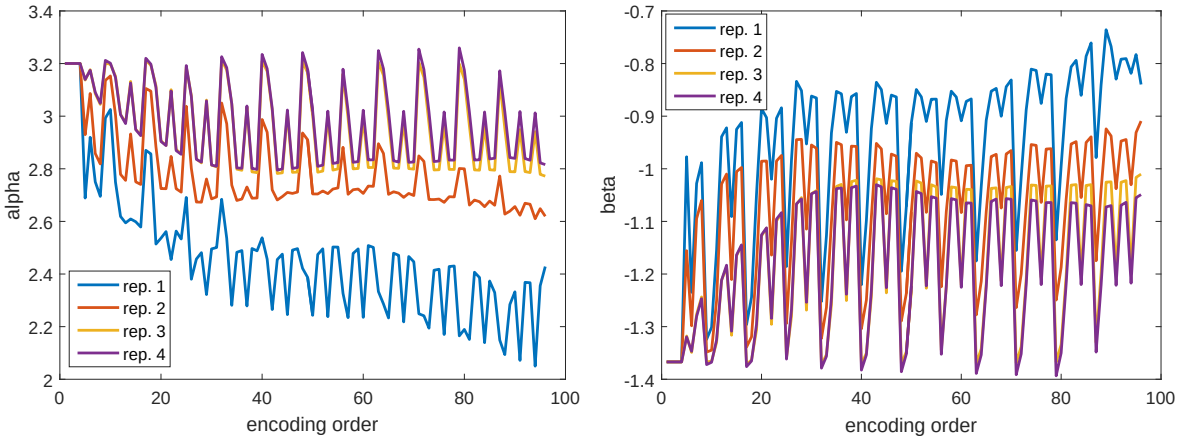
The bitrate accuracy of the rate control at CTU level is also examined and the mean deviation is presented in Table 6.3 as well. With an average of 2.08%, the deviation is slightly smaller than the deviation in the case of rate control at frame level (2.32%, cf. Table 6.2), which means that the bitrate accuracy is slightly improved.

As the CTU-level rate control performs slightly better than the frame-level rate control both in terms of RD performance and bitrate accuracy, the encoding results from the CTU-level rate control are used as baseline in the rest of the chapter to compare the proposed methods.

6.3.3 Limitations of the model

The preceding RD performance results of the rate control implemented in HM compared to the fixed QP encoding show that the model proposed by Li *et al.* [31] is performing suboptimally. A few limitations can be listed:

- The model is based on an assumption of a hyperbolic RD relationship (cf. Eq. (6.6)) which is supposed to hold for the entire RD curve. However, as can be seen from the



(a) α (b) β

Figure 6.1: Model parameters for the *Kimono* sequence over 96 frames at four different representations.

fact that each hierarchical level has its own model parameters (cf. Section 6.2.3), in practice the model is only an approximation for a specific bitrate range.

- The I-frames are treated separately from the P and B-frames. For example, the initial model parameter values for I-frames are $\alpha = 6.7542$ and $\beta = 1.7860$ whereas the initial values for the other frames are $\alpha = 3.2003$ and $\beta = -1.367$ (cf. Section 6.2.3).

- The model is based on the assumption that frames at the same hierarchical level are very similar, and should thus have approximately the same model parameters. This is not the case when a scene change occurs in the video to be encoded.

- The use of CTU-level model parameters influences the number of bits achieved at frame level. This in turn influences the calculation of the frame-level parameters (Equations (6.10) to (6.12)). This means that the frame-level parameters α and β for the same video at the same target bitrate will differ depending on whether frame-level or CTU-level rate control is applied. This contradicts the assumption that the model parameters are only content-dependent.

6.4 Proposed multi-rate method

Due to the fact that CTU-level and frame-level model parameters are interacting, for simplicity reasons, the focus is on frame-level model parameters only in the following.

6.4.1 Model parameters

The behavior of the model parameters α and β is examined over 100 frames when rate control is applied. The I-frames are left out, as they undergo a different treatment to the P and B-frames. The *random access, main* profile defines 4 hierarchical levels for the frames. The

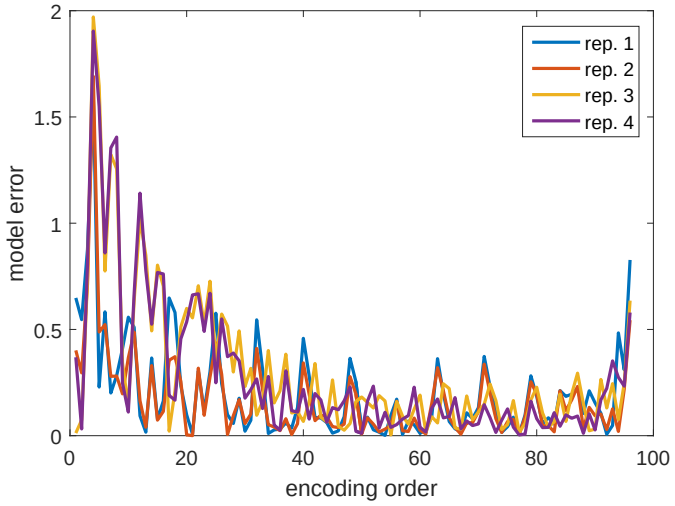


Figure 6.2: Model error at frame level for the *Kimono* sequence at different representations.

model parameters for the first frame at each level are initialized with the values $\alpha = 3.2003$ and $\beta = -1.367$. That is, the first four encoded frames have the same model parameters.

As an example, Figure 6.1 shows the values of α and β for the *Kimono* sequence. From the 100 encoded frames, 4 are I-frames, and thus, 96 frames are represented in encoding order. The first four encoded frames have the same model parameter values, as expected. The pseudo-periodicity of the parameter values is due to the different hierarchical levels, as the parameters are updated using the values of the parameters at the same hierarchical level (cf. Section 6.2.3).

For the different representations, the parameters converge to different values. This confirms that the model by Li *et al.* [31] is sensitive to the bitrate, and is thus not valid for the whole range of bitrates of a video.

Figure 6.2 shows the model error η at frame level as defined in Eq. (6.13) over the 96 B-frames in encoding order of the *Kimono* sequence at four representations. At the beginning of the encoding, the parameters α and β have not been fitted to the actual video content yet, and thus, the model error is large. With the update process of the model parameters, the model error gradually decreases as more and more frames are encoded.

For the last frames, a slight increase in the model error can be observed. This is due to the encoder noticing that the sequence is reaching its end. The remaining bits in the budget are spent on the last frames, and thus the target bits per pixel for the last frames strongly vary from the previous frames. While the model parameters for the last frames only slowly change (cf. Figure 6.1), the Lagrange multiplier λ in Eq. (6.7) is not fitting the video content well anymore due to the bitrate dependency of the model. This leads to an increased model error.

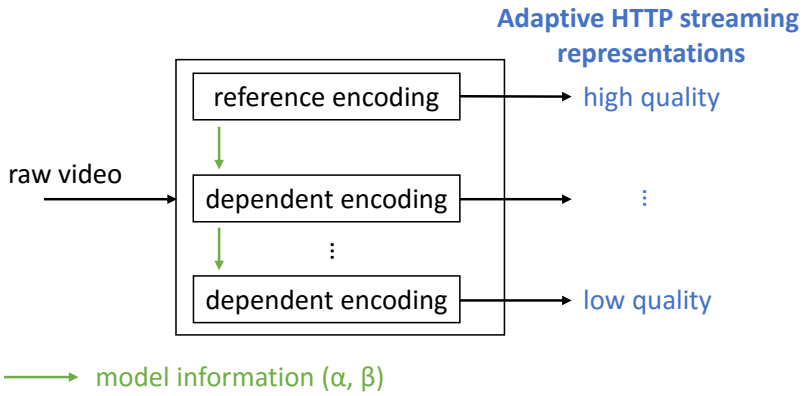


Figure 6.3: Schema of the proposed multi-rate encoder for improved rate control. The model parameters α and β are passed step-by-step from the reference encoding to the next dependent encoding, and then from the dependent encoding to the next dependent encoding.

6.4.2 Proposed parameters reuse method

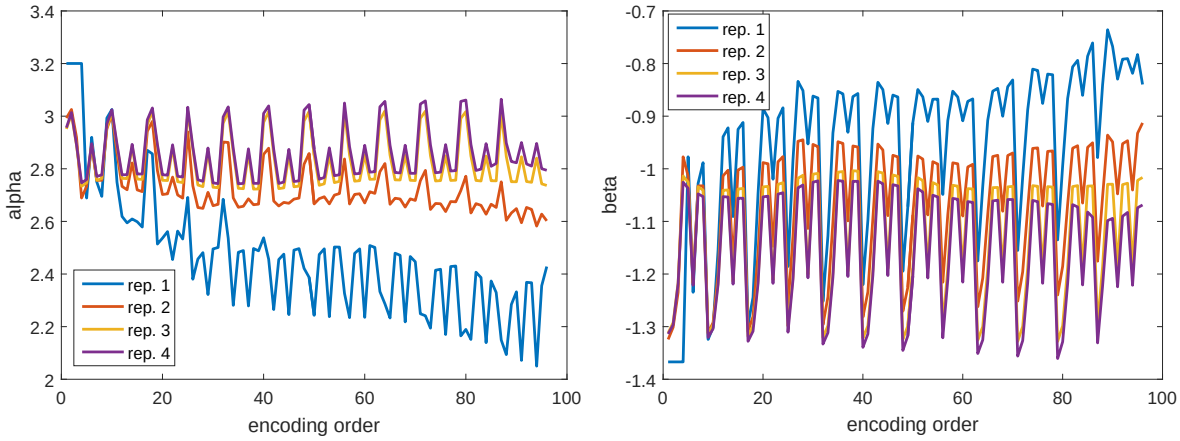
As the model parameters α and β use predefined values for the first frames of a video sequence, the model error at frame level is large for the first encoded frames. This large model error leads to a suboptimal calculation of the Lagrange multiplier λ in Eq. (6.7). This incorrect λ leads to a suboptimal RD-performance, as λ might not correspond to the slope of the tangent to the RD-curve at the targeted bitrate point.

To alleviate the RD-performance loss, in this thesis the model parameter information from a reference encoding is reused to initialize the model parameter values in lower-quality dependent encodings. Specifically, the α and β parameters for the first four non-I-frames are reused after they have been updated, that is, after the specific frame has been encoded in the reference encoding. Even if the model parameters converge to different values for different representations (cf. Figure 6.1), the parameters from the reference encoding are a better initialization than the predetermined values for the first four frames, because they have undergone an update step which adapts the values to the video content characteristics.

Figure 6.3 schematically represents the proposed multi-rate encoder for improved rate control. Unlike the methods proposed in Chapters 4 and 5, the information is now passed step-by-step from the reference encoding to the next dependent encoding, and then from the dependent encoding to the next dependent encoding, and so on. The reason is that the model by Li *et al.* [31] is sensitive to the bitrate. Thus, the updated model parameters α and β from the next closest representation will be a better initialization than always using the model parameters from the highest-quality reference encoding (cf. Figure 6.1).

6.4.3 Results

The proposed multi-rate method for rate control is applied on the main set of video sequences. As an example, Figure 6.4 shows the model parameters α and β over 96 B-frames for the four representations of the *Kimono* sequence. For the reference representation (repre-



(a) α (b) β

Figure 6.4: Model parameters for the *Kimono* sequence over 96 frames at four different representations when the proposed multi-rate method is used.

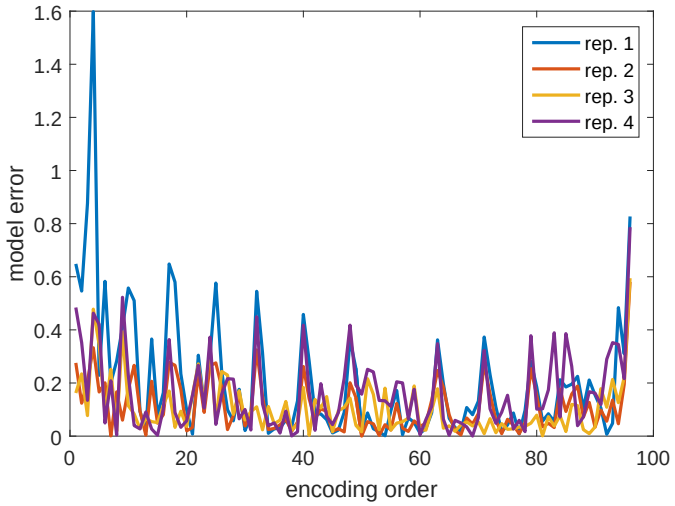


Figure 6.5: Model error at frame level for the *Kimono* sequence at different representations when the proposed multi-rate method is used.

sentation 1), the initial values are the predetermined values from the method by Li *et al.*, and thus, the parameters take the same values as in the original rate control shown in Figure 6.1. On the other hand, the parameter values for the dependent representations (rep. 2 to 4) are initialized with the parameter values from the next higher-quality representation. Compared to Figure 6.1, the parameter values converge more rapidly.

Figure 6.5 shows the model error at frame level over all encoded frames of the *Kimono* sequence when the proposed multi-rate method is applied. Compared to Figure 6.2, the model error for the three dependent representations is lower for the first frames.

Table 6.4 presents the mean model error at frame level over 96 frames for all four representations of the 10 videos of the main set for both the original rate control and the proposed multi-rate method. On average, the mean model error is decreased for the three dependent

Table 6.4: Mean model error at frame level over 96 frames

	original rate control				proposed multi-rate			
Sequence	rep. 1	rep. 2	rep. 3	rep. 4	rep. 1	rep. 2	rep. 3	rep. 4
BlueSky	0.263	0.336	0.667	0.575	0.263	0.249	0.313	0.425
CrowdRun	0.128	0.210	0.250	0.309	0.128	0.148	0.173	0.181
DucksTakeOff	0.334	0.701	0.473	0.374	0.334	0.975	0.472	0.302
Kimono	0.208	0.173	0.301	0.284	0.208	0.114	0.106	0.164
ParkJoy	0.240	0.472	0.368	0.365	0.240	0.434	0.340	0.307
ParkScene	0.156	0.200	0.218	0.488	0.156	0.113	0.136	0.222
PedestrianArea	0.258	0.319	0.499	0.366	0.258	0.235	0.319	0.323
Riverbed	0.090	0.106	0.164	0.228	0.090	0.090	0.122	0.162
RushHour	0.374	0.248	0.496	0.421	0.374	0.188	0.186	0.167
Sunflower	0.453	0.969	0.827	0.735	0.453	0.631	0.490	0.409
Average	**0.250**	**0.373**	**0.426**	**0.415**	**0.250**	**0.318**	**0.266**	**0.266**

Table 6.5: Comparison of encoding results between the original encoding based on CTU-level rate control and the proposed model parameters reuse method, and mean bitrate deviation of the proposed method.

Sequence	**BD-rate**	**BD-PSNR**	**Mean deviation**
BlueSky	−0.51%	0.02 dB	6.32%
CrowdRun	−0.29%	0.01 dB	0.64%
DucksTakeOff	0.41%	−0.01 dB	1.71%
Kimono	−6.89%	0.24 dB	0.70%
ParkJoy	0.18%	−0.01 dB	1.88%
ParkScene	−0.65%	0.02 dB	3.42%
PedestrianArea	−2.41%	0.08 dB	2.08%
Riverbed	−2.52%	0.11 dB	0.06%
RushHour	−8.78%	0.23 dB	1.48%
Sunflower	−1.09%	0.03 dB	8.98%
Average	**−2.25%**	**0.07 dB**	**2.73%**

encodings for the multi-rate method compared to the original rate control.

Finally, Table 6.5 shows the encoding results of the proposed multi-rate encoding method, compared to the original rate control at CTU level, which was chosen as baseline for comparison due to its better RD performance. On average, the BD-rate is decreased by 2.25%. This shows that the proposed method is able, on average, to improve the RD performance of the rate control in a multi-rate scenario. The average deviation of the rate control when using the proposed method is 2.73%. This is slightly worse than the average deviation of the original rate control (2.32%, cf. Table 6.2). This result is somehow unexpected, because the mean model error at frame level is reduced on average for the dependent encodings (cf.

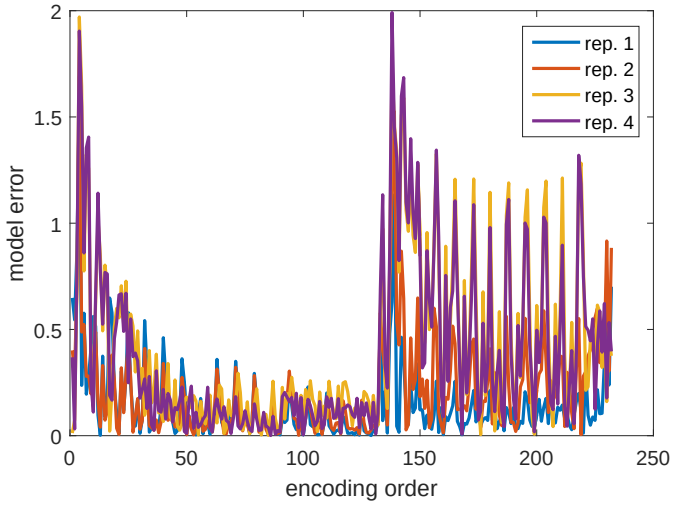


Figure 6.6: Model error at frame level for the entire *Kimono* sequence at different representations with the original rate control.

Table 6.4), and thus, similar to the improved RD performance, an improved bitrate accuracy is expected (smaller deviation).

The encoding time for four representations is also measured and the encoding time difference between the original encodings and the multi-rate encodings using the proposed method is calculated. The average encoding time difference is 0.12%. A *t-test* [91] on the encoding time differences (0.11%, 0.30%, −1.20%, 0.71%, 0.39%, −0.14%, 1.68%, −0.47%, −0.72%, 0.62% in alphabetical order) does not reject the hypothesis that the time difference comes from a normal distribution with mean equal to zero. Thus, there is statistically no noticeable complexity difference due to the proposed method. The small encoding time differences are thus just common small variations of the execution time on a multi-task operating system such as Ubuntu server.

6.4.4 Scene change

In the case of a scene change during a video, the "true" model parameters α and β change, because the content characteristics of the video change as well. However, in the method proposed by Li *et al.* [31], the parameters keep using the same update process described in Section 6.2.3. As a result, the update of the model parameters is too slow to react to the scene change.

As an example, the original HM encoder is used to encode the entire *Kimono* sequence (240 frames), which contains a scene change at frame number 140. With the *random access, main* profile, the encoded video contains 8 I-frames that are left out of this analysis. Figure 6.6 shows the model error ϵ at frame level for four representations of the *Kimono* sequence. The model error is large after the scene change, because the parameter update process is slow to converge to the new "true" parameters.

The multi-rate reuse method is now applied to scene changes as well: the updated model

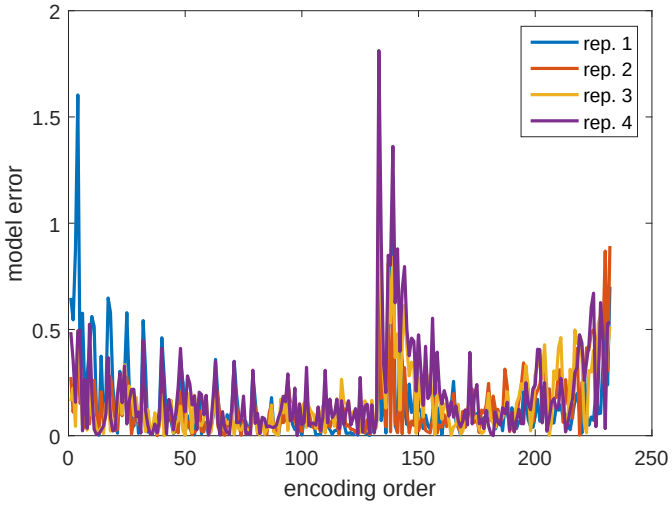


Figure 6.7: Model error at frame level for the entire *Kimono* sequence at different representations with the proposed multi-rate method, also applied at the scene change.

Table 6.6: Mean model error at frame level for the entire *Kimono* sequence.

original rate control				**proposed multi-rate**			
rep. 1	rep. 2	rep. 3	rep. 4	rep. 1	rep. 2	rep. 3	rep. 4
0.147	0.210	0.409	0.387	0.147	0.136	0.159	0.197

parameters of the reference encoding are passed step-by-step to the dependent encodings for the first four frames at each hierarchical level after the scene change.

Figure 6.7 shows the model error at frame level for the entire *Kimono* sequence, when the proposed multi-rate method is applied. Specifically, the parameters of the first four non-I-frames are passed step-by-step from the reference encoding to the dependent encodings. Additionally, the parameters of the frames number 144 (hierarchical level 1), 140 (level 2), 142 (level 3), and 141 (level 4) are passed as part of the multi-rate method. Compared to Figure 6.6, the model error for the dependent encodings is reduced, especially after the scene change.

Table 6.6 presents the mean model error at frame level over all non-I-frames of the *Kimono* sequence. The numbers confirm that the model error is smaller on average for the proposed multi-rate method compared to the original rate control. From an RD perspective, the BD-rate is decreased by 2.05% with the proposed multi-rate method. The mean bitrate deviation for the original rate control is 0.31%, and the bitrate accuracy is improved in this case with the multi-rate method to a mean deviation of 0.14%.

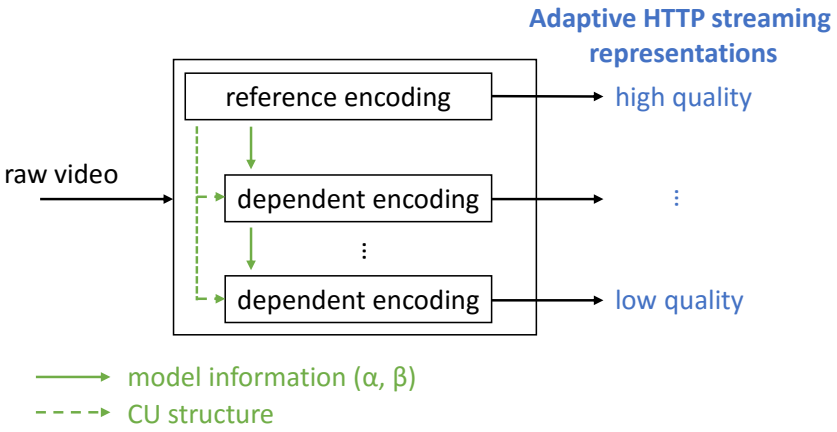


Figure 6.8: Schema of the proposed combined multi-rate encoder. The model parameters α and β are passed step-by-step from the reference encoding to the dependent encodings, while the CU structure information is passed directly from the reference encoding to all dependent encodings.

6.5 Combination of the proposed method with the CU structure reuse method

So far, in this chapter it has been shown that the RD performance of HEVC encoding can be improved in a multi-rate scenario where rate control is used. This improved RD performance does not come with an increased complexity. On the other hand, in Chapters 4 and 5, the proposed methods were targeted at reducing the overall encoding complexity. In this section, the combination of the proposed method to improve the RD performance with a method to reduce the encoding complexity is examined. Therefore, the CU structure reuse method from Chapter 4 is selected, as it is the method which brings the largest encoding time reduction.

The schema of the combined multi-rate encoder is shown in Figure 6.8. The reference encoding is the encoding for the representation at highest quality. The model parameters α and β are passed step-by-step from the reference encoding through the dependent encodings. On the other hand, the CU structure information from the reference is directly passed to all dependent encodings.

The combined multi-rate encoder is applied to the main set using rate control and the target bitrates from Table 6.1. The encoding results compared with the unmodified HM encoder and original rate control at CU level are presented in Table 6.7. Due to the CU structure reuse, the average encoding time is reduced by 35.05%. Furthermore, the multi-rate method to improve the rate control leads to an average BD-rate decrease of 1.84%. The RD performance improvement is slightly less than for the case where the CU structure reuse method is not used (2.25%, cf. Table 6.5). This is due to the CU structure reuse method, which slightly degrades the RD performance (cf. Section 4.3.3). In general, the results show that the combination of the proposed methods leads to both an encoding complexity reduction and an RD performance improvement, in the case where rate control is used. The combination of

Table 6.7: Comparison of encoding results between the original rate control and the combination of CU structure reuse method and rate control model parameters reuse method, and accuracy of the proposed method.

Sequence	BD-rate	BD-PSNR	ΔT	Mean deviation
BlueSky	−0.74%	0.02 dB	−43.90%	9.06%
CrowdRun	0.10%	−0.01 dB	−19.06%	0.61%
DucksTakeOff	0.42%	−0.01 dB	−21.95%	1.68%
Kimono	−6.25%	0.22 dB	−39.40%	0.71%
ParkJoy	0.50%	−0.02 dB	−21.71%	1.65%
ParkScene	−0.44%	0.01 dB	−32.92%	2.99%
PedestrianArea	−1.66%	0.05 dB	−40.43%	1.89%
Riverbed	−2.09%	0.09 dB	−40.18%	0.06%
RushHour	−7.92%	0.21 dB	−39.20%	1.49%
Sunflower	−0.29%	0.02 dB	−51.73%	8.83%
Average	**−1.84%**	**0.06 dB**	**−35.05%**	**2.90%**

the two methods also leads to a very small increase of the mean bitrate deviation with an average of 2.90% instead of 2.73%.

6.6 Summary

In this chapter, the scenario where rate control is applied on the different adaptive HTTP streaming representations has been considered. For HEVC, the current rate control algorithm in the reference encoder HM performs suboptimally, as the RD performance compared to encodings with QP is degraded. After a summary of the rate control algorithm by Li *et al.*, the practical limitations of the underlying model have been identified. The behavior of the internal model parameters has been analyzed, and a metric for the model error, which characterizes the mismatch of the current model parameters compared with the optimal model parameters has been proposed.

To alleviate the RD performance loss when using rate control, a multi-rate method which passes the encoder internal model parameters from a video sequence step-by-step from a high-quality reference encoding to the dependent encodings has been proposed. The proposed method is shown to reduce the average model error. Encoding results compared to the original rate control show that the RD performance can be improved, at the price of a minor decrease in bitrate accuracy of the rate control. The encoding complexity is not increased by the proposed method. Although the proposed method is specific to the rate control implemented in HM, the results show that reusing information that characterizes the video content can be beneficial in terms of RD performance in the case of rate control, which in general relies on modeling the RD characteristics of the video sequence.

In a final step, the proposed multi-rate method has been combined with a multi-rate

method from Chapter 2.3 which reduces the encoding complexity. Results show that, in the case of rate control, a multi-rate encoder can achieve a better RD performance at a lower encoding complexity than an encoder which treats the different representations independently.

Chapter 7

Conclusion and future work

7.1 Conclusion

The continuous rise in video streaming over the internet has driven the demand for efficient video compression. Currently, HEVC offers the best RD performance among available video codecs. However, its increased encoding complexity as well as the need to encode at several representations for adaptive HTTP streaming are challenging the video providers as the encoding costs are rising. This thesis focuses on multi-rate HEVC encoding and proposes solutions to decrease the video encoding complexity for adaptive HTTP streaming as well as improve the RD performance of the encoding under certain conditions. The contributions of the thesis are summarized in the following.

First, the case of an adaptive HTTP streaming scenario with representations at a single spatial resolution and different SNR qualities has been considered. Observations of the encoding decisions similarities have shown that the encoding decisions from a high-quality reference cannot be directly reused as encoding decisions in the lower-quality dependent encodings, as this would notably harm the RD performance of the encodings. On the contrary, it has been shown that the encoding information from the high-quality reference can be used to constrain the RDO of the dependent encodings. This leads to an encoding complexity reduction without significantly decreasing the RD performance. Specifically, methods to reuse the CU structure, the prediction mode, the intra mode, and the motion vectors have been proposed. Finally, these methods have been combined to form a multi-rate encoder capable of reducing the overall encoding complexity by 37% on average at the cost of a 0.46% BD-rate increase.

Second, the case of representations at different spatial resolutions has been studied. The main challenge to share information between encodings of different representations is that the HEVC encoding decisions are taken at different block levels, but these blocks may not be corresponding depending on the downsampling ratio between the different representations. Still, methods to extract the relevant information from a high-resolution reference encoding have been proposed and the information is used to constrain the RDO of lower-resolution dependent encodings. The methods reusing the CU structure, the prediction mode, and the

intra mode have shown to decrease the encoding complexity at the price of a slightly higher BD-rate. Finally, a multi-rate encoder spanning representations at different resolutions and different signal qualities has been presented. Combining the various proposed methods, the multi-rate encoder can reduce the overall encoding time by 42% while the BD-rate increase is approximately 1%, which outperforms a state-of-the-art method in terms of RD performance.

Finally, a practical scenario where the representations are encoded using rate control has been considered. It has been shown that the RD performance of the rate control algorithm in the HEVC reference software is suboptimal due to approximations in the underlying model and a metric to measure the model error has been proposed. A multi-rate method where the model information is shared among representations at different SNR qualities has been presented. The RD performance of the multi-rate system has been improved with the proposed method without increasing the encoding complexity. At last, the proposed method has been combined with a previously presented method constraining the RDO, leading to a multi-rate encoder capable of both reducing the encoding complexity and increasing the RD performance.

7.2 Future work

The research work presented in this thesis can be extended in various directions.

1. HEVC is currently the video compression standard achieving the best RD performance [92]. However, the field of video coding is moving forward and new compression techniques as well as new codecs arise. For example, the Alliance for Open Media is planning to release its first codec AV1 by the end of 2016 or beginning of 2017 [93]. AV1 is aiming to attain a better RD performance than HEVC, with the significant advantage of being royalty-free. A new codec means that new multi-rate methods need to be developed, based on the observed similarities between different representations and based on the specific encoding decisions that can be shared within the multi-rate system.

2. The HM software is the reference HEVC encoder, but is not optimized for encoding time/complexity. On the other hand, there are many other HEVC encoders (e.g., the open-source x265 encoder) that use different techniques such as the ones presented in the state-of-the-art description of HEVC (cf. Section 2.1.5) to reduce the encoding complexity. Depending on the implemented techniques, the effect of the proposed multi-rate methods should be studied. In particular, as the implemented techniques can already constrain the RDO, the complexity reduction achieved with the multi-rate methods could be lower.

3. The proposed methods are based on one reference encoding, which is the encoding at the best SNR quality and the highest spatial resolution. While this choice makes sense

if the goal is to provide an RD performance close to the original single-layer encoder (cf. Section 4.3.2), targeting the highest encoding complexity reduction could benefit from using a different reference. The effect on both the RD performance and the encoding complexity reduction of choosing different references for the proposed methods should be evaluated. In multi-rate systems with a large number of representations, one could also think of having more than one reference encoding.

4. The current algorithm for rate control in HM has been shown to perform suboptimally in terms of RD performance and in terms of bitrate accuracy and could be potentially replaced in HEVC encoders. Based on different rate control models, the possibilities of improving both the RD performance and the bitrate accuracy with multi-rate methods should be further examined. Besides the model parameters, additional information gained from encoding the reference encoding could be shared with the dependent encodings, similar to the case of two-pass rate control.

5. This thesis is based on classical distortion measures (MSE and PSNR) for comparing the RD performance of two encoders. However, these metrics have been shown to poorly represent the human perceptual quality in certain scenarios [51]. As the target of adaptive HTTP streaming are generally human viewers, the perceptual quality should be part of the evaluation process of a multi-rate encoder. For example, the goal could be to decrease the overall encoding complexity while keeping a given perceptual quality constant. Similarly, the perceptual quality could be improved without increasing the encoding complexity.

Bibliography

Publications by the author

Journal publications

[1] D. Schroeder, A. Ilangovan, M. Reisslein, and E. Steinbach, "Efficient multi-rate video encoding for HEVC-based adaptive HTTP streaming," *IEEE Transactions on Circuits and Systems for Video Technology, accepted for publication*, 2016. DOI: 10.1109/TCSVT.2016.2599028.

[2] C. Lottermann, D. Schroeder, and E. Steinbach, "Low-complexity and context-aware estimation of spatial and temporal activity parameters for automotive camera rate control," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 7, pp. 1376–1390, Jul. 2016. DOI: 10.1109/TCSVT.2015.2455771.

[3] J. Chao, R. Huitl, E. Steinbach, and D. Schroeder, "A novel rate control framework for SIFT/SURF feature preservation in H.264/AVC video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 6, pp. 958–972, Jun. 2015. DOI: 10.1109/TCSVT.2014.2367354.

[4] A. El Essaili, D. Schroeder, E. Steinbach, D. Staehle, and M. Shehada, "QoE-based traffic and resource management for adaptive HTTP video delivery in LTE," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 6, pp. 988–1001, Jun. 2015. DOI: 10.1109/TCSVT.2014.2367355.

Conference publications

[5] D. Schroeder, A. El Essaili, and E. Steinbach, "Fast converging auction-based resource allocation for QoE-driven wireless video streaming," in *Proc. IEEE International Conference on Communications (ICC), Workshop on Quality of Experience-based Management for Future Internet Applications and Services*, Kuala Lumpur, Malaysia, May 2016. DOI: 10.1109/ICCW.2016.7503843.

[6] D. Schroeder, A. Ilangovan, and E. Steinbach, "Multi-rate encoding for HEVC-based adaptive HTTP streaming with multiple resolutions," in *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Xiamen, China, Oct. 2015. DOI: 10.1109/MMSP.2015.7340822.

[7] D. Schroeder, P. Rehm, and E. Steinbach, "Block structure reuse for multi-rate high efficiency video coding," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Quebec City, Canada, Sep. 2015, pp. 3972 –3976. DOI: 10.1109/ICIP.2015.7351551.

[8] C. Lottermann, S. Gül, D. Schroeder, and E. Steinbach, "Network-aware video level encoding for uplink adaptive HTTP streaming," in *Proc. IEEE International Conference on Communications (ICC)*, London, UK, Jun. 2015, pp. 6861–6866. DOI: 10.1109/ICC.2015.7249419.

[9] C. Lottermann, A. Machado, D. Schroeder, Y. Peng, and E. Steinbach, "Bit rate estimation for H.264/AVC video encoding based on temporal and spatial activities," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Paris, France, Oct. 2014, pp. 3195–3199. DOI: 10.1109/ICIP.2014.7025646.

[10] C. Lottermann, A. Machado, D. Schroeder, W. Hintermaier, and E. Steinbach, "Camera context based estimation of spatial and temporal activity parameters for video quality metrics in automotive applications," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, Chengdu, China, Jul. 2014. DOI: 10.1109/ICME.2014.6890223.

[11] S. Khan, D. Schroeder, A. El Essaili, and E. Steinbach, "Energy-efficient and QoE-driven adaptive HTTP streaming over LTE," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Istanbul, Turkey, Apr. 2014, pp. 2354–2359. DOI: 10.1109/WCNC.2014.6952717.

[12] D. Schroeder, A. El Essaili, E. Steinbach, D. Staehle, and M. Shehada, "Low-complexity no-reference PSNR estimation for H.264/AVC encoded video," in *Proc. International Packet Video Workshop*, San Jose, CA, USA, Dec. 2013. DOI: 10.1109/PV.2013.6691445.

[13] A. El Essaili, D. Schroeder, D. Staehle, M. Shehada, W. Kellerer, and E. Steinbach, "Quality-of-experience driven adaptive HTTP media delivery," in *Proc. IEEE International Conference on Communications (ICC)*, Budapest, Hungary, Jun. 2013, pp. 2480 – 2485. DOI: 10.1109/ICC.2013.6654905.

[14] D. Schroeder, A. El Essaili, E. Steinbach, Z. Despotovic, and W. Kellerer, "A quality-of-experience driven bidding game for uplink video transmission in next generation mobile networks," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Orlando, Florida, USA, Sep. 2012, pp. 2281 –2284. DOI: 10.1109/ICIP.2012.6467351.

[15] A. El Essaili, L. Zhou, D. Schroeder, E. Steinbach, and W. Kellerer, "QoE-driven live and on-demand LTE uplink video transmission," in *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Hangzhou, China, Oct. 2011. DOI: 10.1109/MMSP.2011.6093821.

General publications

[16] Sandvine, "Global Internet Phenomena Report, Africa, Middle East and North America," Dec. 2015.

[17] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[18] Netflix. (Dec. 2015). Per-Title Encode Optimization, [Online]. Available: http : / / techblog.netflix.com/2015/12/per-title-encode-optimization.html.

[19] T. Stockhammer, "Dynamic adaptive streaming over HTTP–: standards and design principles," in *Proc. ACM conference on Multimedia systems (MMSys)*, San Jose, CA, USA, Feb. 2011, pp. 133–144.

[20] C. Timmerer, C. Griwodz, A. C. Begen, T. Stockhammer, and B. Girod, "Guest editorial - adaptive media streaming," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 681–683, Apr. 2014.

[21] G. K. Walker, T. Stockhammer, G. Mandyam, Y.-K. Wang, and C. Lo, "ROUTE/DASH IP Streaming-Based System for Delivery of Broadcast, Broadband, and Hybrid Services," *IEEE Transactions on Broadcasting*, vol. 62, no. 1, pp. 328–337, Mar. 2016.

[22] D. H. Finstad, H. K. Stensland, H. Espeland, and P. Halvorsen, "Improved multirate video encoding," in *Proc. IEEE International Symposium on Multimedia (ISM)*, Dana Point, CA, USA, Dec. 2011.

[23] ITU-T, "High efficiency video coding," International Telecommunication Union, Recommendation H.265, Apr. 2013.

[24] ISO/IEC, "High efficiency video coding," International Organization for Standardization/International Electrotechnical Commission, International Standard 23008-2:2013, Dec. 2013.

[25] M. Wien, *High Efficiency Video Coding, Coding Tools and Specifications*. Springer, 2015.

[26] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1697–1706, Dec. 2012.

[27] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra Coding of the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1792–1801, Dec. 2012.

[28] M. Budagavi, A. Fuldseth, G. Bjontegaard, V. Sze, and M. Sadafale, "Core transform design in the high efficiency video coding (HEVC) standard," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1029–1041, Dec. 2013.

[29] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

[30] B. Bing, *Next-Generation Video Coding and Streaming*. John Wiley & Sons, Inc, 2015.

[31] B. Li, H. Li, L. Li, and J. Zhang, "Lambda domain rate control algorithm for high efficiency video coding," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3841–3854, Sep. 2014.

[32] Z. He, Y. K. Kim, and S. K. Mitra, "Low-delay rate control for DCT video coding via ρ-domain source modeling," *IEEE transactions on Circuits and Systems for Video Technology*, vol. 11, no. 8, pp. 928–940, Aug. 2001.

[33] JCT-VC. (2015). HEVC reference software HM 16.5, [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.5/.

[34] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012.

[35] N. Purnachand, L. N. Alves, and A. Navarro, "Improvements to TZ search motion estimation algorithm for multiview video coding," in *Proc. 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Vienna, Austria, Apr. 2012, pp. 388–391.

[36] XEVC. (Jan. 2014). Motion estimation of HM encoder, [Online]. Available: https://xevc.net/2014/01/23/motion-estimation-of-hm-encoder/.

[37] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An effective CU size decision method for HEVC encoders," *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 465–470, Feb. 2013.

[38] S. Cho and M. Kim, "Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 9, pp. 1555–1564, Sep. 2013.

[39] H.-S. Kim and R.-H. Park, "Fast CU Partitioning Algorithm for HEVC Using an Online-Learning-Based Bayesian Decision Rule," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 130–138, Jan. 2016.

[40] L. Zhao, L. Zhang, S. Ma, and D. Zhao, "Fast mode decision algorithm for intra prediction in HEVC," in *Proc. IEEE Visual Communications and Image Processing (VCIP)*, Tainan, Taiwan, Nov. 2011, pp. 1–4.

[41] W. Jiang, H. Ma, and Y. Chen, "Gradient based fast mode decision algorithm for intra prediction in HEVC," in *Proc. 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, Yichang, China, Apr. 2012, pp. 1836–1840.

[42] H. Zhang and Z. Ma, "Fast intra mode decision for high efficiency video coding (HEVC)," *IEEE Transactions on circuits and systems for video technology*, vol. 24, no. 4, pp. 660–668, Apr. 2014.

[43] N. Hu and E.-H. Yang, "Fast motion estimation based on confidence interval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 8, pp. 1310–1322, Aug. 2014.

[44] K.-Y. Kim, H.-Y. Kim, J.-S. Choi, and G.-H. Park, "MC complexity reduction for generalized P and B pictures in HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 10, pp. 1723–1728, Oct. 2014.

[45] Z. Pan, S. Kwong, M.-T. Sun, and J. Lei, "Early merge mode decision based on motion estimation and hierarchical depth correlation for HEVC," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 405–412, Jun. 2014.

[46] C. Yan, Y. Zhang, J. Xu, F. Dai, J. Zhang, Q. Dai, and F. Wu, "Efficient parallel framework for HEVC motion estimation on many-core processors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 12, pp. 2077–2089, Dec. 2014.

[47] S. Radicke, J.-U. Hahn, Q. Wang, and C. Grecos, "A Parallel HEVC Intra Prediction Algorithm for Heterogeneous CPU+GPU Platforms," *IEEE Transactions on Broadcasting*, vol. 62, no. 1, pp. 103–119, Mar. 2016.

[48] K. Chen, J. Sun, Y. Duan, and Z. Guo, "A novel wavefront-based high parallel solution for HEVC encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 181–194, Jan. 2016.

[49] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," *Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA*, Apr. 2001.

[50] ——, "Improvements of the BD-PSNR model," *ITU-T SG16 Q*, vol. 6, p. 35, 2008.

[51] B. Girod, "Digital images and human vision," in, A. B. Watson, Ed., Cambridge, MA, USA: MIT Press, 1993, ch. What's Wrong with Mean-squared Error? Pp. 207–220.

[52] A. C. Begen, T. Akgul, and M. Baugher, "Watching video over the web: Part 1: Streaming protocols," *IEEE Internet Computing*, vol. 15, no. 2, pp. 54–63, Mar. 2011.

[53] R. Pantos, "HTTP Live Streaming," IETF, Informational Internet Draft, May 2009.

[54] Microsoft, *[MS-SSTR]: Smooth Streaming Protocol*, 2010. [Online]. Available: https://msdn.microsoft.com/en-us/library/ff469518.aspx.

[55] ISO/IEC, "Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats," International Standard 23009-1:2012, Apr. 2012.

[56] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *Proc. ACM conference on Multimedia systems (MMSys)*, San Jose, CA, USA, Feb. 2011, pp. 169–174.

[57] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation algorithm for adaptive streaming over HTTP," in *Proc. 19th International Packet Video Workshop (PV)*, Munich, Germany, May 2012, pp. 173–178.

[58] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE," in *Proc. 8th international conference on Emerging networking experiments and technologies (CoNEXT)*, Nice, France, Dec. 2012, pp. 97–108.

[59] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. Begen, and D. Oran, "Probe and adapt: Rate adaptation for HTTP video streaming at scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719–733, Apr. 2014.

[60] C. Timmerer, M. Maiero, and B. Rainer, "Which Adaptation Logic? An Objective and Subjective Performance Evaluation of HTTP-based Adaptive Media Streaming Systems," *arXiv preprint arXiv:1606.00341*, Jun. 2016.

[61] R. K. Mok, E. W. Chan, and R. K. Chang, "Measuring the quality of experience of HTTP video streaming," in *Proc. IFIP/IEEE International Symposium on Integrated Network Management (IM)*, Dublin, Ireland, May 2011, pp. 485–492.

[62] O. Oyman and S. Singh, "Quality of experience for HTTP adaptive streaming services," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 20–27, Apr. 2012.

[63] K. D. Singh, Y. Hadjadj-Aoul, and G. Rubino, "Quality of experience estimation for adaptive HTTP/TCP video streaming using H.264/AVC," in *Proc. IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, USA, Jan. 2012, pp. 127–131.

[64] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. ACM conference on Multimedia systems (MMSys)*, San Jose, CA, USA, Feb. 2011, pp. 157–168.

[65] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What happens when HTTP adaptive streaming players compete for bandwidth?" In *Proc. International workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Toronto, Canada, Jun. 2012, pp. 9–14.

[66] R. Houdaille and S. Gouache, "Shaping HTTP adaptive streams for a better user experience," in *Proc. ACM 3rd Multimedia Systems Conference (MMSys)*, Chapel Hill, NC, USA, Feb. 2012, pp. 1–9.

[67] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "QoE-driven cache management for HTTP adaptive bit rate streaming over wireless networks," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1431–1445, Oct. 2013.

[68] C. Mueller, S. Lederer, C. Timmerer, and H. Hellwagner, "Dynamic adaptive streaming over HTTP/2.0," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, San Jose, CA, USA, Jul. 2013, pp. 1–6.

[69] V. Adzic, H. Kalva, and B. Furht, "Optimizing video encoding for adaptive streaming over HTTP," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 397–403, May 2012.

[70] L. Toni, R. Aparicio-Pardo, K. Pires, G. Simon, A. Blanc, and P. Frossard, "Optimal selection of adaptive streaming representations," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 11, no. 2s, p. 43, Feb. 2015.

[71] S. Wang, S. Ma, S. Wang, D. Zhao, and W. Gao, "Rate-GOP based rate control for high efficiency video coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1101–1111, Dec. 2013.

[72] Y. Chen, Z. Wen, J. Wen, M. Tang, and P. Tao, "Efficient Software H. 264/AVC to HEVC Transcoding on Distributed Multi-Core Processors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 6, pp. 1423–1434, Aug. 2015.

[73] A. Diaz-Honrubia, J. Martinez, P. Cuenca, J. Gamez, and J. Puerta, "Adaptive Fast Quadtree Level Decision Algorithm for H. 264/HEVC Video Transcoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 154–168, Jan. 2016.

[74] L. Pham Van, J. De Praeter, G. Van Wallendael, S. Van Leuven, J. De Cock, and R. Van de Walle, "Efficient bit rate transcoding for high efficiency video coding," *IEEE Transactions on Multimedia*, vol. 18, no. 3, pp. 364–378, Mar. 2016.

[75] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H. 264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[76] J. Boyce, Y. Ye, J. Chen, and A. Ramasubramonian, "Overview of SHVC: Scalable Extensions of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 20–34, Jan. 2015.

[77] A. Zaccarin and B.-L. Yeo, "Multi-rate encoding of a video sequence in the DCT domain," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 2, Scottsdale, AZ, USA, May 2002, pp. II–680–II–683.

[78] J. Song and B.-L. Yeo, "A fast algorithm for DCT-domain inverse motion compensation based on shared information in a macroblock," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 767–775, Aug. 2000.

[79] J. Bankoski, P. Wilkins, and Y. Xu, "VP8 data format and decoding guide," IETF, Informational RFC 6386, 2011.

[80] J. De Praeter, A. J. Diaz-Honrubia, N. Van Kets, G. Van Wallendael, J. De Cock, P. Lambert, and R. Van de Walle, "Fast simultaneous video encoder for adaptive streaming," in *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Xiamen, China, Oct. 2015.

[81] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[82] G. Cebrián-Márquez, A. J. Diaz-Honrubia, J. De Praeter, G. Van Wallendael, J. L. Martinez, and P. Cuenca, "A motion vector re-use algorithm for H. 264/AVC and HEVC simultaneous video encoding," in *Proc. 13th International Conference on Advances in Mobile Computing and Multimedia*, Brussels, Belgium, Dec. 2015, pp. 241–245.

[83] F. Bossen, "Common test conditions and software reference configurations," JCT-VC, Tech. Rep. L1100, Jan. 2013.

[84] Xiph.org Foundation. (2016). Xiph.org video test media, [Online]. Available: http://media.xiph.org/video/derf/.

[85] Y. Peng and E. Steinbach, "A novel full-reference video quality metric and its application to wireless video transmission," in *Proc. 18th IEEE International Conference on Image Processing (ICIP)*, Brussels, Belgium, Sep. 2011, pp. 2517–2520.

[86] G. R. Iversen and H. Norpoth, *Analysis of variance*, 1. Sage, 1987.

[87] F. Bossen, D. Flynn, K. Sharman, and K. Sühring. (2015). HM Software Manual.

[88] D. Zhang, K. N. Ngan, and Z. Chen, "A two-pass rate control algorithm for H.264/AVC high definition video coding," *Signal Processing: Image Communication*, vol. 24, no. 5, pp. 357–367, 2009.

[89] H. Choi, J. Yoo, J. Nam, D. Sim, and I. V. Bajić, "Pixel-wise unified rate-quantization model for multi-level rate control," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1112–1123, Dec. 2013.

[90] B. Li, H. Li, L. Li, and J. Zhang, "Rate Control by R-Lambda Model for HEVC," JCT-VC, Shanghai, China, Tech. Rep. K0103, Oct. 2012.

[91] Student, "The probable error of a mean," *Biometrika*, pp. 1–25, 1908.

[92] J. De Cock, A. Mavlankar, A. Moorthy, and A. Aaron, "A large-scale video codec comparison of x264, x265 and libvpx for practical VOD applications," in *Proc. SPIE 9971, Applications of Digital Image Processing XXXIX*, Sep. 2016.

[93] J. Ozer. (Apr. 2016). A Progress Report: The Alliance for Open Media and the AV1 Codec, [Online]. Available: http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/A-Progress-Report-The-Alliance-for-Open-Media-and-the-AV1-Codec-110383.aspx.

List of Figures

List of Tables